SILVACO

Atlas User's Manual

DEVICE SIMULATION SOFTWARE

Silvaco, Inc.
4701 Patrick Henry Drive, Bldg. 2
Santa Clara, CA 95054
Phone: (408) 567-1000
Web: www.silvaco.com

August 26, 2016

The information contained in this document is subject to change without notice.

Silvaco, Inc. MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

Silvaco, Inc. shall not be held liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

AccuCell, AccuCore, Athena, Athena 1D, Atlas, Blaze, C-Interpreter, Catalyst AD, Catalyst DA, Clarity RLC, Clever, Clever Interconnect, Custom IC CAD, DeckBuild, DevEdit, DevEdit 3D, Device 3D, DRC Assist, Elite, Exact, Expert, Expert C++, Expert 200, ExpertViews, Ferro, Gateway, Gateway 200, Giga, Giga 3D, Guardian, Guardian DRC, Guardian LVS, Guardian NET, Harmony, Hipex, Hipex C, Hipex NET, Hipex RC, HyperFault, Interconnect Modeling, IWorkBench, Laser, LED, LED 3D, Lisa, Luminous, Luminous 3D, Magnetic, Magnetic 3D, MaskViews, MC Etch & Depo, MC Device, MC Implant, Mercury, MixedMode, MixedMode XL, MultiCore, Noise, OLED, Optolith, Organic Display, Organic Solar, OTFT, Quantum, Quantum 3D, Quest, RealTime DRC, REM 2D, REM 3D, SEdit, SMovie, S-Pisces, SSuprem 3, SSuprem 4, SDDL, SFLM, SIPC, SiC, Silvaco, Silvaco Management Console, SMAN, Silvaco Relational Database, Silos, Simulation Standard, SmartSpice, SmartSpice 200, SmartSpice API, SmartSpice Debugger, SmartSpice Embedded, SmartSpice Interpreter, SmartSpice Optimizer, SmartSpice RadHard, SmartSpice Reliability, SmartSpice Rubberband, SmartSpice RF, SmartView, SolverLib, Spayn, SpiceServer, Spider, Stellar, TCAD Driven CAD, TCAD Omni, TCAD Omni Utility, TCAD & EDA Omni Utility, TFT, TFT 3D, Thermal 3D, TonyPlot, TonyPlot 3D, TurboLint, Universal Token, Universal Utility Token, Utmost III, Utmost III Bipolar, Utmost III Diode, Utmost III GaAs, Utmost III HBT, Utmost III JFET, Utmost III MOS, Utmost III MultiCore, Utmost III SOI, Utmost III TFT, Utmost III VBIC, Utmost IV, Utmost IV Acquisition Module, Utmost IV Model Check Module, Utmost IV Optimization Module, Utmost IV Script Module, VCSEL, Verilog-A, Victory, Victory Cell, Victory Device, Victory Device Single Event Effects, Victory Process, Victory Process Advanced Diffusion & Oxidation, Victory Process Monte Carlo Implant, Victory Process Physical Etch & Deposit, Victory Stress, Virtual Wafer Fab, VWF, VWF Automation Tools, VWF Interactive Tools, and Vyper are trademarks of Silvaco, Inc.

All other trademarks mentioned in this manual are the property of their respective owners.

Copyright © 1984 - 2016, Silvaco, Inc.

Style Conventions		
Font Style/Convention	**Description**	**Example**
•	This represents a list of items or terms.	• Bullet A • Bullet B • Bullet C
1. 2. 3.	This represents a set of directions to perform an action.	To open a door: 1. Unlock the door by inserting the key into keyhole. 2. Turn key counter-clockwise. 3. Pull out the key from the keyhole. 4. Grab the doorknob and turn clockwise and pull.
→	This represents a sequence of menu options and GUI buttons to perform an action.	**File→Open**
`Courier`	This represents the commands, parameters, and variables syntax.	`HAPPY BIRTHDAY`
Times Roman Bold	This represents the menu options and buttons in the GUI.	**File**
New Century Schoolbook Italics	This represents the variables of equations.	$x + y = 1$
New Century Schoolbook Italics Bold	This represents the vectors.	$x + \mathbf{y} = 1$
Note:	This represents the additional important information.	**Note:** Make sure you save often when working on a manual.

Acknowledgements

Atlas contains third-party software.

Atlas contains the QD library, the use of which is governed as follows:

This work was supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract numbers DE-AC03-76SF00098 and DE-AC02-05CH11231.

Copyright (c) 2003-2009, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(1) Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) Neither the name of the University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

Chapter 1
Introduction

1.1 Atlas Overview

Atlas provides general capabilities for physically-based two (2D) and three-dimensional (3D) simulation of semiconductor devices. If you're new to Atlas, read this chapter and Chapter 2 "Getting Started with Atlas" to understand how Atlas works. Once you've read these chapters, you can refer to the remaining chapters for a detailed understanding of the capabilities of each Atlas product.

Those who have used earlier versions of Atlas may find it helpful to review the updated version history in Appendix D: "ATLAS Version History".

Atlas is designed to be used with the VWF Interactive Tools. The VWF Interactive Tools are DeckBuild, TonyPlot, DevEdit, MaskViews, and Optimizer. See their respective manuals on how to use these products. See Section 1.3 "Using Atlas With Other Silvaco Software" for more information about using Atlas with other Silvaco tools.

Atlas is supplied with numerous examples that can be accessed through DeckBuild. These examples demonstrate most of Atlas's capabilities. The input files that are provided with the examples are an excellent starting point for developing your own input files. To find out how to access the example, see Section 2.4 "Accessing The Examples".

1.2 Features And Capabilities of Atlas

1.2.1 Comprehensive Set of Models

Atlas provides a comprehensive set of physical models, including:

- DC, AC small-signal, and full time-dependency.
- Drift-diffusion transport models.
- Energy balance and Hydrodynamic transport models.
- Lattice heating and heatsinks.
- Graded and abrupt heterojunctions.
- Optoelectronic interactions with general ray tracing.
- Amorphous and polycrystalline materials.
- General circuit environments.
- Stimulated emission and radiation
- Fermi-Dirac and Boltzmann statistics.
- Advanced mobility models.
- Heavy doping effects.
- Full acceptor and donor trap dynamics
- Ohmic, Schottky, and insulating contacts.
- SRH, radiative, Auger, and surface recombination.
- Impact ionization (local and non-local).
- Floating gates.
- Band-to-band and Fowler-Nordheim tunneling.
- Hot carrier injection.
- Quantum transport models
- Thermionic emission currents.

1.2.2 Fully Integrated Capabilities

Atlas works well with other software from Silvaco. For example, Atlas

- Runs in the DeckBuild interactive run-time environment.
- Is interfaced to TonyPlot, the interactive graphics and analysis package.
- Accepts input from the Athena and SSuprem3 process simulators.
- Is interfaced to Utmost parameter extraction and device modeling software.
- can be used in experiments with the VWF Automation Tools.

1.2.3 Sophisticated Numerical Implementation

Atlas uses powerful numerical techniques, including:

- Accurate and robust discretization techniques.
- Gummel, Newton, and block-Newton nonlinear iteration strategies.
- Efficient solvers, both direct and iterative, for linear subproblems.
- Powerful initial guess strategies.
- Small-signal calculation techniques that converge at all frequencies.
- Stable and accurate time integration.

1.3 Using Atlas With Other Silvaco Software

Atlas is best used with the VWF Interactive Tools. These include DeckBuild, TonyPlot, DevEdit, MaskViews, and Optimizer. DeckBuild provides an interactive run time environment. TonyPlot supplies scientific visualization capabilities. DevEdit is an interactive tool for structure and mesh specification and refinement. MaskViews is an IC Layout Editor. The Optimizer supports black box optimization across multiple simulators.

Atlas, however, is often used with the Athena process simulator. Athena predicts the physical structures that result from processing steps. The resulting physical structures are used as input by Atlas, which then predicts the electrical characteristics associated with specified bias conditions. The combination of Athena and Atlas makes it possible to determine the impact of process parameters on device characteristics.

The electrical characteristics predicted by Atlas can be used as input by the Utmost device characterization and SPICE modeling software. Compact models based on simulated device characteristics can then be supplied to circuit designers for preliminary circuit design. Combining Athena, Atlas, Utmost, and SmartSpice makes it possible to predict the impact of process parameters on circuit characteristics.

Atlas can also be used as one of the simulators within the VWF Automation Tools. VWF makes it convenient to perform highly automated simulation-based experimentation. VWF is used in a way that reflects experimental research and development procedures using split lots. It therefore links simulation very closely to technology development, resulting in significantly increased benefits from simulation use.

1.4 The Nature Of Physically-Based Simulation

Atlas is a physically-based device simulator. Physically-based device simulation is not a familiar concept for all engineers. This section will briefly describe this type of simulation.

Physically-based device simulators predict the electrical characteristics that are associated with specified physical structures and bias conditions. This is achieved by approximating the operation of a device onto a two or three dimensional grid, consisting of a number of grid points called nodes. By applying a set of differential equations, derived from Maxwell's laws, onto this grid you can simulate the transport of carriers through a structure. This means that the electrical performance of a device can now be modeled in DC, AC or transient modes of operation.

There are three physically-based simulation. These are:

- It is predictive.
- It provides insight.
- It conveniently captures and visualizes theoretical knowledge.

Physically-based simulation is different from empirical modeling. The goal of empirical modeling is to obtain analytic formulae that approximate existing data with good accuracy and minimum complexity. Empirical models provide efficient approximation and interpolation. They do not provide insight, or predictive capabilities, or encapsulation of theoretical knowledge.

Physically-based simulation has become very important for two reasons. One, it is almost always much quicker and cheaper than performing experiments. Two, it provides information that is difficult or impossible to measure.

The drawbacks of physically-based simulation are that all the relevant physics must be incorporated into a simulator. Also, numerical procedures must be implemented to solve the associated equations. These tasks have been taken care of for Atlas users.

Those who use physically-based device simulation tools must specify the problem to be simulated. In Atlas, specify device simulation problems by defining:

- The physical structure to be simulated.
- The physical models to be used.
- The bias conditions for which electrical characteristics are to be simulated.

The subsequent chapters of this manual describe how to perform these steps.

Chapter 2
Getting Started with Atlas

2.1 Overview

Atlas is a physically-based two and three dimensional device simulator. It predicts the electrical behavior of specified semiconductor structures and provides insight into the internal physical mechanisms associated with device operation.

Atlas can be used standalone or as a core tool in Silvaco's Virtual Wafer Fab simulation environment. In the sequence of predicting the impact of process variables on circuit performance, device simulation fits between process simulation and SPICE model extraction.

This chapter will show you how to use Atlas effectively. It is a source of useful hints and advice. The organization of topics parallels the steps that you go through to run the program. If you have used earlier versions of Atlas, you will still find this chapter useful because of the new version.

This chapter concentrates on the core functionality of Atlas. If you're primarily interested in the specialized capabilities of a particular Atlas tool, read this chapter first. Then, read the chapters that describe the Atlas tools you wish to use.

2.2 Atlas Inputs and Outputs

Figure 2-1 shows the types of information that flow in and out of Atlas. Most Atlas simulations use two input files. The first input file is a text file that contains commands for Atlas to execute. The second input file is a structure file that defines the structure that will be simulated.

Atlas produces three types of output files. The first type of output file is the run-time output, which gives you the progress and the error and warning messages as the simulation proceeds. The second type of output file is the log file, which stores all terminal voltages and currents from the device analysis. The third type of output file is the solution file, which stores 2D and 3D data relating to the values of solution variables within the device at a given bias point.

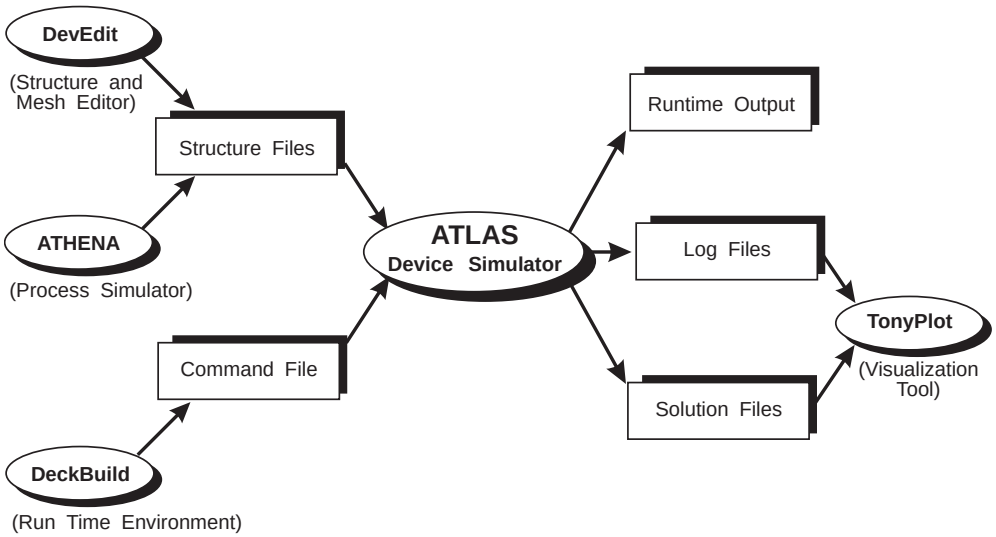


Figure 2-1 Atlas Inputs and Outputs

2.3 Modes of Operation

Atlas is normally used in conjunction with the DeckBuild run-time environment, which supports both interactive and batch mode operation. We strongly recommend that you always run Atlas within DeckBuild. In this section, we present the basic information you need to run Atlas in DeckBuild. The DeckBuild User's Manual provides a more detailed description of the features and capabilities of DeckBuild.

2.3.1 Interactive Mode With DeckBuild

To start Atlas in DeckBuild, type:

```
deckbuild -as
```

at the UNIX system command prompt. The command line option, `-as`, instructs DeckBuild to start Atlas as the default simulator.

If you want to start from an existing input file, start DeckBuild by typing:

```
deckbuild -as  <input filename>
```

The run-time output shows the execution of each Atlas command and includes error messages, warnings, extracted parameters, and other important output for evaluating each Atlas run. When Atlas runs in this mode, the run-time output is sent to the output section of the DeckBuild Window and can be saved as needed. Therefore, you don't need to save the run-time output explicitly. The following command line, however, specifies the name of a file that will be used for storing the run-time output.

```
deckbuild -as  <input filename>  -outfile  <output filename>
```

In this case, the run-time output is sent to the output file and to the output section of the DeckBuild Window.

2.3.2 Batch Mode With DeckBuild

To use DeckBuild in a non-interactive or batch mode, add the `-run` parameter to the command that invokes DeckBuild. A prepared command file is required for running in batch mode. We advise you to save the run-time output to a file, since error messages in the run-time output would otherwise be lost when the batch job completes. For example:

```
deckbuild -run -as  <input filename>  -outfile  <output filename>
```

Using this command requires a local X-Windows system to be running. The job runs inside a DeckBuild icon on the terminal and quits automatically when the Atlas simulation is complete. You can also run DeckBuild using a remote display. For example:

```
deckbuild -run -as <input file> -outfile <output file>
          -display<hostname>:0.0
```

2.3.3 No Windows Batch Mode With DeckBuild

For completely non-X Windows operation of DeckBuild, use the `-ascii` parameter. For example:

```
deckbuild -run -ascii -as <input filename>
          -outfile <output filename>
```

This command directs DeckBuild to run the Atlas simulation without the DeckBuild Window or icon. This is useful for remote execution without an X Windows emulator or for replacing UNIX-based Atlas runs within framework programs.

When using batch mode, use the UNIX command suffix, `&`, to detach the job from the current command shell. To run a remote Atlas simulation under DeckBuild without display and then logout from the system, use the UNIX command, `nohup` , before the DeckBuild command line. For example:

```
nohup deckbuild -run -ascii -as <input filename>
                -outfile <output filename> &
```

2.3.4 Running Atlas inside Deckbuild

Each Atlas run inside DeckBuild should start with the line:

```
go atlas
```

A single input file may contain several Atlas runs each separated with a `go atlas` line. Input files within DeckBuild may also contain runs from other programs such as Athena or DevEdit along with the Atlas runs.

Running a given version number of Atlas

The `go` statement can be modified to provide parameters for the Atlas run. To run version 5.14.0.R, the syntax is:

```
go atlas simflags="-V 5.14.0.R"
```

Starting Parallel Atlas

The `-P` option is used to set the number of processors to use in a parallel Atlas run. If the number set by `-P` is greater than the number of processors available or than the number of parallel thread licenses, the number is automatically reduced to this cap number. To run on 4 processors, use:

```
go atlas simflags="-V 5.14.0.R -P 4"
```

2.3.5 Batch Mode Without DeckBuild

You can run Atlas outside the DeckBuild environment. But this isn't recommended by Silvaco. If you don't want the overhead of the DeckBuild Window, use the No Windows Mode. Many important features such as variable substitution, automatic interfacing to process simulation, and parameter extraction are unavailable outside the DeckBuild environment. To run Atlas directly under UNIX, use the command:

```
atlas <input filename>
```

To save the run-time output to a file, don't use the UNIX redirect command (>). Simply specify the name of the output file. For example:

```
atlas <input filename> -logfile  <output filename>
```

Note: The standard examples supplied with Atlas will not run correctly outside of DeckBuild.

2.3.6 TMA Compatibility Mode

You can add the `-TMA` command line flag to the `atlas` command to direct the Atlas simulator to operate in TMA Compatibility Mode. In this mode, the default material models and parameters are altered to closely agree with those of TMA's Medici© simulator.

2.3.7 ISE Compatibility Mode

You can add the `-ISE` command line flag to the `atlas` command to direct the Atlas simulator to operate in ISE Compatibility Mode. In this mode, the default material models and parameters are altered to closely agree with those of ISE's Dessis© simulator.

2.4 Accessing The Examples

Atlas has a library of standard examples that demonstrate how the program is used to simulate different technologies. These examples are a good starting point for creating your own simulations. The examples are accessed from the menu system in DeckBuild. To select and load an example:

1. Start DeckBuild with Atlas as the simulator, which is described in the previous section.
2. Use left mouse button to pull down the **Main Control** menu.
3. Select **Examples**. An index will then appear in a Deckbuild Examples Window (see Figure 2-2).

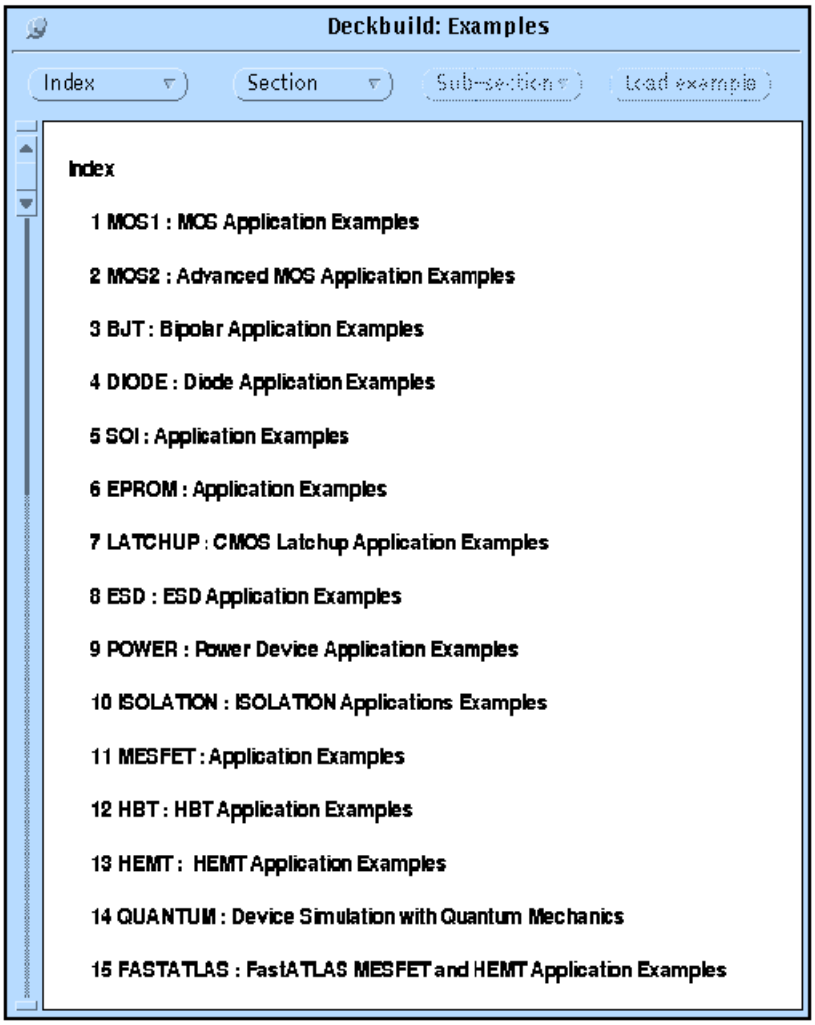


Figure 2-2 Examples Index in DeckBuild

The examples are divided by technology or technology group. For instance, the most common technologies are individually listed (e.g., MOS are under BJT), while others are grouped with similar devices (e.g., IGBT and LDMOS are under POWER, and solar cell and photodiode are under OPTOELECTRONICS).

4. Choose the technology by double clicking the left mouse button over that item. A list of examples for that technology will appear. These examples typically illustrate different devices, applications, or types of simulation.

   You can also search for an example by selecting the **Index** button. Wildcards can be used in the search.

5. Choose a particular example by double clicking the left mouse button over that item in the list. A text description of the example will appear in the window. This text describes the important physical mechanisms in the simulation and the details of the Atlas syntax used. You should read this information before proceeding.

6. Press the **Load example** button. The input command file for the example will be copied into your current working directory together with any associated files. A copy of the command file will be loaded into DeckBuild. Note that the **Load example** button remains faded out until this step is performed correctly.

7. Press the **run** button in the middle frame of the DeckBuild application window to run the example. Alternatively, most examples are supplied with results that are copied into the current working directory along with the input file. To view the results, select (highlight) the name of the results file and select **Tools→Plot**. See the TonyPlot User's Manual for details on using TonyPlot.

2.5 The Atlas Syntax

An Atlas command file is a list of commands for Atlas to execute. This list is stored as an ASCII text file that can be prepared in DeckBuild or in any text editor. Preparing an input file in DeckBuild is preferred. You can create an input file by using the DeckBuild Commands menu in the DeckBuild Window.

2.5.1 Statements and Parameters

The input file contains a sequence of statements. Each statement consists of a keyword that identifies the statement and a set of parameters. The general format is:

```
<STATEMENT>    <PARAMETER>=<VALUE>
```

With a few exceptions, the input syntax is not case sensitive. One important exception is that commands described in this manual as being executed by DeckBuild rather than Atlas are case sensitive. These include **EXTRACT**, **SET**, **GO**, and **SYSTEM**. Also, filenames for input and output under UNIX are case sensitive.

For any <STATEMENT>, Atlas may have four different types for the <VALUE> parameter. These are: Real, Integer, Character, and Logical.

An example of a statement line is:

```
DOPING UNIFORM N.TYPE CONCENTRATION=1.0e16 REGION=1 OUTFILE=my.dop
```

The statement is **DOPING**. All other items are parameters of the **DOPING** statement. UNIFORM and N.TYPE are Logical parameters. Their presence on the line sets their values to true. Otherwise, they take their default values (usually false). CONCENTRATION is a Real parameter and takes floating point numbers as input values. REGION is an Integer parameter taking only integer numbers as input. OUTFILE is a Character parameter type taking strings as input.

The statement keyword must come first but the order of parameters within a statement is unimportant.

You only need to use enough letters of any parameter to distinguish it from any other parameter on the same statement. Thus, CONCENTRATION can be shortened to CONC. REGION. It can't be shortened to R, however, since there's a parameter called RATIO associated with the DOPING statement.

You can set logical parameters to false by preceding them with the ^ symbol. Any line beginning with a # is ignored. These lines are used as comments.

Atlas can read up to 256 characters on one line. But it is better to spread long input statements over several lines to make the input file more readable. The \ character at the end of a line indicates continuation.

For more information about statements and parameters in Atlas, see Chapter 22 "Statements".

2.5.2 The Order of Atlas Commands

The order in which statements occur in an Atlas input file is important. There are five groups of statements that must occur in the correct order (see Figure 2-3). Otherwise, an error message will appear, which may cause incorrect operation or termination of the program. For example, if the material parameters or models are set in the wrong order, then they may not be used in the calculations.

The order of statements within the mesh definition, structural definition, and solution groups is also important. Otherwise, it may also cause incorrect operation or termination of the program.

.

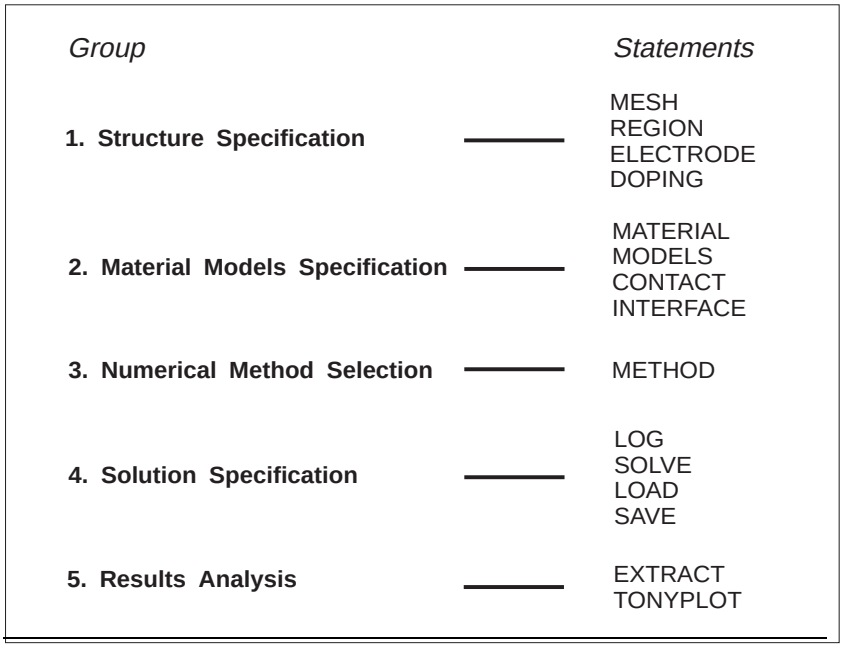


Figure 2-3 Atlas Command Groups with the Primary Statements in each Group

2.5.3 The DeckBuild Command Menu

The DeckBuild Command Menu (Command Menu) can help you to create input files. This menu is found under the **Commands** button on DeckBuild's main screen. The Commands Menu is configured for Atlas whenever Atlas is the currently active simulator in DeckBuild. When Atlas is active, which is indicated in the lower bar of the DeckBuild Window, an Atlas command prompt will appear in the DeckBuild output section.

The Command Menu gives you access to pop-up windows where you type information. When you select the **Write** button, syntactically correct statements are written to the DeckBuild text edit region. The DeckBuild Command Menu does not support all possible Atlas syntax, but aims to cover the most commonly used commands.

2.5.4 PISCES-II Quick Start

This section is a quickstart for those who may be familiar with the syntax and use of the Stanford University PISCES-II program or other device simulators derived from this program.

The major differences between Atlas and PISCES-II are:

- all graphics are handled by a separate interactive graphics program, TonyPlot. By using TonyPlot, you no longer need to run the device simulator simply to plot or alter graphics.
- no need to separate individual Atlas simulations into separate input files. Multiple runs of Atlas are possible in the same input file separated by the line `go atlas`. There's also no need to separate process and device simulation runs of Silvaco products into separate input files. A single file containing Athena and Atlas syntax is permitted in DeckBuild.
- the interface from process to device simulation is handled though a single file format compatible with other programs. The file read by Atlas is the default output file format of Athena. No special file format for the interface is required.
- when defining a grid structure within Atlas, the `NODE` and `LOCATION` syntax to define exact grid line numbers in X and Y is not recommended. A more reliable and easier to use syntax using `LOCATION` and `SPACING` is available.
- using the `REGRID` command is not recommended due to the creation of obtuse triangles. A standalone program, such as DevEdit, can be used as a grid pre-processor for Atlas.
- all numerical method selection commands and parameters are on the `METHOD` statement. The `SYMBOLIC` statement is not used. Historically, `SYMBOLIC` and `METHOD` were used as a coupled pair of statements, but it is more convenient to use a single statement (`METHOD`) instead. Most of the old parameters of the `SYMBOLIC` statement have the same meaning and names, despite this move to a single statement. One notable change in Atlas is that you can combine numerical methods together.

  See the "Pisces-II Compatibility" on page 83 for more information concerning the translation of PISCES-II numerics statements.
- various general purpose commands are actually part of the DeckBuild user environment. These include `SET`, `EXTRACT`, `GO`, `SYSTEM`, and `SOURCE`. These commands can be interspersed inside Atlas syntax.
- Variable substitution is supported for both numerical and string variables using the `SET` statement and the `$` symbol. To avoid confusion, the # symbol is preferred over the `$` symbol for comment statements.

In addition to these changes, the physical models are generally different in Atlas. Most of the original PISCES-II models have been preserved but often are not the default or the recommended models to use. See the on-line examples for technology specific information about models.

2.6 Defining A Structure

There are three ways to define a device structure in Atlas.

The first way is to read an existing structure from a file. The structure is created either by an earlier Atlas run or another program such as Athena or DevEdit. A **MESH** statement loads in the mesh, geometry, electrode positions, and doping of the structure. For example:

```
MESH INFILE=<filename>
```

The second way is to use the **Automatic Interface** feature from DeckBuild to transfer the input structure from Athena or DevEdit.

The third way is create a structure by using the Atlas command language. See Chapter 22 "Statements" for more information about the Atlas syntax.

2.6.1 Interface From Athena

When Athena and Atlas are run under DeckBuild, you can take advantage of an automatic interface between the two programs. Perform the following steps to load the complete mesh, geometry, and doping from Athena to Atlas.

1. Deposit and pattern electrode material in Athena.
2. Use the **ELECTRODE** statement in Athena to define contact positions. Specify the X and Y coordinates as cross-hairs to pin-point a region. The whole region is then turned into electrode. In many cases, only the X coordinate is needed. For example:

   ```
   ELECTRODE NAME=gate X=1.3 [Y=-0.1])
   ```

   There is a special case to specify a contact on the bottom of the structure. For example:

   ```
   ELECTRODE NAME=substrate BACKSIDE
   ```
3. Save a structure file while Athena is still the active simulator. For example:

   ```
   STRUCTURE OUTF=nmos.str
   ```
4. Start Atlas with the `go atlas` command written in the same input deck. This will automatically load the most recent structure from Athena into Atlas.

If you need to load the structure saved in step 4 into Atlas without using the auto-interface capability, use the **MESH** command. For example:

```
MESH INF=nmos.str
```

Atlas inherits the grid used most recently by Athena. With a careful choice of initial mesh or by using the grid manipulation techniques in Athena, you can produce a final mesh from Athena that will give good results in Atlas. But, a grid that is appropriate for process simulation isn't always appropriate for device simulation. If the final Athena mesh is inappropriate for Atlas, use either DevEdit to re-mesh the structure or the **REGRID** command.

Note: There's no need to specify a **MESH** command in Atlas when using the Automatic Interface of DeckBuild.

2.6.2 Interface From DevEdit

A 2D or 3D structure created by DevEdit can be read into Atlas using the following statement.

```
MESH INF=<structure filename>
```

This statement loads in the mesh, geometry, electrode positions, and doping of the structure. Atlas will automatically determine whether the mesh is 2D for S-PISCES or Blaze, or 3D for Device 3D or Blaze3D.

If the structure coming from DevEdit were originally created by Athena, then define the electrodes in Athena as described in the previous section. If the structure is created in DevEdit, the electrode regions should be defined in the **Region/Add region** menu of DevEdit.

2.6.3 Using The Command Language To Define A Structure

To define a device through the Atlas command language, you must first define a mesh. This mesh or grid covers the physical simulation domain. The mesh is defined by a series of horizontal and vertical lines and the spacing between them. Then, regions within this mesh are allocated to different materials as required to construct the device. For example, the specification of a MOS device requires the specification of silicon and silicon dioxide regions. After the regions are defined, the location of electrodes is specified. The final step is to specify the doping in each region.

When using the command language to define a structure, the information described in the following four sub-sections must be specified in the order listed.

Specifying The Initial Mesh

The first statement must be:

```
MESH SPACE.MULT=<VALUE>
```

This is followed by a series of X.MESH and Y.MESH statements.

```
X.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
Y.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
```

The SPACE.MULT parameter value is used as a scaling factor for the mesh created by the X.MESH and Y.MESH statements. The default value is 1. Values greater than 1 will create a globally coarser mesh for fast simulation. Values less than 1 will create a globally finer mesh for increased accuracy. The X.MESH and Y.MESH statements are used to specify the locations in microns of vertical and horizontal lines, respectively, together with the vertical or horizontal spacing associated with that line. You must specify at least two mesh lines for each direction. Atlas automatically inserts any new lines required to allow for gradual transitions in the spacing values between any adjacent lines. The X.MESH and Y.MESH statements must be listed in the order of increasing x and y. Both negative and positive values of x and y are allowed.

Figure 2-4 illustrates how these statements work. On the left hand plot, note how the spacing of the vertical lines varies from 1 μm at x=0 and x=10 μm to 0.5 μm at x=5 μm. On the right hand plot, note how specifying the SPACE.MULT parameter to have a value of 0.5 has doubled the density of the mesh in both the X and Y directions.

You can specify the PERIODIC parameter on the **MESH** statement to mean that the structure and mesh are periodic in the x direction.

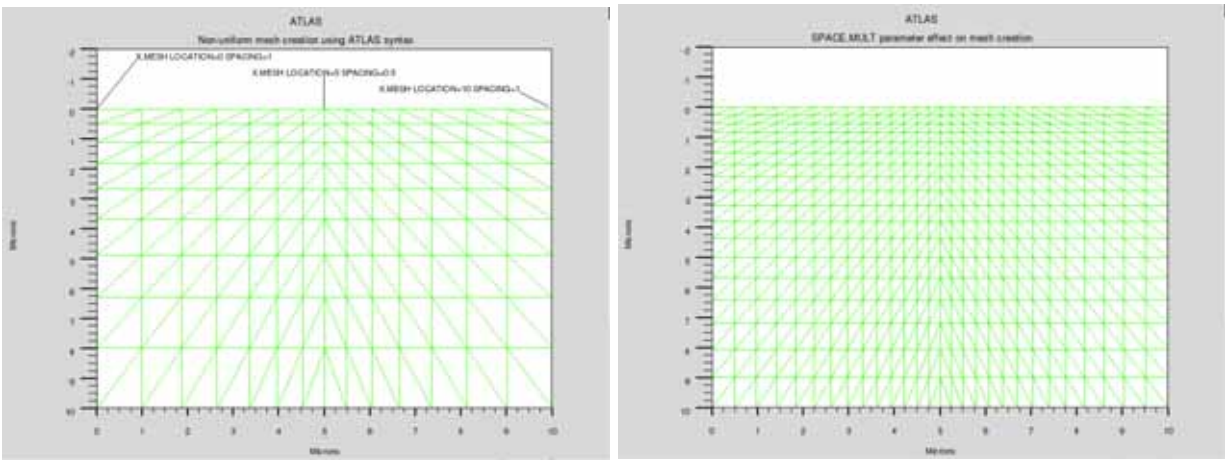


Figure 2-4 Non-uniform Mesh Creation using Atlas Syntax

After an initial mesh has been defined, you can remove grid lines in specified regions. This is typically done in regions of the device where a coarse grid is expected to be sufficient such as the substrate. The removal of grid lines is accomplished using the **ELIMINATE** statement. The **ELIMINATE** statement removes every second mesh line in the specified direction from within a specified rectangle. For example, the statement:

```
ELIMINATE COLUMNS X.MIN=0 X.MAX=4 Y.MIN=0.0 Y.MAX=3
```

removes every second vertical grid line within the rectangle bounded by x=0, x=4, y=0 and y=3 microns.

Specifying Regions And Materials

Once the mesh is specified, every part of it must be assigned a material type. This is done with **REGION** statements. For example:

```
REGION number=<integer> <material_type> <position parameters>
```

Region numbers must start at 1 and are increased for each subsequent region statement. You can have up to 15000 different regions in Atlas. A large number of materials is available. If a composition-dependent material type is defined, the x and y composition fractions can also be specified in the **REGION** statement.

The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. If the position parameters of a new statement overlap those of a previous **REGION** statement, the overlapped area is assigned as the material type of the new region. Make sure that materials are assigned to all mesh points in the structure. If this isn't done, error messages will appear and Atlas won't run successfully.

You can use the **MATERIAL** statement to specify the material properties of the defined regions. But you must complete the entire mesh and doping definition before any **MATERIAL** statements can be used. The specification of material properties is described in Section 2.7.2 "Specifying Material Properties".

Cylindrical Coordinates

Cylindrical coordinates are often used when simulating discrete power devices. In this mode, Atlas operates with x=0 as the axis of symmetry around which the cylindrical geometry is placed. Many of the default units change when cylindrical coordinates are used. The calculated current is in Amps rather than the usual Amps per micron. External elements are specified in absolute units (e.g., Farads, not Farads/micron for capacitors).

The **MESH** statement must be used to specify cylindrical symmetry. The following statement creates a mesh, which contains cylindrical symmetry.

```
MESH NX=20 NY=20 CYLINDRICAL
```

There are 20 mesh nodes along the X axis and 20 mesh nodes along the Y axis.

The following statement imports a mesh, which contains cylindrical symmetry.

```
MESH INF=mesh0.str CYLINDRICAL
```

Note: The CYLINDRICAL parameter setting isn't stored in mesh files. Therefore, this parameter must be specified each time a mesh file, which contains cylindrical symmetry, is loaded.

Specifying Electrodes

Once you have specified the regions and materials, define at least one electrode that contacts a semiconductor material. This is done with the **ELECTRODE** statement. For example:

```
ELECTRODE NAME=<electrode name> <position_parameters>
```

You can specify up to 50 electrodes. The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. Multiple electrode statements may have the same electrode name. Nodes that are associated with the same electrode name are treated as being electrically connected.

Some shortcuts can be used when defining the location of an electrode. If no Y coordinate parameters are specified, the electrode is assumed to be located on the top of the structure. You also can use the RIGHT, LEFT, TOP, and BOTTOM parameters to define the location. For example:

```
ELECTRODE NAME=SOURCE LEFT LENGTH=0.5
```

specifies the source electrode starts at the top left corner of the structure and extends to the right for the distance LENGTH.

Specifying Doping

You can specify analytical doping distributions or have Atlas read in profiles that come from either process simulation or experiment. You specify the doping using the **DOPING** statement. For example:

```
DOPING <distribution_type> <dopant_type> <position_parameters>
```

Analytical Doping Profiles

Analytical doping profiles can have uniform, gaussian, or complementary error function forms. The parameters defining the analytical distribution are specified in the **DOPING** statement. Two examples are shown below with their combined effect shown in Figure 2-5.

```
DOPING UNIFORM CONCENTRATION=1E16 N.TYPE REGION=1

DOPING GAUSSIAN CONCENTRATION=1E18 CHARACTERISTIC=0.05 P.TYPE \
        X.LEFT=0.0 X.RIGHT=1.0 PEAK=0.1
```

The first **DOPING** statement specifies a uniform n-type doping density of 10^{16} cm^{-3} in the region that was previously labelled as region #1. The position parameters X.MIN, X.MAX, Y.MIN, and Y.MAX can be used instead of a region number.

The second **DOPING** statement specifies a p-type Gaussian profile (see Equation 22-3) with a peak concentration of 10^{18} cm^{-3}. This statement specifies that the peak doping is located along a line from x = 0 to x = 1 microns. Perpendicular to the peak line, the doping drops off according to a Gaussian distribution with a standard deviation of $(0.05/\sqrt{2})$ µm. At $x < 0$ or $x > 1$, the doping drops off laterally with a default standard deviation that is $(70/\sqrt{2})$% of CHARACTERISTIC. This lateral roll-off can be altered with the RATIO.LATERAL parameter. If a Gaussian profile is being added to an area that was already defined with the opposite dopant type, you can use the JUNCTION parameter to specify the position of the junction depth instead of specifying the standard deviation using the CHARACTERISTIC parameter.

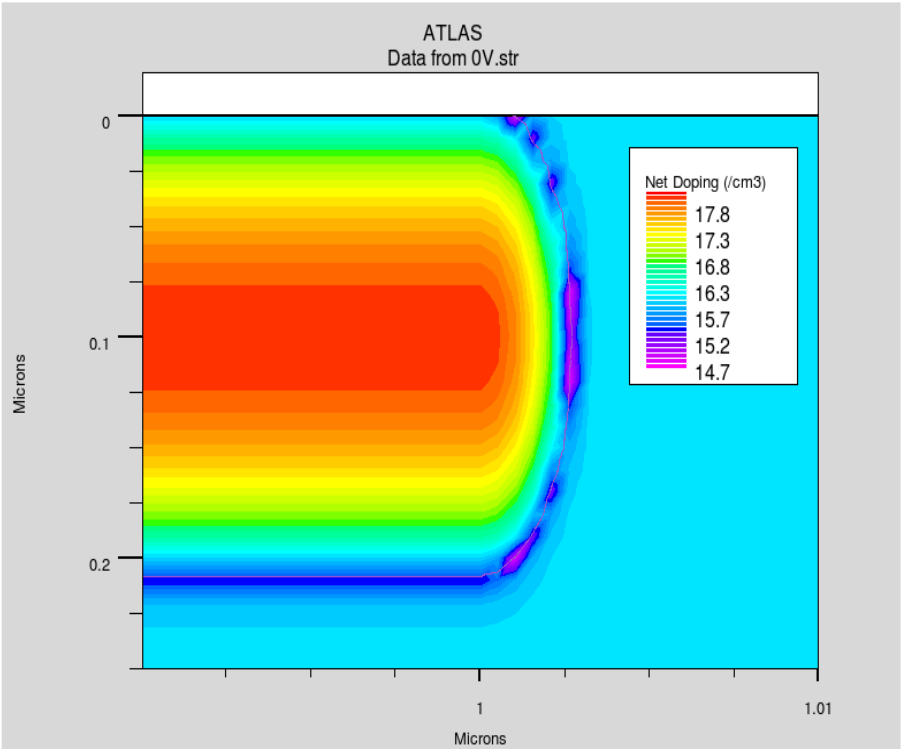


Figure 2-5 Analytical specification of a 2D Profile

The other analytical doping profile available is the complementary error function. This is defined as

$$erfc(z) = \frac{2}{\sqrt{\pi}}\int_{z}^{\infty} exp(-y^2)dy \qquad 2\text{-}1$$

where the z variable is the distance scaled by the characteristic distance defined by the CHAR parameter.

The following example show **DOPING** statements that use this analytical form.

```
DOPING ERFC N.TYPE PEAK=0.5 JUNCTION=1.0 CONC=1.0E19 X.MIN=0.25 \
       X.MAX=0.75 RATIO.LAT=0.3 ERFC.LAT
DOPING P.TYPE CONC=1E18 UNIFORM
```

This sets up a donor profile with a peak concentration of 1.0E19 cm^{-3} at X = 0.5 microns. The CHAR parameter, which determines the rate of change of the doping level with distance, is not directly set on the **DOPING** profile. Instead, it is calculated so that the net doping level at the position given by the JUNCTION parameter is zero. In this example, the acceptor concentration is 1.0×10^{18} cm^{-3} everywhere and so we require a donor density of 1.0×10^{18} cm^{-3} at a position of 1 micron to create the p-n junction there. The value of CHAR is calculated from the formula

$$erfc([\mathtt{JUNCTION} - \mathtt{PEAK}]/\mathtt{CHAR}) = 0.1 \qquad 2\text{-}2$$

which results in a value of CHAR of approximately 0.43 microns.

Additionally, the donor concentration falls off in the lateral direction outside the range of 0.25 to 0.75 microns. The lateral falloff parameter is defined to be 0.3 times the principal falloff parameter and has the shape of the complementary error function.

Importing 1D SSUPREM3 Doping Profiles

One-dimensional doping profiles can be read into Atlas from a SSuprem3 output file. The doping data must have been saved from SSuprem3 using the statement:

```
STRUCTURE OUTFILE=<output filename>
```

at the end of the SSuprem3 run.

In Atlas, the MASTER parameter of the **DOPING** statement specifies that a SSuprem3 file will be read by Atlas. Since this file will usually contain all the dopants from the SSuprem3 simulation, the desired dopant type must also be specified. For example, the statement:

```
DOPING MASTER INFILE=mydata.dat BORON REGION=1
```

specifies that the boron profile from the file mydata.dat should be imported and used in region #1. SSUPREM3 profiles are imported into Atlas one at a time (i.e., one DOPING statement is used for each profile or dopant). The statements:

```
DOPING MASTER INFILE=mydata.dat BORON OUTFILE=doping.dat
DOPING MASTER INFILE=mydata.dat ARSENIC X.RIGHT=0.8 RATIO=0.75
DOPING MASTER INFILE=mydata.dat ARSENIC X.LEFT=2.2 RATIO=0.75
```

offset the arsenic doping from boron to create a 2D doping profile from a single SSuprem3 result.

It is advisable to include the OUTFILE parameter on the first **DOPING** statement to create a 2D3D doping file. This file will then be used in the next section to interpolate doping on a refined mesh after a **REGRID**. This file, however, can't be plotted in TonyPlot. The position parameters and the RATIO.LATERAL parameter are used in the same manner as for analytical doping profiles to set the extent of the 1D profile.

2.6.4 Automatic Meshing (Auto-meshing) Using The Command Language

Automatic meshing provides a simpler method for defining device structures and meshs than the standard method described in Section 2.6.3 "Using The Command Language To Define A Structure". Auto-meshing is particularly suited for epitaxial structures, especially device structures with many layers (for example, a VCSEL device). Auto-meshing unburdens you from the delicate bookkeeping involved in ensuring that the locations of mesh lines in the Y direction are consistently aligned with the edges of regions. This is done by specifying the locations of Y mesh lines in the **REGION** statements. The following sections will show how auto-meshing is done, using a few simple examples.

Specifying The Mesh And Regions

In the first example, we use a simple device to show you the fundamental concepts of auto-meshing. The first statements in this example are as follows:

```
MESH AUTO

X.MESH LOCATION=-1.0 SPACING=0.1

X.MESH LOCATION=1.0  SPACING=0.1
```

These statements are similar to the ones used in the standard method that described a mesh using the command language as shown in Section 2.6.3 "Using The Command Language To Define A Structure". There are, however, two key differences. The first difference is the inclusion of the AUTO parameter in the **MESH** statement. You need this parameter to indicate that you want to use auto-meshing. The second and more important difference is that in this example we will not specify any Y.MESH statements. This is because the locations of Y mesh lines will be automatically determined by the parameters of the **REGION** statements.

You can still specify one or more Y.MESH statements. Such defined mesh lines will be included in the mesh. But including Y.MESH statements is optional in auto-meshing.

In the next few statements of the example, we will show you several new concepts that will explain how auto-meshing works. The following four lines describe the regions in the example device:

```
REGION TOP    THICKNESS=0.02 MATERIAL=GaN   NY=5  DONOR=1E16

REGION  BOTTOM  THICKNESS=0.1    MATERIAL=AlGaN  NY=5    DONOR=1E17
X.COMP=0.2

REGION  TOP     THICKNESS=0.08  MATERIAL=AlGaN  NY=4   ACCEPTOR=1E17
X.COMP=0.2

REGION  BOTTOM  THICKNESS=0.5    MATERIAL=AlGaN  NY=10  DONOR=1E18
X.COMP=0.2
```

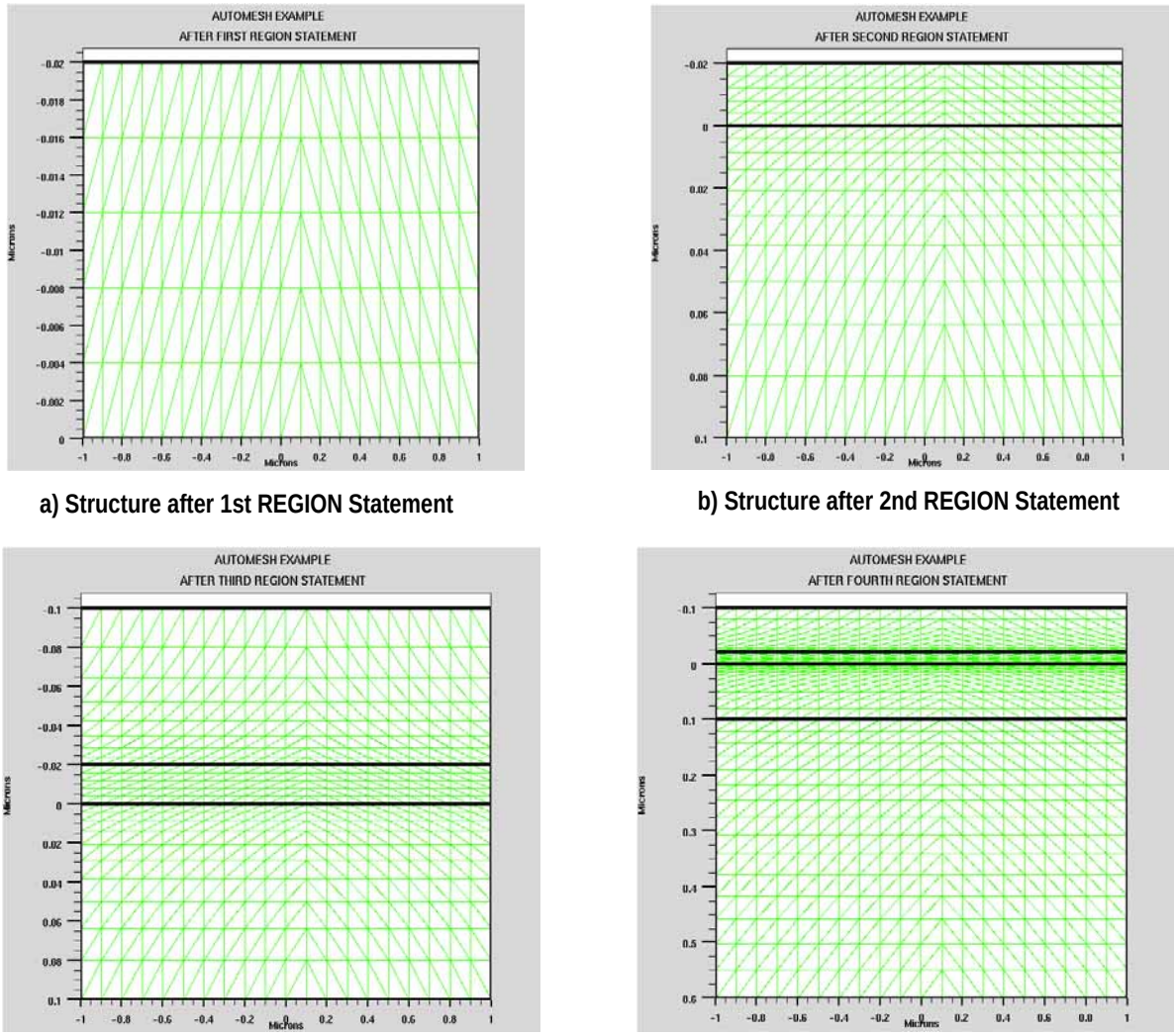
New Concepts

First, it appears that composition and doping are being specified in the **REGION** statement. This is the case for the DONOR, ACCEPTOR, X.COMPOSITION, and Y.COMPOSITION parameters in the **REGION** statement that specify uniform doping or composition or both over the specified region. These parameters are also available to the standard methods described in Section 2.6.3 "Using The Command Language To Define A Structure" but are more amenable to specification of epitaxial structures such as we are describing in this example. Next, you should notice several other new parameters. These are the TOP, BOTTOM, THICKNESS, and NY parameters. All of these are used to describe the relative locations and thicknesses of the layers as well as the locations of the Y mesh lines. The most intuitive of these parameters is the THICKNESS parameter, which describes the thickness in microns, in the Y direction of each layer. As for the extent in the X direction, in the absence of any specified X.MIN or X.MAX parameters, it is assumed that the region extends over the full range of the X mesh described above in the X.MESH statements.

The NY parameter describes how many Y mesh lines are contained in the region so that the Y mesh lines are evenly spaced over the region. You can use the SY parameter instead of NY to specify the spacing in microns between Y mesh lines in the region. Make sure the value of SY does not exceed the value of THICKNESS. Generally, the relationship between SY, NY and THICKNESS can be expressed by the following:

```
SY = THICKNESS/NY
```

Figure 2-6 shows the meanings of the TOP and BOTTOM parameters.



a) Structure after 1st REGION Statement

b) Structure after 2nd REGION Statement

c) Structure after 3rd REGION Statement

d) Structure after 4th REGION Statement

Figure 2-6 Simple Example of Auto-meshing Showing Sequence of Structure Development

This figure shows a progression of representations of how the structure's mesh and region outlines appear after each **REGION** statement. This figure should give you an intuitive feel of how the regions are alternately placed on the top or bottom of the structure according to the specification of the TOP or BOTTOM parameters. It is important to keep in mind that the Atlas coordinate convention for the Y axis is that positive Y is directed down into the device. This is similar to using the TOP and BOTTOM parameters of the **ELECTRODE** statement.

One thing you might notice in this figure is that the number of Y mesh lines in each region does not always match the number specified. This is because at each interface between regions, the Y mesh line spacing is ambiguously defined and the auto-meshing algorithm will always pick the smaller spacing between the two at each interface. Then, the spacing between Y mesh lines varies continuously between subsequent interfaces in a similar way as it does for mesh spacings specified by the LOCATION and SPACING parameters in the X.MESH and Y.MESH statements.

The auto-meshing algorithm maintains the "notion" of the current Y locations of the "top" and "bottom". Let's call these locations "Ytop" and "Ybottom".

Before any **REGION** statements are processed, "Ytop" and "Ybottom" are both defined to be equal to zero. As the **REGION** statements are processed, the following cases are addressed:

- If you place the region on the top, as specified by the TOP parameter, the region will extend from "Ytop" to "Ytop"-THICKNESS (remember positive Y points down) and "Ytop" will move to the new location "Ytop"-THICKNESS.
- If you place the region on the bottom, as specified by the BOTTOM parameter, the region will extend from "Ybottom" to "Ybottom"+THICKNESS and "Ybottom" will move to the new location "Ybottom"+THICKNESS.

The auto-meshing algorithm will ensure that all regions are perfectly aligned to their neighboring regions and there are no inconsistencies between the locations of Y mesh lines and the region edges that they resolve.

Non-uniformity In The X Direction and Auto-meshing

In some cases, you may want to define a device with material discontinuities in the X direction. Such discontinuities may represent etched mesa structures or oxide apertures. There are a couple of ways to do this in auto-meshing. For example, you want to etch out a region from the structure of the previous example, from `X=0` to the right and from `Y=0` to the top. You can add the statement

```
REGION MATERIAL=Air X.MIN=0.0 Y.MAX=0.0
```

In this statement, we are not using the auto-meshing features but are using syntax of the standard meshing methods described in Section 2.6.3 “Using The Command Language To Define A Structure”. Atlas supports mixing the standard syntax with auto-meshing but be careful when doing this as we will explain shortly. Figure 2-7 shows the resulting structure and mesh after adding this statement. In this figure, we see that we have obtained the desired result.

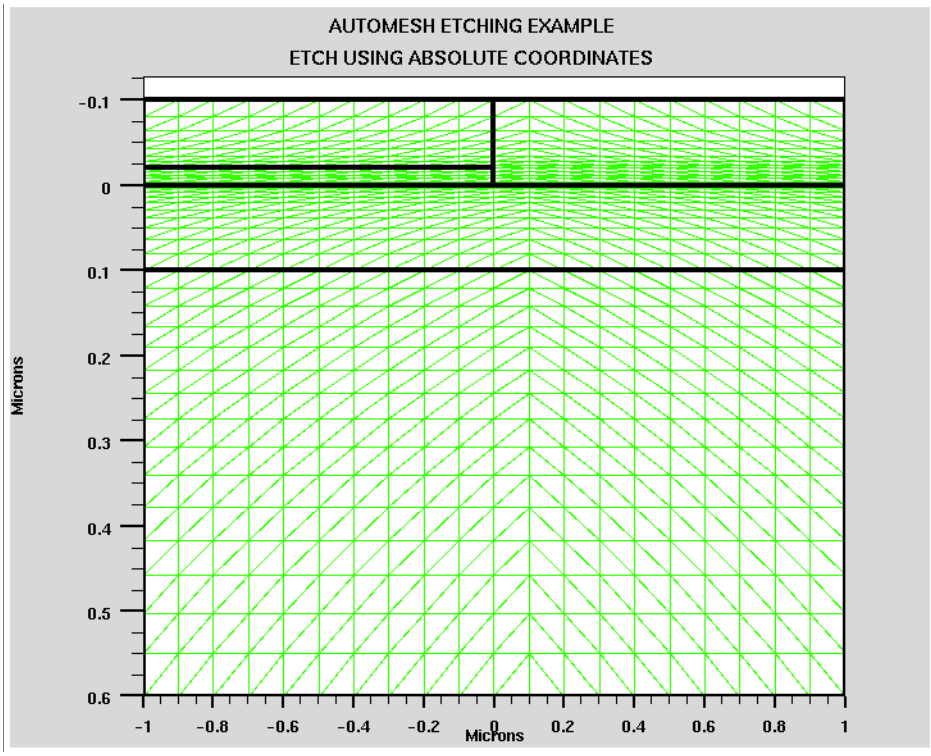


Figure 2-7 Structure Etch Example in Auto-meshing

The potential pitfall in using absolute Y coordinates in auto-meshing is that the location you choose, for example, by summing up thicknesses may not match the location your computer has calculated with its inherent numerical imprecision. The result is not only that the resulting structure may not exactly match what you desire. More importantly, you may end up accidently creating a mesh with Y mesh lines closely spaced (usually in the order of the machine precision). This can cause numerical instability (poor convergence) and can overflow or underflow in certain models. What’s worse is that this situation is difficult to detect.

There is, however, one situation where we can absolutely predict the location of an edge or Y mesh line. That is at Y=0. This is exactly what we have done in our example. So if you want to use auto-meshing with specifications of an absolute value of Y, then arrange your device structure so that the specification of Y will be at zero.

This also applies to the location of a recessed electrode. Make sure it is located at Y=0 if you are using auto-meshing.

There is another method of providing for discontinuities in material in the X direction that is absolutely safe from the previously discussed problems. In this approach, we use another parameter called STAY in the **REGION** statement in conjunction with the TOP or BOTTOM parameters.

The following describes the effect of the STAY parameter in the **REGION** statement.

- If you place the region on the top, as specified by the TOP parameter, and the STAY parameter is specified, the region will extend from "Ytop" to "Ytop"-THICKNESS and "Ytop" will remain in its current position.
- If you place the region on the bottom, as specified by the BOTTOM parameter, and the STAY parameter is specified, the region will extend from "Ybottom" to "Ybottom"+THICKNESS and "Ybottom" will remain in its current position.

The use of the STAY parameter can best be illustrated by the following example. In this example, we will reproduce the same structure discussed in the last example but this time using only STAY parameters and by not using any specification of absolute Y coordinates. The new **REGION** specifications are as follows:

```
REGION  BOTTOM  THICKNESS=0.1    MATERIAL=AlGaN  NY=5    DONOR=1E17
X.COMP=0.2

REGION  BOTTOM  THICKNESS=0.5    MATERIAL=AlGaN  NY=10  DONOR=1E18
X.COMP=0.2

REGION TOP STAY  THICKNESS=0.02 MATERIAL=GaN   NY=5  DONOR=1E16

REGION TOP       THICKNESS=0.02 MATERIAL=Air   NY=5  X.MIN=0.0

REGION TOP STAY THICKNESS=0.08 MATERIAL=AlGaN NY=4   ACCEPTOR=1E17
X.COMP=0.2

REGION TOP      THICKNESS=0.08 MATERIAL=Air   NY=4  X.MIN=0.0
```

In this example, we slightly rearranged the **REGION** statements for clarity and split one region into two. Figure 2-8 shows the results.

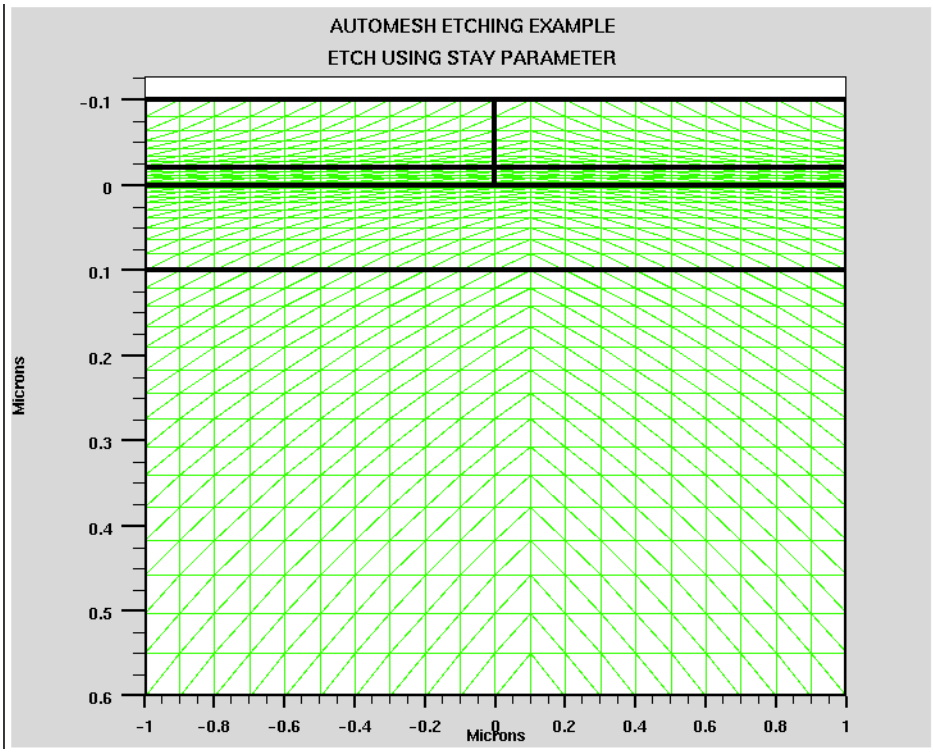


Figure 2-8 Structure Etching Example Using The STAY Parameter

The following describes the operation of the STAY parameter in this example:

- The STAY parameter in the third **REGION** statement means that the fourth REGION will start at the same Y coordinate as the third.
- Since the THICKNESS and NY parameters are the same in the third and fourth regions, they will have the same range of Y values and the same Y mesh.
- The specification of X.MIN in the fourth **REGION** statement means that the region will not completely overlap the third region but will only overlap to a minimum X value of 0.
- The lack of a STAY parameter in the fourth **REGION** statement means that the fifth region will lie directly atop the third and fourth regions.
- The STAY parameter in the fifth **REGION** statement means that the sixth **REGION** will start at the same Y coordinate as the fifth.
- Since the THICKNESS and NY parameters are the same in the fifth and sixth regions, they will have the same range of Y values and the same Y mesh.
- The specification of X.MIN in the sixth **REGION** statement means that the region will not completely overlap the fifth region but will only overlap to a minimum X value of 0.

As you can see, using the STAY parameter carefully avoids the problems of specifying values of Y coordinates and the potential problems involved. By doing so, you can specify arbitrary stepped structures.

Grading of Composition and Doping

We have provided another method for specifying composition or doping or both in the **REGION** statement that is available for both auto-meshing and meshing using the standard method described in Section 2.6.3 “Using The Command Language To Define A Structure”. This method allows linear grading of rectangular regions. Eight new parameters of the REGION statement support this function. They are ND.TOP, ND.BOTTOM, NA.TOP, NA.BOTTOM, COMPX.TOP, COMPX.BOTTOM, COMPY.TOP, and COMPY.BOTTOM. In the syntax of these parameter names, "TOP" refers to the "top" or extreme Y coordinate in the negative Y direction, and "BOTTOM" refers to the "bottom" or extreme Y coordinate in the positive Y direction. With this in mind, the following rules apply:

- If you specify ND.TOP and ND.BOTTOM in a REGION statement, the donor doping in the region will vary linearly from the value specified by ND.TOP at the "top" of the device to the value specified by ND.BOTTOM at the "bottom" of the device.
- If you specify NA.TOP and NA.BOTTOM in a REGION statement, the acceptor doping in the region will vary linearly from the value specified by NA.TOP at the "top" of the device to the value specified by NA.BOTTOM at the "bottom" of the device.
- If you specify COMPX.TOP and COMPX.BOTTOM in a REGION statement, the X composition fraction in the region will vary linearly from the value specified by COMPX.TOP at the "top" of the device to the value specified by COMPX.BOTTOM at the "bottom" of the device.
- If you specify COMPY.TOP and COMPY.BOTTOM in a REGION statement, the Y composition fraction in the region will vary linearly from the value specified by COMPY.TOP at the "top" of the device to the value specified by COMPY.BOTTOM at the "bottom" of the device.

Note: Any subsequent **DOPING** statements will add to the doping specified by the doping related parameters on the **REGION** statement.

Superlattices and Distributed Bragg Reflectors DBRs

For auto-meshing, we have provided a convenient way of specifying a certain class of superlattices or as they are commonly used distributed Bragg reflectors (DBRs). This class of superlattice includes any superlattice that can be described as integer numbers of repetitions of two different layers. By different, we mean that the two layers may have different thicknesses, material compositions, or dopings, or all. These "conglomerate" structures are specified by the **DBR** or SUPERLATTICE statement. Actually, SUPERLATTICE is a synonym for **DBR**. Therefore in the following discussion, we will use the **DBR** syntax and recognize that SUPERLATTICE and **DBR** can be used interchangeably.

Most of the syntax of the **DBR** statement is similar to the syntax of the **REGION** statement, except the syntax is set up to describe two regions (or two sets of regions). As you will see, we differentiate these regions (or sets of regions) by the indices 1 and 2 in the parameter’s name.

The following example should make this concept clear.

```
MESH AUTO
X.MESH LOCATION=-1.0 SPACING=0.1
X.MESH LOCATION=1.0  SPACING=0.1
```

```
DBR TOP LAYERS=4 THICK1=0.1 THICK2=0.2 N1=2 N2=3 MAT1=GaAs
MAT2=AlGaAs \

X2.COMP=0.4

DBR BOTTOM LAYERS=3 THICK1=0.05 THICK2=0.075 N1=3 N2=3 MAT1=AlGaAs
\

MAT2=GaAs X1.COMP=0.4
```

In this example, we can see we are using auto-meshing because of the appearance of the AUTO parameter in the **MESH** statement. The **DBR** statements should be familiar since the functionality and syntax are similar to the **REGION** statement. We will not discuss those similarities here. We will only discuss the important differences. For more information about the similarities, see Chapter 22 “Statements” .

The main new parameter is the LAYERS parameter. This parameter specifies the total number of regions added. The region indexed "1" is always added first, then the region indexed "2" is added, and depending on the value of LAYERS, the sequence of regions continues 1, 2, 1, 2, 1, ...

Figure 2-9 shows the functionality of the **DBR** statement, which gives the resulting structure and mesh from the example.

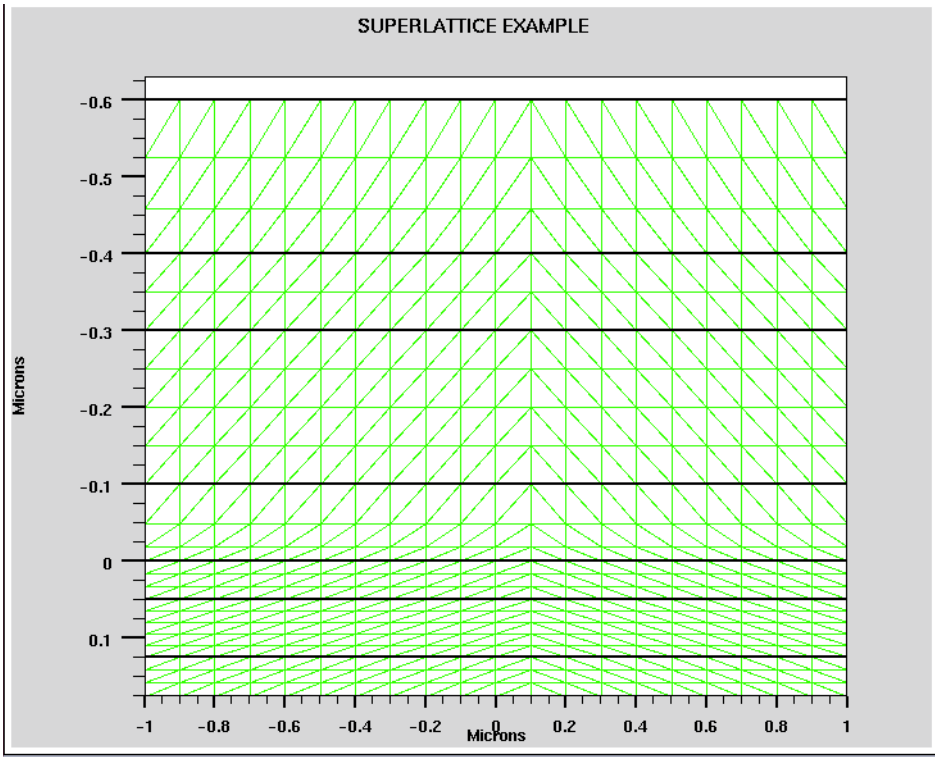


Figure 2-9 Supperlattice Example Structure

2.6.5 Modifying Imported Regions

If you import a device structure from a file using the INFILE parameter of the **MESH** statement, you may want to modify some of the characteristics of one or more of the regions in the structure. To do this, specify a **REGION** statement with the MODIFY parameter and the NUMBER or NAME parameter assigned to the region number of interest. You can specify/respecify any of the following **REGION** statement parameters: ACCEPTORS, DONORS, LED, MATERIAL, NAME, POLAR.SCALE, POLARIZATION, QWELL, STRAIN, VNBS.GAIN, WELL, WELL.FIELD, WELL.GAIN, X.COMP, and Y.COMP.

2.6.6 Remeshing Using The Command Language

It can be difficult to define a suitable grid when specifying a structure with the command language. The main problem is that the meshes required to resolve 2D doping profiles and curved junctions are quite complicated. Simple rectangular meshes require an excessive number of nodes to resolve such profiles. If a device structure only includes regions of uniform doping, then there's usually no need to regrid. But when realistic 2D doping profiles are present, a regrid may be necessary.

Note: The recommended solution for defining complex mesh structures for Atlas is to use the standalone program, DevEdit.

Regrid On Doping

Atlas includes a regridding capability that generates a fine mesh only in a localized region. You specify a quantity on which the regrid is to be performed. The mesh is then refined in regions where the specified quantity varies rapidly. Whenever a specified quantity (usually doping) changes quickly, the regridding will automatically grade the mesh accordingly. You can get a regrid on doping before any solutions are obtained. You can do this by using the statement:

```
REGRID LOGARITHM DOPING RATIO=2 SMOOTH.KEY=4 DOPFILE=<filename1> \
       OUTFILE=<filename2>
```

This statement must be used after the **MESH**, **REGION**, **MATERIAL**, **ELECTRODE**, and **DOPING** statements described previously. The effects of this **REGRID** statement on a simple diode structure are shown in Figure 2-10.

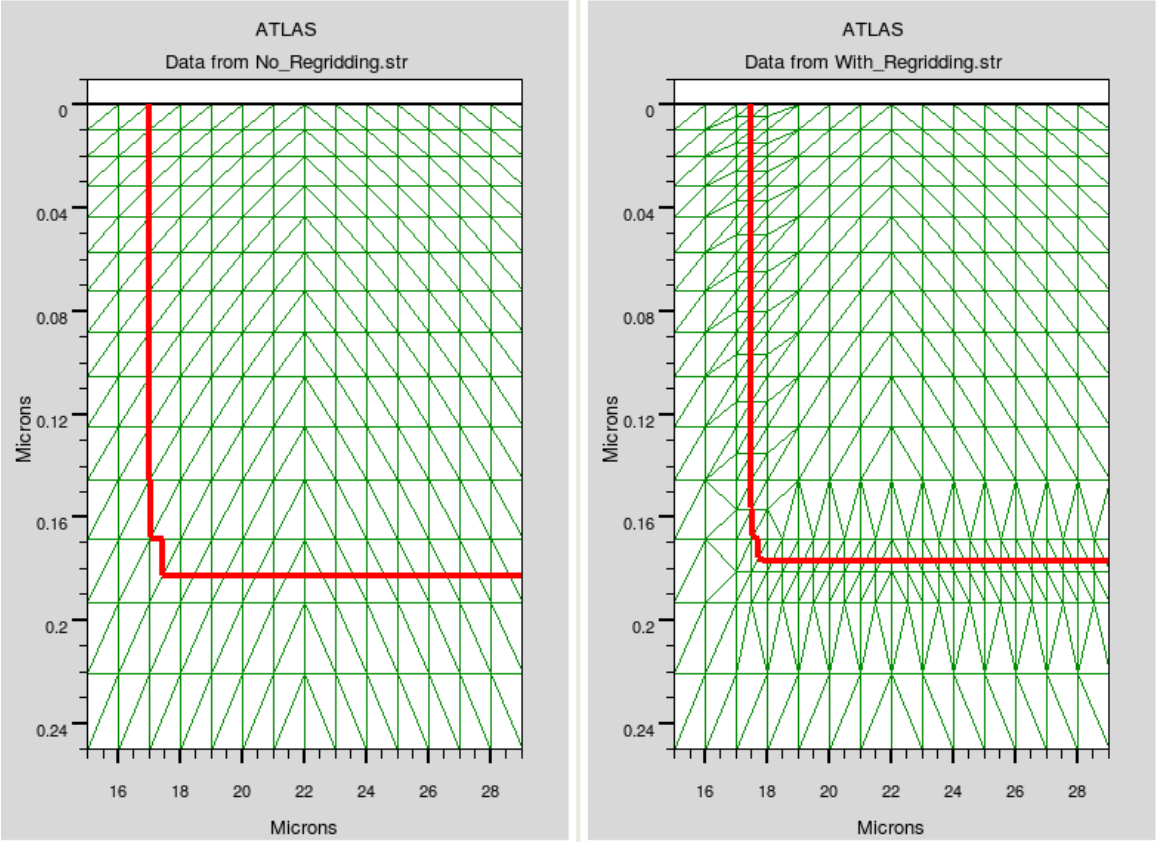


Figure 2-10 Regrid on doping provides improved resolution of junction

In this statement, the regridding will resolve doping profiles to two orders of magnitude in change.

The doping file, *filename1*, must be specified in the first **DOPING** statement with the OUTFILE parameter. The results of the regrid are saved in the file, *filename2*. The SMOOTH.KEY parameter value selects a smoothing algorithm. A value of 4 is typically best as this algorithm tends to produce the fewest obtuse triangles. For a complete description of the various smoothing algorithms, see Chapter 21 "Numerical Techniques".

Regrid Using Solution Variables

The **REGRID** statement can use a wide range of solution variables as the basis for mesh refinement. A regrid on solution variables can only be used after a solution has been obtained. After a regrid on a solution variable, the solution must be re-solved at the same bias in Atlas.

For example:

```
REGRID      POTENTIAL      RATIO=0.2      MAX.LEVEL=1      SMOOTH.K=4
DOPFILE=<filename1>

SOLVE PREV
```

Regrid on potential is often used for high voltage power devices.

Note: You can use the **REGRID** statement any number of times on a structure. But we advise you to quit and restart Atlas between regrids on electrical quantities. You can use the `go atlas` statement to do this. This should be followed by a **MESH** statement, loading the output file of the **REGRID** command, and a re-setting of all material and model parameters.

2.6.7 Specifying 2D Circular Structures

In some situations, it may be desirable to construct a device with a circular cross section. This is possible by using DevEdit and by using the **MESH** statement in the Atlas command language. Make sure to put the **MESH** statement first in the input deck and specify the `CIRCULAR` parameter.

The radial mesh spacing is then given by a series of `R.MESH` statements and the angular mesh given by a series of `A.MESH` statements. The angles are specified in degrees between 0 and 360.

For example

```
MESH CIRCULAR


R.MESH LOC=0.0  SPAC=0.05
R.MESH LOC=0.2  SPAC=0.05
R.MESH LOC=0.35 SPAC=0.025
R.MESH LOC=0.39 SPAC=0.01
R.MESH LOC=0.4  SPAC=0.01


A.MESH LOC=0   SPAC=30
A.MESH LOC=360 SPAC=30
```

would create a circular mesh with major angular spacing of 30° and a radial spacing, which is relatively coarse near to the origin and becomes finer towards the edge of the device.

The angular spacing defines a set of major spokes radiating out from the origin with the number of elements between each spoke increasing with distance from the origin. The mesh spacing may be irregular, although a regular mesh spacing suffices for most problems.

The spacing in the radial direction will in general be irregular. This can result in the creation of obtuse elements, particularly if the spacing decreases rapidly with increasing radius. The `MINOBTUSE` parameter on the **MESH** statement deploys an algorithm to reduce the occurence of obtuse elements, particularly obtuse boundary elements. It is set to true by default, but you can clear it by using the syntax `^MINOBTUSE`.

For some device structures, you can reduce the size of the solution domain by using symmetry arguments. Atlas allows you to create a wedge shaped device by one of two methods. In the first method, you specify more than one `A.MESH` statement, and the maximum angle specified in these statements is the angle of the wedge. In the second method, you only specify one `A.MESH` statement with a location of zero degrees and the required angular spacing. The `MAX.ANGLE` parameter specified on the `MESH` statement gives the angle of the wedge.

For example

```
MESH CIRCULAR
...
A.MESH LOC=0    SPAC=30
A.MESH LOC=150  SPAC=30
```

or alternatively

```
MESH CIRCULAR MAX.ANGLE=150
...
A.MESH LOC=0 SPAC=30
```

would both create an angular wedge with an angle at the origin of 150° and a major angular spacing of 30°. The R.MESH statements have been omitted for clarity.

Note: MAXANGLE should not exceed 180°.

Once a circular mesh has been created, it may be subdivided into regions using the R.MIN, R.MAX, A.MIN and A.MAX parameters on the REGION statement.

R.MIN is the inner radius of the region. R.MAX is the outer radius of the region. Both are in the units of microns. A.MIN and A.MAX define the minimum and maximum angular limits of the region respectively, both in degrees.

For example

```
REGION NUM=1 MATERIAL=SILICON A.MIN=0 A.MAX=360.0 R.MAX=0.4
REGION  NUM=2  MATERIAL=OXIDE  A.MIN=30  A.MAX=150.0  R.MIN=0.35
R.MAX=0.4
```

will enforce the area between angular limits of 30 and 150 degrees and radial limits of 0.35 and 0.4 microns to be an oxide region, and the rest of the mesh defined above to be a silicon region.

Similarly, the **ELECTRODE**, **DOPING**, and **INTERFACE** statements have R.MIN, R.MAX, A.MIN, and A.MAX parameters. Support for doping of circular meshes is extended from that described in Section 2.6.3 “Using The Command Language To Define A Structure” to allow analytical doping profiles in the radial direction (UNIFORM, GAUSS, ERFC) with optional lateral fall off in the angular (i.e., at constant radius) direction. The lateral falloff is GAUSSIAN unless ERFC.LAT is specified to change it to the complementary error function. The usual parameters for specifying the analytical profile apply with the principal direction being the radial direction.

If you set the maximum angle to 180° and add the following lines, you will obtain the structure as shown in Figure 2-11.

```
ELECTRODE NAME=DRAIN  R.MIN=0.35 A.MIN=0.0 A.MAX=30.0
ELECTRODE NAME=SOURCE R.MIN=0.35 A.MIN=150.0 A.MAX=180.0
ELECTRODE NAME=GATE   A.MIN=60.0 A.MAX=120.0 R.MIN=0.39 R.MAX=0.4
```

If you further add the DOPING statement below, then you will obtain the doping profile as shown in Figure 2-12.

```
DOPING GAUSS N.TYPE CONC=1.0e19 CHAR=0.05 R.MIN=0.15 R.MAX=0.2
A.MIN=0.0 A.MAX=180.0
```

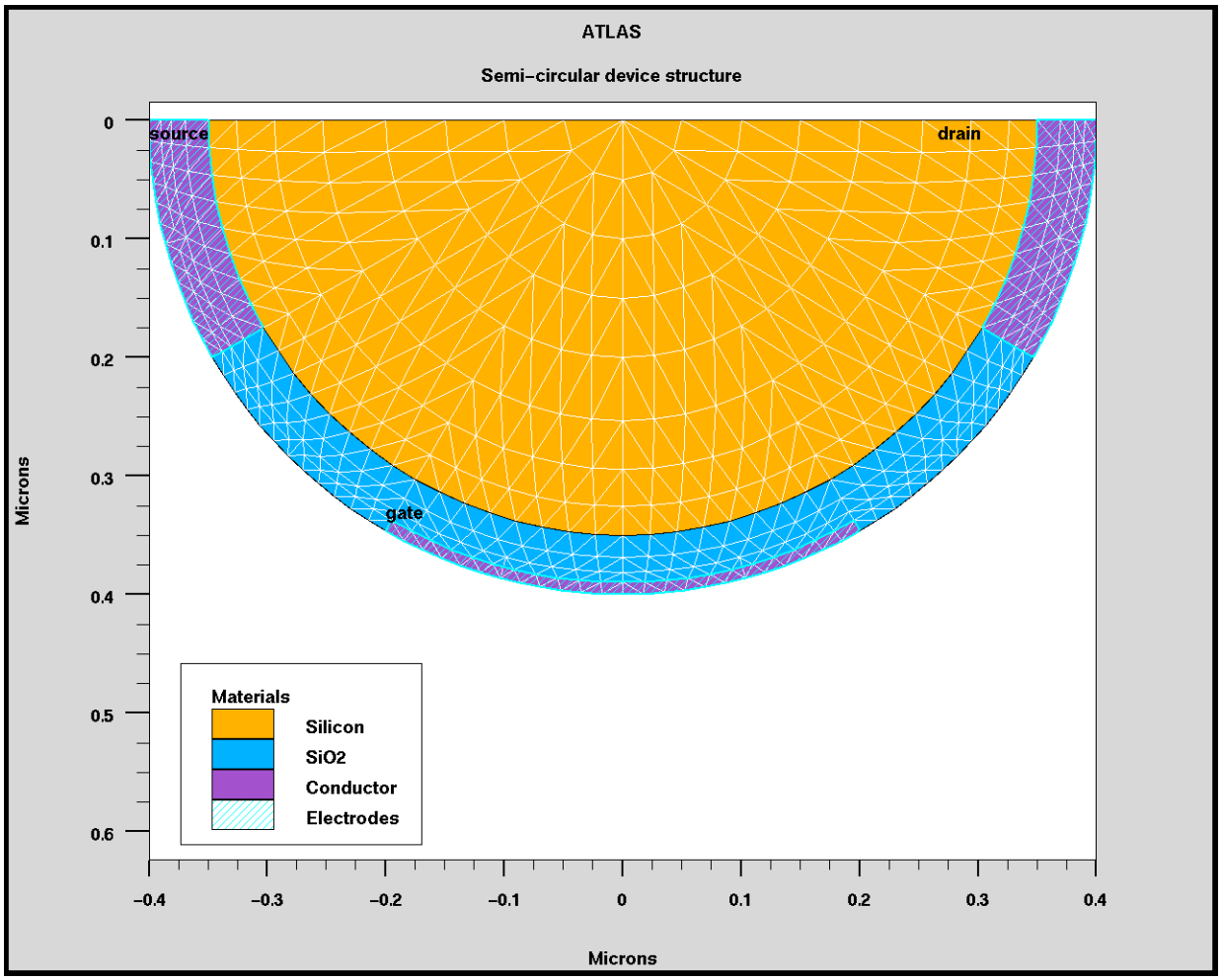


Figure 2-11 Semi-circular structure created using Atlas command syntax

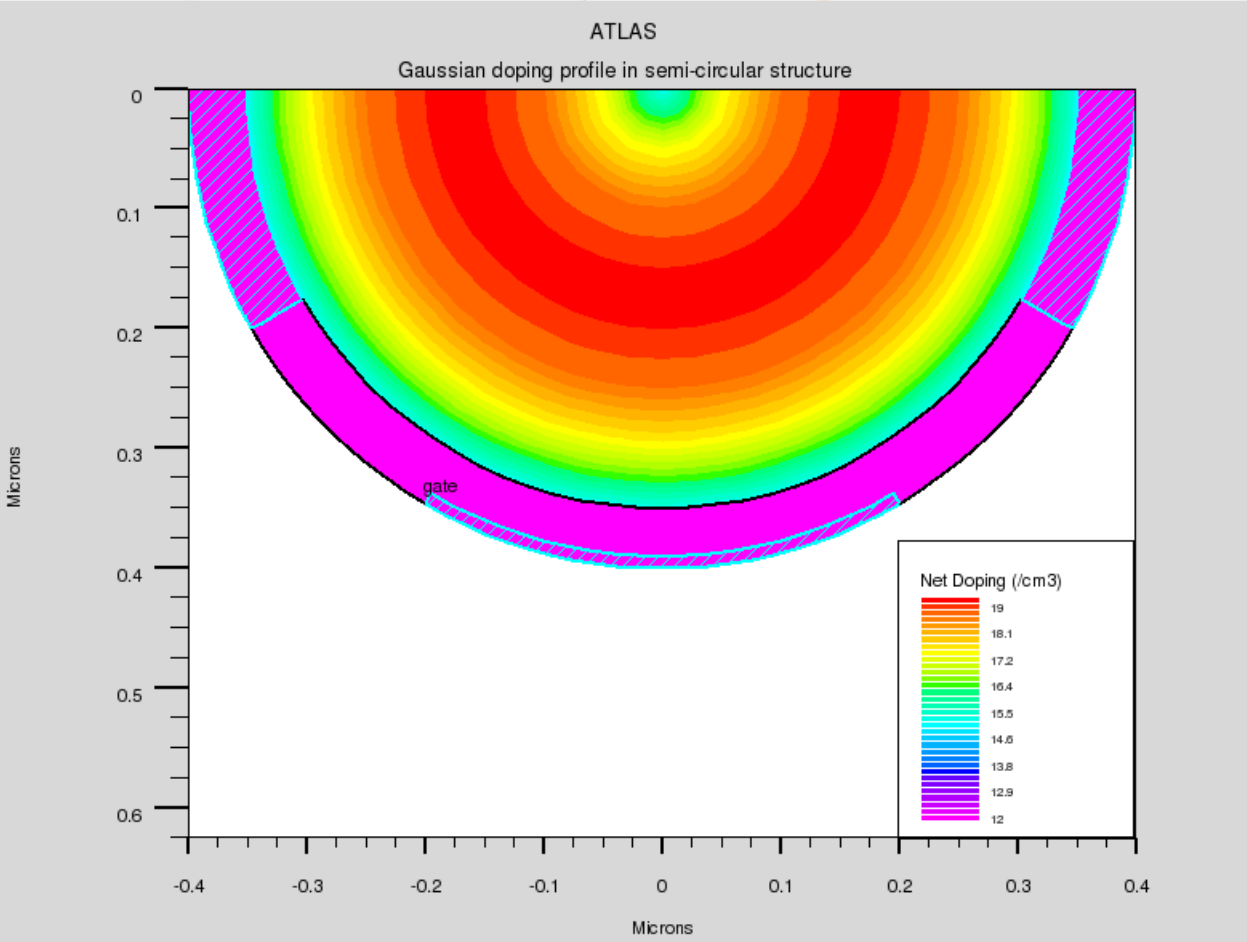


Figure 2-12 Gaussian doping profile applied to the semi-circular structure of Figure 2-11

Note: To obtain the best results, align region and electrode boundaries with existing meshlines (radial or major spokes).

2.6.8 Specifying 3D Structures

The syntax for forming 3D device structures is an extension of the 2D syntax described in the previous section. The **MESH** statement should appear as:

```
MESH THREE.D
```

The THREE.D parameter tells Atlas that a three dimensional grid will be specified. The other statements used to specify 3D structures and grids are the same as for 2D with the addition of Z direction specifications. The statements:

```
MESH THREE.D

X.MESH LOCATION=0 SPACING=0.15

X.MESH LOCATION=3 SPACING=0.15

Y.MESH LOCATION=0 SPACING=0.01

Y.MESH LOCATION=0.4 SPACING=0.01

Y.MESH LOCATION=3 SPACING=0.5

Z.MESH LOCATION=0 SPACING=0.1

Z.MESH LOCATION=3 SPACING=0.1
```

define a 3D mesh that is uniform in the X and Z directions and varies in the Y direction.

Position parameters for the Z direction (Z.MIN and Z.MAX) are also used on **REGION**, **ELECTRODE**, or **DOPING** statements.

2.6.9 Specifying 3D Cylindrical Structures

As mentioned in Section 2.6.3 “Using The Command Language To Define A Structure”, you can model a quasi-3D cylindrical structure in Atlas2D by specifying the CYLINDRICAL parameter on the **MESH** statement. This has the drawback that the resulting device structure and solution has no dependence on the angle around the axis of rotation. In Atlas3D, the CYLINDRICAL parameter enables you to create a general cylindrical structure. The **MESH** statement must appear as:

```
MESH THREE.D CYLINDRICAL
```

where the THREE.D parameter informs the simulator to create a fully 3D mesh.

When specifying the CYLINDRICAL parameter, you must now specify the structure in terms of radius, angle, and cartesian Z coordinates.

There are three mesh statements analogous to those used for general structures that are used to specify mesh in radius, angle and Z directions. The R.MESH statement is used to specify radial mesh. The A.MESH statement is used to specify angular mesh. The Z.MESH statement is used to specify mesh in the Z direction.

For example:

```
MESH THREE.D CYLINDRICAL


R.MESH LOCATION=0.0    SPACING=0.0004

R.MESH LOCATION=0.0028 SPACINg=0.00005
```

```
R.MESH LOCATION=0.004  SPACING=0.0002

A.MESH LOCATION=0   SPACING=45
A.MESH LOCATION=360 SPACING=45

Z.MESH LOCATION=-0.011 SPACING=0.001
Z.MESH LOCATION=0.011  SPACING=0.001
```

Here, the R.MESH lines are similar to the familiar X.MESH, Y.MESH, and Z.MESH except the R.MESH locations and spacings are radial relative to the Z axis in microns.

The locations and spacings on the A.MESH lines specify locations and spacings in degrees of rotation about the Z axis. The Z.MESH lines are exactly the same as have been already discussed.

For 3D cylindrical structure, specifying the **REGION** and **ELECTRODE** statements also are modified to allow specification of ranges in terms of radius, angle and z location. The range parameters are R.MIN, R.MAX, A.MIN, A.MAX, Z.MIN and Z.MAX. R.MIN, R.MAX, Z.MIN and Z.MAX are all in units of microns. A.MIN and A.MAX are in degrees.

To continue the example, we will specify a surround gate nano-tube as follows:

```
REGION NUM=1 MATERIAL=silicon \
   Z.MIN=-0.1 Z.MAX=0.1 A.MIN=0 A.MAX=360.0 R.MAX=0.0028

REGION NUM=2 MATERIAL=oxide \
   R.MIN=0.0028 R.MAX=0.004 Z.MIN=-0.011 Z.MAX=0.011 A.MIN=0
   A.MAX=360.0

ELECTRODE NAME=DRAIN  Z.MIN=-0.011 Z.MAX=-0.01 R.MAX=0.0028

ELECTRODE NAME=SOURCE Z.MIN=0.01   Z.MAX=0.011 R.MAX=0.0028

ELECTRODE NAME=GATE   Z.MIN=-0.004 Z.MAX=0.004 R.MIN=0.0038
MATERIAL=aluminum
```

Here, we defined a central core of silicon surrounded by an oxide cladding. The device has a surrounding gate with a source and drain at either end.

Figure 2-13 shows a cutplane normal to the Z axis at z=0 for this device.

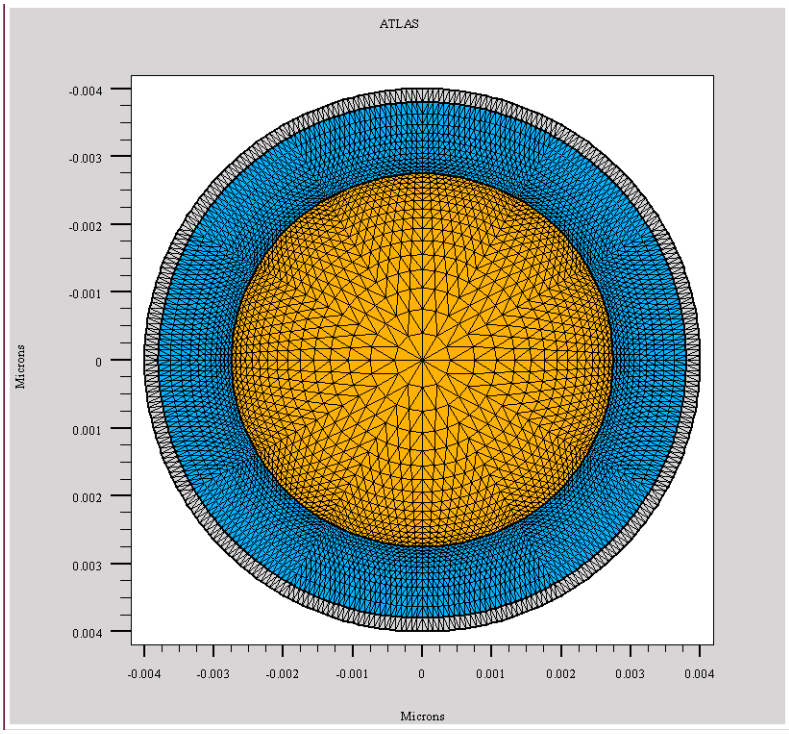


Figure 2-13 Cutplane of 3D Cylindrical Simulation

In some instances, you can reduce the size of the cylindrical device by using symmetry arguments. So if the device has mirror symmetry about some plane through its axis of rotation, it suffices to model only half of the cylinder. Atlas allows you to achieve this by one of two methods. In the first method, you specify more than one A.MESH statement, and the maximum angle specified in these statements is the maximum angle of the device structure. In the second method, you only specify one A.MESH statement with a location of zero degrees and the required angular spacing. The MAX.ANGLE parameter specified on the **MESH** statement gives the angle of the wedge.

For example, to make a semi-cylinder, you use either of the following syntaxes

```
MESH CYLINDRICAL THREE.D
...
A.MESH LOC=0    SPAC=30
A.MESH LOC=180  SPAC=30
...
```

or alternatively

```
MESH CYLINDRICAL THREE.D MAX.ANGLE=180
...
A.MESH LOC=0 SPAC=30
...
```

where statements for radial and vertical meshes are omitted for clarity. The major angular spacing is 30°.

The value of the maximum angle should not be greater than 180° and must satisfy the condition that the angular range of the device must start and end on major spokes of the radial mesh. In other words, the maximum angle must be equal to a integral multiple of angular mesh spacing. For example, if the angular mesh spacing was 45°, then the maximum angle can be 45, 90, 135, or 180°. Atlas will automatically adjust the value of the maximum angle if it is necessary.

If the mesh spacing in the radial direction is regular, then the resulting mesh will usually have no obtuse elements. If the mesh spacing in the radial direction decreases with increasing radius, then it is possible that some obtuse triangular elements will be formed. Use the parameter MINOBTUSE to try to reduce the number of obtuse triangular elements.

The **DOPING** statement has been modified to accept the parameters R.MIN, R.MAX, A.MIN and A.MAX. These apply to the analytic doping PROFILES, namely UNIFORM, GAUSSIAN, and ERFC. R.MIN and R.MAX specify the minimum and maximum radial extents in microns. A.MIN and A.MAX specify the minimum and maximum angular extents in degrees. The principal direction for the GAUSSIAN and ERFC dependence is the Z direction. Thus, the usual DOPING parameters, such as CHAR, PEAK, DOSE, START and JUNCTION all apply to the Z direction with R.MIN, R.MAX, A.MIN, A.MAX defining the extents at which lateral fall off occurs.

The lateral fall-off in the radial and angular directions can be controlled by the LAT.CHAR or RATIO.LAT parameters.

For example:

```
DOPING GAUSSIAN N.TYPE START=0.25 CONC=1.0e19 CHAR=0.2 \
LAT.CHAR=0.2 \

R.MIN=0.4 R.MAX=0.6 A.MIN=0.0 A.MAX=360
```

produces a gaussian doping profile in the Z direction, with a peak at Z=0.25 microns, and with a lateral gaussian fall-off in the radial direction when it is outside a radius of between 0.4 and 0.6 microns. Figure 2-14 shows the radial and angular distribution. Figure 2-15 shows the radial and z-variation. The Z direction corresponds to the vertical direction on the cutplane.

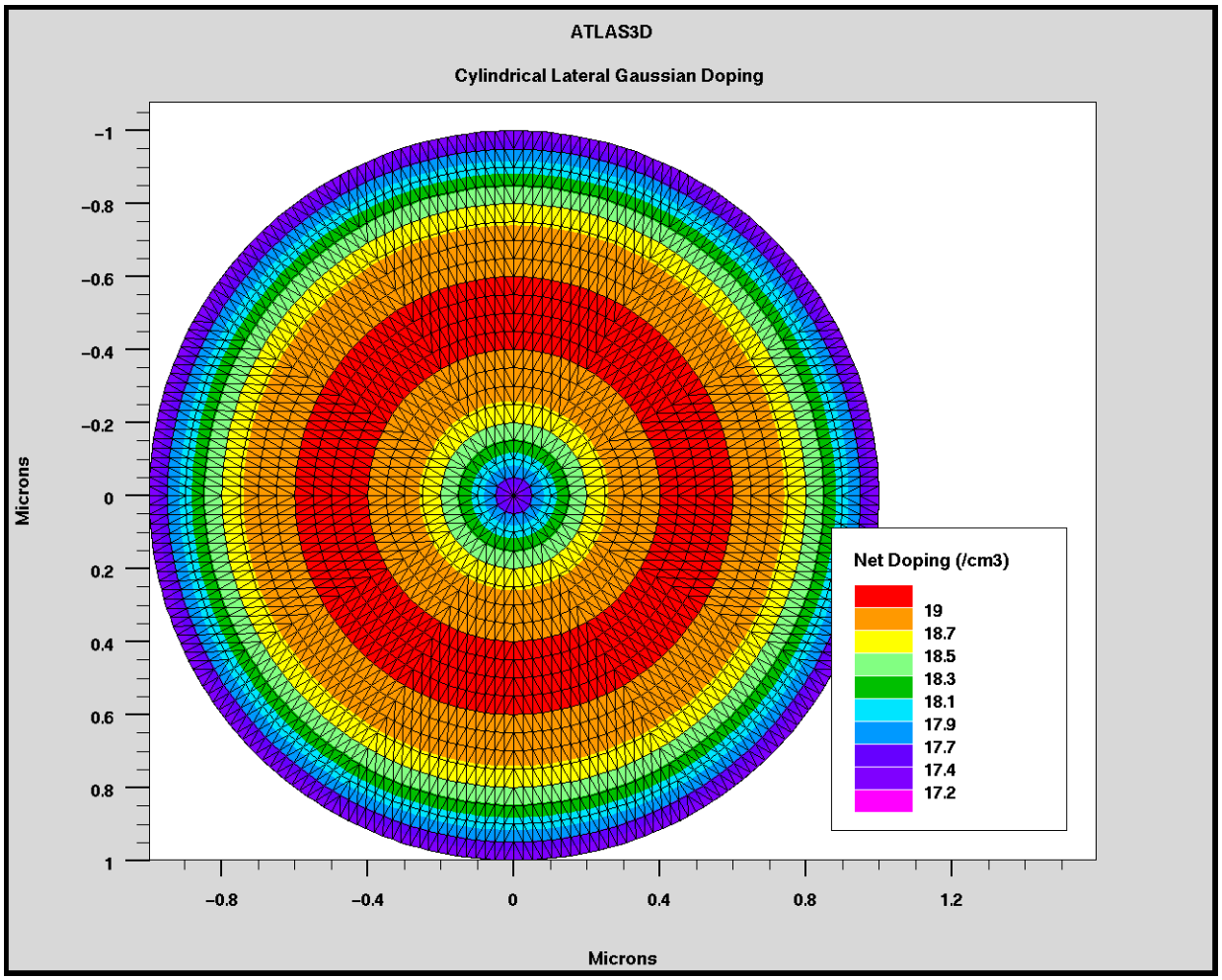


Figure 2-14 Cylindrical gaussian doping profile on a cutplane perpendicular to the Z axis

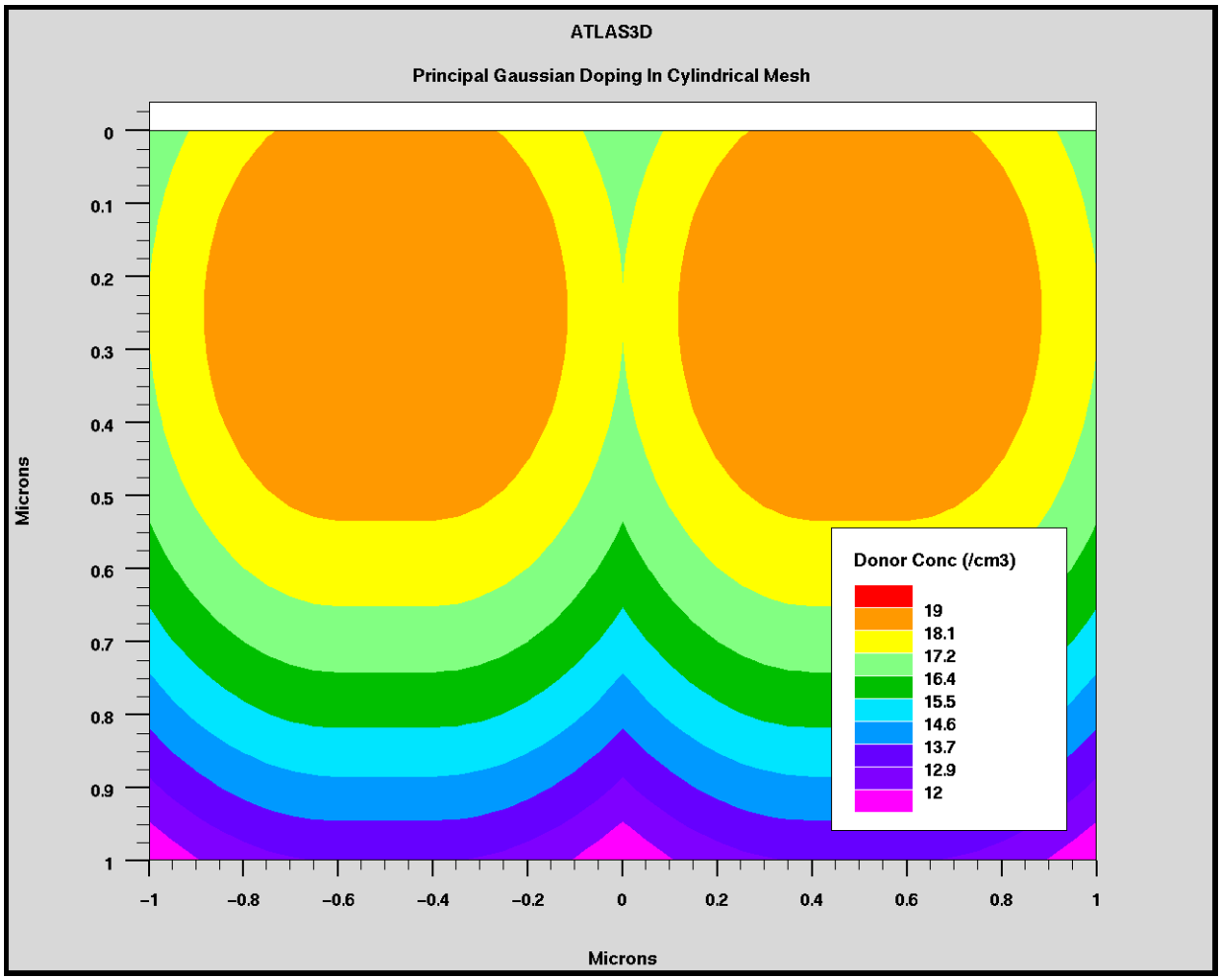


Figure 2-15 Cylindrical gaussian doping profile on a cutplane containing the Z axis

2.6.10 Extracting 2D Circular Structures From 3D Cylindrical Structures

To simulate circular structures in 2D that are analogous to the 3D meshing described above, use the cutplane feature in structure saving. To do this, specify the CUTPLANE and Z.CUTPLANE parameters of the **SAVE** statement. The logical parameter CUTPLANE specifies that you are interested in outputting a 2D cutplane from a 3D structure. The Z.CUTPLANE specifies the Z coordinate value in microns of the location where you want the cutplane.

For example, Figure 2-13 was generated by the following syntax:

```
SAVE CUTPLANE Z.CUTPLANE=0.0
```

2.6.11 General Comments Regarding Grids

Specifying a good grid is a crucial issue in device simulation but there is a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine grid that resolves the structure in solutions. Numerical efficiency is greater when fewer grid points are used. The critical areas to resolve are difficult to generalize because they depend on the technology and the transport phenomena. The only generalization possible is that most critical areas tend to coincide with reverse-biased metallurgical junctions. Typical critical areas are:

- High electric fields at the drain/channel junction in MOSFETs
- The transverse electric field beneath the MOSFET gate
- Recombination effects around the emitter/base junction in BJTs
- Areas of high impact ionization
- Around heterojunctions in HBT's and HEMTs.

The CPU time required to obtain a solution is typically proportional to N^{α}, where N is the number of nodes and α varies from 2 to 3 depending on the complexity of the problem. Thus, the most efficient way is to allocate a fine grid only in critical areas and a coarser grid elsewhere.

The three most important factors to look for in any grid are:

- Ensure adequate mesh density in high field areas
- Avoid obtuse triangles in the current path or high field areas
- Avoid abrupt discontinuities in mesh density

For more information about grids, see Chapter 21 "Numerical Techniques", Section 21.3 "Meshes".

2.6.12 Maximum Numbers Of Nodes, Regions, and Electrodes

Atlas sets some limits on the maximum number of grid nodes that can be used. But this shouldn't be viewed as a bottleneck to achieving simulation results. In the default version, 2D Atlas simulations have a maximum node limit of 100,000. 3D Atlas simulations have an upper limit of 40,000,000 nodes with no more than 100,000 nodes in any one Z plane and a maximum number of Z planes of 2,000.

This limit is high enough that for almost all simulations of conventional devices, running out of nodes is never an issue. For most 2D simulations, you can obtain accurate results with somewhere between 2,000 and 4,000 node points properly located in the structure.

If you exceed the node limits, error messages will appear and Atlas will not run successfully. There are two options to deal with this problem. The first option is to decrease the mesh density because simulations with the maximum nodes take an extremely long time to complete. The second option is to contact your local Silvaco office (support@silvaco.com).

A node point limitation below these values might be seen due to virtual memory constraints on your hardware. For each simulation, Atlas dynamically allocates the virtual memory. See the Silvaco Installation Guide for information about virtual memory requirements. The virtual memory used by the program depends on the number of nodes and on items, such as the models used and the number of equations solved.

Also, there is a node limit for the number of nodes in the X or Y directions in 2D and 3D Atlas. In the standard version, this limit is 20,000 nodes. This is applicable to meshes defined in the Atlas syntax using X.MESH and Y.MESH statements.

The maximum number of regions defined in both 2D and 3D Atlas is 1,000. The maximum number of definable electrodes is 100. Again, if you need to include more than the maximum number of regions or electrodes, contact your local Silvaco office (support@silvaco.com).

2.6.13 Quadrilateral REGION Definition

In addition to the rectangular shaped regions, which we have considered so far, you can create regions with sloping sides. Although it is possible to make complicated overall region shapes using rectangular regions as building blocks, this task is made easier by being able to specify the co-ordinates of each of the four corners of the **REGION**. To do this, use the P1.X, P1.Y, P2.X, P2.Y, P3.X, P3.Y, P4.X, and P4.Y parameters on the **REGION** statement. Then, (P1.X, P1.Y) are the XY coordinates of the first corner of the region and so on.

The region must lie entirely within the device limits as defined by the mesh. To obtain best results, these corner coordinates must coincide with valid mesh points as created by the X.MESH and Y.MESH statements.

Because the overall mesh shape will still be rectangular, you may need to use VACUUM regions as padding to get a non-rectangular shaped device. Alternatively, use DevEdit to get a non-rectangular overall device shape.

Example:

```
REGION   NUMBER=3   MATERIAL=SILICON   P1.X=0.0   P1.Y=1.0   P2.X=3.0
P2.Y=1.0 P3.X=3.5

P3.Y=2.5 P4.X=0.0 P4.Y=2.0
```

Table 2-1 Quadrilateral Shaped REGION Definition

Statement	Parameter	Type	Default	Units
REGION	P1.X	Real		Microns
REGION	P1.Y	Real		Microns
REGION	P2.X	Real		Microns
REGION	P2.Y	Real		Microns
REGION	P3.X	Real		Microns
REGION	P3.Y	Real		Microns
REGION	P4.X	Real		Microns
REGION	P4.Y	Real		Microns

2.7 Defining Material Parameters And Models

Once you define the mesh, geometry, and doping profiles, you can modify the characteristics of electrodes, change the default material parameters, and choose which physical models Atlas will use during the device simulation. To accomplish these actions, use the **CONTACT**, **MATERIAL**, and **MODELS** statements respectively. Impact ionization models can be enabled using the **IMPACT** statement. Interface properties are set by using the **INTERFACE** statement.

Many parameters are accessible through the Silvaco C-Interpreter (SCI), which is described in Appendix A "C-Interpreter Functions". This allows you to define customized equations for some models.

2.7.1 Specifying Contact Characteristics

Workfunction for Gates or Schottky Contacts

An electrode in contact with semiconductor material is assumed by default to be ohmic. If a work function is defined, the electrode is treated as a Schottky contact. The **CONTACT** statement is used to specify the metal workfunction of one or more electrodes. The NAME parameter is used to identify which electrode will have its properties modified.

The WORKFUNCTION parameter sets the workfunction of the electrode. For example, the statement:

```
CONTACT NAME=gate WORKFUNCTION=4.8
```

sets the workfunction of the electrode named gate to 4.8eV. The workfunctions of several commonly used contact materials can be specified using the name of the material. You can specify workfunctions for ALUMINUM, N.POLYSILICON, P.POLYSILICON, TUNGSTEN, and TU.DISILICIDE in this way. The following statement sets the workfunction for a n-type polysilicon gate contact.

```
CONTACT NAME=gate N.POLYSILICON
```

Aluminum contacts on heavily doped silicon is usually ohmic. For this situation, don't specify a workfunction. For example, MOS devices don't specify:

```
CONTACT NAME=drain ALUMINUM    /* wrong */
```

The **CONTACT** statement can also be used to specify barrier and dipole lowering of the Schottky barrier height. To enable barrier lowering, specify the BARRIER parameter, while specifying dipole lowering using the ALPHA parameter. For example, the statement:

```
CONTACT NAME=anode WORKFUNCTION=4.9 BARRIER ALPHA=1.0e-7
```

sets the work function of the Schottky contact named anode to 4.9eV enables barrier lowering and sets the dipole lowering coefficient to 1 nm.

Note: When a Schottky barrier is defined at a contact, we recommend that a fine y mesh is present just beneath the contact inside the semiconductor. This allows the Schottky depletion region to be accurately simulated.

Setting Current Boundary Conditions

The **CONTACT** statement is also used to change an electrode from voltage control to current control. Current controlled electrodes are useful when simulating devices, where the current is highly sensitive to voltage or is a multi-valued function of voltage (e.g., post-breakdown and when there is snap-back).

The statement:

```
CONTACT NAME=drain CURRENT
```

changes the drain electrode to current control. The BLOCK or NEWTON solution methods are required for all simulations using a current boundary condition. For more information about these methods, see Section 21.5 “Non-Linear Iteration”.

Defining External Resistors, Capacitors, or Inductors

Lumped resistance, capacitance, and inductance connected to an electrode can be specified using the RESISTANCE, CAPACITANCE, and INDUCTANCE parameters in the **CONTACT** statement. For example, the statement:

```
CONTACT NAME=drain RESISTANCE=50.0 CAPACITANCE=20e-12
INDUCTANCE=1e-6
```

specifies a parallel resistor and capacitor of 50 ohms and 20 pF respectively in series with a 1 μH inductor. Note that in 2D simulations, these passive element values are scaled by the width in the third dimension. Since in 2D Atlas assumes a 1μm width, the resistance becomes 50 Ω-μm.

Distributed contact resistance for an electrode can be specified using the CON.RESIST parameter. For example, the statement:

```
CONTACT NAME=source CON.RESISTANCE=0.01
```

specifies that the source contact has a distributed resistance of 0.01 Ωcm^2.

Note: Simulations with external resistors, capacitors, or inductors must be solved using the BLOCK or NEWTON solution method.

Floating Contacts

The **CONTACT** statement is also used to define a floating electrode. There are two distinctly different situations where floating electrodes are important. The first situation is for floating gate electrodes used in EEPROM and other programmable devices. The second situation is for contacts directly onto semiconductor materials such as floating field plates in power devices.

To enable floating gates, specify the FLOATING parameter on the **CONTACT** statement. For example, the statement:

```
CONTACT NAME=fgate FLOATING
```

specifies that the electrode named fgate will be floating and that charge boundary conditions will apply.

For contacts directly onto semiconductor, the FLOATING parameter cannot be used. This type of floating electrode is best simulated by specifying current boundary conditions on the CONTACT statement. For example, the statement:

```
CONTACT NAME=drain CURRENT
```

specifies current boundary conditions for the electrode named drain. On subsequent **SOLVE** statements, the drain current boundary condition will default to zero current. Therefore, floating the contact.

You can also make a floating contact to a semiconductor using a very large resistor attached to the contact instead. For example:

```
CONTACT NAME=drain RESIST=1e20
```

Note that extremely large resistance values must be used to keep the current through the contact insignificant. Using a lumped resistor will allow the tolerance on potential to move slightly above zero. For example, if the tolerance is 10^{-5}V and the defined resistance was only 10MΩμm, then a current of 10^{-12} A/μm may flow through the contact, which may be significant in breakdown simulations.

Shorting Two Contacts Together

It is possible in Atlas to tie two or more contact together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initially defined.

If the structure is defined using Atlas syntax, you can have multiple **ELECTRODE** statements with the same NAME parameter defining separate locations within the device structure. In this case, the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both **ELECTRODE** areas.

Also, if two separate metal regions in Athena are defined using the ATHENA **ELECTRODE** statement to have the same name, then in Atlas these two electrodes will be considered as shorted together.

If the electrodes are defined with different names, the following syntax can be used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base

.

SOLVE VBASE=0.1
```

Here, the electrode, base1, will be linked to the electrode, base. The applied 0.1V on base will then appear on base1. Atlas, however, will calculate and store separate currents for both base and base1. This can be a useful feature. But in some cases, such as where functions of the currents are required in **EXTRACT** or TonyPlot, it is undesirable. You can add the SHORT parameter to the **CONTACT** statement above to specify so that only a single base current will appear combining the currents from base and base1. Note that the electrode specified by the COMMON parameter should preferably have a name that exactly matches one of those specified in the **ELECTRODE** statement (see the NAME description). Additionally, the electrode specified by the NAME parameter should not have a bias applied directly to it. You should always apply the bias to the electrode specified by the COMMON parameter.

You can also specify the voltages on the linked electrodes to be different, but have a constant OFFSET relative to the bias on the electrode specified by the COMMON parameter.

The statement

```
CONTACT name=base1 COMMON=base  FACTOR=0.5
```

ensures the bias on base1 will be equal to the bias on base + 0.5 V. If you also specify the MULT parameter on this statement, then FACTOR will be interpreted as multiplicative rather than additive. In this case, the bias applied to base1 will be one-half of the bias applied to base. If FACTOR (and optionally MULT) is applied to a contact for which current boundary conditions are being used, then it has no effect. Any existing bias differences between the linked electodes are, however, maintained under current boundary conditions. To apply additive or multiplicative relationships between currents on linked electrodes, remove the **CONTACT** statements linking them. Then, directly specify the required values of the currents on the **SOLVE** statement.

When loading a structure from Athena or DevEdit, where two defined electrode regions are touching, Atlas will automatically short these and use the electrode name that was defined first.

Making an Open Circuit Contact

It is often required to perform a simulation with an open circuit on one of the defined electrodes. There are three different methods to make an open circuit contact. The first method is to entirely deleting an electrode from the structure file. The second method is to add an extremely large lumped resistance. For example, $10^{20}\Omega$ onto the contact to be made open circuit. The third method is to switch the boundary conditions on the contact to be made open circuit from voltage controlled to current controlled. Then, specifying a very small or zero current through that electrode.

Each of these methods are feasible. If a floating region, however, is created within the structure, then numerical convergence may be affected. As a result, we normally recommend that you use the second method because it ensures better convergence.

2.7.2 Specifying Material Properties

Semiconductor, Insulator, or Conductor

All materials are split into three classes: semiconductors, insulators and conductors. Each class requires a different set of parameters to be specified. For semiconductors, these properties include electron affinity, band gap, density of states and saturation velocities. There are default parameters for material properties used in device simulation for many materials.

Appendix B “Material Systems” lists default material parameters and includes a discussion on the differences between specifying parameters for semiconductors, insulators, and conductors.

Setting Parameters

The **MATERIAL** statement allows you to specify your own values for these basic parameters. Your values can apply to a specified material or a specified region. For example, the statement:

```
MATERIAL MATERIAL=Silicon EG300=1.12 MUN=1100
```

sets the band gap and low field electron mobility in all silicon regions in the device. If the material properties are defined by region, the region is specified using the REGION or NAME parameters in the **MATERIAL** statement. For example, the statement:

```
MATERIAL REGION=2 TAUN0=2e-7 TAUP0=1e-5
```

sets the electron and hole Shockley-Read-Hall recombination lifetimes for region number two (see the "Shockley-Read-Hall (SRH) Recombination" section on page 3-215 for more information about this type of recombination). If the name, base, has been defined using the NAME parameter in the **REGION** statement, then the statement:

```
MATERIAL NAME=base NC300=3e19
```

sets the conduction band density of states at 300 K for the region named base.

The description of the **MATERIAL** statement provides a complete list of all the material parameters that are available.

Heterojunction Materials

The material properties of heterojunctions can also be modified with the **MATERIAL** statement. In addition to the regular material parameters, you can define composition dependent material parameters. For example, composition dependent band parameters, dielectric constants, and saturation velocities.

For heterojunction material systems, the bandgap difference between the materials is divided between conduction and valence bands. The ALIGN parameter specifies the fraction of this difference applied to the conduction band edge. This determines the electron and hole barrier height and overrides any electron affinity specification. For example, the statement:

```
MATERIAL MATERIAL=InGaAs ALIGN=0.36

MATERIAL MATERIAL=InP    ALIGN=0.36
```

specifies that 36% of the bandgap difference between InGaAs and InP is applied to the conduction band and 64% is applied to the valence band. For example, if the band gap difference (Δ Eg) for this material system is 0.6 eV, then the conduction band barrier height is 0.216 eV and the valence band barrier height is 0.384 eV.

For heterojunction devices, the transport models may be different for each material. You can specify these models and their coefficients for each material using the **MODELS** statement. See Section 2.7.4 "Specifying Physical Models" for a description of this option.

2.7.3 Specifying Interface Properties

The **INTERFACE** statement is used to define the interface charge density and surface recombination velocity at interfaces between semiconductors and insulators. For example, the statement:

```
INTERFACE QF=3e10
```

specifies that all interfaces between semiconductors and insulators have a fixed charge of $3.10^{10}cm^{-2}$. In many cases, the interface of interest is restricted to a specific region. This can be accomplished with the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters on the **INTERFACE** statement. These parameters define a rectangle, where the interface properties apply. For example, the statement:

```
INTERFACE QF=3e10 X.MIN=1.0 X.MAX=2 Y.MIN=0.0 Y.MAX=0.5
```

restricts the interface charge to the semiconductor-insulator boundary within the specified rectangle. In addition to fixed charge, surface recombination velocity and thermionic emission are enabled and defined with the **INTERFACE** statement.

2.7.4 Specifying Physical Models

Physical models are specified using the **MODELS** and **IMPACT** statements. Parameters for these models appear on many statements including: **MODELS**, **IMPACT**, **MOBILITY**, and **MATERIAL**. The physical models can be grouped into five classes: mobility, recombination, carrier statistics, impact ionization, and tunneling. Section 3.6 “Physical Models” contains details for each model.

Tables 2-2 to 2-6 give summary descriptions and recommendations on the use of each model. Table 2-7 is a guide for compatibility between models.

All models with the exception of impact ionization are specified on the **MODELS** statement. Impact ionization is specified on the **IMPACT** statement. For example, the statement:

```
MODELS CONMOB FLDMOB SRH FERMIDIRAC

IMPACT SELB
```

specifies that the standard concentration dependent mobility, parallel field mobility, Shockley-Read-Hall recombination with fixed carrier lifetimes, Fermi Dirac statistics and Selberherr impact ionization models should be used.

Atlas also provides an easy method for selecting the correct models for various technologies. The MOS, BIP, PROGRAM, and ERASE parameters for the **MODELS** statement configure a basic set of mobility, recombination, carrier statistics, and tunneling models. The MOS and BIP parameters enable the models for MOSFET and bipolar devices, while PROGRAM and ERASE enable the models for programming and erasing programmable devices. For example, the statement:

```
MODELS MOS PRINT
```

enables the CVT, SRH, and FERMIDIRAC models, while the statement:

```
MODELS BIPOLAR PRINT
```

enables the CONMOB, FLDMOB, CONSRH, AUGER, and BGN models. If LAT.TEMP is also specified on the **MODELS** statement, or the TEMPERATURE parameter differs from 300 Kelvin by more than 10 Kelvin, then the ANALYTIC model is used instead of CONMOB.

Note: The PRINT parameter lists to the run time output the models and parameters, which will be used during the simulation. This allows you to verify models and material parameters. We highly recommend that you include the PRINT parameter in the **MODELS** statement.

Physical models can be enabled on a material by material basis. This is useful for heterojunction device simulation and other simulations where multiple semiconductor regions are defined and may have different characteristics. For example, the statement:

```
MODEL MATERIAL=GaAs FLDMOB EVSATMOD=1 ECRITN=6.0e3 CONMOB
MODEL MATERIAL=InGaAs SRH FLDMOB EVSATMOD=1 \
ECRITN=3.0e3
```

change both the mobility models and critical electric field used in each material. For devices based on advanced materials, these model parameters should be investigated carefully.

Energy Balance Models

The conventional drift-diffusion model of charge transport neglects non-local effects, such as velocity overshoot and reduced energy dependent impact ionization. Atlas can model these effects through the use of an energy balance model, which uses a higher order approximation of the Boltzmann Transport Equation (see Section 3.1.3 “The Transport Equations”). In this equation, transport parameters, such as mobility and impact ionization, are functions of the local carrier temperature rather than the local electric field.

To enable the energy balance transport model, use the HCTE, HCTE.EL, or HCTE.HO parameters in the **MODELS** statement. These parameters enable the energy transport model for both carriers, electrons only, or holes only respectively. For example, the statement:

```
MODELS MOS HCTE
```

enables the energy balance transport model for both electrons and holes in addition to the default MOSFET models.

2.7.5 Summary Of Physical Models

Table 2-2 Carrier Statistics Models

Model	Syntax	Notes
Boltzmann	BOLTZMANN	Default model
Fermi-Dirac	FERMI	Reduced carrier concentrations in heavily doped regions (statistical approach).
Incomplete Ionization	INCOMPLETE	Accounts for dopant freeze-out. Typically, it is used at low temperatures.
Silicon Ionization Model	IONIZ	Accounts for full ionization for heavily doped Si. Use with INCOMPLETE.
Bandgap Narrowing	BGN	Important in heavily doped regions. Critical for bipolar gain. Use Klaassen Model.

Table 2-3 Mobility Models

Model	Syntax	Notes
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora's Model	ARORA	Alternative to ANALYTIC for Si.
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity saturation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.
Klaassen Model	KLA	Includes N, T, and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices
Shirahata Model	SHI	Includes N, $E_{\perp}$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_{\perp}$ effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E//, and $E_{\perp}$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_{\perp}$ effects. Only calibrated for 300K.

Table 2-4 Recombination Models

Model	Syntax	Notes
Shockley-Read-Hall	SRH	Uses fixed minority carrier lifetimes. Should be used in most simulations.
Concentration Dependent	CONSRH	Uses concentration dependent lifetimes. Recommended for Si.
Auger	AUGER	Direct transition of three carriers. Important at high current densities.
Optical	OPTR	Band-band recombination. For direct materials only.
Surface	S.N S.P	Recombination at semiconductor to insulator interfaces. This is set in the INTERFACE statement.

Table 2-5 Impact Ionization

Model	Syntax	Notes
Selberherr's Model	IMPACT SELB	Recommended for most cases. Includes temperature dependent parameters.
Grant's Model	IMPACT GRANT	Similar to Selberherr's model but with different coefficients.
Crowell-Sze	IMPACT CROWELL	Uses dependence on carrier scattering length.
Toyabe Model	IMPACT TOYABE	Non-local model used with Energy Balance. Any IMPACT syntax is accepted.
Concannon	N.CONCAN P.CONCAN	Non-local model developed in Flash EEPROM technologies.

Table 2-6 Tunneling Models and Carrier Injection Models

Model	Syntax	Notes
Band-to-Band (standard)	BBT.STD	For direct transitions. Required with very high fields.
Concannon Gate Current Model	N.CONCAN P.CONCAN	Non-local gate model consistent with Concannon substrate current model.
Direct Quantum tunneling (Electrons)	QTUNN.EL	Quantum tunneling through conduction band barrier due to an insulator.

Table 2-6 Tunneling Models and Carrier Injection Models		
Model	**Syntax**	**Notes**
Direct Quantum tunneling (Hole)	QTUNN.HO	Quantum tunneling through valence band barrier due to an insulator.
Fowler-Nordheim (electrons)	FNORD	Self-consistent calculation of tunneling through insulators. Used in EEPROMs.
Fowler-Nordheim (holes)	FNHOLES	Same as FNORD for holes.
Klaassen Band-to-Band	BBT.KL	Includes direct and indirect transitions.
Hot Electron Injection	HEI	Models energetic carriers tunneling through insulators. Used for gate current and Flash EEPROM programming.
Hot Hole Injection	HHI	HHI means hot hole injection.

Note: In the notes in Tables 2-2 through 2-6, n is electron concentration, p is hole concentration, T is lattice temperature, N is dopant concentration, Ell is parallel electric field, and E⊥ is perpendicular electric field.

Table 2-7 Model Compatibility Chart											
	CONMOB	**FLDMOB**	**TFLDMB2**	**YAMAGUCHI**	**CVT**	**ARORA**	**ANALYTIC**	**CCSMOB**	**SURMOB**	**LATTICE H**	**E.BALANCE**
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	CV	AR	—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK

Table 2-7 Model Compatibility Chart

	CONMOB	FLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURMOB	LATTICE H	E.BALANCE
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	[2]

Key To Table Entries

MODEL ABBREVIATION = The model that supersedes when a combination is specified. In some cases, a warning message is issued when a model is ignored.

OK = This combination is allowed.

NO = This combination is not allowed.

NOTES:

1.Uses internal model similar to FLDMOB

2.When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

Using the C-Interpreter to Specify Models

One of the Atlas products is a C language interpreter that allows you to specify many of the models used by Atlas. To use these functions, implement the model in C as equations in a special file called an Atlas lib file. You can access the default Atlas template file by typing:

```
atlas -T <filename>
```

at the UNIX command prompt. This creates a default template file with the name specified by the `<filename>` parameter. See Appendix A "C-Interpreter Functions" for a listing of the default C-Interpreter functions. To use the interpreter functions, give the corresponding parameters in the statements containing the name of the C language file with the model given as the parameter value. For example, the statement:

```
MATERIAL NAME=Silicon F.MUNSAT=munsat.lib
```

specifies that the file, *munsat.lib*, contains the C-Interpreter function for the specification of the parallel field dependent electron mobility model.

2.8 Choosing Numerical Methods

2.8.1 Numerical Solution Techniques

Several different numerical methods can be used for calculating the solutions to semiconductor device problems. Numerical methods are given in the **METHOD** statements of the input file. Some guidelines for these methods will be given here. For full details, however, see Chapter 21 "Numerical Techniques".

Different combinations of models will require Atlas to solve up to six equations. For each of the model types, there are basically three types of solution techniques: (a) decoupled (GUMMEL), (b) fully coupled (NEWTON) and (c) BLOCK. The GUMMEL method will solve for each unknown in turn keeping the other variables constant, repeating the process until a stable solution is achieved. The NEWTON method solve the total system of unknowns together. The BLOCK methods will solve some equations fully coupled while others are de-coupled.

Generally, the GUMMEL method is useful where the system of equations is weakly coupled but has only linear convergence. The NEWTON method is useful when the system of equations is strongly coupled and has quadratic convergence. The NEWTON method may, however, spend extra time solving for quantities, which are essentially constant or weakly coupled. NEWTON also requires a more accurate initial guess to the problem to obtain convergence. Thus, a BLOCK method can provide for faster simulations times in these cases over NEWTON. GUMMEL can often provide better initial guesses to problems. It can be useful to start a solution with a few GUMMEL iterations to generate a better guess. Then, switch to NEWTON to complete the solution. Specification of the solution method is carried out as follows:

```
METHOD GUMMEL BLOCK NEWTON
```

The exact meaning of the statement depends on the particular models it applied to. This will be discussed in the following sections.

Basic Drift Diffusion Calculations

The isothermal drift diffusion model requires the solution of three equations for potential, electron concentration, and hole concentration. Specifying GUMMEL or NEWTON alone will produce simple Gummel or Newton solutions as detailed above. For almost all cases, the NEWTON method is preferred and it is the default.

Specifying:

```
METHOD GUMMEL NEWTON
```

will cause the solver to start with GUMMEL iterations. Then, switch to NEWTON if convergence is not achieved. This is a robust but a more time consuming way of obtaining solutions for any device. This method, however, is highly recommended for all simulations with floating regions such as SOI transistors. A floating region is defined as an area of doping, which is separated from all electrodes by a pn junction.

BLOCK is equivalent to NEWTON for all isothermal drift-diffusion simulations.

Drift Diffusion Calculations with Lattice Heating

When the lattice heating model is added to drift diffusion, an extra equation is added. The BLOCK algorithm solves the three drift diffusion equations as a NEWTON solution. Follows this with a GUMMEL solution of the heat flow equation. The NEWTON algorithm solves all four equations in a coupled manner. NEWTON is preferred once the temperature is high. BLOCK, however, is quicker for low temperature gradients. Typically, the combination used is:

```
METHOD BLOCK NEWTON
```

Energy Balance Calculations

The energy balance model requires the solution of up to 5 coupled equations. GUMMEL and NEWTON have the same meanings as with the drift diffusion model. In other words, GUMMEL specifies a de-coupled solution and NEWTON specifies a fully coupled solution.

But BLOCK performs a coupled solution of potential, carrier continuity equations followed by a coupled solution of carrier energy balance, and carrier continuity equations.

You can switch from BLOCK to NEWTON by specifying multiple solution methods on the same line. For example:

```
METHOD BLOCK NEWTON
```

will begin with BLOCK iterations. Then, switch to NEWTON if convergence is still not achieved. This is the most robust approach for many energy balance applications.

The points where the algorithms switch is pre-determined, but can be changed in the METHOD statement, the default values set by Silvaco work well for most circumstances.

Energy Balance Calculations with Lattice Heating

When non-isothermal solutions are performed in conjunction with energy balance models, a system of up to six equations must be solved. GUMMEL or NEWTON solve the equations iteratively or fully coupled respectively. BLOCK initially performs the same function as with energy balance calculations, then solves the lattice heating equation in a de-coupled manner.

Setting the Number of Carriers

Atlas can solve both electron and hole continuity equations, or only for one or none. You can make this choice by using the CARRIERS parameter. For example:

```
METHOD CARRIERS=2
```

specifies a solution for both carriers is required. This is the default. With one carrier, the ELEC or HOLE parameter is needed. For example, for hole solutions only:

```
METHOD CARRIERS=1 HOLE
```

To select a solution for potential, only specify:

```
METHOD CARRIERS=0
```

Note: Setting the number of carriers using the syntax, MODEL NUMCARR=<n>, is obsolete and should not be used.

Important Parameters of the METHOD Statement

You can alter all of the parameters relevant to the numerical solution process. This, however, isn't recommended unless you have expert knowledge of the numerical algorithms. All of these parameters have been assigned optimal values for most solution conditions. For more information about numerical algorithms, see Chapter 21 "Numerical Techniques".

The following parameters, however, are worth noting at this stage:

- CLIMIT or CLIM.DD specify minimal values of concentrations to be resolved by the solver. Sometimes, you need to reduce this value to aid solutions of breakdown characteristics. A value of CLIMIT=1e-4 is recommended for all simulations of breakdown, where the pre-breakdown current is small. CLIM.DD is equivalent to CLIMIT but uses the more convenient units of cm^{-3} for the critical concentration. CLIM.DD and CLIMIT are related by the following expression.

$$\mathtt{CLIM\_DD} = \mathtt{CLIMIT} * \sqrt{N_C N_V} \qquad \text{2-3}$$

- DVMAX controls the maximum update of potential per iteration of Newton's method. The default corresponds to 1V. For power devices requiring large voltages, you may need an increased value of DVMAX. DVMAX=1e8 can improve the speed of high voltage bias ramps.
- CLIM.EB controls the cut-off carrier concentration below, which the program will not consider the error in the carrier temperature. This is applied in energy balance simulations to avoid excessive calculations of the carrier temperature at locations in the structure where the carrier concentration is low. Setting this parameter too high, where Atlas ignores the carrier temperature errors for significant carrier concentrations, will lead to unpredictable and mostly incorrect results .

Restrictions on the Choice of METHOD

The following cases require METHOD NEWTON CARRIERS=2 to be set for isothermal drift-diffusion simulations:

- current boundary conditions
- distributed or lumped external elements
- AC analysis
- impact ionization

Both BLOCK or NEWTON or both are permitted for lattice heat and energy balance.

Note: Simulations using the GUMMEL method in these cases may lead to non-convergence or incorrect results.

Pisces-II Compatibility

Previous releases of Atlas (2.0.0.R) and other PISCES-II based programs use the **SYMBOLIC** command to define the solution method and the number of carriers included in the solution. In this version of Atlas, the solution method is specified completely on the **METHOD** statement.

The COMB parameter, which was available in earlier Atlas versions, is no longer required. It has been replaced with either the BLOCK method or the combination of GUMMEL and NEWTON parameters. Table 2-8 identifies direct translations of old syntax to new.

Note: These are direct translations and not necessarily the best choices of numerical methods.

Table 2-8 Parameter Syntax Replacements

Old Syntax (V2.0.0.R)	New Syntax
`symbolic newton carriers=2`	`method newton`
`symbolic newton carriers=1 elec`	`method newton carriers=1 electron`
`symbolic gummel carriers=0`	`method gummel carriers=0`
`symbolic newton carriers=2` `method comb`	`method gummel newton`

`models lat.temp`
`symbolic newton carriers=2`
`method comb`
	`models lat.temp`
	`method block`

`models hcte`
`symbolic gummel carriers=2`
`method comb`
	`models hcte`
	`method block`

2.9 Obtaining Solutions

Atlas can calculate DC, AC small signal, and transient solutions. Obtaining solutions is similar to setting up parametric test equipment for device tests. You usually define the voltages on each of the electrodes in the device. Atlas then calculates the current through each electrode. Atlas also calculates internal quantities, such as carrier concentrations and electric fields throughout the device. This is information that is difficult or impossible to measure.

In all simulations, the device starts with zero bias on all electrodes. Solutions are obtained by stepping the biases on electrodes from this initial equilibrium condition. As will be discussed, due to the initial guess strategy, voltage step sizes are limited. This section concentrates on defining solution procedures. To save results, use the **LOG** or **SAVE** statements. Section 2.10 "Interpreting The Results" on how to analyze and display these results.

2.9.1 DC Solutions

In DC solutions, the voltage on each electrode is specified using the **SOLVE** statement. For example, the statements:

```
SOLVE VGATE=1.0

SOLVE VGATE=2.0
```

solves a single bias point with 1.0V and then 2.0V on the gate electrode. One important rule in Atlas is that when the voltage on any electrode is not specified in a given **SOLVE** statement, the value from the last **SOLVE** statement is assumed.

In the following case, the second solution is for a drain voltage of 1.0V and a gate voltage of 2.0V.

```
SOLVE VGATE=2.0

SOLVE VDRAIN=1.0
```

When the voltage on a particular electrode is never defined on any **SOLVE** statement and voltage is zero, you don't need to explicitly state the voltage on all electrodes on all **SOLVE** statements. For example, in a MOSFET, if VSUBSTRATE is not specified, then Vbs defaults to zero.

Sweeping The Bias

For most applications, a sweep of one or more electrodes is usually required. The basic DC stepping is inconvenient and a ramped bias should be used. To ramp the base voltage from 0.0V to 1.0V with 0.05V steps with a fixed collector voltage of 2.0V, use the following syntax:

```
SOLVE VCOLLECTOR=2.0

SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base
```

The NAME parameter is required and the electrode name is case-sensitive. Make sure the initial voltage, VSTEP and VFINAL, are consistent. A badly specified ramp from zero to 1.5V in 0.2V steps will finish at 1.4V or 1.6V.

Generating Families of Curves

Many applications such as MOSFET Id/Vds and bipolar Ic/Vce simulations require that a family of curves is produced. This is done by obtaining solutions at each of the stepped bias points first, and then solving over the swept bias variable at each stepped point. For example, in MOSFET Id/Vds curves, solutions for each Vgs value are obtained with Vds=0.0V. The output from these solutions are saved in Atlas solution files. For each gate bias, the solution file is loaded and the ramp of drain voltage performed.

The family of curves for three 1V gate steps and a 3.3V drain sweep would be implemented in Atlas as follows:

```
SOLVE VGATE=1.0  OUTF=solve_vgate1
SOLVE VGATE=2.0  OUTF=solve_vgate2
SOLVE VGATE=3.0  OUTF=solve_vgate3

LOAD INFILE=solve_vgate1
LOG OUTFILE=mos_drain_sweep1
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate2
LOG OUTFILE=mos_drain_sweep2
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate3
LOG OUTFILE=mos_drain_sweep3
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3
```

The **LOG** statements are used to save the Id/Vds curve from each gate voltage to separate files. We recommend that you save the data in this manner rather than to a single LOG file (see Section 2.10 “Interpreting The Results”).

2.9.2 The Importance Of The Initial Guess

To obtain convergence for the equations used, supply a good initial guess for the variables to be evaluated at each bias point. The Atlas solver uses this initial guess and iterates to a converged solution. For isothermal drift diffusion simulations, the variables are the potential and the two carrier concentrations. If a reasonable grid is used, almost all convergence problems in Atlas are caused by a poor initial guess to the solution.

During a bias ramp, the initial guess for any bias point is provided by a projection of the two previous results. Problems tend to appear near the beginning of the ramp when two previous results are not available. If one previous bias is available, it is used alone. This explains why the following two examples eventually produce the same result. The first will likely have far more convergence problems than the second.

```
1.  SOLVE VGATE=1.0 VDRAIN=1.0 VSUBSTRATE=-1.0
2.  SOLVE VGATE=1.0
    SOLVE VSUBSTRATE=-1.0
    SOLVE VDRAIN=1.0
```

In the first case, one solution is obtained with all specified electrodes at 1.0V. In the second case, the solution with only the gate voltage at 1.0V is performed first. All other electrodes are at zero bias. Next, with the gate at 1.0V, the substrate potential is raised to -1.0V and another solution is obtained. Finally, with the substrate and the gate biased, the drain potential is added and the system solved again. The advantage of this method over the first case is that the small incremental changes in voltage allow for better initial guesses at each step.

Generally, the projection method for the initial guess gives good results when the I-V curve is linear. But it may encounter problems if the IV curve is highly non-linear or if the device operating mode is changing. Typically, this may occur around the threshold or breakdown voltages. At these biases, smaller voltage steps are required to obtain convergence. As will be described, Atlas contains features such as the TRAP parameter and the curve tracer to automatically cut the voltage steps in these highly non-linear area.

Numerical methods are described in Section 2.8 “Choosing Numerical Methods” and Chapter 21 “Numerical Techniques”.

In many cases, these methods are designed to overcome the problems associated with the initial guess. This is particularly important in simulations involving more than the three drift diffusion variables. Generally, coupled solutions require a good initial guess, whereas de-coupled solutions can converge with a poor initial guess.

The Initial Solution

When no previous solutions exist, the initial guess for potential and carrier concentrations must be made from the doping profile. This is why the initial solution performed must be the zero bias (or thermal equilibrium) case. This is specified by the statement:

```
SOLVE INIT
```

But if this syntax isn’t specified, Atlas automatically evaluates an initial solution before the first **SOLVE** statement. To aid convergence of this initial guess, the solution is performed in the zero carrier mode solving only for potential.

The First and Second Non-Zero Bias Solutions

From experience with Atlas, it is found that the first and second non-zero bias solutions are the most difficult to obtain good convergence. Once these two solutions are obtained, the projection algorithm for the initial guess is available and solutions should all have a good initial guess.

These first two solutions, however, must use the result of the initial solution as the basis of their initial guess. Since the initial solution is at zero bias, it provides a poor initial guess.

The practical result of this is that the first and second non-zero bias solutions should have very small voltage steps. In the following example, the first case will likely converge whereas the second case may not.

```
1.  SOLVE INIT
    SOLVE VDRAIN=0.1
    SOLVE VDRAIN=0.2
    SOLVE VDRAIN=2.0
2.  SOLVE INIT
    SOLVE VDRAIN=2.0
```

The Trap Parameter

Although Atlas provides several methods to overcome a poor initial guess and other convergence problems, it is important to understand the role of the initial guess in obtaining each solution. The simplest and most effective method to overcome poor convergence is by using:

```
METHOD TRAP
```

This is enabled by default. Its effect is to reduce the bias step if convergence problems are detected. Consider the example from the previous section:

```
SOLVE INIT
SOLVE VDRAIN=2.0
```

If the second SOLVE statement does not converge, TRAP automatically cuts the bias step in half and tries to obtain a solution for Vd = 1.0V. If this solution does not converge, the bias step will be halved again to solve for Vd = 0.5V. This procedure is repeated up to a maximum number of tries set by the METHOD parameter MAXTRAPS. Once convergence is obtained, the bias steps are increased again to solve up to 2.0V. The default for MAXTRAPS is 4 and it is not recommended to increase it, since changing the syntax to use smaller bias steps is generally much faster.

This trap facility is very useful during bias ramps in overcoming convergence difficulties around transition points such as the threshold voltage. Consider the following syntax used to extract a MOSFET Id/Vgs curve.

```
SOLVE VGATE=0.0 VSTEP=0.2 VFINAL=5.0 NAME=gate
```

Assume the threshold voltage for the device being simulated is 0.7V and that Atlas has solved for the gate voltages up to 0.6V. The next solution, at 0.8V, may not converge at first. This is because the initial guess was formed from the two sub-threshold results at Vgs=0.4V and 0.6V and the solution has now become non-linear. The trap facility will detect the problems in the 0.8V solution and cut the bias step in half to 0.7V and try again. This will probably converge. The solution for 0.8V will then be performed and the bias ramp will continue with 0.2V steps.

2.9.3 Small-Signal AC Solutions

Specifying AC simulations is a simple extension of the DC solution syntax. AC small signal analysis is performed as a post-processing operation to a DC solution. Two common types of AC simulation in Atlas are outlined here. The results of AC simulations are the conductance and capacitance between each pair of electrodes. Tips on interpreting these results is described in Section 2.10 “Interpreting The Results”.

Single Frequency AC Solution During A DC Ramp

The minimum syntax to set an AC signal on an existing DC ramp is just the AC flag and the setting of the small signal frequency. For example:

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6
```

Other AC syntax for setting the signal magnitude and other parameters are generally not needed as the defaults suffice. One exception is in 1D MOS capacitor simulations. To obtain convergence in the inversion/deep depletion region, add the `DIRECT` parameter to access a more robust solution method.

Ramped Frequency At A Single Bias

For some applications, such as determining bipolar gain versus frequency, you need to ramp the frequency of the simulation. This is done using the following syntax:

```
1.  SOLVE VBASE=0.7 AC FREQ=1e9 FSTEP=1e9 NFSTEPS=10
2.  SOLVE VBASE=0.7 AC FREQ=1e6 FSTEP=2 MULT.F NFSTEPS=10
```

The first case ramps the frequency from 1GHz to 11GHz in 1GHz steps. A linear ramp of frequency is used and `FSTEP` is in Hertz. In the second example, a larger frequency range is desired and so a geometrical step of frequency is used. The `MULT.F` parameter is used to specify that `FSTEP` is a unitless multiplier for the frequency. This doubles the frequency in successive steps from 1MHz to 1.024GHz.

You can combine the following syntax for ramping the frequency of the AC signal with the syntax for ramping the bias. The frequency ramps are done at each bias point during the DC ramp.

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6 \
      FSTEP=2 MULT.F NFSTEPS=10
```

2.9.4 Transient Solutions

Transient solutions can be obtained for piecewise-linear, exponential, and sinusoidal bias functions. Transient solutions are used when a time dependent test or response is required. To obtain transient solutions for a linear ramp, specify the TSTOP, TSTEP, and RAMPTIME parameters.

Figure 2-16 shows the syntax of the transient voltage ramp. The RAMPTIME specifies the time that the linear ramp should obtain its final value. TSTOP specifies the time that solutions will stop. TSTEP specifies the initial step size.

Subsequent time steps are calculated automatically by Atlas. For example, the statement:

```
SOLVE VGATE=1.0 RAMPTIME=1E-9 TSTOP=10e-9 TSTEP=1e-11,
```

specifies that the voltage on the gate electrode will be ramped in the time domain from its present value to 1.0V over a period of 1 nanoseconds.

Time domain solutions are obtained for an additional 9 nanoseconds. An initial time step of 10 picoseconds is specified. Note that if you specify subsequent transient solutions, don't reset the time to zero.

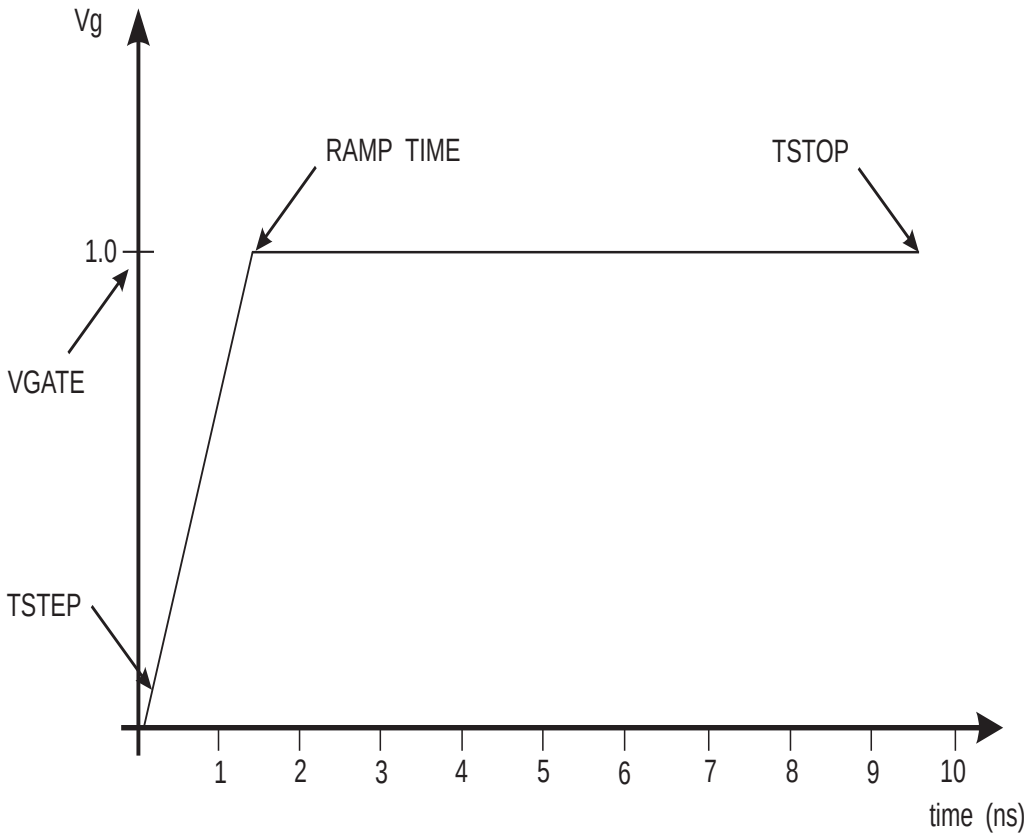


Figure 2-16 Diagram showing syntax of Transient Voltage Ramp in Atlas

2.9.5 Advanced Solution Techniques

Obtaining Solutions Around The Breakdown Voltage

Obtaining solutions around the breakdown voltage can be difficult using the standard Atlas approach. It requires special care when choosing voltage steps and interpreting the results. The curve tracer described in "The Curvetrace Capability" on page 91 is the most effective method in many cases.

A MOSFET breakdown simulation might be performed using the following standard syntax for ramping the drain bias. Note that the setting of CLIMIT is recommended for breakdown simulations when the pre-breakdown leakage is low.

```
IMPACT SELB

METHOD CLIMIT=1e-4

SOLVE VDRAIN=1.0 VSTEP=1.0 VFINAL=20.0 NAME=drain
```

If the breakdown were 11.5V, then convergence problems will be expected for biases higher than 11.0V. Although it depends on technology used, it is common for the breakdown curve to be flat up to a voltage very close to breakdown and then almost vertical. The current changes by orders of magnitude for very small bias increments.

This produces some problems for Atlas using the syntax described above. If the breakdown occurs at 11.5V, there are no solutions for voltages greater than this value. Atlas is trying to ramp to 20.0V. Therefore, it is inevitable that Atlas will fail to converge at some point. This is usually not a problem since by that point the breakdown voltage and curve have been obtained.

Above 11V, bias step reduction will take place due to the TRAP parameter. Atlas will continually try to increase the drain voltage above 11.5V and those points will fail to converge. But it will solve points asymptotically approaching Vds=11.5V until it reaches the limit set by the MAXTRAPS parameter. If you use the default of 4 traps, the minimum allowed voltage step will be $1.0\times(0.5)^4$ or 0.004V. This is normally enough accuracy for determining the breakdown point. But the simulation might not allow the current to reach a sufficiently high level before MAXTRAPS is needed.

Typically in device simulation, the breakdown point is determined once the current is seen to increase above the flat pre-breakdown leakage value by two orders of magnitude in a small voltage increment. If you want to trace the full breakdown curve up to high current values, apply more advanced techniques than the simple voltage ramp. Two of these techniques are described in the following subsections. These methods may use extra CPU time.

Using Current Boundary Conditions

In all of the examples considered in the basic description of the **SOLVE** statement, it was assumed that voltages were being forced and currents were being measured. Atlas also supports the reverse case through current boundary conditions. The current through the electrode is specified in the **SOLVE** statement and the voltage at the contact is calculated. Current boundary conditions are set using the **CONTACT** statement as described in Section 2.7 "Defining Material Parameters And Models".

The syntax of the **SOLVE** statement is altered once current boundary conditions are specified.

```
SOLVE IBASE=1e-6
```

The syntax above specifies a single solution at a given current.

```
SOLVE IBASE=1e-6 ISTEP=1e-6 IFINAL=5e-6 NAME=base
```

This sets a current ramp similar in syntax to the voltage ramp described earlier.

```
SOLVE IBASE=1e-10 ISTEP=10 IMULT IFINAL=1e-6 NAME=base
```

This is similar to the previous case. The IMULT parameter, however, is used to specify that ISTEP should be used as a multiplier for the current rather than a linear addition. This is typical for ramps of current since linear ramps are inconvenient when several orders of magnitude in current may need to be covered.

Important points to remember about current boundary conditions are that the problems of initial guess are more acute when very small (noise level) currents are used. Often, it is best to ramp the voltage until the current is above 1pA/μm and then switch to current forcing.

When interpreting the results, it is important to remember the calculated voltage on the electrode with current boundary conditions is stored as the internal bias. In other words, base int.bias in TonyPlot or vint.base in DeckBuild's extract syntax.

The Compliance Parameter

Compliance is a parameter used to limit the current or voltage through or on an electrode during a simulation. You can set an electrode compliance. After reaching this limit, the bias sweep will stop. This is similar to parametric device testing when we stop a device from being over-stressed or destroyed. The compliance refers to the maximum resultant current or voltage present after obtaining a solution. If you set an electrode voltage, the compliance will then refer to the electrode current. If there are current boundary conditions, you can set a voltage compliance.

The statements:

```
SOLVE VGATE=1.0

SOLVE NAME=drain VDRAIN=0 VFINAL=2 VSTEP=0.2 COMPL=1E-6 CNAME=drain
```

solve for IV on the gate and then ramps the drain voltage towards 2V in 0.2V steps. The simulation will stop if it reaches 1μA/μm of drain current before Vd=2V. Thus, as in parametric testing, you can define a particular level and set the simulation to solve up to that point. Once Atlas reaches the compliance limit, it will simulate the next statement line in the command file.

The Curvetrace Capability

The automatic curve tracing algorithm can be invoked to enable Atlas to trace out complex IV curves. The algorithm can automatically switch from voltage to current boundary conditions and vice versa. You can use a single SOLVE statement to trace out complex IV curves, such as breakdown curves and CMOS latch-up including the snapback region and second breakdown. The algorithm is based upon a dynamic load line approach.

For example, typical **CURVETRACE** and **SOLVE** statements to trace out an IV curve for the breakdown of a diode would look like:

```
CURVETRACE CONTR.NAME=cathode STEP.INIT=0.5 NEXTST.RATIO=1.2 \
           MINCUR=1e-12 END.VAL=1e-3 CURR.CONT
SOLVE CURVETRACE
```

CONTR.NAME specifies the name of the electrode, which is to be ramped. STEP.INIT specifies the initial voltage step. NEXTST.RATIO specifies the factor used to increase the voltage step in areas on the IV curve away from turning points. MINCUR sets a small current value above that activates the dynamic load line algorithm. Below the MINCUR level, using the STEP.INIT and NEXTST.RATIO determines the next solution bias. END.VAL stops the tracing if the voltage or current of the ramped electrode equals or exceeds END.VAL. Using VOLT.CONT or CURR.CONT specify whether END.VAL is a voltage or current value.

When you plot the log file created by the **CURVETRACE** statement in TonyPlot, select the internal bias labeled int.bias for the ramped electrode instead of plotting the applied bias, which is labeled Voltage.

2.9.6 Using DeckBuild To Specify SOLVE Statements

The DeckBuild Solve menu can be used generate **SOLVE** statements. The menu has a spreadsheet style entry. To access this menu, select the **Command/Solutions/Solve...** button in DeckBuild. To define a test, click the right mouse button in the worksheet and select the **Add new row** option. This will add a new row to the worksheet. This procedure should be repeated once per electrode in your device structure. The entry for each cell can then be edited to construct a **SOLVE** statement.

Some cells require the selection using a pop menu or the entry of numerical values. The electrode name is specified in the first cell. To edit the electrode name, use a popup menu by pressing the right mouse button on the cell. The second cell specifies whether the electrode will be a voltage (V), current (I) or charge (Q) controlled. The third cell specifies whether the **SOLVE** statement is to be a single DC solve (CONST), a swept DC variable (VAR1), a stepped DC variable (VAR2), or a transient solution (PULSE). The remaining cells specify the parameter values that are required for the type of solution desired.

The pop-up window to specify the solution file names are accessed through the **Props...** button. You can construct several **SOLVE** statements to create solve sequences, which define a test. This test may be saved in a file and read in using the **Save...** and **Load...** buttons.

2.10 Interpreting The Results

As already described in Section 2.2 “Atlas Inputs and Outputs”, Atlas produces three different types of output files. These files are also described in the following sections.

2.10.1 Run-Time Output

Run-time output is provided at the bottom of the DeckBuild Window. If it’s run as a batch job, the run-time output can be stored to a file.

Errors occurring in the run-time output will be displayed in this window. Note that not all errors will be fatal (as DeckBuild tries to interpret the file and continue). This may cause a statement to be ignored, leading to unexpected results. We recommend that you check the run-time output of any newly created input file the first time it runs to intercept any errors.

If you specify the PRINT option within the **MODELS** statement, the details of material parameters and constants and mobility models will be specified at the start of the run-time output. This is a useful way of checking what parameters values and models are being applied in the simulation. We recommend that you always specify **MODELS** PRINT in input files.

During **SOLVE** statements, the error numbers of each equation at each iteration are displayed (this is a change from the previous Atlas version). It isn’t vital for you to understand the iteration information. It can, however, provide important insights in the case of convergence problems.

The output can be interpreted as follows:

```
proj      psi     n      p       psi      n         p
direct    x       x      x       rhs      rhs       rhs

 i   j  m  -5.00*  -5.00*  -5.00*  -26.0*  -17.3*  -17.3*

 - - - - - - - - - - - - - - - - - - - - - - - - - - -

 1      N   -      -2.934 -1.932   -25.2  -10.19 -9.876
            1.932

 2      N   -      -5.64* -4.267   -28.8* -16.67 -15.47
            4.741

 3      A   -      -11.7* -9.63*   -28.8* -16.67 -18.0*
            11.3*

Electrode  Va(V)         Jn(A/um)      Jp(A/um)      Jc(A/um)      Jt(A/um)
=============================================================================
gate       0.000e+00     -0.000e+00    -0.000e+00    0.000e+00     0.000e+00
source     0.000e+00     -3.138e-13    -1.089e-35    -3.138e-13    -3.138e-13
drain      1.000e-01     3.139e-13     1.076e-23     3.139e-13     3.139e-13
substrate  0.000e+00     -6.469e-19    -8.853e-17    -8.918e-17    -8.918e-17
```

The top left value, proj, indicates the initial guess methodology used. The default projection method is used here. Alternatives are previous, local, or init. The second value, direct, indicates the solver type. This will either be direct or iterative.

The first three column headings: i, j, and m indicate the iteration numbers of the solution and the solution method. i indicates the outer loop iteration number for decoupled solutions. j indicates the inner loop number. m indicates the solution method by a single letter, which are:

- G = gummel
- B = block
- N = newton
- A = newton with autonr
- S = coupled Poisson-Schrodinger solution

The remaining column headings indicate which column lists the XNORM and RHSNORM errors for the equations being solved. See Chapter 21 "Numerical Techniques", Section 21.5.7 "Error Measures" for a full description of these errors. The values printed in each error column under the hashed line are the logarithm to base 10 of the error. Earlier PISCES versions would print the floating point value. The values printed above the dashed line in each column are the tolerances used.

When the star * symbol appears as the least significant digit in the number, it means this error measure has met its tolerance.

After achieving convergence, Atlas lists the results by electrode. The column, *Va*, lists the voltage at the contact surface. This will differ from the applied voltage if external resistors or the curvetracer are used. All relevant current components are listed. Here, only electron, hole, conduction, and total currents are given. In other modes of simulation, these columns may differ.

The output of AC analysis, MixedMode, and 3D simulations differ from this standard. See Chapter 13 "MixedMode: Mixed Circuit and Device Simulator" for more information about MixedMode.

Atlas may produce a very large amount of run-time output for complex simulations. You can save run-time output to a file as shown in Section 2.3 "Modes of Operation".

2.10.2 Log Files

Log files store the terminal characteristics calculated by Atlas. These are current and voltages for each electrode in DC simulations. In transient simulations, the time is stored. In AC simulations, the small signal frequency and the conductances and capacitances are saved. For example, the statement:

```
LOG OUTF=<FILENAME>
```

is used to open a log file. Terminal characteristics from all **SOLVE** statements after the **LOG** statement are then saved to this file along with any results from the **PROBE** statement.

To not save the terminal characteristics to this file, use another **LOG** statement with either a different log filename or the OFF parameter.

Typically, a separate log file should be used for each bias sweep. For example, separate log files are used for each gate bias in a MOS Id/Vds simulation or each base current in a bipolar Ic/Vce simulation. These files are then overlaid in TonyPlot.

Log files contain only the terminal characteristics. They are typically viewed in TonyPlot. Parameter extraction of data in log files can be done in DeckBuild. Log files cannot be loaded into Atlas to re-initialize the simulation.

Units of Currents In Log files

Generally, the units of current are written into the log file. In TonyPlot, it is Amperes per micron. This is because Atlas is a two-dimensional simulator. It sets the third dimension (or Z direction) to be one micron. Thus, if you compare Atlas 2D simulation results for a MOSFET versus the measured data from a MOSFET of width 20 micron, you need to multiply the current in the log file by 20.

There are four exceptions:

- In the 3D modules of Atlas, the width is defined in the 3D structure. Therefore, the units of the current are Amperes.
- In MixedMode, set the width of devices. The current is also in Amperes.
- When cylindrical coordinates are used, the current written to the log file is integrated through the cylinder and is also in Amperes.
- When the `WIDTH` parameter in the **MESH** statement is used, the current is scaled by this factor and is in Amperes.

Similar rules apply for the capacitance and conductance produced by AC simulations. These are usually in 1/(ohms.microns) and Farads/micron respectively.

Utmost Interface

Atlas log files can be read directly into the batch mode, Utmost. The following commands in Utmost are used to read in a set of IV curves stored in separate log files.

```
INIT INF=<filename> MASTER

INIT INF=<filename> MASTER APPEND
```

Note: The **UTMOST** statement in Atlas is no longer recommended for interfacing to UTMOST.

2.10.3 Parameter Extraction In DeckBuild

The **EXTRACT** command is provided within the DeckBuild environment. It allows you to extract device parameters. The command has a flexible syntax that allows you to construct specific **EXTRACT** routines. **EXTRACT** operates on the previous solved curve or structure file. By default, **EXTRACT** uses the currently open log file. To override this default, supply the name of a file to be used by **EXTRACT** before the extraction routine. For example:

```
EXTRACT INIT INF="<filename>"
```

A typical example of using **EXTRACT** is the extraction of the threshold voltage of an MOS transistor. In the following example, the threshold voltage is extracted by calculating the maximum slope of the I_d / V_g curve, finding the intercept with the X axis and then subtracting half of the applied drain bias.

```
EXTRACT     NAME="nvt"     XINTERCEPT(MAXSLOPE(CURVE     (V."GATE",
(I."DRAIN"))) \

-(AVE(V."DRAIN"))/2.0)
```

The results of the extraction will be displayed in the run-time output and will be by default stored in the file `results.final`. To store the results in a different file at the end of **EXTRACT** command, use the following option:

```
EXTRACT....DATAFILE="<filename>"
```

Cut-off frequency and forward current gain are of particular use as output parameters. These functions can be defined as follows:

```
# MAXIMUM CUTOFF FREQUENCY

EXTRACT            NAME="FT_MAX"            MAX(G."COLLECTOR""BASE"/
(6.28*C."BASE""BASE"))


#FORWARD CURRENT GAIN

EXTRACT NAME="PEAK GAIN" MAX(I."COLLECTOR"/ I."BASE")
```

Note: Over 300 examples are supplied with Atlas to provide many practical examples of the use of the **EXTRACT** statement.

AC Parameter Extraction

You can perform basic analysis of the capacitance and conductance matrix produced by Atlas using DeckBuild or TonyPlot. The capacitance between gate and drain will be labeled as `Cgate>drain` in TonyPlot or `c."gate""drain"` in DeckBuild's **EXTRACT**.

The total capacitance on any electrode is defined as `Celectrode>electrode`. Thus, the magnitude of `Cgate>gate` is the total gate capacitance.

The **LOG** statement also includes options for small-signal, two-port RF analysis including s-parameter extraction. The solutions are saved into the log file and in the run-time output. The list of options for RF analysis are:

```
s.param, y.param, h.param, z.param, abcd.param gains
```

Terminal impedance and parasitics are accounted for by adding any of the following:

```
impedance=<val>, rin=<val>, rout=<val>, rcommon=<val> or

rground=<val>, lin=<val>, lout=<val>,

lcommon=<val> or lground=<val>, width=<val>
```

The width defaults to 1μm and impedance defaults to 50Ω. All parasitics default to zero.

The Stern stability factor, `k`, is calculated along with current gain (h21), GUmax, and GTmax when the `GAINS` option is added to the `LOG` statement.

The run-time output for AC analysis has been modified to only list the analysis frequency and electrode conductance/capacitance values. If one of the two-port options is added to the `LOG` statement (e.g., `S.PARAM`), the two-port parameters are also included in the run-time output.

Note: AC parameter conversion utilities in the **UTMOST** statement have been discontinued.

See Appendix C "RF and Small Signal AC Parameter Extraction" for more information.

Additional Functions

EXTRACT has two additional important functions. The first function is to provide data for the VWF database. In other words, to store device parameters to the VWF database, you must evaluate them using **EXTRACT**. The second function is when using the DeckBuild Optimizer to tune parameters, use the **EXTRACT** statements as the optimization targets.

2.10.4 Functions In TonyPlot

The **Edit/Functions** Menu in TonyPlot allows you to specify and plot functions of the terminal characteristics in the **Graph Function** text fields. For example, you can calculate transconductance using the following function:

```
dydx (drain current, gate voltage)
```

Current gain can be evaluated as:

```
collector current / base current
```

When creating functions, the key to correct syntax is that the name for any variable in a function is the same as that in the **Y Quantities** list on the **Display** menu.

2.10.5 Solution Files

Solution files or structure files provide an image of the device at a particular bias point (DC solution or transient solution point). This gives you the ability to view any evaluated quantity within the device structure in question, from doping profiles and band parameters to electron concentrations and electric fields. These files should be plotted using TonyPlot.

The syntax used to generate these files is of two forms:

```
(1)   SAVE OUTFILE=<filename>

(2)   SOLVE .... OUTFILE=<filename>.sta MASTER [ONEFILEONLY]
```

Here in the second form, a file named `<filename>` will be saved with data from the previously solved bias point.

In this case, a structure file will be saved at each bias point solved in the solve statement. The last letter of the file name will be automatically incremented alphabetically (i.e., `*.sta`, `*.stb`, `*.stc...`, and so on). If you need the solution for the last bias point, you can add the `onefileonly` parameter to the command. The `<filename>.sta` file will be overwritten at each solution point.

Structure files can be large (1 - 2 MB), depending on the mesh density and quantities saved. It is recommended that unwanted structure files be deleted.

If many solution files have been written from a long simulation, it is often confusing to find out which solution file belongs to which bias point or transient time. The solution files should be plotted in TonyPlot. In TonyPlot, you can create 2D contour plots and 1D cutlines. To find out the bias of any solution file in TonyPlot, select the plot, and press **B** on the keyboard.

Interpreting Contour Plots

Some quantities saved in the solution files are not evaluated at the node points during solutions. They are evaluated at the center of the sides of each triangle in the mesh. Values of quantities at each node are derived from averaging the values from the sides of triangles connected to that node. The weighting method used to do the averaging can be selected with options in the **OUTPUT** statement. It is possible that for some meshes, smoother contour plots can be obtained by choosing a non-default averaging method.

When interpreting the contour plots, it's important to remember that the solution file contains values only at each node point. The color fills seen in TonyPlot are simply interpolations based on the node values. This may lead to occasional strange contour values. In these cases, check the node values using the probe in TonyPlot.

The primary solution variables (potential, carrier concentration, lattice temperature, and carrier temperatures) are calculated on the nodes of the Atlas mesh. Therefore, they are always correct in TonyPlot. But since Atlas doesn't use nodal values of quantities, such as electric field and mobility, the actual values used in calculations cannot be determined from TonyPlot or the structure files. The **PROBE** statement allows you to directly probe values at given locations in the structure. This provides the most accurate way to determine the actual values used in Atlas calculations.

Customizing Solution Files (OUTPUT Statement)

Several quantities are saved by default within a structure file. For example, doping, electron concentration, and potential. You can also specify additional quantities (such as conduction band potential) by using the **OUTPUT** statement. This must precede the **SAVE** statement in question. For example, to save the conduction and valence band potentials, the following command would be used at some point before the relevant **SAVE**.

```
OUTPUT CON.BAND VAL.BAND
```

Saving Quantities from the Structure at each Bias Point (PROBE statement)

Structure files provide all data from the structure at a single bias point. The log files provide terminal characteristics for a set of bias points. Use the **PROBE** statement to combine these and allow certain structural quantities to be saved at each bias point.

The **PROBE** statement allows you to specify quantities to be saved at given XY locations. You can also save the maximum or minimum of certain quantities. The value from the **PROBE** at each bias point in DC or timestep in transient mode is saved to the log file. The syntax:

```
PROBE NAME=mycarriers N.CONC X=1 Y=0.1
```

saves the electron concentration at (1, 0.1) for each solution in the log file. When TonyPlot displays the log file, the value will be labelled `mycarriers`. You can then plot `mycarriers` versus terminal bias or current or other probed quantities.

Certain direction dependent quantities, such as electric field and mobility, can be probed. In these cases, specify a component of the vector quantity using one of the `DIR`, `V.X`, `V.Y`, `V.Z`, `V.MAG`, `V.XANG`, or `V.ZANG` parameters.

The **PROBE** statement provides the only way to extract the actual values of quantities, which are calculated along the sides of each triangle in Atlas. The **PROBE** statement actually stored the triangle side value closest to the probed location, while taking into account the direction for vector quantities.

Note: Specifying the probe location exactly at a material or region interface will often lead to erroneous results. It is best to slightly offset the location of the probe inside the material or region of interest.

Re-initializing Atlas at a Given Bias Point

Each **SOLVE** statement will begin with the device biased at the previous value solved. To begin a solution at a previously solved bias point, re-load the structure file saved at that point. This is accomplished in the following manner:

```
LOAD INFILE=<filename> MASTER
```

Information about that solution point will be displayed in the Output Window.

This command is useful for solving a set of I/V curves. For example, to solve a family of Id / Vd (at various Vg), ramp the gate with zero drain bias. A structure file is then saved at each desired value of Vg. These structure files can be reloaded in turn while a Vd sweep is performed.

Note: An Atlas input file cannot start with a **LOAD** statement. Before loading the structure file, use the **MESH** statement to load the device mesh for the same structure. Also, the same **MODELS**, **MATERIAL**, and **CONTACT** settings are required when the files are saved by Atlas.

2.10.6 Technology Specific Issues in Atlas

This chapter was designed to give an overview to the basic use of Atlas without regard to the details required for a given technology. Requirements for Atlas simulation vary considerably. The needs of the sub-micron MOS device engineer, the 1000V power device engineer, and the III-V RF device engineer differ and cannot all be covered in one chapter. Silvaco provides many references to individual technology problems using Atlas. These are:

- A library of standard examples that can be accessed on-line from DeckBuild. Look at these examples not only for their technology but also related ones. For example, different aspects of high frequency analysis is covered in the MESFET and silicon bipolar example sections.
- The chapters for each Atlas module in this manual.
- The Simulation Standard, a newsletter distributed by Silvaco. To make sure you're on the mailing list, contact your local Silvaco office or go to www.silvaco.com.
- The Silvaco website also provides detailed information. It contains on-line versions of the articles in our newsletter, on-line searchable index of the examples, links to other TCAD web sites, and a section on solutions to known problems with all Silvaco programs.
- For more information about suggested technology specific strategies, contact your local Silvaco support engineer at support@silvaco.com.

Chapter 3
Physics

3.1 Basic Semiconductor Equations

Years of research into device physics have resulted in a mathematical model that operates on any semiconductor device [244]. This model consists of a set of fundamental equations, which link together the electrostatic potential and the carrier densities, within some simulation domain. These equations, which are solved inside any general purpose device simulator, have been derived from Maxwell's laws and consist of Poisson's Equation (see Section 3.1.1 "Poisson's Equation"), the continuity equations (see Section 3.1.2 "Carrier Continuity Equations") and the transport equations (see Section 3.1.3 "The Transport Equations"). Poisson's Equation relates variations in electrostatic potential to local charge densities. The continuity and the transport equations describe the way that the electron and hole densities evolve as a result of transport processes, generation processes, and recombination processes.

This chapter describes the mathematical models implemented in Atlas. Note that a discretization of the equations is also performed so that they can be applied to the finite element grid used to represent the simulation domain.

3.1.1 Poisson's Equation

Poisson's Equation relates the electrostatic potential to the space charge density:

$$div(\varepsilon \nabla \psi) = -\rho \quad \text{3-1}$$

where ψ is the electrostatic potential, ε is the local permittivity, and ρ is the local space charge density. The reference potential can be defined in various ways. For Atlas, this is always the intrinsic Fermi potential ψ_i which is defined in the next section. The local space charge density is the sum of contributions from all mobile and fixed charges, including electrons, holes, and ionized impurities.

The electric field is obtained from the gradient of the potential (see Equation 3-2).

$$\vec{E} = -\nabla \psi \quad \text{3-2}$$

3.1.2 Carrier Continuity Equations

The continuity equations for electrons and holes are defined by equations:

$$\frac{\partial n}{\partial t} = \frac{1}{q} div \vec{J_n} + G_n - R_n \quad \text{3-3}$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} div \vec{J_p} + G_p - R_p \quad \text{3-4}$$

where n and p are the electron and hole concentration, $\vec{J_n}$ and $\vec{J_p}$ are the electron and hole current densities, G_n and G_p are the generation rates for electrons and holes, R_n and R_p are the recombination rates for electrons and holes, and q is the magnitude of the charge on an electron.

By default Atlas includes both Equations 3-3 and 3-4. In some circumstances, however, it is sufficient to solve only one carrier continuity equation. The specification of which continuity equations are to be solved is performed in the **METHOD** statement by turning off any equation that is not to be solved. The syntax, ^ELECTRONS or ^HOLES, turns off the electron continuity equation and the hole continuity equation respectively.

3.1.3 The Transport Equations

Equations 3-1, 3-3, and 3-4 provide the general framework for device simulation. But further secondary equations are needed to specify particular physical models for: $\vec{J}_n$, $\vec{J}_p$, G_n, R_n, G_p and R_p.

The current density equations, or charge transport models, are usually obtained by applying approximations and simplifications to the Boltzmann Transport Equation. These assumptions can result in a number of different transport models such as the drift-diffusion model, the Energy Balance Transport Model or the hydrodynamic model. The choice of the charge transport model will then have a major influence on the choice of generation and recombination models.

The simplest model of charge transport that is useful is the Drift-Diffusion Model. This model has the attractive feature that it does not introduce any independent variables in addition to ψ, n and p. Until recently, the drift-diffusion model was adequate for nearly all devices that were technologically feasible. The drift-diffusion approximation, however, becomes less accurate for smaller feature sizes. More advanced energy balance and hydrodynamic models are therefore becoming popular for simulating deep submicron devices. Atlas supplies both drift-diffusion and advanced transport models.

The charge transport models and the models for generation and recombination in Atlas make use of some concepts associated with carrier statistics. These concepts are summarized in a further section of this chapter that deals with the carrier statistics.

Drift-Diffusion Transport Model

Derivations based upon the Boltzmann transport theory have shown that the current densities in the continuity equations may be approximated by a drift-diffusion model [286]. In this case, the current densities are expressed in terms of the quasi-Fermi levels ϕ_n and ϕ_p as:

$$\vec{J}_n = -q\mu_n n \nabla \phi_n \qquad 3\text{-}5$$

$$\vec{J}_p = -q\mu_p p \nabla \phi_p \qquad 3\text{-}6$$

where μ_n and μ_p are the electron and hole mobilities. The quasi-Fermi levels are then linked to the carrier concentrations and the potential through the two Boltzmann approximations:

$$n = n_{ie} exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \qquad 3\text{-}7$$

$$p = n_{ie} exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \qquad 3\text{-}8$$

where n_{ie} is the effective intrinsic concentration and T_L is the lattice temperature. These two equations may then be re-written to define the quasi-Fermi potentials:

$$\phi_n = \psi - \frac{kT_L}{q}\ln\frac{n}{n_{ie}} \quad 3\text{-}9$$

$$\phi_p = \psi + \frac{kT_L}{q}\ln\frac{p}{n_{ie}} \quad 3\text{-}10$$

By substituting these equations into the current density expressions, the following adapted current relationships are obtained:

$$\vec{J}_n = qD_n\nabla n - qn\mu_n\nabla\psi - \mu_n n(kT_L\nabla(\ln n_{ie})) \quad 3\text{-}11$$

$$\vec{J}_p = -qD_p\nabla p - qp\mu_p\nabla\psi + \mu_p p(kT_L\nabla(\ln n_{ie})) \quad 3\text{-}12$$

The final term accounts for the gradient in the effective intrinsic carrier concentration, which takes account of bandgap narrowing effects. Effective electric fields are normally defined whereby:

$$\vec{E}_n = -\nabla\left(\psi + \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}13$$

$$\vec{E}_p = -\nabla\left(\psi - \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}14$$

which then allows the more conventional formulation of drift-diffusion equations to be written (see Equations 3-15 and 3-16).

$$\vec{J}_n = qn\mu_n\vec{E}_n + qD_n\nabla n \quad 3\text{-}15$$

$$\vec{J}_p = qp\mu_p\vec{E}_p - qD_p\nabla p \quad 3\text{-}16$$

It should be noted that this derivation of the drift-diffusion model has tacitly assumed that the Einstein relationship holds. In the case of Boltzmann statistics this corresponds to:

$$D_n = \frac{kT_L}{q}\mu_n \quad 3\text{-}17$$

$$D_p = \frac{kT_L}{q}\mu_p \quad 3\text{-}18$$

If Fermi-Dirac statistics are assumed for electrons, Equation 3-17 becomes:

$$D_n = \frac{\left(\frac{kT_L}{q}\mu_n\right)F_{1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn}-\varepsilon_C]\right\}}{F_{-1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn}-\varepsilon_C]\right\}} \quad 3\text{-}19$$

where F_α is the Fermi-Dirac integral of order α and ε_{Fn} is given by $-q\phi_n$. An analogous expression is used for holes with Fermi-Dirac statistics.

Note: See Section 3.2.9 "Bandgap Narrowing" for more information on the effects resulting from bandgap narrowing and their implementation.

Energy Balance Transport Model

A higher order solution to the general Boltzmann Transport Equation consists of an additional coupling of the current density to the carrier temperature, or energy. The current density expressions from the drift-diffusion model are modified to include this additional physical relationship. The electron current and energy flux densities are then expressed as:

$$\vec{J_n} = qD_n\nabla n - q\mu_n n\nabla\psi + qnD_n^T\nabla T_n \quad 3\text{-}20$$

$$\vec{S_n} = -K_n\nabla T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}_n T_n \quad 3\text{-}21$$

$$\vec{J_p} = -qD_p\nabla p - q\mu_p p\nabla\psi - qpD_p^T\nabla T_p \quad 3\text{-}22$$

$$\vec{S_p} = -K_p\nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}_p T_p \quad 3\text{-}23$$

where T_n and T_p represent the electron and hole carrier temperatures and S_n and S_p are the flux of energy (or heat) from the carrier to the lattice. The energy balance transport model includes a number of very complex relationships and therefore a later section of this chapter has been devoted to this model.

3.1.4 Displacement Current Equation

For time domain simulation, the displacement current is calculated and included in the structure, log file and the run time output. The expression for displacement current is given as:

$$\overrightarrow{J_{dis}} = \varepsilon\left(\frac{\partial\bar{E}}{\partial t}\right) \quad 3\text{-}24$$

3.2 Basic Theory of Carrier Statistics

3.2.1 Fermi-Dirac and Boltzmann Statistics

Electrons in thermal equilibrium at temperature T_L with a semiconductor lattice obey Fermi-Dirac statistics. That is the probability $f(\varepsilon)$ that an available electron state with energy ε is occupied by an electron is:

$$f(\varepsilon) = \frac{1}{1 + \exp\left(\frac{\varepsilon - E_F}{kT_L}\right)} \quad 3\text{-}25$$

where E_F is a spatially independent reference energy known as the Fermi level and k is Boltzmann's constant.

In the limit, $\varepsilon - E_F >> kT_L$, Equation 3-25 can be approximated as:

$$f(\varepsilon) = \exp\left(\frac{E_F - \varepsilon}{kT_L}\right) \quad 3\text{-}26$$

Statistics based on the use of Equation 3-26 are referred to as Boltzmann statistics [371, 146]. The use of Boltzmann statistics instead of Fermi-Dirac statistics makes subsequent calculations much simpler. The use of Boltzmann statistics is normally justified in semiconductor device theory, but Fermi-Dirac statistics are necessary to account for certain properties of very highly doped (degenerate) materials.

The Fermi-Dirac statistics have been implemented in Atlas in a similar form to Boltzmann statistics.

The remainder of this section outlines derivations and results for the simpler case of Boltzmann statistics which are the default in Atlas. You can have Atlas use Fermi-Dirac statistics by specifying the FERMIDIRAC parameter in the **MODELS** statement.

3.2.2 Effective Density of States

Integrating the Fermi-Dirac statistics over a parabolic density of states in the conduction and valence bands, whose energy minimum is located at energies E_C and E_V respectively, yields the following expressions for the electron and hole concentrations:

$$n = N_C F_{1/2}\left(\frac{E_F - E_C}{kT_L}\right) \quad 3\text{-}27$$

$$p = N_V F_{1/2}\left(\frac{E_v - E_F}{kT_L}\right) \quad 3\text{-}28$$

where $F_{1/2}(\eta)$ is referred to as the Fermi-Dirac integral of order 1/2. If Equation 3-26 is a good approximation, then Equations 3-27 and 3-28 can be simplified to

$$n = N_C \exp\left(\frac{E_F - E_C}{kT_L}\right) \quad 3\text{-}29$$

$$p = N_V \exp\left(\frac{E_v - E_F}{kT_L}\right) \quad 3\text{-}30$$

which are referred to as the Boltzmann approximation.

N_C and N_V are referred to as the effective density of states for electrons and holes and are given by:

$$N_C(T_L) = 2\left(\frac{2\pi\, m_e^* kT_L}{h^2}\right)^{\frac{3}{2}} M_c = \left(\frac{T_L}{300}\right)^{\mathtt{NC.F}} \mathtt{NC300} \quad 3\text{-}31$$

$$N_V(T_L) = 2\left(\frac{2\pi\, m_h^* kT_L}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_L}{300}\right)^{\mathtt{NV.F}} \mathtt{NV300} \quad 3\text{-}32$$

where M_c is the number of equivalent conduction band minima. NC300 and NV300 are user-definable on the **MATERIAL** statement as shown in Table 3-1.

In some circumstances, the lattice temperature, T_L, is replaced by the electron temperature, T_n, in Equation 3-31 and hole temperature, T_p, in Equation 3-32.

Table 3-1 User-Definable Parameters for the Density of States

Statement	Parameter	Default	Units
MATERIAL	NC300	2.8×10^{19}	cm^{-3}
MATERIAL	NV300	1.04×10^{19}	cm^{-3}
MATERIAL	NC.F	1.5	
MATERIAL	NV.F	1.5	

3.2.3 Intrinsic Carrier Concentration

Multiplying Equations 3-29 and 3-30 yields:

$$np = n_{ie}^2 \quad 3\text{-}33$$

where n_{ie} is the intrinsic carrier concentration and is given for Boltzmann statistics by:

$$n_{ie} = \sqrt{N_C N_V} \exp\left(\frac{-E_g}{2kT_L}\right) \quad 3\text{-}34$$

$E_g = E_C - E_V$ is the band-gap energy.

For intrinsic (undoped) material, $p = n$. By equating Equations 3-29 and 3-30 and solving for E_F yields:

$$E_F = E_i = -q\psi_i = \frac{E_C + E_V}{2} + \left(\frac{kT_L}{2}\right) 1n\left(\frac{N_v}{N_c}\right) \quad 3\text{-}35$$

where E_i is the Fermi level for intrinsic doped silicon, and ψ_i is the intrinsic potential. Equation 3-35 also defines the intrinsic potential under non-equilibrium conditions. As indicated previously, for Atlas the ψ used in Equation 3-1 is the intrinsic potential.

The electron and hole concentrations can be expressed in terms of the intrinsic carrier concentration as:

$$n = n_{ie} \exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \quad 3\text{-}36$$

$$p = n_{ie} \exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \quad 3\text{-}37$$

where ψ is the intrinsic potential and ϕ is the potential corresponding to the Fermi level (i.e., $E_F = q\phi$).

The expression for intrinsic carrier concentration, n_{ie}, can be generalized to Fermi-Dirac statistics using Equations 3-27 and 3-28. Specifying the `NI.FERMI` parameter in the **`MODELS`** statement will cause Atlas to calculate n_{ie} using Fermi-Dirac statistics.

3.2.4 Evaluation of Fermi-Dirac Integrals

In addition to the Fermi-Dirac integral of order $\frac{1}{2}$ as used in Equations 3-27 and 3-28, Atlas also needs to evaluate the Fermi-Dirac integrals of order $\frac{-3}{2}, \frac{-1}{2}, \frac{3}{2}$. There are simple, quickly calculated, but relatively poor approximations available for these integrals. You can also evaluate them to machine precision; however, the amount of computation required makes the evaluations relatively slow. Atlas uses a Rational Chebyshev approximation scheme, which is efficient in terms of CPU use and also has accuracy close to the machine precision. In the worst case, the approximation differs from the actual values by 1 part in 10^{10} and is typically much better.

3.2.5 Rules for Evaluation of Energy Bandgap

In the Atlas simulator, there are several ways to evaluate the value of energy bandgap for a given material. To uniquely define the approach used, there is a set of rules that are followed in a hierarchical manner. The following is a description of this hierarchy:

1. For each material or material system, there may be a default approach. To determine the default for a given material or material system, refer to the descriptions given in Section 6.4 "Material Dependent Physical Models". If the material system is not listed in this section, then look for default energy bandgap model constants in Appendix B "Material Systems".
2. If you set the EG300 parameter of the **MATERIAL** statement to a value greater than zero and you do not set the PASSLER parameter of the **MODELS** statement, the universal energy bandgap model described in Section 3.2.6 "The Universal Energy Bandgap Model" will be used.
3. If you set the EG300 parameter of the **MATERIAL** statement to a value greater than zero and you set the PASSLER parameter of the **MODELS** statement, then Passler's model for temperature dependent energy bandgap described in Section 3.2.7 "Passler's Model for Temperature Dependent Bandgap" will be used.
4. If you specify the K.P parameter of the **MODELS** statement for materials in the InAlGaN system, then the bandgap is taken as the minimum transition energy calculated following the approach given by the strained zincblende tow band model described in Section 3.9.8 "Strained Two-Band Zincblende Model for Gain and Radiative Recombination" or the strained wurtzite 3 band model described in Section 3.9.10 "Strained Wurtzite Three-Band Model for Gain and Radiative Recombination".
5. If you assign the F.BANDCOMP parameter of the **MATERIAL** statement to a valid filename, the C interpreter function for energy bandgap will be used.
6. If you assign the EG12BOW parameter to a non-zero value, the general ternary bowing model described in Section 3.2.8 "General Ternary Bandgap Model with Bowing" will be used.

3.2.6 The Universal Energy Bandgap Model

The temperature dependence of the bandgap energy is modeled in Atlas as follows [307]:

$$E_g(T_L) = E_g(0) - \frac{\mathrm{EGALPHA}(T_L^2)}{T_L + \mathrm{EGBETA}} = \mathrm{EG300} + \mathrm{EGALPHA}\left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}}\right] \quad \text{3-38}$$

You can specify EG300, EGALPHA and EGBETA parameters in the **MATERIAL** statement (see Table 3-2).

Table 3-2 User-Specifiable Parameters for Equation 3-38

Statement	Parameter	Default	Units
MATERIAL	EG300	1.08	eV
MATERIAL	EGALPHA	4.73×10^{-4}	eV/K
MATERIAL	EGBETA	636	K

Table 3-2 User-Specifiable Parameters for Equation 3-38			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	NC300	2.8×10^{19}	cm^{-3}
MATERIAL	NV300	1.04×10^{19}	cm^{-3}

The default values are material dependent, which can be found in Appendix B "Material Systems". Table 3-2 displays the defaults for Silicon only.

3.2.7 Passler's Model for Temperature Dependent Bandgap

An alternative temperature dependent bandgap model by Passler [238] can be enabled by the PASSLER parameter of the **MODELS** statement.

The bandgap is given by

$$Eg(T) = Eg(0) - \frac{A.PASSLER \cdot T.PASSLER}{2} \left\{ \left[\left(\frac{2 \cdot T}{T.PASSLER} \right)^{1/P.PASSLER} + 1 \right]^{P.PASSLER} - 1 \right\} \quad 3\text{-}39$$

where $Eg(T)$ is the lattice temperature dependent bandgap, T is the lattice temperature, A.PASSLER, T.PASSLER and P.PASSLER are user-specified parameters on the **MATERIAL** statement and Eg(0) is given by

$$Eg(0) = EG300 - \frac{A.PASSLER \cdot T.PASSLER}{2} \left\{ \left[\left(\frac{2 \cdot 300}{T.PASSLER} \right)^{1/P.PASSLER} + 1 \right]^{P.PASSLER} - 1 \right\} \quad 3\text{-}40$$

where EG300 is a user specified parameter on the **MATERIAL** statement.

3.2.8 General Ternary Bandgap Model with Bowing

For Ternary compound semiconductors, you can use a bandgap model that continuously and non-linearly interpolates between the bandgaps of the two binary extremes in composition. To enable this model, you must specify the EG12BOW, EG1300 and EG2300 parameters of the **MATERIAL** statement. EG1300 is the 300K bandgap of the binary compound at a composition fraction of x=0. EG2300 is the 300K bandgap of the binary compound at a composition fraction of x=1. EG12BOW is the bowing factor. The bandgap as a function of composition is given by Equation 3-41.

$$Eg = \mathtt{EG2300}*x + \mathtt{EG1300}*(1-x) - \mathtt{EG12BOW}*x*(1-x) \quad (3\text{-}41)$$

where x is the composition fraction.

You can include the effects of temperature in the Bowing model by specifying the EG1ALPH, EG1BETA, EG2ALPH and EG2BETA parameters of the **MATERIAL** statement. These parameters are used in Equations 3-72 through 3-74 to calculate the bandgap.

$$Eg_1 = \mathtt{EG1300} + \mathtt{EG1ALPH}\left[\frac{300^2}{300 + \mathtt{EG1BETA}} - \frac{T_L^2}{T_L + \mathtt{EG1BETA}}\right] \quad (3\text{-}42)$$

$$Eg_2 = \mathtt{EG2300} + \mathtt{EG2ALPH}\left[\frac{300^2}{300 + \mathtt{EG2BETA}} - \frac{T_L^2}{T_L + \mathtt{EG2BETA}}\right] \quad (3\text{-}43)$$

$$Eg = Eg_2*x + Eg_1(1-x) - \mathtt{EG12BOW}*x*(1-x) \quad (3\text{-}44)$$

3.2.9 Bandgap Narrowing

In the presence of heavy doping, greater than 10^{18}cm^{-3}, experimental work has shown that the pn product in silicon becomes doping dependent [296]. As the doping level increases, a decrease in the bandgap separation occurs, where the conduction band is lowered by approximately the same amount as the valence band is raised. In Atlas this is simulated by a spatially varying intrinsic concentration n_{ie} defined according to Equation 3-45.

$$n_{ie}^2 = n_i^2 \exp\left(\frac{\Delta Eg}{kT}\right) \quad (3\text{-}45)$$

Bandgap narrowing effects in Atlas are enabled by specifying the BGN parameter of the **MODELS** statement. These effects may be described by an analytic expression relating the variation in bandgap, ΔE_g, to the doping concentration, N. The expression used in Atlas is from Slotboom and de Graaf [297]:

$$\Delta E_g = \mathtt{BGN.E}\left\{ ln\frac{N}{\mathtt{BGN.N}} + \left[\left(ln\frac{N}{\mathtt{BGN.N}}\right)^2 + \mathtt{BGN.C}\right]^{\frac{1}{2}}\right\} \quad (3\text{-}46)$$

You can specify the BGN.E, BGN.N, and BGN.C parameters in the **MATERIAL** statement. The default values from Slotboom [297] and Klaassen [160] are shown in Table 3-3. Specify BGN or BGN.SLOTBOOM on the **MODELS** statement to choose the Slotboom set of defaults for use in Equation 3-46. Alternatively, specify BGN2 or BGN.KLAASSEN on the **MODELS** statement to use the Klaassen set of defaults in Equation 3-46. You can verify which values are being used with the PRINT parameter on the **MODELS** statement.

Table 3-3 User-Definable Parameters of Bandgap Narrowing Model

Statement	Parameter	Defaults (Slotboom)	Defaults (Klaassen)	Units
MATERIAL	BGN.E	9.0×10^{-3}	6.92×10^{-3}	eV
MATERIAL	BGN.N	1.0×10^{17}	1.3×10^{17}	cm^{-3}
MATERIAL	BGN.C	0.5	0.5	

The variation in bandgap is introduced to the other physical models by subtracting the result of Equation 3-47 from the bandgap, E_g. In addition an adjustment is also made to the electric field terms in the transport models as described earlier. The adjustment takes the form:

$$\vec{E}_n = -\nabla\left(\psi + \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}47$$

$$\vec{E}_p = -\nabla\left(\psi - \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}48$$

The variation in the energy bandgap is also applied partially to the electron affinity, χ. The effective electron affinity, χ_{eff} given as follows:

$$\chi_{eff} = \chi + \Delta E_g \times \text{ASYMMETRY} \quad 3\text{-}49$$

where ASYMMETRY is a user-specifiable asymmetry factor. You can specify the value of the asymmetry factor using the ASYMMETRY parameter of the **MATERIAL** statement.

Note: In addition to this in-built model for bandgap narrowing, Atlas allows you to use its C-Interpreter module. You can write an external file with C-code that defines an equation for bandgap narrowing. The filename is specified with the F.BGN parameter in the **MODELS** statement. See Appendix A "C-Interpreter Functions" for more information on C-Interpreter.

3.2.10 The Universal Bandgap Narrowing Model

A universal bandgap narrowing model has been suggested [246]. This model given in Equation 3-50 relates the change in bandgap to the material effective masses, m_c and m_v, and the static dielectric constant ε_s. The UBGN.C and UBGN.B parameters are fitting parameters to define in the **MATERIAL** statement. To enable the model, specify UBGN on the **MODELS** statement.

$$\Delta E_g = -\mathrm{UBGN.C}\left[\frac{\varepsilon_S^5}{N}\left(m_0\frac{m_c+m_v}{m_c m_v}+\mathrm{UBGN.B}T^2\frac{\varepsilon_s}{N}\right)\right]^{-\frac{1}{4}} \quad 3\text{-}50$$

In Equation 3-50, N is the net doping concentration. The default values for UBGN.C and UBGN.B are 3.9×10^{-5} eV $cm^{3/4}$ and 3.10×10^{12} cm^{-3} K^{-2}.

3.2.11 Bennett-Wilson and del Alamo Bandgap Models

These are variations of the Slotboom and de Graaf model. The Bennett-Wilson formula [27] is

$$\Delta E_g = -\mathrm{BGN.E}\left[log(N/\mathrm{BGN.N})\right]^2 \quad 3\text{-}51$$

and the del-Alamo formula [5], [306] is

$$\Delta E_g = -\mathrm{BGN.E}\left[log(N/\mathrm{BGN.N})\right] \quad 3\text{-}52$$

where N is the total doping concentration. In both cases, the formula is only used for N>BGN.N, otherwise it is set at zero. Table 3-4 shows the default values of BGN.E and BGN.N for Silicon.

.

Table 3-4 Silicon Default Values for Bennett-Wilson and del Alamo Bandgap models

Statement	Parameter	Units	Default (Bennett)	Default (Del Alamo)
MATERIAL	BGN.E	eV	6.84×10^{-3}	1.87×10^{-2}
MATERIAL	BGN.N	cm^{-3}	3.162×10^{18}	7.0×10^{17}

To enable the Bennett-Wilson model, use the BGN.BENNETT flag on the **MODELS** statement. To enable the del Alamo model, use the BGN.ALAMO flag on the **MODELS** statement.

3.2.12 Schenk Bandgap Narrowing Model

Some bandgap narrowing models are described in previous sections. In those models, the bandgap narrowing depends only on the doping level and temperature. High doping levels are required in these models to produce significant bandgap narrowing. It has been observed experimentally that electron-hole plasma densities in regions with low doping levels can also produce bandgap narrowing effects [221]. An analytical model of bandgap narrowing in Silicon was subsequently developed by Schenk [279], which takes into account both the contribution due to the electron-hole plasma and the contribution due to dopant-carrier interactions. These are referred to as the exchange-correlation (*xc*) contributions and the ionic contributions (*i*) respectively, and are denoted by Δ_t^{xc} and Δ_t^i, where *t* can be either *e* for electrons or *h* for holes. Schenk uses the excitonic Bohr radius as a length scale in his formulae. This is simply the radius of the orbit of the ground state of an electron and hole electrostatically bound to each other. It is obtained by multiplying the Bohr radius for the hydrogen atom (5.29 nanometers) by the relative dielectric permittivity of Silicon and dividing by the reduced mass of the excitonic system. The effective masses of electrons and holes in this model can be set by the parameters BGN.SHNK.ME and BGN.SHNK.MH on the MATERIAL statement. The reduced mass is given by

$$\mu = \frac{\text{BGN.SHNK.ME} \times \text{BGN.SHNK.MH}}{(\text{BGN.SHNK.ME} + \text{BGN.SHNK.MH})} \quad \text{3-53}$$

and the excitonic Bohr radius in centimetres is calculated from

$$\alpha_{ex} = \frac{5.29 \times 10^{-9} \text{BGN.SHNK.EPS}}{\mu} \quad \text{3-54}$$

The excitonic Rydberg energy is used as an energy scale. This is obtained by multiplying the binding energy of the ground state of the Bohr hydrogen atom (13.6 eV) as follows

$$E_{ex} = \frac{13.6\mu}{\text{BGN.SHNK.EPS}^2} \quad \text{3-55}$$

Default values of these quantities are given in Table 3-5.

Table 3-5 Parameters for the Schenk Band Gap Narrowing Model

Parameter	Default	Units
BGN.SHNK.ME	0.321	
BGN.SHNK.MH	0.346	
BGN.SHNK.EPS	11.7	
BGN.SHNK.GE	12	
BGN.SHNK.GH	4	
μ	0.1665	
α_{ex}	3.719×10^{-7}	cm
E_{ex}	1.655×10^{-2}	eV

The scaled, dimensionless, carrier concentrations used in the expressions for Δ_t^{xc} are

$$n_e = n\alpha_{ex}^3 \quad 3\text{-}56$$

$$n_h = p\alpha_{ex}^3 \quad 3\text{-}57$$

$$n_s = n_e + n_h \quad 3\text{-}58$$

$$n_w = \frac{\mu}{\texttt{BGN.SHNK.ME}} n_e + \frac{\mu}{\texttt{BGN.SHNK.MH}} n_h \quad 3\text{-}59$$

where n is the free electron concentration and p is the free hole concentration. Similarly, the dimensionless thermal energy parameter is

$$\Lambda = \frac{KT}{E_{ex}} \quad 3\text{-}60$$

Then, the carrier concentration and temperature dependent energy band shifts are given by

$$\Delta_t^{xc} = -\frac{(4\pi)^3 n_s^2 \left[\left(\frac{48 n_t}{\pi g_t}\right)^{1/3} + c_t \log(1 + d_t n_w^{p_t})\right] + \left(\frac{8\pi\mu}{g_t m_t}\right) n_t \Lambda^2 + \sqrt{8\pi n_s}\Lambda^{5/2}}{(4\pi)^3 n_s^2 + \Lambda^3 + b_t \sqrt{n_s}\Lambda^2 + 40 n_s^{3/2}\Lambda} \quad 3\text{-}61$$

where t = [e,h] and g_e= BGN.SHNK.GE, g_h = BGN.SHNK.GH, m_e =BGN.SHNK.ME, m_h= BGN.SHNK.MH. The overall bandgap narrowing due to the electron-hole plasma is obtained from multiplying the sum $\Delta_e^{xc} + \Delta_h^{xc}$ by the energy scale E_{ex}. In order to obtain an analytical expression which is accurate over a large range of dopant and carrier densities, Padé approximation theory was used by Schenk. This results in the parameters b_t, c_t, d_t, and p_t. Because of the complexity of the fitting process, these are set to the values given in [279] and you cannot modify them. Their values are shown in Table 3-6.

The other component of the bandgap narrowing is the ionic contribution, which we denote as Δ_t^i. This depends on the local dopant level, and the carrier densities which would occur under charge neutrality conditions. Following the simplifications in [279], the following parameters are defined

$$n_d = N_d\alpha_{ex}^3 \quad 3\text{-}62$$

$$n_a = N_a\alpha_{ex}^3 \quad 3\text{-}63$$

$$n_s = n_d + n_a \quad 3\text{-}64$$

$$n_w = \frac{\mu}{\texttt{BGN.SHNK.ME}} n_d + \frac{\mu}{\texttt{BGN.SHNK.ME}} n_a \quad 3\text{-}65$$

where N_d is the donor density and N_a is the acceptor density. The ionic contributions are

$$\Delta_t^i = \frac{-(n_d + n_a)[1 + \Omega]}{\sqrt{\Lambda n_s/2\pi}[1 + h_t \log(1 + \sqrt{n_s}/\Lambda)] + j_t \Omega n_w^{3/4}(1 + k_t n_w^{q_t})} \quad 3\text{-}66$$

where

$$\Omega = \frac{n_s^2}{\Lambda^3} \quad \text{3-67}$$

The Pade′ fitting parameters h_t, j_t, k_t, and q_t have the same values used in [279] and you cannot change them. Their values are given in Table 3-6.

The overall bandgap narrowing in electron Volts is obtained by multiplying the sum of $\Delta_e^i + \Delta_h^i$ by the energy scaling factor E_{ex} and adding the result to the plasma contributions to the bandgap narrowing. A fraction of this bandgap narrowing is also applied to the electron affinity. The default value of this fraction is 0.5. This can be changed by using the ASYMMETRY parameter on the **MATERIAL** statement.

The model is enabled by specifying the parameter BGN.SCHENK on the **MODELS** statement.

The carrier density dependence of the bandgap and electron affinity calls for a more complicated numerical implementation. It was found that convergence on the left hand side norm could in some cases be achieved when the right hand side convergence had not fully occurred. It is therefore recommended that you reduce the the left hand side convergence criterion by using the CX.TOL parameter on the **METHOD** statement when using this model. Alternatively, you can require right hand side convergence by specifying RHSNORM on the **METHOD** statement. In this latter case, it may be necessary to adjust the CR.TOL parameter on the **METHOD** statement to achieve convergence.

Table 3-6 Fitting Parameters

	Value		Value		Value		Value		Value		Value		Value		Value
b_e	8	c_e	1.3346	d_e	0.893	p_e	$0.233\dot{3}$	h_e	3.91	j_e	2.8585	k_e	0.012	q_e	0.75
b_h	1	c_h	1.2365	d_h	1.153	p_h	$0.233\dot{3}$	h_h	4.20	j_h	2.9307	k_h	0.19	q_h	0.25

3.2.13 Lindefelt Bandgap Narrowing Model

A model for the doping induced bandgap narrowing in 3C-SiC, 4H-SiC, and 6H-SiC due to Lindefelt [185] is available. You implement the model by specifying BGN.LIND on the **MODELS** statement.

According to this model, the downward shift of the conduction band in n-doped material is given by

$$\Delta E_c = \text{BGN.LIND.ANC}\left(\frac{N_d^+}{\text{BGN.LIND.ND}}\right)^{1/3} + \text{BGN.LIND.BNC}\left(\frac{N_d^+}{\text{BGN.LIND.ND}}\right)^{1/2} \quad \text{3-68}$$

and the upward shift of the valence band is given by

$$\Delta E_v = \text{BGN.LIND.ANV}\left(\frac{N_d^+}{\text{BGN.LIND.ND}}\right)^{1/4} + \text{BGN.LIND.BNV}\left(\frac{N_d^+}{\text{BGN.LIND.ND}}\right)^{1/2} \quad \text{3-69}$$

where N_d^+ is the ionized donor concentration. The corresponding quantities in p-doped material are

$$\Delta E_c = \texttt{BGN.LIND.APC}\left(\frac{N_a^-}{\texttt{BGN.LIND.NA}}\right)^{1/4} + \texttt{BGN.LIND.BPC}\left(\frac{N_a^-}{\texttt{BGN.LIND.NA}}\right)^{1/2} \quad \text{3-70}$$

and

$$\Delta E_v = \texttt{BGN.LIND.APV}\left(\frac{N_a^-}{\texttt{BGN.LIND.NA}}\right)^{1/3} + \texttt{BGN.LIND.BPV}\left(\frac{N_a^-}{\texttt{BGN.LIND.NA}}\right)^{1/2} \quad \text{3-71}$$

where N_a^- is the ionized acceptor concentration. The bandgap narrowing is obtained from the sum of ΔE_c and ΔE_v. The default values for these parameters are given in Table 3-7 for four different materials.

Because the bandgap narrowing depends on the ionized impurity densities rather than the physical density, it is recommended to take incomplete ionisation into account by specifying INCOMPLETE on the **MODELS** statement when using this model. In SiC, polytypes dopant ionization energies are relatively large and incomplete ionization can be significant [170].

Table 3-7 Default values of Parameters for Lindefelt Bandgap Narrowing Model.

Parameter	Type	Default (Si)	Default (3C-SiC)	Default (4H-SiC)	Default (6H-SiC)	Units
BGN.LIND.ANC	Real	9.74×10^{-3}	1.48×10^{-2}	1.5×10^{-2}	1.12×10^{-2}	eV
BGN.LIND.BNC	Real	1.39×10^{-3}	3.06×10^{-3}	2.93×10^{-3}	1.01×10^{-3}	eV
BGN.LIND.ANV	Real	1.27×10^{-2}	1.75×10^{-2}	1.9×10^{-2}	2.11×10^{-2}	eV
BGN.LIND.BNV	Real	1.4×10^{-3}	6.85×10^{-3}	8.74×10^{-3}	1.73×10^{-3}	eV
BGN.LIND.APC	Real	1.14×10^{-2}	1.5×10^{-2}	1.57×10^{-2}	1.74×10^{-2}	eV
BGN.LIND.BPC	Real	2.05×10^{-3}	6.41×10^{-4}	3.87×10^{-4}	6.64×10^{-4}	eV
BGN.LIND.APV	Real	1.11×10^{-2}	1.3×10^{-2}	1.3×10^{-2}	1.3×10^{-2}	eV
BGN.LIND.BPV	Real	2.06×10^{-3}	1.43×10^{-3}	1.15×10^{-3}	1.14×10^{-3}	eV

Although the model was developed for SiC polytypes, it has also been applied to Silicon. It is therefore active in regions where the material is either Silicon, 3C-SiC, 4H-SiC, or 6H-SiC when BGN.LIND is enabled on the **MODELS** statement. Other bandgap narrowing models can be used in other regions so long as the REGION parameter is used on the **MODELS** statement to localize each model.

This model has also been fitted to the Slotboom expression, Equation 3-46, for 4H-SiC and 6H-SiC, by Lades [170]. The coefficients are different according to whether the material is n-type or p-type. Thus, BGN.N is replaced by BGN.N.DON for n-type material and BGN.N.ACC for p-type material. BGN.E is replaced by BGN.E.DON and BGN.E.ACC. BGN.C is fixed at 0.5 in all cases. If the BGN or BGN2 flags are specified on the **MODELS** statement, then the default parameters for the Slotboom expression in 4H-SiC and 6H-SiC are as listed in Table 3-8.

Table 3-8 Default values of parameters for Slotboom Bandgap narrowing model for 4H-SiC and 6H-SiC.

Parameter	Type	Default (4H-SiC)	Default (6H-SiC)	Units
BGN.N.DON	Real	1×10^{17}	1×10^{17}	cm^{-3}
BGN.N.ACC	Real	1×10^{17}	1×10^{17}	cm^{-3}
BGN.E.DON	Real	2×10^{-2}	9×10^{-3}	eV
BGN.E.ACC	Real	9×10^{-3}	9×10^{-3}	eV

3.3 Space Charge from Incomplete Ionization, Traps, and Defects

Poisson's Equation (Equation 3-1) including the carrier concentrations, the ionized donor and acceptor impurity concentrations N_D^+ and N_A^-, charge due to traps and defects, Q_T, has the form:

$$div(\varepsilon\nabla\psi) = q(n - p - N_D^+ + N_A^-) - Q_T \quad 3\text{-}72$$

In Atlas the default is to assume full impurity ionization (i.e., $N_D^+ = N_{D,total}$ and $N_A^- = N_{A,total}$), and to set Q_T equal to zero.

Atlas also provides the options of accounting for incomplete ionization of impurities and accounting for additional charge associated with traps and defects.

3.3.1 Incomplete Ionization of Impurities

Atlas can account for impurity freeze-out [144] with appropriate degeneracy factors GCB and GVB for conduction and valence bands. The ionized donor and acceptor impurity concentrations are then given by:

$$N_D^+ = \frac{N_D}{1 + \mathrm{GCB}\,exp\left(\dfrac{\varepsilon_{F_n} - (E_C - \mathrm{EDB})}{kT_L}\right)} \quad 3\text{-}73$$

$$N_A^- = \frac{N_A}{1 + \mathrm{GVB}\,exp\left(\dfrac{E_V + \mathrm{EAB} - \varepsilon_{F_p}}{kT_L}\right)} \quad 3\text{-}74$$

where EDB and EAB are the dopant activation energies and N_D and N_A are net compensated n-type and p-type doping, respectively. Net compensated doping is defined as follows:

If

$$N_{total} \equiv (N_{D,total} - N_{A,total}) > 0 \quad 3\text{-}75$$

then

$$N_D = |\, N_{total} \,| \text{ and } N_A = 0 \quad 3\text{-}76$$

Otherwise

$$N_D = 0 \text{ and } N_A = |\, Ntotal \,|\,) \quad 3\text{-}77$$

The INCOMPLETE parameter of the **MODELS** statement is used to select incomplete ionization and the parameters.

Table 3-9 User-Specifiable Parameters for Equations 3-73 and 3-74		
Statement	**Parameter**	**Units**
MATERIAL	GCB	
MATERIAL	EDB	eV
MATERIAL	GVB	
MATERIAL	EAB	eV

To properly handle incomplete ionization in silicon for high doping levels, a new incomplete ionization model has been added.

The models that form incomplete ionization of impurities given by Equations 3-73 and 3-74 give good physical results for low to moderately doped semiconductors. For heavily (greater than $3\times10^{18}/cm^3$) doped semiconductors, these models fail to predict experimental results of complete ionization. For silicon, an optional model has been introduced that better matches experimental results. This model is set by the IONIZ parameter of the **MODELS** statement.

In this model, the activation energies of the dopants in Equations 3-73 and 3-74 have been modified for doping dependence as given in the following equations:

$$EDB(eV) = A.EDB - B.EDB N_D^{1/3} \quad \text{3-78}$$

$$EAB(eV) = A.EAB - B.EAB N_A^{1/3} \quad \text{3-79}$$

At doping concentrations above 3×10^{18} cm^{-3}, the model predicts complete ionization. At doping concentrations between 10^{18} cm^{-3} and 3×10^{18} cm^{-3}, the model predicts a linearly interpolated value between the above expressions and complete ionization.

The default value for the parameters of Equations 3-78 and 3-79 were chosen to represent reasonable values for silicon and are given in Table 3-10.

Table 3-10 User-Specifiable Parameters for Equations 3-78 and 3-79		
Statement	**Parameter**	**Units**
MATERIAL	A.EDB	0.044
MATERIAL	B.EDB	3.6e-8
MATERIAL	A.EAB	0.0438
MATERIAL	B.EAB	3.037e-5

For general semiconductors, you can use Equations 3-78 and 3-79 without the transitions to complete ionization. To enable this model, specify INC.ION in the **MODELS** statement.

3.3.2 Low Temperature Simulations

In conjunction with Fermi-Dirac statistics and impurity freeze-out, Atlas simulates device behavior under low operating temperatures. In general, simulations can be performed at temperatures as low as 50K without loss of quadratic convergence. Below this temperature, carrier and ionization statistics develop sharp transitions which cause slower convergence. Since many more iterations will probably be required if temperatures below 50K are specified, the ITLIMIT parameter of the **METHOD** statement should be increased.

Due to the limited exponent range on some machines, Atlas can have trouble calculating the quasi-Fermi level of minority carriers. As the temperature decreases, the intrinsic carrier concentration also decreases. In quasi-neutral regions, the minority carrier concentration can easily underflow. Such situations were handled in the past by setting these concentrations to zero. This method does not allow an accurate subsequent calculation of minority carrier quasi-Fermi levels. In order to accurately calculate quasi-Fermi levels, majority carrier concentration and the relation, $np=n_{ie}^2$ is used to obtain minority carrier concentrations in case of an underflow. Despite these efforts, spurious glitches are occasionally observed at low temperatures in the minority quasi-Fermi levels.

3.3.3 Traps and Defects

Semiconductor materials exhibit crystal flaws, which can be caused by dangling bonds at interfaces or by the presence of impurities in the substrate. The presence of these defect centers, or traps, in semiconductor substrates may significantly influence the electrical characteristics of the device. Trap centers, whose associated energy lies in a forbidden gap, exchange charge with the conduction and valence bands through the emission and capture of electrons. The trap centers influence the density of space charge in semiconductor bulk and the recombination statistics.

Device physics has established the existence of three different mechanisms, which add to the space charge term in Poissons's equation in addition to the ionized donor and acceptor impurities. These are interface fixed charge, interface trap states and bulk trap states. Interface fixed charge is modeled as a sheet of charge at the interface and therefore is controlled by the interface boundary condition. Interface traps and bulk traps will add space charge directly into the right hand side of Poisson's equation. This section describes the definition of bulk trap states and the implementation of these bulk trap states into Atlas for both steady state and transient conditions.

A donor-type trap can be either positive or neutral like a donor dopant. An acceptor-type trap can be either negative or neutral like an acceptor dopant. A donor-like trap is positively charged (ionized) when empty and neutral when filled (with an electron). An empty donor-type trap, which is positive, can capture an electron or emit a hole. A filled donor-type trap, which is neutral, can emit an electron or capture a hole. An acceptor-like trap is neutral when empty and negatively charged (ionized) when filled (with an electron). A filled acceptor-like trap can emit an electron or capture a hole. An empty acceptor-like trap can capture an electron or emit a hole. Unlike donors, donor-like traps usually lie near the valence band. Likewise, acceptor-like traps usually lie near the conduction band.

Figure 3-1 shows the terminology used within Atlas to define the type of trap. The position of the trap is defined relative to the conduction or valence bands using E.LEVEL so for instance, an acceptor trap at 0.4eV would be 0.4eV below the conduction band.

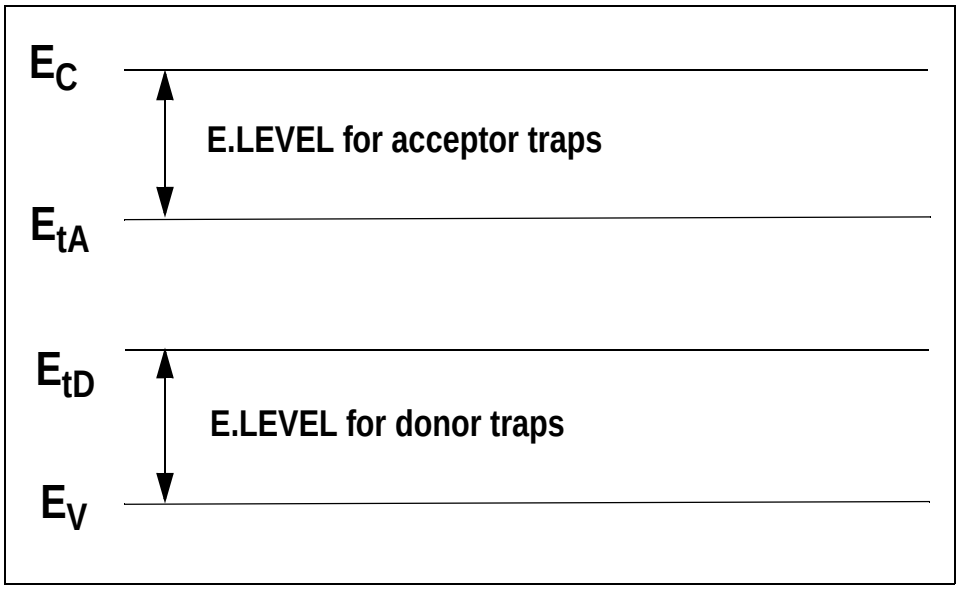


Figure 3-1: Definition of the trap energy level for acceptor and donor traps in reference to the conduction and valence band edges.

Calculation of Trapped Charge in Poisson's Equation

The total charge caused by the presence of traps is subtracted from the right hand side of Poisson's equation. The total charge value is defined by:

$$Q_T = q(N_{tD}^{+} - N_{tA}^{-}) \quad \text{3-80}$$

where N_{tD}^{+} and N_{tA}^{-} are the densities of ionized donor-like and acceptor-like traps respectively. The ionized density depends upon the trap density, DENSITY, and its probability of ionization, F_{tA} and F_{tD}. For donor-like and acceptor-like traps respectively, the ionized densities are calculated by the equations:

$$N_{tD}^{+} = \texttt{DENSITY} \times F_{tD} \quad \text{3-81}$$

$$N_{tA}^{-} = \texttt{DENSITY} \times F_{tA} \quad \text{3-82}$$

In the case where multiple traps at multiple trap energy levels are defined the total charge becomes:

$$N_{tD}^{+} = \sum_{\alpha=1}^{k} N_{tD\alpha}^{+} \quad , N_{tA}^{-} = \sum_{\beta=1}^{m} N_{tA\beta}^{-} \quad \text{3-83}$$

where k is the number of donor-like traps and m is the number of acceptor-like traps.

The probability of ionization assumes that the capture cross sections are constant for all energies in a given band and follows the analysis developed by Simmons and Taylor [295]. The probability of ionization is given by the following equations for acceptor and donor-like traps.

$$F_{tA} = \frac{v_n \texttt{SIGN}\, n + e_{pA}}{v_n \texttt{SIGN}\, n + v_p \texttt{SIGP}\, p + e_{nA} + e_{pA}} \quad \text{3-84}$$

$$F_{tD} = \frac{v_p \texttt{SIGP}\, p + e_{nD}}{v_n \texttt{SIGN}\, n + v_p \texttt{SIGP}\, p + e_{nD} + e_{pD}} \quad \text{3-85}$$

where SIGN and SIGP are the carrier capture cross sections for electrons and holes respectively, v_n and v_p are the thermal velocities for electrons and holes. For donor like traps, the electron and hole emission rates, e_{nD} and e_{pD}, are defined by:

$$e_{nD} = \frac{1}{\texttt{DEGEN.FAC}}\, v_n\, \texttt{SIGN}\, n_i exp\frac{E_t - E_i}{kT_L} \quad \text{3-86}$$

$$e_{pD} = \texttt{DEGEN.FAC}\, v_p\, \texttt{SIGP}\, n_i exp\frac{E_i - E_t}{kT_L} \quad \text{3-87}$$

where E_i is the intrinsic Fermi level position, E_t is the trap energy level as defined by E.LEVEL and DEGEN.FAC is the degeneracy factor of the trap center. The latter term takes into account that spin degeneracy will exist, that is the “empty” and “filled” conditions of a defect will normally have different spin and orbital degeneracy choices. For acceptor like traps, the electron and hole emission rates, e_{nA} and e_{pA}, are defined by:

$$e_{nA} = \texttt{DEGEN.FAC}\, v_n\, \texttt{SIGN}\, n_i \exp\frac{E_t - E_i}{kT_L} \quad \text{3-88}$$

$$e_{pA} = \frac{1}{\texttt{DEGEN.FAC}}\, v_p\, \texttt{SIGP}\, n_i exp\frac{E_i - E_t}{kT_L} \quad \text{3-89}$$

Table 3-11 User-Specifiable Parameters for Equations 3-81 to 3-87

Statement	Parameter	Units
TRAP	E.LEVEL	eV
TRAP	DENSITY	cm^{-3}
TRAP	DEGEN.FAC	
TRAP	SIGN	cm^2
TRAP	SIGP	cm^2

Trap Implementation into Recombination Models

To maintain self-consistency, you need to take into account that electrons are being emitted or captured by the donor and acceptor-like traps. Therefore, the concentration of carriers will be affected. This is accounted for by modifying the recombination rate in the carrier continuity equations.

The standard SRH recombination term (see "Shockley-Read-Hall (SRH) Recombination" on page 215) is modified as follows:

$$R = \sum_{\alpha = 1}^{l} R_{D_\alpha} + \sum_{\beta = 1}^{m} R_{A_\beta} \quad \text{3-90}$$

where l is the number of donor-like traps, m is the number of acceptor-like traps. For donor-like traps, the function R is:

$$R_{D\alpha} = \frac{pn - n_{ie}^2}{\mathrm{TAUN}\left[p + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \mathrm{TAUP}\left[n + \frac{1}{\mathrm{DEGEN.FAC}}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-91}$$

For acceptor like traps the function R is:

$$R_{A\beta} = \frac{pn - n_{ie}^2}{\mathrm{TAUN}\left[p + \frac{1}{\mathrm{DEGEN.FAC}} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \mathrm{TAUP}\left[n + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-92}$$

The electron and hole lifetimes TAUN and TAUP are related to the carrier capture cross sections SIGN and SIGP through the equations:

$$\mathrm{TAUN} = \frac{1}{\mathrm{SIGN}\, v_n \mathrm{DENSITY}} \quad \text{3-93}$$

$$\mathrm{TAUP} = \frac{1}{\mathrm{SIGP}\, v_p\, \mathrm{DENSITY}} \quad \text{3-94}$$

The thermal velocities v_n and v_p are calculated from the following electron and hole effective masses.

$$v_n = \left(\frac{3kT}{\mathrm{M.VTHN}\, m_0}\right)^{1/2} \quad \text{3-95}$$

$$v_p = \left(\frac{3kT}{\mathrm{M.VTHP}\, m_0}\right)^{1/2} \quad \text{3-96}$$

To specify the effective masses directly, use the M.VTHN and M.VTHP parameters from the **MATERIAL** statement. If M.VTHN or M.VTHP or both are not specified, the density of states effective mass is extracted from the density of states (N_c or N_v) using Equations 3-31 and 3-32. In the case of silicon if M.VTHN or M.VTHP are not specified, the effective masses are calculated from:

$$\texttt{M.VTHN} = 1.045 + 4.5 \times 10^{-4} T \quad \text{3-97}$$

$$\texttt{M.VTHP} = 0.523 + 1.4 \times 10^{-3} T - 1.48 \times 10^{-6} T^2 \quad \text{3-98}$$

where T is the lattice temperature.

Table 3-12 User-Specifiable Parameters for Equations 3-91-3-94

Statement	Parameter	Units
TRAP	TAUN	s
TRAP	TAUP	s

The **TRAP** statement activates the model and is used to:

- Specify the trap type DONOR or ACCEPTOR
- Specify the energy level E.LEVEL parameter
- Specify the density of the trap centers DENSITY
- Specify the degeneracy factor DEGEN.FAC
- Specify either the cross sections, SIGN and SIGP, or the electron and hole lifetimes TAUN and TAUP.

Trap-Assisted Tunneling

Trap-Assisted Tunneling models the trap-to-band phonon-assisted tunneling effects for Dirac wells. At high electric fields, tunneling of electrons from the valence band to the conduction band through trap or defect states can have an important effect on the current.

Trap-assisted tunneling is modeled by including appropriate enhancement factors [133] (Γ_n^{DIRAC} and Γ_p^{DIRAC}) in the trap lifetimes in Equation 3-91. These enhancement factors modify the lifetimes so that they include the effects of phonon-assisted tunneling on the emission of electrons and holes from a trap. This model is enabled by specifying TRAP.TUNNEL in the **MODELS** statement.

For donor like traps, the recombination term for traps becomes:

$$R_D = \frac{pn - n_{ie}^2}{\frac{TAUN}{1+\Gamma_n^{DIRAC}}\left[p + DEGEN.FAC\, n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{TAUP}{1+\Gamma_p^{DIRAC}}\left[n + \frac{1}{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}99$$

For acceptor like traps, the recombination term becomes:

$$R_A = \frac{pn - n_{ie}^2}{\frac{TAUN}{1+\Gamma_n^{DIRAC}}\left[p + \frac{1}{DEGEN.FAC} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{TAUP}{1+\Gamma_p^{DIRAC}}\left[n + DEGEN.FAC\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}100$$

The field-effect enhancement term for electrons is given by:

$$\Gamma_n^{DIRAC} = \frac{\Delta En}{kT_L}\int_0^1 \exp\left(\frac{\Delta En}{kT_L}u - K_n u^{3/2}\right)du \quad 3\text{-}101$$

while the field-effect enhancement term for hole is:

$$\Gamma_p^{DIRAC} = \frac{\Delta Ep}{kT_L}\int_0^1 \exp\left(\frac{\Delta En}{kT_L}u - K_p u^{3/2}\right)du \quad 3\text{-}102$$

where u is the integration variable, ΔE_n is the energy range where tunneling can occur for electrons, ΔE_p is the tunneling energy range for holes, and K_n and K_p are defined as:

$$K_n = \frac{4}{3}\frac{\sqrt{2m_0 MASS.TUNNEL \Delta E_n^3}}{qh|E|} \quad 3\text{-}103$$

$$K_p = \frac{4}{3}\frac{\sqrt{2m_0 MASS.TUNNEL \Delta Ep^3}}{qh|E|} \quad 3\text{-}104$$

h is the reduced Planck's constant, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. You can specify MASS.TUNNEL by setting the MASS.TUNNEL parameter in the **MODELS** statement.

Table 3-13 shows the user-specifiable parameter for Equations 3-103 and 3-104.

Table 3-13 User-Specifiable Parameters for Equations 3-103 and 3-104

Statement	Parameter	Default	Units
MODELS	MASS.TUNNEL	0.25	

Non-local Trap Assisted Tunneling

The Trap-assisted-tunneling model in the previous section uses values of the field effect enhancement factors I_n^{DIRAC} and I_p^{DIRAC} which are approximate. They are evaluated using an approximation for the tunneling probability. It is assumed that the value at each node point depends only on the local field strength there, and that the field is assumed to be constant over the range of the tunneling. This gives Airy function solutions for the tunneling probabilities. A further approximation is also made by replacing the Airy function with the leading term of its high field asymptotic behavior. The integration over allowed tunneling energies is also evaluated approximately. A more accurate expression for I_n^{DIRAC} is

$$\Gamma_n^{DIRAC} = \frac{1}{KT}\int_0^{\Delta E_n} e^{E/kT} T(E)dE \qquad 3\text{-}105$$

where *T(E)* is the probability that an electron with energy *E* tunnels elastically to the conduction band from the trap position.

The exponential term gives the probability that the electron will gain energy ($\Delta E_n - E$) from phonon interactions before tunneling. In Equation 3-105, the reference of energy is the conduction band edge and ΔE_n is the maximum energy below this for which the tunneling is possible, or the trap depth if less than this maximum (see Figure 3-2). There is a similar expression for holes. The tunneling probability is evaluated using a Wentzel-Kramers-Brillouin (WKB) method. The integral in Equation 3-105 is evaluated numerically in order to give the Γ_n^{DIRAC} and Γ_p^{DIRAC} terms.

The trap energy is at the Intrinsic Fermi energy in each material, unless you specify the parameter TAT.NLDEPTH on the **MODELS** statement. TAT.NLDEPTH has units of energy in Electron Volts. If you specify TAT.NLDEPTH and TAT.RELEI on the **MODELS** statement, then the value of TAT.NLDEPTH will be added to the Intrinsic Fermi energy to give the trap energy level. If TAT.NLDEPTH is specified without TAT.RELEI, then the trap energy is calculated by subtracting TAT.NLDEPTH from the conduction band energy as follows:

$$E_T = E_c - \texttt{TAT.NLDEPTH} \qquad 3\text{-}106$$

To model Trap Assisted Tunneling through heterojunctions, you must use the TAT.RELEI parameter if you want to use the TAT.NLDEPTH parameter.

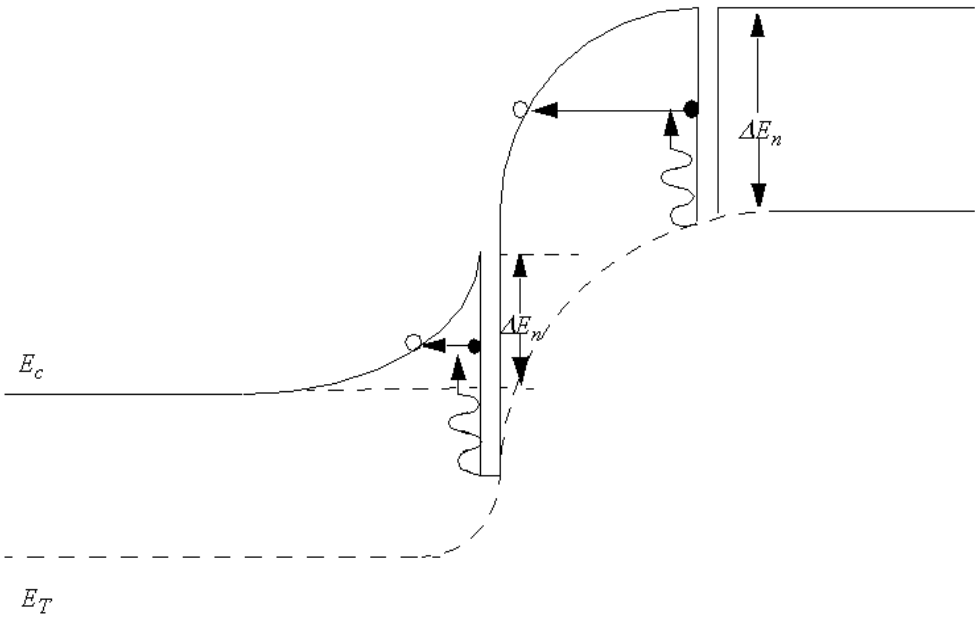


Figure 3-2: Schematic of Phonon assisted electron tunneling from a trap to the conduction band

The Γ_n^{DIRAC} and Γ_p^{DIRAC} terms are then used in Equations 3-99 and 3-100. The values of wavevector used in the tunneling probability calculation are evaluated using the density of states effective mass. To use different masses, then use the ME.TUNNEL and MH.TUNNEL parameters on the **MATERIAL** statement. Using ME and MV on the **MATERIAL** statement will also change the masses used but may affect the behavior of other models. For example, if you specify both ME and ME.TUNNEL, the evaluation of the Γ factor will use the value of ME.TUNNEL.

To use this model, specify an extra rectangular mesh encompassing the region where the Γ terms are to be evaluated. To position the special mesh, use the QTX.MESH and QTY.MESH statement. You must then set the required direction of the quantum tunneling using the QTUNN.DIR parameter on the **MODELS** statement. Outside this mesh, the local model of the previous section will be used to evaluate the Γ terms.

For example

```
qtx.mesh loc=0.0 spac=0.01
qtx.mesh loc=1.0 spac=0.01
qty.mesh loc=1.0 spac=1.0
qty.mesh loc=9.0 spac=1.0
```

will set up a mesh in the rectangle bounded by x=0.0, x=1.0, y=1.0 and y=9.0. The tunneling will be assigned to be in the X direction using the QTUNN.DIR (=1) parameter on the **MODELS** statement. All the required values are interpolated from the values on the Atlas mesh. In this case, they are interpolated onto each slice in the X direction.

You can also use one or more **QTREGION** statements to set up the special meshes required for this model. The syntax is exactly the same as described in the "Non-local Band-to-Band Tunneling" on page 260. This enables you to model non-local Trap-Assisted tunneling for more than one p-n junction and for p-n junctions with complicated shapes.

To enable the model, specify TAT.NONLOCAL on the **MODELS** statement. You also need to specify QTUNN.DIR on the **MODELS** statement to determine the direction of tunneling if you have used QTX.MESH and QTY.MESH to set up the special mesh.

To output the calculated values of Γ_n^{DIRAC} and Γ_p^{DIRAC} to a structure file, specify NLTAT.GAMMA on the **OUTPUT** statement. This allows you to visualize their values when interpolated back onto the device mesh.

Table 3-14 Non-local Trap Assisted Tunneling Parameters for the MODELS Statements

Parameter	Type	Default
TAT.RELEI	Logical	False
TAT.NONLOCAL	Logical	False
TAT.NLDEPTH	Real	0
QTUNN.DIR	Real	0

Table 3-15 Non-local Trap Assisted Tunneling Parameters for the MATERIAL Statement

Parameter	Type	Default
ME.TUNNEL	Real	
MH.TUNNEL	Real	

Poole-Frenkel Barrier Lowering for Coulombic Wells

The Poole-Frenkel barrier lowering effect enhances the emission rate for trap-to-band phonon-assisted tunneling as well as pure thermal emissions at low electric fields. The Poole-Frenkel effect occurs when the Coulombic potential barrier is lowered due to the electric field, and only occurs when there is a Coulomb interaction between the trap and the carrier.

The following interactions are Coulombic.

- Between an empty (positive) donor-type trap and an electron.
- Between a filled (negative) acceptor-type trap and a hole.

The following interactions are short-range (Dirac).

Between a filled (neutral) donor-type trap and a hole.

Between an empty (neutral) acceptor-type trap and an electron.

The Poole-Frenkel effect is modeled by including field-effect enhancement for Coulombic wells (Γ_n^{COUL} and Γ_p^{COUL}) and thermal emission (χ_F) [189] in the trap lifetimes in Equation 3-91. The trap-assisted tunneling effects for Dirac wells remains unmodified. To enable this model, specify TRAP.COULOMBIC in the **MODELS** statement.

The recombination term for traps now becomes:

$$R_A = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{\chi_F + \Gamma_n^{COUL}}\left[p + \frac{1}{\mathrm{DEGEN.FAC}} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{1 + \Gamma_p^{DIRAC}}\left[n + \mathrm{DEGEN.FAC}\ n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}107$$

$$R_D = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{1 + \Gamma_n^{DIRAC}}\left[p + \mathrm{DEGEN.FAC} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{\chi_F + \Gamma_p^{COUL}}\left[n + \frac{1}{\mathrm{DEGEN.FAC}}\ n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}108$$

where the Poole-Frenkel thermal emission enhancement factor, χ_F, is given by:

$$\chi_F = \mathrm{A.TRAPCOULOMBIC} exp\left(\frac{\Delta E_{fp}}{kT_L}\right) \quad 3\text{-}109$$

ΔE_{fp} is the barrier lowering term for a Coulombic well (see Equation 3-110).

$$\Delta E_{fp} = \sqrt{\frac{\mathrm{B.TRAPCOULOMBIC} q^3 |E|}{\pi\varepsilon}} \quad 3\text{-}110$$

The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_p^{COUL}, are defined as:

$$\Gamma_n^{COUL} = \frac{\Delta En}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} \exp\left(\frac{\Delta En}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_n}\right)^{5/3}\right]\right) du \quad 3\text{-}111$$

$$\Gamma_p^{COUL} = \frac{\Delta E_p}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} \exp\left(\frac{\Delta E_p}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_p}\right)^{5/3}\right]\right) du \quad 3\text{-}112$$

Table 3-16 User-Specifiable Parameters for Equations 3-109 and 3-110

Statement	Parameter	Default	Units
MATERIAL	A.TRAPCOULOMBIC	1.0	
MATERIAL	B.TRAPCOULOMBIC	1.0	

Transient Traps

In the time domain the acceptor and donor traps do not reach equilibrium instantaneously but require time for electrons to be emitted or captured. This is taken account of inside Atlas by solving an additional differential rate equation whenever a transient simulation is performed. These rate equations for donor and acceptor traps are given in Equations 3-113 and 3-114 [350]:

$$\frac{dN_{tD}^{+}}{dt} = \mathtt{DENSITY}\left\{\upsilon_p \mathtt{SIGP}\left[p(1-F_{tD}) - F_{tD} n_i \mathtt{DEGEN.FAC}\ exp\frac{E_i - E_t}{kT}\right]\right.$$
$$\left. - \upsilon_n \mathtt{SIGN}\left[nF_{tD} - \frac{(1-F_{tD}) n_i exp}{\mathtt{DEGEN.FAC}}\ \frac{E_t - E_i}{kT}\right]\right\} \quad (3\text{-}113)$$

$$\frac{dN_{tA}^{-}}{dt} = \mathtt{DENSITY}\left\{\upsilon_n \mathtt{SIGN}\left[n(1-F_{tA}) - \mathtt{DEGEN.FAC}\ F_{tA} n_i exp\frac{E_t - E_i}{kT}\right]\right.$$
$$\left. - \upsilon_p \mathtt{SIGP}\left[pF_{tA} - \frac{(1-F_{tA}) n_i}{\mathtt{DEGEN.FAC}}\ exp\frac{E_i - E_t}{kT}\right]\right\} \quad (3\text{-}114)$$

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. If this is the case, a flag (FAST) on the TRAP statement will neglect the trap rate equation from the simulation.

As may be seen from Equations 3-113 and 3-114, each trap species has a net electron capture rate and a net hole capture rate. The former arise from the terms proportional to υ_nSIGN and the latter from terms proportional to υ_pSIGP. These capture rates enter the transient current continuity Equations 3-3 and 3-4 as components of the R_n and R_p terms respectively. In some circumstances, they will be generation rates rather than recombination rates, but by convention they are still referred to as R_n and R_p with negative values. In steady state, the trap occupation is constant and therefore the electron and hole capture rates are equal for each trap species. By substituting Equations 3-84 to 3-89 into the right hand sides of Equations 3-113 and 3-114, you obtain the expressions for steady state recombination, namely Equations 3-91 and 3-92. Under transient simulation conditions R_n and R_p have different values. This difference leads to the change over time of the trap occupation densities.

To view the values of R_n and R_p summed over the trap species present in a transient simulation, Atlas provides two options. To output the rates to a standard structure file for **SAVE** statements, you specify U.TRANTRAP on the **OUTPUT** statement. You specify E.TRANTRAP on the **PROBE** statement to obtain a probe of R_n and H.TRANTRAP on the **PROBE** statement to obtain a probe of R_p.

Table 3-17 Output of Transient Recombination Rates			
Statement	**Parameter**	**Type**	**Default**
OUTPUT	U.TRANTRAP	Logical	False
PROBE	E.TRANTRAP	Logical	False
PROBE	H.TRANTRAP	Logical	False

Hysteresis of Interface Traps

The default implementation of traps at a semiconductor-insulator interface assumes that the traps are located exactly at the interface. Their rate of response to changes in the energy of the Quasi-Fermi level in the semiconductor depends linearly on the values of the carrier capture cross-sections, SIGN and SIGP, as Equations 3-113 and 3-114 show.

In the HEIMAN model, the traps are assumed to have a uniform distribution in depth in the insulator [125]. They are further assumed to have values of SIGN and SIGP that depend on their distance into the insulator, *d*, as

$$\mathrm{SIGN}(d) = \mathrm{SIGN}\ e^{-\kappa_e d} \quad 3\text{-}115$$

$$\mathrm{SIGP}(d) = \mathrm{SIGP}\ e^{-\kappa_h d} \quad 3\text{-}116$$

where

$$\kappa_e^2 = \frac{2m_e^*(E_c - E_{tA})}{\hbar^2} \quad 3\text{-}117$$

and

$$\kappa_h^2 = \frac{2m_h^*(E_{tD} - E_v)}{\hbar^2} \quad 3\text{-}118$$

are the evanescent wavevectors of the semiconductor electron and hole states as they tunnel into the insulator. The capture cross-sections decrease with depth, *d*, in the insulator. Consequently the spatially deeper the trap, the more slowly it will respond to changes of carrier concentration at the semiconductor-insulator interface. For example, if the traps are initially full and the device is biased to empty them, then after a certain time the traps up to a certain depth will be empty and deeper traps will still be full. Heiman demonstrates that the transition distance between the two cases is very narrow with a length scale of approximately $1/(2\kappa_e)$ for electrons and $1/(2\kappa_h)$ for holes [125].

To enable this model you specify HEIMAN on the **INTTRAP** statement. The trap parameters are set up as usual for interface traps with the additional specification of the DEPTH parameter. This determines the extent of the uniform trap density distribution into the insulator and is in units of microns with a default value of 0.005 microns. The number of equally spaced internal points describing the internal trap distribution for each interface node is the HPOINTS parameter. This is optionally specified on the **INTTRAP** statement and has a default value of 10. For each interface node, the trap distribution is calculated at each of the HPOINTS internal points. The average occupancy of the interface charge at each interface point is obtained by taking the arithmetic mean over these internal points.

These averages can be output by using the parameters FTACC.TRAP and FTDON.TRAP on the **PROBE** statement. The **PROBE** must be positioned on the semiconductor-insulator interface where hysteresis traps are present.

The HEIMAN model can be used with either steady-state bias ramps or as a fully transient simulation. To calculate the effect of hysteresis on small-signal admittance values, you must perform a steady-state simulation. To do this, you specify the total simulation timespan for the bias ramp using the TIMESPAN parameter on the **SOLVE** statement. The assumed time to sweep the bias through VSTEP volts is thus VSTEP*TIMESPAN/(VFINAL-V(initial)) seconds. To simulate a hysteresis cycle, you need two consecutive **SOLVE** statements–one to ramp the voltage to the desired final value and the other to return it to the initial value. The value of TIMESPAN can be different for the two **SOLVE** statements. Any steady-state SOLVE statement executed without a TIMESPAN parameter will cause the trap occupation probabilities to be set to their steady-state values consistent with the device bias state.

Note: Changes in the quasi-Fermi levels at the semiconductor interface cause trap occupation changes up to a distance *d* into the insulator. Distance *d* depends on the natural logarithm of the measurement time TIMESPAN divided by the trap capture lifetime. Traps that are situated deep in the insulator therefore require a large value of TIMESPAN. The DEPTH parameter, therefore, should not be set to too high a value.

For a transient simulation of trap hysteresis, you specify HEIMAN on the **INTTRAP** statement and use the transient **SOLVE** statement. You do not specify the TIMESPAN parameter in this case because all necessary information is specified on the transient **SOLVE** statement. You can intermix transient and steady-state hysteresis simulations.

The HEIMAN model can also be used to study Negative Bias Temperature Instability in p-MOSFETS if the SR.HEIMAN option is used. This enhancement is based on the modification of hole trapping and emission rates by Structural Relaxation (SR) of hole traps in SiO_2 [136]. The carrier capture terms in Equations 3-113 and 3-114 (i.e., those that depend on free carrier density) are multiplied by a factor $P_{SR}^{c} = exp\frac{-\mathrm{SR.TRAP}}{KT}$, and the emission terms are multiplied by $P_{SR}^{e} = exp\frac{-\mathrm{SR.EMIT}}{KT}$. The effect of this is to give NBTI degradation a much stronger temperature dependence than is obtained purely from the tunneling factors and introduce a tunable asymmetry between emission and capture processes. You set the values of SR.TRAP and SR.EMIT on the **INTTRAP** statement, and the default values are 0.5 eV [136].

Table 3-18 Parameters for the HEIMAN Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
INTTRAP	HEIMAN	Logical	False	
INTTRAP	DEPTH	Real	5.0e-3	μm
INTTRAP	SR.HEIMAN	Logical	False	
INTTRAP	SR.TRAP	Real	0.5	eV
INTTRAP	SR.EMIT	Real	0.5	eV
INTTRAP	HPOINTS	Real	10	
SOLVE	TIMESPAN	Real		seconds

3.3.4 Multistate Traps

The individual traps considered above can be in one of two internal states, that is they are either occupied by a charge carrier or empty. The occupation probability gives the fraction of traps which are in a given state, and the remaining fraction are in the other state.

It is possible to model traps that have two or more internal states by using the Multistate trap model. In this model, which is currently restricted to interface traps, a trap can have up to 5 internal states.

This allows you to model structural changes of a trap as well as trapping of multiple carrier types and ionic or atomic species. In this model, each state has its own occupation probability with the sum over states of occupation probability being one. The time evolution of each occupation probability depends on the transition rates between the different states. In the multistate trap model, the steady state occupation probability is fixed, and the occupation probabilities are only changed during a transient simulation. The model can be used with electron and hole traps and it can also be coupled with the generic ion transport model (see Section 3.15 "Generic Ion Transport Model") and the hydrogen diffusion model, which forms part of the Reaction-Diffusion degradation model.

For a trap with N-possible internal states, the equation for the time dependence of the occupation probability of state i, f_i, is given by

$$\frac{\partial f_i}{\partial t} = \sum_{j \neq i} (f_j a_{ji} - f_i a_{ij}) \qquad 3\text{-}119$$

where the term $f_j a_{ji}$ represents transfer into state i from state j, and the term $f_i a_{ij}$ represents transfer from state i to state j. If we add together all N of these equations, then the right hand sides cancel so we get

$$\sum_{i=1} \frac{\partial f_i}{\partial t} = 0 \qquad 3\text{-}120$$

so that the sum of occupation probabilities of the state is constrained to be a constant. By convention this constant is 1.0. In general $a_{ij} \neq a_{ji}$. In other words, the forward and reverse transfer rates between two internal states are usually different. The transfer rates typically depend on several simulation variables as well as various external parameters, and are usually time dependent themselves. Atlas solves the set of Equation 3-119 in the TR-BDF2 or BDF1 transient schemes, taking into account the evolution of the interface charge and the recombination/generation processes associated with the change of internal trap state occupations. Any numerical truncation errors causing the $\sum_{i=1}^{N} f_i$ to be different than exactly 1.0 are distributed among the individual probabilities in such a way as to maintain the constraint.

Almost all parameters and flags for this model belong to the **INTTRAP** statement (see Table 3-19). You specify the MSCTRAP flag to enable this model. You must also specify MSC.NSTATES to give the number of internal states that the model has. This can be anywhere between 2 and 5. Details of each internal state can be specified using the MSCx.ELEC, MSCx.HOLE, MSCx.SP1, MSCx.SP2, MSCx.SP3 parameters, where x can be any integer between 1 and 5. The latter three parameters refer to generic ions. To use this functionality, you need to either use the generic ion transport model or specify H.ATOM or H.MOLE or the **MODELS** statement to include atomic and molecular hydrogen respectively. In this case, the parameters MSCx.H (aliased to MSCx.SP1) refer to atomic hydrogen and MSCx.H2 (aliased to MSCx.SP2) to molecular hydrogen. If you use hydrogen transport, the transport parameters and dimerisation rates are specified on the **MATERIAL** statement. If instead you use the generic ion model, the transport parameters are described in Section 3.15 “Generic Ion Transport Model” and reaction parameters are described in Section 3.16 “Generic Ion Reaction Model”.

For example, you would indicate that internal state 3 contained a trapped electron by specifying MSC3.ELEC=1 on the **INTTRAP** statement. If internal state 1 contained 2 ions of ionic species 3, you would specify MSC1.SP3 = 2 on the **INTTRAP** statement. The charge of each internal state must also be specified using the MSCx.CHARGE parameter, and changes in the CHARGE parameter between internal states must be consistent with the changes in individual components.

The CHARGE parameter is used in the contribution of the multistate traps to the interface charge. Atlas will automatically check the CHARGE parameters for consistency.

The initial occupation probabilities of the internal states can be initialized using the MSCx.FT0 parameter. The sum of the MSCx.FT0 parameters must add to 1.0. Atlas will also check this automatically. Each internal state can also be assigned an energy level using the MSCx.ELEVEL parameter. You specify that this energy is relative to the conduction band edge by using the MSCx.ET.CB flag, and relative to the valence band edge using the MSCx.ET.VB flag.

You specify the trap cross-sections, SIGN and SIGP, and the trap density, DENSITY in the usual way. The parameters DONOR, ACCEPTOR, FAST, HEIMAN, E.LEVEL, MIDGAP, and E.ACTIVATION are irrelevant to multistate traps and are ignored. The DEGEN parameter is automatically set to 1 for multistate traps.

You must also specify the transition rates between internal states. They are usually complicated functions of the solution variables and it is impossible to stipulate them directly using parser parameters. Instead of this, you specify a filename for the F.MSCRATES parameter, the file containing a C-Interpreter function. The C-Interpreter function is called msctransitionrates, and the prototype is given in the file template.lib. Among other parameters, it is supplied with two integers, *i* and *j*, which are never simultaneously the same and which both run from 1 up to the number of internal states. This allows a transition rate to be programmed for each combination of the start and ending states. The convention is that *i* is the starting state and *j* the final state. Figure 3-3 shows the direction of the transitions for rates a_{ij} in the case of a 3-state model. The function will give optimum performance if the first derivatives of the rate with respect to the dependent variables are given. The file template.lib gives an example of this for multistate traps with 3 internal states.

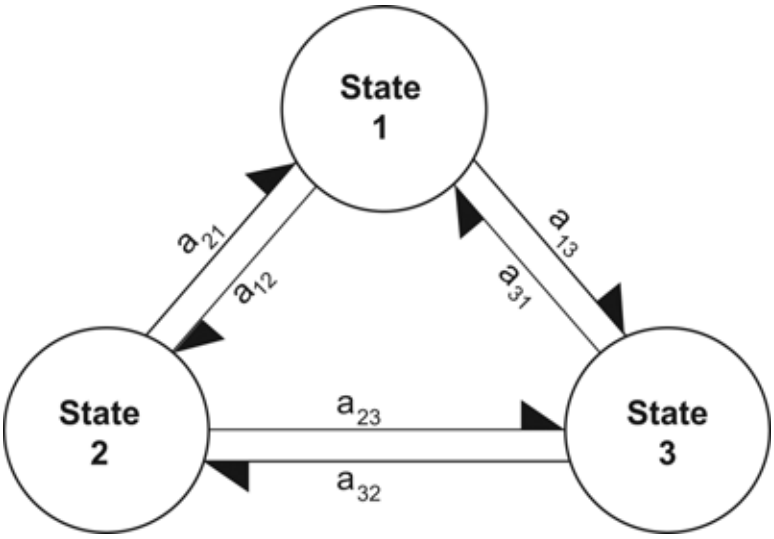


Figure 3-3: Multistate Traps with 3 Internal States

As the trap occupation probabilities evolve over time, they give rise to a recombination rate for the type of carrier they contain. The electron recombination rate, for example, is obtained from the following sum:

$$R_n = \sum_{i=1}^{N} \texttt{MSCi.ELEC} \frac{\partial f_i}{\partial t} \qquad \text{3-121}$$

so that any states containing an electron will contribute. This is then optionally coupled to the electron continuity equation, with analogous results for the other trapped species. In simulations where the coupling to the transport equations can be ignored, you specify ^RG.MSCTRAP on the **METHOD** statement to remove the coupling and thereby improve simulation speed.

Visualization of the trap internal states in a standard log file is obtained using the FT.MSC flag on the **PROBE** statement. This also requires the particular internal state to be specified using the MSC.STATE parameter of the **PROBE** statement. The position parameters of the statement will localize it to the interface with multistate traps. The recombination rates of electrons and holes can be visualized by using the E.TRANTRAP and H.TRANTRAP flags on the **PROBE**

statement respectively. Because the traps are transient, the electron and hole recombination rates will in general differ from each other. For simplicity, any change in a state having a trapped hole is modeled as a hole recombination rate (H.TRANTRAP), and for one containing an electron as an electron recombination rate (E.TRANTAP). If the generic ion transport model is also active, any recombination of ionic species into the multistate traps can also be visualized by using the U.SP1TRAP, U.SP2TRAP, and U.SP3TRAP flags on the **PROBE** statement. If the H.ATOM or H.MOLE flags are enabled, then U.SP1TRAP will output the atomic hydrogen recombination rate, and U.SP2TRAP will output the molecular hydrogen recombination rate. By convention, a recombination rate is given as a positive value and a generation rate is given as a negative value. To visualize the state occupation probabilities in a standard structure file, you specify TRAPS on the **OUTPUT** statement. Five occupation probabilities per multistate configuration trap will then be output. The first MSC.NSTATES values will contain valid data, the rest will be zero.

One possible use of the Multistate trap model is to simulate the degradation behavior of passivated dangling bonds, as an alternative to the built-in DEVDEG.RD, DEVDEG.KN, and DEVDEG.PL models. This is best achieved using the hydrogen diffusion and reaction models enabled with the H.ATOM and H.MOLE models. The most complex of these uses the maximum of 5 internal states.

Table 3-19 Parameters for the Multistate Configuration Trap Model

Statement	Parameter	Type	Default	Units
INTTRAP	F.MSCRATES	Character		
INTTRAP	MSC.NSTATES	Integer	0	
INTTRAP	MSC1.CHARGE	Integer	0	
INTTRAP	MSC1.ELEC	Integer	0	
INTTRAP	MSC1.ELEVEL	Real	0	eV
INTTRAP	MSC1.ET.CB	Logical	False	
INTTRAP	MSC1.ET.VB	Logical	False	
INTTRAP	MSC1.FT0	Real	0	
INTTRAP	MSC1.H1	Integer	0	
INTTRAP	MSC1.H2	Integer	0	
INTTRAP	MSC1.HOLE	Integer	0	
INTTRAP	MSC1.SP1	Integer	0	
INTTRAP	MSC1.SP2	Integer	0	
INTTRAP	MSC1.SP3	Integer	0	
INTTRAP	MSC2.CHARGE	Integer	0	
INTTRAP	MSC2.ELEC	Integer	0	
INTTRAP	MSC2.ELEVEL	Real	0	eV

Table 3-19 Parameters for the Multistate Configuration Trap Model				
INTTRAP	MSC2.ET.CB	Logical	False	
INTTRAP	MSC2.ET.VB	Logical	False	
INTTRAP	MSC2.FT0	Real	0	
INTTRAP	MSC2.HOLE	Integer	0	
INTTRAP	MSC2.H1	Integer	0	
INTTRAP	MSC2.H2	Integer	0	
INTTRAP	MSC2.SP1	Integer	0	
INTTRAP	MSC2.SP2	Integer	0	
INTTRAP	MSC2.SP3	Integer	0	
INTTRAP	MSC3.CHARGE	Integer	0	
INTTRAP	MSC3.ELEC	Integer	0	
INTTRAP	MSC3.ELEVEL	Real	0	eV
INTTRAP	MSC3.ET.CB	Logical	False	
INTTRAP	MSC3.ET.VB	Logical	False	
INTTRAP	MSC3.FT0	Real	0	
INTTRAP	MSC3.H1	Integer	0	
INTTRAP	MSC3.H2	Integer	0	
INTTRAP	MSC3.HOLE	Integer	0	
INTTRAP	MSC3.SP1	Integer	0	
INTTRAP	MSC3.SP2	Integer	0	
INTTRAP	MSC3.SP3	Integer	0	
INTTRAP	MSC4.CHARGE	Integer	0	
INTTRAP	MSC4.ELEC	Integer	0	
INTTRAP	MSC4.FT0	Real	0	
INTTRAP	MSC4.H1	Integer	0	
INTTRAP	MSC4.H2	Integer	0	
INTTRAP	MSC4.HOLE	Integer	0	
INTTRAP	MSC4.SP1	Integer	0	
INTTRAP	MSC4.SP2	Integer	0	

Table 3-19 Parameters for the Multistate Configuration Trap Model				
INTTRAP	MSC4.SP3	Integer	0	
INTTRAP	MSC5.CHARGE	Integer	0	
INTTRAP	MSC5.ELEC	Integer	0	
INTTRAP	MSC4.ELEVEL	Real	0	eV
INTTRAP	MSC4.ET.CB	Logical	False	
INTTRAP	MSC4.ET.VB	Logical	False	
INTTRAP	MSC5.ELEVEL	Real	0	eV
INTTRAP	MSC5.ET.CB	Logical	False	
INTTRAP	MSC5.ET.VB	Logical	False	
INTTRAP	MSC5.FT0	Real	0	
INTTRAP	MSC5.H1	Integer	0	
INTTRAP	MSC5.H2	Integer	0	
INTTRAP	MSC5.HOLE	Integer	0	
INTTRAP	MSC5.SP1	Integer	0	
INTTRAP	MSC5.SP2	Integer	0	
INTTRAP	MSC5.SP3	Integer	0	
INTTRAP	MSCTRAP	Logical	False	
METHOD	RG.MSCTRAP	Logical	True	
MODELS	H.ATOM	Logical	False	
MODELS	H.MOLE	Logical	False	
PROBE	E.TRANTRAP	Logical	False	
PROBE	FT.MSC	Logical	False	
PROBE	H.TRANTRAP	Logical	False	
PROBE	MSC.STATE	Integer	1	
PROBE	U.SP1TRAP	Logical	False	
PROBE	U.SP2TRAP	Logical	False	
PROBE	U.SP3TRAP	Logical	False	

Examples

```
INTTRAP S.I  [Localisation parameters] DENSITY=3.0E12 DEGEN=1 \
SIGN=1.0E-13 SIGP=1.0E-14 MSCTRAP MSC.NSTATES=3 MSC1.FT0=0.5 \
```

```
MSC2.FT0=0.3 MSC3.FT0=0.2 MSC1.CHARGE=1  MSC1.HOLE=1 \
MSC2.CHARGE=-1 MSC2.ELEC=1 MSC3.CHARGE=0 F.MSCRATES="example.lib"
```

This is a 3 state trap State 1 has a trapped hole and a charge of +1 [Q]. State 2 has a trapped electron and a charge of -1 [Q]. State 3 has no trapped carriers and a charge of 0 [Q]. It is initialized with 50% of the traps in state 1, 30% of the traps in state 2 and 20% of the traps in state 3. The transition rate C-Interpreter function is in the file `'example.lib'`.

To obtain the time dependence of the trap state occupancies in a log file, specify

```
PROBE FT.MSC MSC.STATE=1 X=1.5 Y=0
PROBE FT.MSC MSC.STATE=2 X=1.5 Y=0
PROBE FT.MSC MSC.STATE=3 X=1.5 Y=0
INTTRAP [Localisation parameters] DENSITY=1.0E11 SIGN=1.0E-13
SIGP=1.0E-14 MSCTRAP MSC.NSTATES=2 MSC1.FT0=1.0 MSC1.SP1=1
MSC1.CHARGE=1 MSC2.FT0=0.0 MSC2.ELEC=1 MSC2.CHARGE=-1
F.MSCRATES="example2.lib"
MODELS NSPECIES=1 SPECIES1.Z=1
```

The generic ion transport model is enabled on the **MODELS** statement, and there is one ionic species with charge of +1 [Q] Coulombs.

This is a 2-state trap State 1 has a trapped ion of species1, which has a charge of +1 [Q]. State 2 has a trapped electron, which has a charge of -1 [Q]. The initial occupancy condition is that all traps are in state 1. To look at the recombination rates of ionic species 1 and electrons into the trap, specify

```
PROBE E.TRANTRAP [Localisation parameters]
PROBE U.SP1TRAP  [Localisation parameters]
```

In this case, the electron and SP1 recombination rates at any point will be equal and opposite because the rate of change of internal state 1 is equal and opposite to that of internal state 2 (because their sum must be constant).

```
INTTRAP [Localisation parameters] DENSITY=1.0e12 MSCTRAP
MSC.NSTATES=2 MSC1.CHARGE=0 MSC1.H1=1 MSC1.CHARGE=0 MSC2.HOLE=1
MSC2.CHARGE=1 F.MSCRATES="dbond.lib"
MODELS MOS H.ATOM H.MOLE
PROBE U.SP1TRAP [Localisation parameters]
```

In this case, the atomic hydrogen and molecular hydrogen transport and diffusion models are enabled on the **MODELS** statement. State 1 is set to have one hydrogen atom per trap using the `MSC1.H1` parameter (aliased to `MSC1.SP1`). The charge of state 1 is set to 0 [Q], and the charge of state 2 is set to 1 [Q] to be consistent with the fact that state 2 has a trapped hole. The **PROBE** entry will probe the atomic hydrogen concentration in this case. The bulk dimerisation rates can be set on the **MATERIAL** statement to control the creation of molecular hydrogen.

3.4 The Energy Balance Transport Model

The conventional drift-diffusion model of charge transport neglects non-local transport effects such as velocity overshoot, diffusion associated with the carrier temperature and the dependence of impact ionization rates on carrier energy distributions. These phenomena can have a significant effect on the terminal properties of submicron devices. As a result Atlas offers two non-local models of charge transport, the energy balance, and hydrodynamic models.

The Energy Balance Transport Model follows the derivation by Stratton [303,304] which is derived starting from the Boltzmann Transport Equation. By applying certain assumptions, this model decomposes into the hydrodynamic model [209,12,13].

The Energy Balance Transport Model adds continuity equations for the carrier temperatures, and treats mobilities and impact ionization coefficients as functions of the carrier temperatures rather than functions of the local electric field.

3.4.1 The Energy Balance Equations

The Energy Balance Transport Model introduces two new independent variables T_n and T_p, the carrier temperature for electrons and holes. The energy balance equations consist of an energy balance equation with the associated equations for current density and energy flux $S_{n,p}$.

For electrons the Energy Balance Transport Model consists of:

$$div\vec{S}_n = \frac{1}{q}\vec{J}_n \cdot \vec{E} - W_n - \frac{3k}{2}\frac{\partial}{\partial t}\left(\lambda_n^* \, nT_n\right) \quad \text{3-122}$$

$$\vec{J}_n = qD_n\nabla n - q\mu_n n\nabla\psi + qnD_n^T\nabla T_n \quad \text{3-123}$$

$$\vec{S}_n = -K_n\vec{\nabla}T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}_n T_n \quad \text{3-124}$$

and for holes:

$$div\vec{S}_p = \frac{1}{q}\vec{J}_p \cdot \vec{E} - W_p - \frac{3k}{2}\frac{\partial}{\partial t}\left(\lambda_p^* \, pT_p\right) \quad \text{3-125}$$

$$\vec{J}_p = -qD_p\nabla p - q\mu_p p\nabla\psi - qpD_p^T\nabla T_p \quad \text{3-126}$$

$$\vec{S}_p = -K_p\nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}_p T_p \quad \text{3-127}$$

where $\vec{S}_n$ and $\vec{S}_p$ are the energy flux densities associated with electrons and holes, and μ_n and μ_p are the electron and hole mobilities.

The remaining terms, D_n and D_p, are the thermal diffusivities for electrons and holes as defined in Equations 3-128 and 3-135 respectively. W_n and W_p are the energy density loss rates for electrons and holes as defined in Equations 3-151 and 3-152 respectively. K_n and K_p are the thermal conductivities of electrons and holes as defined in Equations 3-132 and 3-139 respectively.

$$D_n = \frac{\mu_n k T_n}{q} \lambda_n^* \quad (3\text{-}128)$$

$$\lambda_n^* = \frac{F(1/2)(\eta_n)}{F{-}1/2(\eta_n)}, \qquad \eta_n = \frac{\varepsilon_{F_n} - \varepsilon c}{kT_n} = F_{1/2}^{-1}\left(\frac{n}{N_c}\right) \quad (3\text{-}129)$$

$$D_n^T = \left(\mu_{2n} - \frac{3}{2}\lambda_n^* \mu_n\right)\frac{k}{q} \quad (3\text{-}130)$$

$$\mu_{2n} = \mu_n\left(\frac{5}{2} + \xi_n\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)} \quad (3\text{-}131)$$

$$K_n = q\ n\ \mu_n\left(\frac{k}{q}\right)^2 \Delta_n T_n \quad (3\text{-}132)$$

$$\Delta_n = \delta_n\left[\left(\xi_n + \frac{7}{2}\right)\frac{F_{\xi_n} + 5/2(\eta_n)}{F_{\xi_n} + 3/2(\eta_n)} - \left(\xi_n + \frac{5}{2}\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)}\right] \quad (3\text{-}133)$$

$$\delta_n = \frac{\mu 2n}{\mu_n} \quad (3\text{-}134)$$

Similar expressions for holes are as follows:

$$D_p = \frac{\mu_p k T_p}{q} \lambda_p^* \quad (3\text{-}135)$$

$$\lambda_p^* = \frac{F(1/2)(\eta_p)}{F{-}1/2(\eta_p)}, \qquad \eta_p = \frac{\varepsilon_v - \varepsilon_{f_p}}{kT_p} = F_{1/2}^{-1}\left(\frac{p}{N_V}\right) \quad (3\text{-}136)$$

$$D_p^T = \left(\mu_{2p} - \frac{3}{2}\lambda_p^* \mu_p\right)\frac{k}{q} \quad (3\text{-}137)$$

$$\mu_{2p} = \mu_p\left(\frac{5}{2} + \xi_p\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)} \quad (3\text{-}138)$$

$$K_p = q\ p\ \mu_p\left(\frac{k}{q}\right)^2 \Delta_p T_p \quad (3\text{-}139)$$

$$\Delta_p = \delta_p\left[\left(\xi_p + \frac{7}{2}\right)\frac{F_{\xi_p} + 5/2(\eta_p)}{F_{\xi_p} + 3/2(\eta_p)} - \left(\xi_p + \frac{5}{2}\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)}\right] \quad 3\text{-}140$$

$$\delta_p = \frac{\mu 2p}{\mu_p} \quad 3\text{-}141$$

If Boltzmann statistics are used in preference to Fermi statistics, the above equations simplify to:

$$\lambda_n^* = \lambda_p^* = 1 \quad 3\text{-}142$$

$$\Delta_n = \delta_n = \left(\frac{5}{2} + \xi_n\right) \quad 3\text{-}143$$

$$\Delta_p = \delta_p = \left(\frac{5}{2} + \xi_p\right) \quad 3\text{-}144$$

$$\xi_n = \frac{d(\ln\mu_n)}{d(\ln T_n)} = \frac{T_n}{\mu_n}\frac{\partial\mu_n}{\partial T_n} \quad 3\text{-}145$$

$$\xi_p = \frac{d(\ln\mu_p)}{d(\ln T_p)} = \frac{T_p}{\mu_p}\frac{\partial\mu_p}{\partial T_p} \quad 3\text{-}146$$

The parameters ξ_n and ξ_p are dependent on the carrier temperatures. Different assumptions concerning ξ_n and ξ_p correspond to different non-local models. In the high-field saturated-velocity limit, that corresponds to velocity saturation, the carrier mobilities are inversely proportional to the carrier temperatures.

$$\xi_n = \xi_p = -1 \quad 3\text{-}147$$

and this corresponds to the Energy Balance Transport Model.

If instead the choice ξ_n=ξ_p=0 was chosen this would correspond to the simplified Hydrodynamic Model. The parameters, ξ_n and ξ_p, can be specified using the KSN and KSP parameters on the **MODELS** statement.

Boundary conditions for n, p, and ψ are the same as for the drift diffusion model. Energy balance equations are solved only in the semiconductor region. Electron and hole temperatures are set equal to the lattice temperature on the contacts. On the other part of the boundary, the normal components of the energy fluxes vanish.

Hot carrier transport equations are activated by the **MODELS** statement parameters: HCTE.EL (electron temperature), HCTE.HO (hole temperature), and HCTE (both carrier temperatures).

3.4.2 Density of States

The calculation of the effective density of states is modified for the Energy Balance Transport Model. The electron and hole temperatures replace the lattice temperature in Equations 3-31 and 3-32. For example:

$$N_c = \left(\frac{2\pi m_e^* kT_n}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_n}{300}\right)^{\frac{3}{2}} NC(300) \quad 3\text{-}148$$

$$N_v = \left(\frac{2\pi m_n^* kT_p}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_p}{300}\right)^{\frac{3}{2}} NV(300) \quad 3\text{-}149$$

3.4.3 Energy Density Loss Rates

The energy density loss rates define physical mechanisms by which carriers exchange energy with the surrounding lattice environment. These mechanisms include carrier heating with increasing lattice temperature as well as energy exchange through recombination processes (SRH and Auger) and generation processes (impact ionization). If the net generation-recombination rate is written in the form:

$$U = R_{srh} + R_n^A + R_p^A - G_n - G_p \quad 3\text{-}150$$

where R_{srh} is the SRH recombination rate, R_n^A are R_p^A Auger recombination rates related to electrons and holes, G_n and G_p are impact ionization rates, then the energy density loss rates in Equations 3-122 and 3-125 can be written in the following form:

$$W_n = \frac{3}{2} n \frac{k(T_n - T_L)}{\texttt{TAUREL.EL}} \lambda_n + \frac{3}{2} kT_n \lambda_n R_{SRH} + E_g(G_n - R_n^A) \quad 3\text{-}151$$

$$W_p = \frac{3}{2} p \frac{k(T_p - T_L)}{\texttt{TAUREL.HO}} \lambda_p + \frac{3}{2} k\ T_p\ \lambda_p R_{SRH} + E_g(G_p - R_p^A) \quad 3\text{-}152$$

where

$$\lambda_n = F_{\frac{3}{2}}(h_n)/F_{\frac{1}{2}}(h_n) \quad 3\text{-}153$$

$$\lambda_p = F_{\frac{3}{2}}(h_p)/F_{\frac{1}{2}}(h_p) \quad 3\text{-}154$$

(and are equal to 1 for Boltzmann statistics), TAUREL.EL and TAUREL.HO are the electron and hole energy relaxation times, E_g is the bandgap energy of the semiconductor. The relaxation parameters are user-definable on the **MATERIAL** statement, which have their defaults shown in Table 3-20.

The relaxation times are extremely important as they determine the time constant for the rate of energy exchange and therefore precise values are required if the model is to be accurate.

But, this is an unmeasurable parameter and Monte Carlo analysis is the favored method through which values can be extracted for the relaxation time.

It's also important to take into consideration that different materials will have different values for the energy relaxation time but within Atlas, the relaxation time will always default to the value for silicon.

Table 3-20 User-Specifiable Parameters for Equations 3-151 and 3-152

Statement	Parameter	Default	Units
MATERIAL	TAUREL.EL	4×10^{-13}	s
MATERIAL	TAUREL.HO	4×10^{-13}	s

3.4.4 Temperature Dependence of Relaxation Times

To obtain more accurate energy balance simulations, the energy relaxation time can be modeled as a variable quantity. Atlas provides three ways of doing this. The first is a built-in model in which the energy relaxation time is a function of carrier temperature only. The second is a more advanced built-in model in which the energy relaxation time is a function of both carrier and lattice temperatures. The final way is a C-Interpreter function for your specific carrier and lattice temperature dependent model. In all cases, the actual variables used are the carrier and lattice energies, which are obtained by multiplying the temperatures by $\frac{3}{2}K$, where K is the Boltzmann constant. They are described below.

Carrier Temperature Dependent Model

For the first method, use a built-in model by specifying the TRE.T1, TRE.T2, TRE.T3, TRE.W1, TRE.W2, and TRE.W3 parameters in the **MATERIAL** statement. Then, activate the electron temperature (electron energy) dependent energy relaxation time by using E.TAUR.VAR in the **MODELS** statement. Electron energy relaxation time will then be:

$$\tau_e = \begin{cases} \texttt{TRE.T1}, & W < \texttt{TRE.W1} \\ \texttt{TRE.T2}, & W = \texttt{TRE.W2} \\ \texttt{TRE.T3}, & W > \texttt{TRE.W3} \end{cases} \quad \text{3-155}$$

where:

$$W = \frac{3}{2}kT_n \quad \text{3-156}$$

For TRE.W1<W<TRE.W2 the energy relaxation time varies quadratically between TRE.T1 and TRE.T2. For TRE.W2<W<TRE.W3 energy relaxation time varies quadratically between TRE.T2 and TRE.T3. The corresponding parameter for hole energy relaxation time in the **MODELS** statement is H.TAUR.VAR. Other parameters are listed in Table 3-21.

Table 3-21 User- Specifiable Parameters for Variable Energy Relaxation Time.			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	TRE.T1	8×10^{-13}	s
MATERIAL	TRE.T2	1.92×10^{-12}	s
MATERIAL	TRE.T3	1×10^{-12}	s
MATERIAL	TRE.W1	0.06	eV
MATERIAL	TRE.W2	0.3	eV
MATERIAL	TRE.W3	0.45	eV
MATERIAL	TRH.T1	1×10^{-12}	s
MATERIAL	TRH.T2	1×10^{-12}	s
MATERIAL	TRH.T3	1×10^{-12}	s
MATERIAL	TRH.W1	1×10^{10}	eV
MATERIAL	TRH.W2	1×10^{10}	eV
MATERIAL	TRH.W3	1×10^{10}	eV

Carrier and Lattice Temperature Dependent Model

This model is based on an analytical approximation to energy relaxation times obtained from Monte-Carlo simulations, and is due to Gonzalez et al [103, 235]. It assumes that the energy relaxation time for electrons is given by

$$\tau_{wn} = \texttt{GONZALEZ.TAUN0} + \texttt{GONZALEZ.TAUN1}\exp(F(T_n, T_L)) \quad \text{3-157}$$

where

$$F(T_n, T_L) = \texttt{GONZALEZ.C1}(T_n/300 + \texttt{GONZALEZ.C0})^2 + \texttt{GONZALEZ.C2}$$
$$(T_n/300 + \texttt{GONZALEZ.C0}) + \texttt{GONZALEZ.C3}T_L/300 \quad \text{3-158}$$

where T_n is the electron temperature and T_L is the lattice temperature. You set the flag E.TAUR.GONZALEZ on the **MODELS** statement to enable this model for electrons. The energy relaxation time for holes is assumed to be constant

$$\tau_{wp} = \texttt{GONZALEZ.TAUP0} \quad \text{3-159}$$

You set the flag H.TAUR.GONZALEZ on the **MODELS** statement to enable this model for holes. The parameter values are provided for a range of materials. It is possible to specify your own values of parameters on the **MATERIAL** statement. If a material does not have its own specific values, and there are no user-supplied values, then the values for Silicon will be used as the defaults. These are shown in Table 3-22.

Table 3-22 Generic Default Parameters for the Gonzalez Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
MATERIAL	GONZALEZ.TAUN0	Real	1.0×10^{-12}	s
MATERIAL	GONZALEZ.TAUN1	Real	-0.538×10^{-12}	s
MATERIAL	GONZALEZ.C0	Real	0.0	
MATERIAL	GONZALEZ.C1	Real	0.0015	
MATERIAL	GONZALEZ.C2	Real	-0.09	
MATERIAL	GONZALEZ.C3	Real	0.17	
MATERIAL	GONZALEZ.TAUP0	Real	0.4×10^{-12}	s

Specific defaults are available for a range of elemental and binary semiconductors and will be used for those materials, unless you specify a value on the MATERIAL statement instead. The material specific values are shown in Table 3-23.

Table 3-23 Material Specific Parameters for the Gonzalez Model							
Parameter	**Si**	**Ge**	**GaAs**	**AlAs**	**InAs**	**InP**	**GaP**
GONZALEZ.TAUN0 ($\times10^{-12}$s)	1.0	0.26	0.48	0.17	0.08	0.5	0.04
GONZALEZ.TAUN1 ($\times10^{-12}$s)	-0.538	1.49	0.025	0.025	0.025	0.21	0.21
GONZALEZ.C0	0	0	0	61	3	-18	10
GONZALEZ.C1	0.0015	-0.434	-0.053	-0.053	-0.053	-0.04	-0.04
GONZALEZ.C2	-0.09	1.322	0.853	0.853	0.853	0.0	0.0
GONZALEZ.C3	0.17	0.0	0.5	0.5	0.5	0.0	0.0
GONZALEZ.TAUP0 ($\times10^{-12}$s)	0.4	0.4	1.0	1.0	1.0	1.0	1.0

Specific default parameters are also available for some ternary alloys. In this case, the quantities GONZALEZ.TAUN0 and GONZALEZ.C0 are interpolated between the constituent binary values.

For the III-V ternaries of the form $A_xB_{1-x}C$, where A and B are group III elements, we have

$$\text{GONZALEZ.TAUN0}(x) = x\,\text{GONZALEZ.TAUN0}(A) + (1-x)\,\text{GONZALEZ.TAUN0}(B) + x(1-x)\,\text{GONZALEZ.TAUBOW} \quad (3\text{-}160)$$

$$\text{GONZALEZ.C0}(x) = x\,\text{GONZALEZ.C0}(A) + (1-x)\,\text{GONZALEZ.C0}(B) + x(1-x)\,\text{GONZALEZ.CEEBOW} \quad (3\text{-}161)$$

where the bowing parameters, GONZALEZ.TAUBOW and GONZALEZ.CEEBOW are available for some materials, and are shown in Table 3-24. Apart from these materials, the default values of GONZALEZ.TAUBOW and GONZALEZ.CEEBOW are zero.

Table 3-24 Ternary Bowing Parameters for the Gonzalez model

Parameter	AlGaAs	InGaAs	InGaP
GONZALEZ.TAUBOW($\times 10^{-12}$s)	-0.35	1.8	-0.4
GONZALEZ.CEEBOW	-61.0	-34.0	-5.2

C-interpreter Model

For the third method, use a C-Interpreter function to specify a user-defined function of carrier energy and lattice energy for the energy relaxation time. For electrons, you specify E.TAUR.VAR on the **MODELS** statement and F.TAURN on the **MATERIAL** statement. F.TAURN should be the name of the external file containing the C-function of the user-defined implementation. For holes, you specify H.TAUR.VAR on the **MODELS** statement and F.TAURP on the **MATERIAL** statement. For more details of C-Interpreter functions see Appendix A "C-Interpreter Functions".

3.4.5 Energy Dependent Mobilities

The Energy Balance Transport Model requires the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated, which causes the carriers in a homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields, $E_{eff,n}$ and $E_{eff,p}$, are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\texttt{TAUMOB.EL}} \quad \text{3-162}$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\texttt{TAUMOB.HO}} \quad \text{3-163}$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived from the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introduced into the relevant field dependent mobility model. A full description of the available models is given in Section 3.6.1 “Mobility Modeling”.

Atlas2d provides a general C-interpreter function allowing you to specify carrier mobility as a function of Perpendicular field, carrier temperature, Lattice temperature, carrier concentration and donor and acceptor concentrations.

For electron mobility, the parameter is `F.ENMUN`, which can be set in either the **`MATERIAL`** or **`MOBILITY`** statement. The corresponding parameter for hole mobility is `F.ENMUP`, which can be set in either the **`MATERIAL`** or **`MOBILITY`** statement. The respective C-functions are `endepmun()` and `endepmup()`. The examples of these functions are provided in Atlas.

3.5 Boundary Physics

Atlas supports several boundary conditions: Ohmic contacts, Schottky contacts, insulated contacts, and Neumann (reflective) boundaries. Voltage boundary conditions are normally specified at contacts. Current boundary conditions can also be specified. Additional boundary conditions have been implemented to address the needs of specific applications. You can connect lumped elements between applied biases and semiconductor device contacts. A true distributed contact resistance is included to account for the finite resistivity of semiconductor contacts.

3.5.1 Ohmic Contacts

Ohmic contacts are implemented as simple Dirichlet boundary conditions, where surface potential, electron concentration, and hole concentrations (ψ_s, n_s, p_s) are fixed. Minority and majority carrier quasi-Fermi potentials are equal to the applied bias of the electrode (i.e., $\phi_n=\phi_p=V_{applied}$). The potential ψ_s is fixed at a value that is consistent with space charge neutrality. For example:

$$n_s + N_A^- = p_s + N_D^+ \quad \text{3-164}$$

Equation 3-164 can be solved for ψ_s, n_s, and p_s, since ϕ_n and ϕ_p are known. If Boltzmann statistics are used, the substitution of Equations 3-36 and 3-37 into Equation 3-164 will yield:

$$n_s = \frac{1}{2}\left[(N_D^+ - N_A^-) + \sqrt{\left(N_D^+ - N_A^-\right)^2 + 4n_{ie}^2}\right] \quad \text{3-165}$$

$$p_s = \frac{n_{ie}^2}{n_s} \quad \text{3-166}$$

$$\psi_s = \phi_n + \frac{kT_L}{q}\ln\frac{n_s}{n_{ie}} = \phi_p - \frac{kT_L}{q}\ln\frac{p_s}{n_{ie}} \quad \text{3-167}$$

Note: If you don't specify a work function, the contacts will be Ohmic regardless of its material.

3.5.2 Schottky Contacts

To specify a Schottky contact [263], specify a workfunction using the WORKFUN parameter of the CONTACT statement. The surface potential of the Schottky contact is given by:

$$\psi_s = \text{AFFINITY} + \frac{E_g}{2q} + \frac{kT_L}{2q}\ln\frac{N_C}{N_V} - \text{WORKFUN} + V_{applied} \quad \text{3-168}$$

where AFFINITY is the electron affinity of the semiconductor material, E_g is the bandgap, N_C is the conduction band density of states, N_V is the valence band density of states, and T_L is the ambient temperature. In practice, the workfunction is defined as:

$$\text{WORKFUN} = \text{AFFINITY} + \phi_B \quad \text{3-169}$$

where ϕ_B is the barrier height at the metal-semiconductor interface in eV.

Table 3-25 User-Specifiable Parameters for Equation 3-169

Statement	Parameter	Units
CONTACT	WORKFUN	eV
MATERIAL	AFFINITY	eV

For example, if the Schottky contact were aluminum with a workfunction difference to the silicon of 4.2 eV and a barrier height of 0.7eV, then you would define the Schottky contact with the statement:

```
CONTACT NAME=GATE WORKFUN=4.9
```

Note: You can also define the workfunction using a C-Interpreter function. This function is referenced by the F.WORKF parameter of the **CONTACT** statement. The function defines workfunction as a function of electric field.

You can enable the thermionic emission model by specifying any of the following parameters of the **CONTACT** statement: SURF.REC, E.TUNNEL, VSURFN, VSURFP, or BARRIERL. In this case, the quasi-Fermi levels, ϕ_n and ϕ_p, are no longer equal to $V_{applied}$. Instead, these parameters are defined by a current boundary conditions at the surface [68]:

$$J_{sn} = q\text{VSURFN}(n_s - n_{eq})exp\left(\frac{\Delta\phi_b}{kT}\right) \quad \text{3-170}$$

$$J_{sp} = q\text{VSURFP}(p_s - p_{eq})exp\left(\frac{\Delta\phi_b}{kT}\right) \quad \text{3-171}$$

Table 3-26 User-Specifiable Parameters for Equations 3-170 to 3-171		
Statement	**Parameter**	**Units**
CONTACT	VSURFN	cm/s
CONTACT	VSURFP	cm/s

where J_{sn} and J_{sp} are the electron and hole currents at the contact, n_s, is the surface electron concentration and p_s is the surface hole concentrations. The terms, n_{eq} and p_{eq}, are the equilibrium electron and hole concentrations assuming infinite surface recombination velocity ($\phi_n = \phi_p = V_{applied}$). If VSURFN and VSURFP are not specified on the **CONTACT** statement, their values will be calculated using:

$$\text{VSURFN} = \frac{\text{ARICHN}\ T_L^{\,2}}{q\ N_C} \qquad 3\text{-}172$$

$$\text{VSURNP} = \frac{\text{ARICHP}\ T_L^{\,2}}{q\ N_V} \qquad 3\text{-}173$$

Table 3-27 User-Specifiable Parameters for Equations 3-172 to 3-173			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	ARICHN	110	$A/cm^2/K^2$
MATERIAL	ARICHP	30	$A/cm^2/K^2$

Here, ARICHN and ARICHP are the effective Richardson constants for electrons and holes, taking account of quantum mechanical reflections and tunneling, N_C and N_V are the conduction and valence band density of states. The ARICHN and ARICHP parameters are user-definable as shown in Table 3-27 and N_C and N_V are functions of the lattice temperature, T_L, according to Equations 3-31 and 3-32.

The schottky thermionic emission model also accounts for field-dependent barrier-lowering mechanisms. These mechanisms are caused by image forces and possible static dipole layers at the metal-semiconductor interface [307]. If the barrier height is defined as:

$$\phi_{bn} = \text{WORKFUN} - \text{AFFINITY} \qquad 3\text{-}174$$

$$\phi_{bp} = \text{AFFINITY} + \frac{E_g}{q} - \text{WORKFUN} \qquad 3\text{-}175$$

With barrier lowering, the amount of energy by which these barrier heights are lowered is defined by:

$$\Delta\phi_b = \mathrm{BETA}\left[\frac{q}{4\pi\varepsilon_s}\right]^{\frac{1}{2}} E^{1/2} + \mathrm{ALPHA} \times E^{\mathrm{GAMMA}} \quad \text{3-176}$$

Table 3-28 User-Specifiable Parameters for Equation 3-176

Statement	Parameter	Default	Units
CONTACT	ALPHA	0.0	cm
CONTACT	BETA	1.0	
CONTACT	GAMMA	1.0	

Here, E is the magnitude of the electric field at the interface and ALPHA is the linear, dipole barrier lowering coefficient. The Barrier Lowering Model can be turned on with the BARRIER parameter in the **CONTACT** statement. Typical values of ALPHA may be found in [10]. Note that the term with the square root dependence on electric field corresponds to the image force, while the linear term corresponds to the Dipole Effect [307].

Thermionic emission is implemented on a triangle-by-triangle basis, which means using the surface recombination velocity and geometrical data. A thermionic emission component is then calculated for each triangle so that an element of interest is connected. Using the electric field for each triangle, an adjusted recombination thermionic emission term can be computed if barrier lowering is incorporated. This is in contrast to where a single field value for the electrode node is used to compute total thermionic emission value [275].

You can take electron or hole tunneling through the barrier into account by specifying the E.TUNNEL or H.TUNNEL parameter in the CONTACT statement. The electron tunneling current (J_{tn}) is given by the Tsu-Esaki Model (Equation 3-177) and is applied to the contact as a current boundary condition.

$$J_{tn} = -\frac{4\,\pi\,q\,\mathrm{ME.TUNNEL}\,m_0\,k\,T_L}{h^3}\int_0^{\phi_{bn}} P(E_Z)N(E_Z)dEz \quad \text{3-177}$$

Here, ϕ_{bn} is the barrier height, ME.TUNNEL is the relative effective mass for electrons, m_0 is the electron rest mass, h is Planck's constant, k is Boltzmann's constant, q is the electron charge, and $N(E_Z)$ is given by:

$$N(E_Z) = \ln\left(\frac{1+e^{\left(\frac{E_F - E_Z}{kT_L}\right)}}{1+e^{\left(\frac{E_F - E_Z - qV}{kT_L}\right)}}\right) \quad \text{3-178}$$

The transmission probability $P(E_Z)$ is given by the *WKB* [307] approximation for a triangular barrier and is expressed as:

$$P(E_Z) = exp\left(\frac{-8\pi(2\text{ME.TUNNEL } m^*)^{\frac{1}{2}}(\phi_{bn}-E_Z)^{\frac{3}{2}}}{3qhE}\right) \quad 3\text{-}179$$

The ME.TUNNEL parameter is user-definable in the CONTACT statement.

Similar expressions for Equations 3-177 through 3-179 exist for hole tunneling. You can specify the MH.TUNNEL parameter on the **CONTACT** statement for the hole relative mass for tunneling.

For example, to specify a thermoinic emission schottky contact with tunneling and barrier lowering, the command would be:

```
CONTACT NAME=GATE SURF.REC E.TUNNEL BARRIER
```

Parabolic Field Emission Model

For wide bandgap materials such as SiC the parabolic field emission model is suggested [67,85]. Equation 3-180 describes the current as a function of applied bias voltage *V*. Equation 3-181 describes the tunneling probability *T(E)*.

$$J = \frac{A^*T}{k_B}\int_0^{\infty} T(E)\ln\left[\frac{1+\exp\left(\frac{-E-E_n}{k_BT}\right)}{1+\exp\left(\frac{-E-qV-E_n}{k_BT}\right)}\right] \quad 3\text{-}180$$

$$T(E) = \begin{cases} \exp\left\{\frac{2}{hq}\sqrt{\frac{m\varepsilon}{N}}\left[E\ \ln\left(\frac{\sqrt{E_b^*}+\sqrt{E_b^*-E}}{\sqrt{E}}\right)-\sqrt{E_b^*}\sqrt{E_b^*-E}\right]\right\} & for\ (E<E_b^*) \\ 1 & for\ (E>E_b^*) \end{cases} \quad 3\text{-}181$$

Here A^* is Richardson's constant, E_b is the barrier height on the metalside, N is the semiconductor doping concentration, $En=Ec-Ef=kT^*ln(Nc/N)$ is the Fermi level energy $Vbi=(Eb-En)/q$ is the built in voltage, and $Eb^* = q(Vbi-V) = Eb-En-qV$ is the barrier height on the semiconductor side.

Richardson's constant A^* is given by $\frac{4\pi m q k^2}{h^3}$ where m is the effective mass.

The effective mass is generally taken from the density of states. You can, however, specify a mass for this tunneling calculation only by the ME.TUNNEL and MH.TUNNEL parameters of the **CONTACT** statement. To enable the model, use the PARABOLIC parameter of the **CONTACT** statement. To turn on the tunneling components independently for electrons and holes, specify the E.TUNNEL and H.TUNNEL parameters of the **CONTACT** statement. To enable the model, specify the NSURF.REC or PSURF.REC parameters of the **CONTACT** statement.

The energy step size used in numeric integration of Equation 3-180 is controlled by the DE.TUNNEL parameter of the **CONTACT** statement. This parameter secifies proportion of the barrier height used in each step. In other words, the energy step is obtained by multiplying DE.TUNNEL by the barrier height. The default value for DE.TUNNEL is 10.

In evaluation of the parabolic model Equation 3-180 is only used to evaluate the tunneling part of the current. For pure thermionic emission, the expression in Equation 3-182 may be used:

$$J = qv_s N_c \exp(-\phi b / kT)(V / KT + 1) \quad \text{3-182}$$

The parameter THERM of the **CONTACT** statement controls whether the thermionic emission component of Equation 3-182 is included. By default, the value of the THERM parameter is TRUE.

You should notice that the parabolic Schottky model is essentially a voltage controlled current source where the voltage applied to the contact, *V*, appears in the various expressions without regard to the potentials applied anywhere else in the device. To promote flexibility, we have added the ability to specify a reference electrode either by name using the REF.ELEC parameter or by index using the REF.ENUM parameter of the **CONTACT** statement. When either of these parameters are assigned, the parameter, *V*, in the above expressions can be interpreted as the voltage of the contact minus the voltage at the reference electrode.

Universal Schottky Tunneling (UST) Model

In the Universal Schottky Tunneling (UST) Model [139, 202], the tunneling current is represented by localized tunneling rates at grid locations near the Schottky contact. For electrons, this is illustrated in Figure 3-4. The tunneling component of current can be described by Equation 3-183.

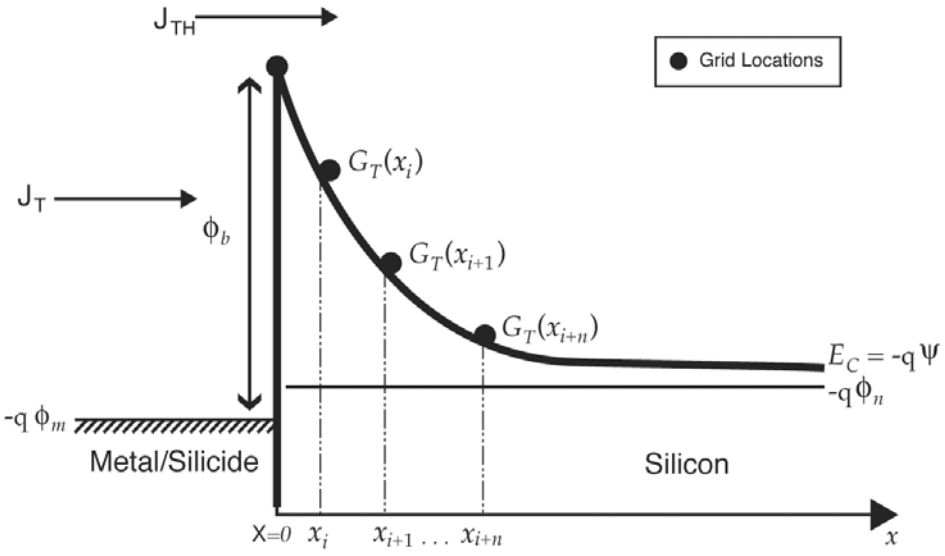


Figure 3-4: Local tunneling generation rate representation of the universal Schottky tunneling model

$$J_T = \frac{A^* T_L}{k} \int_E^{\infty} I(E') \ln\left[\frac{1 + fs(E')}{1 + fm(E')}\right] dE' \quad \text{3-183}$$

Here, J_T is the tunneling current density, A^* is the effective Richardson's coefficient, T_L is the lattice temperature, $\Gamma(E)$ is the tunneling probability, $fs(E)$ and $fm(E)$ are the Maxwell-Boltzmann distribution functions in the semiconductor and metal and E is the carrier energy.

You can then apply the transformation [139] given in Equation 3-184 to obtain the localized tunneling rates, G_T.

$$G_T = \frac{1}{q} \nabla J_T \quad \text{3-184}$$

Applying the transformation given in Equations 3-183 and 3-184, you can obtain the expression given in Equation 3-185.

$$G_T = \frac{A^* T_L \vec{E}}{k} \Gamma(x) ln\left[\frac{1 + n/\gamma_n N_c}{1 + exp[-(E_c - E_{FM})/kT_L]}\right] \quad \text{3-185}$$

Here, $\vec{E}$ is the local electric field, n is the local electron concentration, N_c is the local conduction band density of states, γ_n is the local Fermi-Dirac factor, E_c is the local conduction band edge energy and E_{FM} is the Fermi level in the contact.

The tunneling probability $\Gamma(x)$ can be described by Equation 3-186.

$$\Gamma(x) = exp\left[\frac{-2\sqrt{2m}}{\hbar}\int_0^x (E_c(x') - E_c(x))dx'\right] \quad \text{3-186}$$

In Equation 3-186, m is the electron effective mass for tunneling, and $E_c(x)$ is the conduction band edge energy as a function of position.

Assuming linear variation of E_c around a grid location, Equation 3-186 can be reduced to Equation 3-187.

$$\Gamma(x) = exp\left[\frac{-4\sqrt{2m}x}{3\hbar}(E_{FM} + q\phi_b - E_c(x))^{1/2}\right] \quad \text{3-187}$$

In Equation 3-187, ϕ_b is the barrier height. Similar expressions to Equations 3-183 through 3-187 exist for holes.

To enable the universal Schottky tunneling model you should specify `UST` on the **`MODELS`** statement. You are also required to specify `SURF.REC` on the **`CONTACT`** statement for each contact that you wish the model to apply. Once enabled, this model automatically applies to both electrons and holes. For a given grid point, however, the model will only apply to one carrier according to the relative direction of the local electric field.

Once you enable the model, for each electrode with thermionic emission, the model will be applied to all grid points within a specified distance of the electrode, To specify this distance, use `D.TUNNEL` parameter of the **`MATERIAL`** statement. The default value of `D.TUNNEL` is 10^{-6} in units of cm.

Phonon-assisted Tunneling Model for GaN Schottky Diodes

It has been found that the reverse I-V characteristics of some Gallium Nitride (GaN) diodes can best be explained by using a phonon-assisted electron tunneling model [245]. The electrons are assumed to be emitted from local levels in the metal-semiconductor interface and give a contribution to the current of

$$J = Q\ \mathtt{PIP.NT}\ W \quad (3\text{-}188)$$

where PIP.NT gives the occupied state density near the interface in units of cm^{-2}. The rate of phonon-assisted tunneling, *W*, is given by

$$W = \frac{qF}{(8m_o\ \mathtt{ME.TUNNELPIP.ET})^{1/2}}[(1+\gamma^2)^{1/2}-\gamma]^{1/2}[1+\gamma^2]^{-1/4} \times \exp\left\{\frac{-4}{3}\frac{(2m_o\ \mathtt{ME.TUNNEL})^{1/2}}{qFh}PIP.ET^{3/2}[(1+\gamma^2)^{1/2}-\gamma]^2 \times \left[(1+\gamma^2)^{1/2}+\frac{\gamma}{2}\right]\right\} \quad (3\text{-}189)$$

where

$$\gamma = \frac{(2m_o\ \mathtt{ME.TUNNEL})^{1/2}\Gamma^2}{8q\hbar F(\mathtt{PIP.ET})^{1/2}} \quad (3\text{-}190)$$

F is the field strength at the interface and Γ is the energy width of the absorption band given by

$$\Gamma^2 = 8\mathtt{PIP.ACC}(\mathtt{PIP.OMEGA})^2(2n+1) \quad (3\text{-}191)$$

and *n* is the phonon density.

To enable the model, use the keyword PIPINYS on the **CONTACT** statement. The default value of PIP.NT is 0 and so you must set a finite value for this quantity in order for the model to have any effect. The value of ME.TUNNEL must be set on the **CONTACT** statement. Any value of ME.TUNNEL specified on the **MATERIAL** statement is not used in this model. The other parameters may be specified on the **MATERIAL** statement. Although the model was developed specifically for reverse bias current in GaN, Atlas does not restrict use of the model to that material. PIP.OMEGA is the phonon energy, PIP.ACC is the electron-phonon interaction constant, and PIP.ET is the trap depth.

Table 3-29 User-Specifiable Parameters for the Pipinys model.

Statement	Parameter	Type	Default	Units
MATERIAL	ME.TUNNEL	Real	1.0	
MATERIAL	PIP.OMEGA	Real	0.07	eV
MATERIAL	PIP.ACC	Real	2.0	
MATERIAL	PIP.ET	Real	1.0	eV
MATERIAL	PIP.NT	Real	0.0	cm^{-2}

Scott-Malliaras Boundary Conditions

The Scott-Malliaras boundary conditions [283] define a reverse bias contact charge density of

$$n = 4\Psi^2 N_C \exp\left(\frac{-\phi_b}{kT}\right) \exp f^{1/2} \quad \text{3-192}$$

where the normalized electric field is

$$f = \frac{q^3 E}{4\pi\varepsilon\varepsilon_0 k^2 T^2} \quad \text{3-193}$$

and the factor Ψ is

$$\Psi = \frac{1}{f} + \frac{1}{f^{1/2}} - \frac{1}{f}(1 + 2f^{1/2})^{1/2} \quad \text{3-194}$$

In forward bias, the contact charge density remains at the zero bias value, $N_C \exp(-\phi_b / kT)$.

You can activate Scott-Malliaras boundary conditions by specifying SCOTT.MALLIARAS, in addition to the zero bias work-function, on a **CONTACT** statement. For example

```
CONTACT NAME=ANODE WORKFUNCTION=4.47 SCOTT.MALLIARAS
```

A "quality" parameter SCOM.QUALITY, on the **CONTACT** command, can be used to tune the Scott-Malliaras boundary conditions. A SCOM.QUALITY=1 (the default) gives the equations from [283]. A SCOM.QUALITY=0 gives a Schottky contact (where the contact charge density is given by $N_C \exp(-\Phi_b / kT)$.

A control parameter SCOM.RANGE, on the **CONTACT** command, can be used to limit the carrier density at the contact. The electron density, for example, is $n = N_C \exp(-\Phi_{b,\,eff} / kT)$. The SCOM.RANGE gives the range that $\Phi_{b,\,eff}$ can take. If SCOM.RANGE=1 (the default), then $\Phi_{b,\,eff}$ can lie anywhere in the bandgap. If SCOM.RANGE=0.8, then $\Phi_{b,\,eff}$ can only lie in 80% of the bandgap (centered on the middle of the bandgap). This can be used to limit the contact carrier density at high electric fields.

3.5.3 Floating Contacts

A contact that isn't connected to a current or voltage source and is totally insulated by dielectric is called a floating contact. Atlas accounts for floating contacts such as floating gates in EPROM devices by using a distributed charge boundary condition, which is applied to all nodes of the floating electrode (see Equation 3-195).

$$\int_s D \, dS = Q_{FG} \quad \text{3-195}$$

where D is the electric displacement vector, s represents the external surface of the floating gate, and Q_{FG} is the injected charge.

Atlas performs an integration over the entire surface of the electrode and forces the potential on the floating nodes to produce the correct total charge on the electrode. The total charge when performing a simulation is by default zero but can be defined using the **SOLVE** statement. The total charge may change if you activate a Charge Injection Model. For more information about this model, see Section 3.6.7 "Gate Current Models". To define a contact as a floating contact, use:

```
CONTACT NAME=fgate FLOATING
```

Note: When specifying a floating contact, use the Newton scheme as the numerical technique. See Section 21.5.1 "Newton Iteration".

3.5.4 Current Boundary Conditions

In some devices, the terminal current is a multi-valued function of the applied voltage. This means that for some voltage boundary conditions, the solution that is obtained depends on the initial guess. An example of this is the CMOS latch-up trigger point. At the trigger point the I-V curve changes from being flat to vertical and may exhibit a negative slope. The solution will then have three different solutions of current for one applied bias. The particular solution which the model finishes in will depend upon the initial conditions.

This trigger point is difficult to determine using a simple voltage boundary condition. In addition, it is almost impossible to compute any solutions in the negative resistance regime when using voltage boundary conditions. Some of these problems can be overcome using current boundary conditions.

Calculation of current boundary conditions is activated by the CURRENT parameter in the **CONTACT** statement.

The current boundary conditions are described by the Kirkhoff relation:

$$J_s - \Sigma(J_n + J_p) = 0 \qquad \text{3-196}$$

where J_s is the applied current, J_n and J_p are given by Equations 3-11 and 3-12, and the summation is taken over all grid points incident on the boundary. This relation is solved for the boundary potential. The ohmic conditions in Equations 3-165 and 3-166 apply for electron and hole concentrations.

The voltage boundary condition should be used in regions where dI/dV is small. The current boundary condition may be preferable for operating regimes where dI/dV is large. It is common for the negative resistance regime of a device to have a slope dI/dV very close to 0. Such behavior should be considered when using a current source to trace out an entire I-V curve. In these cases, use the CURVETRACE option in Atlas.

Note: When a current boundary condition has been specified, choose the Newton numerical scheme on the **METHOD** statement. You can perform ac small signal analysis with current boundary conditions.

3.5.5 Insulating Contacts

Insulating contacts are contacts that are completely surrounded by insulator. They may be connected to a voltage source (tied contact) or they may be floating.

Insulating contacts that are connected to a voltage source generally have a work function that dictates a value for ψ_s similar to that given by Equation 3-168. Electron and hole concentrations within the insulator and at the insulating contact are forced to be zero (i.e., $n_s=p_s=0$).

3.5.6 Neumann Boundaries

Along the outer (non-contact) edges of devices, homogeneous (reflecting) Neumann boundary conditions are imposed so that current only flows out of the device through the contacts. In the absence of surface charge along such edges, the normal electric field component becomes zero. Current isn't permitted to flow from the semiconductor into an insulating region except via oxide tunneling models. At the interface between two different materials, the difference between the normal components of the respective electric displacements must be equal to any surface charge according to:

$$\hat{n} \cdot \varepsilon_1 \nabla \psi_1 - \hat{n} \cdot \varepsilon_2 \nabla \psi_2 = \rho_s \qquad 3\text{-}197$$

where n is the unit normal vector, ε_1 and ε_2 are the permittivities of the materials on either side of the interface and ρ_s is the sheet charge at the interface.

3.5.7 Lumped Element Boundaries

Atlas supports some simple configurations of lumped elements. These are indicated in Figure 3-5.

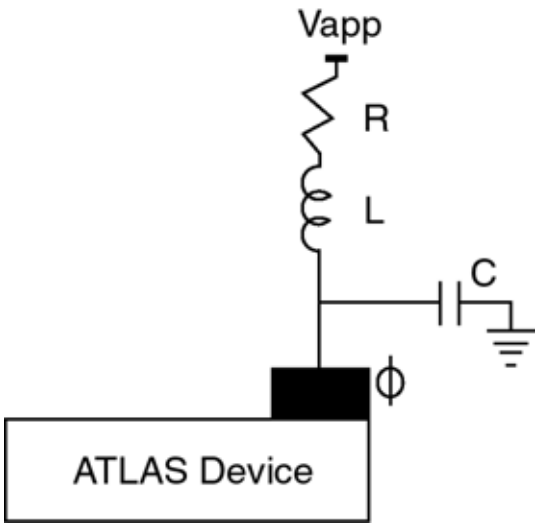


Figure 3-5: The Lumped elements supported by Atlas

Lumped elements can be extremely useful when simulating CMOS or MOSFET structures. For example, the p-tub contact in the CMOS cross-section might be tens or hundreds of microns away from the active area of an embedded vertical npn bipolar transistor, which may only be 10-20 µm on a side. If the whole structure were simulated, a tremendous number of grid points (probably more than half) are essentially wasted in accounting for a purely resistive region of the device.

In the case of a MOSFET substrate, you would not want to include grid points all the way to the back side of a wafer. In either of these cases, a simple lumped resistance can be substituted [255].

Table 3-30 shows the user-specifiable parameters for Figure 3-5.

Table 3-30 User-Specifiable Parameters for Figure 3-2.

Symbol	Statement	Parameter	Units
C	**CONTACT**	CAPACITANCE	F/µm
R	**CONTACT**	RESISTANCE	Ωµm
L	**CONTACT**	INDUCTANCE	Hµm

Resistance is specified in Ωµm, capacitance is specified in F/µ*m*, and inductance is specified in Hµ*m*.

The following combinations are possible: resistance only, resistance and capacitance, inductance and resistance and inductance, capacitance and resistance. For the case of inductance or capacitance only, Atlas adds a small resistance as well. For more complicated circuit configurations, use the MixedMode module of Atlas.

The boundary condition for a lumped resistance boundary is analogous to the current boundary condition described in Section 3.5.4 “Current Boundary Conditions” and is given by:

$$\frac{V_a - V_s}{R} - \Sigma(J_n + J_p) = 0 \qquad 3\text{-}198$$

where V_a is the applied potential, R is the lumped resistance, and J_n and J_p are the electron and hole currents given by Equations 3-11 and 3-12. V_s is the boundary potential or "internal bias" solution. The summation is taken over all grid points incident on the boundary. The ohmic conditions in Equations 3-165 and 3-166 apply for electron and hole concentrations.

Note: Capacitance increases with device width (into the Z plane) while resistance decreases. Except for the case of extremely large resistances where the arrangement becomes similar to a pure current source, no convergence degradation has been observed for a lumped element boundary in comparison to a simple Ohmic contact. Transient simulation therefore becomes easier and is more well-defined.

You should use the simulator to calculate any resistance (or capacitance) components that might be included as lumped elements. When performing CMOS simulations, you could simulate just the p-tub with Ohmic contacts at either end. From the plot of terminal current (in A/μm) versus voltage, resistance can be directly extracted from the slope. Be very careful to consider any three-dimensional effects (e.g., current spreading) before using a resistance value in further simulations.

When looking at the results of simulation with lumped elements it's important to distinguish between the applied voltage (V_{app}) and the internal bias (ϕ) in the log file produced by Atlas.

The lumped element boundary conditions can also be used with AC small signal analysis (see Section 21.10.1 "Frequency Domain Perturbation Analysis"). In this case, the driving voltage is a harmonic signal of small amplitude and the capacitance shown in Figure 3-5 is connected to AC ground. Because some of the AC current goes to this ground, the measured admittances may not always sum to zero.

An alternative circuit arrangement is also available. If you specify the RESONANT parameter on a **CONTACT** statement, then for that contact the capacitance will also connected to the driving voltage.

In this case, the lumped elements form a parallel resonant circuit with the series resistor and inductance in parallel with the capacitance. Also in this case, current continuity will be explicitly satisfied.

The susceptance may be inductive rather than capacitive if inductors are present in the lumped elements. This manifests itself as a negative capacitance (i.e., the conductance and capacitance are of different signs). You can easily convert the output capacitance to an inductance using the formula

$$L = 1/(\omega^2 C) \qquad 3\text{-}199$$

where ω is the angular frequency, C is the capacitance in farads and L the inductance in henrys.

3.5.8 Distributed Contact Resistance

Since contact materials have finite resistivities, the electrostatic potential isn't always uniform along the metal-semiconductor surface. To account for this effect, a distributed contact resistance can be associated with any electrode. This is in contrast to the lumped contact resistance described in the previous section.

Atlas implements this distributed contact resistance by assigning a resistance value to each surface element associated with the contact. If each surface element is a different length, a different value for the contact resistance will be assigned. Atlas calculates a resistance value of R_i for each surface element from the value of CON.RESIST, as specified on the **CONTACT** statement. The units of CON.RESIST are Ωcm^2 and R_i calculated as:

$$R_i = \frac{\text{CON.RESIST}}{d_i \, \text{WIDTH}} \qquad 3\text{-}200$$

where R_i is the resistance at node i, P_c is specified by the CON.RESIST parameter, d_i is the length of the contact surface segment associated with node i and WIDTH is the width of the device. The effect of the resistance, R_i is to add an extra equation to be satisfied to node i.

This equation is given by:

$$\frac{1}{R_i}\left[V_{applied} - \left(\psi_i \pm \frac{kT_L}{q}\ln(N/n_{ie})\right)\right] - (I_n + I_p + I_{disp}) = 0 \quad \text{3-201}$$

where $V_{applied}$ is the external applied voltage, Ψ_i is the surface potential, N is the net doping, n_i is the intrinsic electron concentration, and I_n, I_p, I_{disp} are the electron, hole and displacement currents at node i. This equation simply balances the current in and out of the resistor added to each i node.

As with the case for lumped elements, Atlas can print out a value of contact resistance for each contact in the run time output. Since the actual value depends on the length of each surface segment for distributed contacts, Atlas prints out the value of CON.RESIST/WIDTH which is the same for all contact surface segments. This runtime output is enabled by adding the PRINT option on the **MODELS** statement.

Table 3-31 User-Specifiable Parameters for Equation 3-200

Statement	Parameter	Units
CONTACT	CON.RESIST	Ω-cm^2
MESH	WIDTH	μm

Note: AC small signal analysis cannot be performed when any distributed contact resistance has been specified. Also, only the NEWTON numerical scheme should be chosen on the **METHOD** statement.

3.5.9 Energy Balance Boundary Conditions

When the Energy Balance Transport Model is applied, special boundary conditions are applied for carrier temperatures. By default at the contacts, Dirichlet boundary conditions are used for carrier temperatures:

$$T_n = T_p = T_L \quad \text{3-202}$$

You can treat contacts as Neumann (reflective) boundaries with respect to carrier temperature by specifying REFLECT on the **CONTACT** statement.

Elsewhere on the boundary, the normal components of the energy fluxes vanish. The boundary conditions for (ψ, n, p) are the same as for the drift-diffusion model.

3.5.10 Floating Semiconductor Regions

A semiconductor region that is totally surrounded by insulator can be modeled as a floating contact as described in Section 3.5.3 “Floating Contacts”. The approximations made there are that the floating region is an equipotential and that the value of the equipotential is derived from a macroscopic application of Gauss's flux theorem to the whole region. The Poisson equation is not solved in the interior of the region.

This approximation is appropriate for a metallic contact but highly doped polysilicon is commonly used as a floating contact. In this case, it is more appropriate to make the assumption that the potential is non-uniform and that the Fermi-level is constant in the floating region. The electron and hole densities then vary across the region according to Equations 3-36 and 3-37. The electron and hole quasi-Fermi potentials, ϕ_n and ϕ_p, in this case are equal to the same constant value.

The value of the Fermi level is obtained from the constraint that

$$\int_V Q(\vec{r})dV = Q_{FG} \qquad 3\text{-}203$$

where the local charge density comprises the ionized dopant densities, along with the free carrier densities. The integration is over the volume of the region and Q_{FG} is the total charge associated with the region. Within the floating semiconductor region, the Poisson equation is solved to give the potential distribution. The current-continuity equations are not solved there because the assumption of constant quasi-Fermi level implies that there is no current flow within the region.

It is convenient to refer to these floating semiconductor regions as contacts. The value of Q_{FG} for the region can be set explicitly on the **SOLVE** statement as for the floating contacts in Section 3.5.3 “Floating Contacts”. On transient **SOLVE** statements, if a gate charging model is enabled, then the value of Q_{FG} can evolve with time. The value of Q_{FG} is sent to the runtime output.

To define a floating region, you define an **ELECTRODE** and add the FLOATING parameter. You must also set the **ELECTRODE** to be a semiconductor by using the MATERIAL parameter. For example

```
ELECTRODE NAME=FGATE X.MIN=0.0 X.MAX=1.0 Y.MIN=-0.08  Y.MAX=-0.02
MATERIAL=POLYSILICON FLOATING
```

For a pre-existing electrode read in from a standard structure file, you use the MODIFY parameter on the **ELECTRODE** statement along with the FLOATING parameter. For example

```
ELECTRODE MODIFY NAME="fgate" FLOATING
```

Table 3-32 Parameters for the FLOATING ELECTRODE Model

Statement	Parameter	Type	Default	Units
ELECTRODE	FLOATING	Logical	False	
ELECTRODE	MODIFY	Logical	False	

3.5.11 High-K Effective Workfunction Model

Atlas includes a model for effective metal workfunction in high-k gate insulators. This model is based on the charge neutrality level model [367]. In this model, the effective workfunction is given by:

$$\Phi_{meff} = \texttt{E.HIGHK} + \texttt{S.HIGHK}(\texttt{WORKF} - \texttt{E.HIGHK}) \quad \text{3-204}$$

where Φ_{meff} is the effective gate workfunction, E.HIGHK is the charge neutrality level of the high-k material, S.HIGHK is a slope parameter of the high-k material, and WORKF is the metal workfunction in vacuum. The parameters WORKF, E.HIGHK, and S.HIGHK can all be specified on the **CONTACT** statement.

The following default metal workfunctions can be used by specifying the associated parameter name on the **CONTACT** statement.

Table 3-33 Parameterized Workfunction Defaults

Parameter Name	WORKF (eV)
HIGHK.MG	3.66
HIGHK.TA	4.25
HIGHK.AL	4.28
HIGHK.TIN	4.53
HIGHK.W	4.63
HIGHK.MO	4.95
HIGHK.PT	5.65
HIGHK.NI	5.04
HIGHK.AU	5.39
HIGHK.HF	3.95

The following default high-k parameter values can be used by specifying the associated parameter name on the **CONTACT** statement.

Table 3-34 Parameterized Dielectric Parameter Defaults

Parameter Name	E.HIGHK (eV)	S.HIGHK
ON.SIO2	4.81	0.95
ON.AL2O3	3.43	0.69

Table 3-34 Parameterized Dielectric Parameter Defaults		
Parameter Name	**E.HIGHK (eV)**	**S.HIGHK**
ON.SI3N4	4.16	0.59
ON.HFO2	4.81	0.52
ON.ZRO2	4.53	0.52

Note: Here, the energy values are derived from the charge neutrality level values from [367] and the bandgap and conduction band edge offset values from [268].

For example using Tables 3-33 and 3-34, you can specify an aluminum gate with a HfO_2 dielectric by the following statement:

```
CONTACT NAME=gate HIGHK.AL ON.HFO2
```

This simple statement specifies that Equation 3-204 is used to calculate the effective gate workfunction.

You can also modify the workfunction to include the effects of a second interfacial dielectric layer in an anologous manner to the main dielectric layer. The interfacial layer is always located between the high-k gate dielectric layer and the semiconductor.

In these cases, you can specify one of the logical parameters in Table 3-35. When you include one of these parameters, the effective workfunction is calculated as in Equation 3-204, except that the equation is applied recursively to the interfacial layer using the effective workfunction from Equation 3-204 for WORKF and E.HIGHK and S.HIGHK from the interface layer to obtain a composite effective workfunction.

The parameters used to enable interfacial layers are given in Table 3-35.

Table 3-35 Parameterized Dielectric Interface Layer Parameter Defaults		
Parameter Name	**E.HIGHK (eV)**	**S.HIGHK**
IFL.SIO2	4.81	0.95
IFL.AL2O3	3.43	0.69
IFL.SI3N4	4.16	0.59
IFL.HFO2	4.81	0.52
IFL.ZRO2	4.53	0.52

3.6 Physical Models

3.6.1 Mobility Modeling

Electrons and holes are accelerated by electric fields, but lose momentum as a result of various scattering processes. These scattering mechanisms include lattice vibrations (phonons), impurity ions, other carriers, surfaces, and other material imperfections. Since the effects of all of these microscopic phenomena are lumped into the macroscopic mobilities introduced by the transport equations these mobilities are therefore functions of the local electric field, lattice temperature, doping concentration, and so on.

Mobility modeling is normally divided into: (i) low-field behavior, (ii) high field behavior, (iii) bulk semiconductor regions and (iv) inversion layers.

The low electric field behavior has carriers almost in equilibrium with the lattice and the mobility has a characteristic low-field value that is commonly denoted by the symbol $\mu_{n0,p0}$. The value of this mobility is dependent upon phonon and impurity scattering. Both of which act to decrease the low-field mobility.

The high electric field behavior shows that the carrier mobility declines with electric field because the carriers that gain energy can take part in a wider range of scattering processes. The mean drift velocity no longer increases linearly with increasing electric field, but rises more slowly. Eventually, the velocity doesn't increase any more with increasing field but saturates at a constant velocity. This constant velocity is commonly denoted by the symbol ν_{sat}. Impurity scattering is relatively insignificant for energetic carriers, and so ν_{sat} is primarily a function of the lattice temperature.

Modeling mobility in bulk material involves: (i) characterizing μ_{n0} and μ_{p0} as a function of doping and lattice temperature, (ii) characterizing ν_{sat} as a function of lattice temperature, and (iii) describing the transition between the low-field mobility and saturated velocity regions.

Modeling carrier mobilities in inversion layers introduces additional complications. Carriers in inversion layers are subject to surface scattering, extreme carrier-carrier scattering, and quantum mechanical size quantization effects. These effects must be accounted for in order to perform accurate simulation of MOS devices. The transverse electric field is often used as a parameter that indicates the strength of inversion layer phenomena.

You can define multiple non-conflicting mobility models simultaneously. You also need to know which models are over-riding when conflicting models are defined.

Low-Field Mobility Models

The low-field carrier mobility can be defined in five different ways.

The first way is use the `MUN` and `MUP` parameters to set constant values for electron and hole mobilities and optionally specify temperature dependence. The second way is by using a look-up table model (`CONMOB`) to relate the low-field mobility at 300K to the impurity concentration. The third way is by choosing the analytic low-field mobility models, `ANALYTIC`, `ARORA`, or `MASETTI`, to relate the low-field carrier mobility to impurity concentration and temperature.The fourth way is by choosing a carrier-carrier scattering model (`CCSMOB`, `CONWELL`, or `BROOKS`) that relates the low-field mobility to the carrier concentrations and temperature. The fifth way is to use a unified low-field mobility model (`KLAASSEN`) that relates the low-field mobility to donor, acceptor, lattice, carrier-carrier scattering, and temperature.

Constant Low-Field Mobility Model

In Atlas, the choice of mobility model is specified on the **MODELS** statement. The parameters associated with mobility models are specified on a separate **MOBILITY** statement. One or more mobility models should always be specified explicitly. The default is to use constant low-field mobilities within each region of a device. This default model is independent of doping concentration, carrier densities and electric field. It does account for lattice scattering due to temperature according to:

$$\mu_{n0} = \mathrm{MUN}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUN}} \quad \text{3-205}$$

$$\mu_{p0} = \mathrm{MUP}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUP}} \quad \text{3-206}$$

where T is the lattice temperature. The low-field mobility parameters: MUN, MUP, TMUN, and TMUP can be specified in the **MOBILITY** statement with the defaults as shown in Table 3-36.

Table 3-36 User-Specifiable Parameters for the Constant Low-Field Mobility Model

Statement	Parameter	Default	Units
MOBILITY	MUN	1000	$cm^2/(V \cdot s)$
MOBILITY	MUP	500	$cm^2/(V \cdot s)$
MOBILITY	TMUN	1.5	
MOBILITY	TMUP	1.5	

Concentration-Dependent Low-Field Mobility Tables

Atlas provides empirical data for the doping dependent low-field mobilities of electrons and holes in silicon at T_L=300K only. This data is used if the CONMOB parameter is specified in the **MODELS** statement. The data that is used is shown in Table 3-37.

Table 3-37 Mobility of Electrons and Holes in Silicon at T=300K

Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
1.0×10^{14}	1350.0	495.0
2.0×10^{14}	1345.0	495.0
4.0×10^{14}	1335.0	495.0
6.0×10^{14}	1320.0	495.0
8.0×10^{14}	1310.0	495.0
1.0×10^{15}	1300.0	491.1

Table 3-37 Mobility of Electrons and Holes in Silicon at T=300K		
Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
2.0×10^{15}	1248.0	487.3
4.0×10^{15}	1200.0	480.1
6.0×10^{15}	1156.0	473.3
8.0×10^{15}	1115.0	466.9
1.0×10^{16}	1076.0	460.9
2.0×10^{16}	960.0	434.8
4.0×10^{16}	845.0	396.5
6.0×10^{16}	760.0	369.2
8.0×10^{16}	720.0	348.3
1.0×10^{17}	675.0	331.5
2.0×10^{17}	524.0	279.0
4.0×10^{17}	385.0	229.8
6.0×10^{17}	321.0	2103.8
8.0×10^{17}	279.0	186.9
1.0×10^{18}	252.0	178.0
2.0×10^{18}	182.5	130.0
4.0×10^{18}	140.6	90.0
6.0×10^{18}	113.6	74.5
8.0×10^{18}	99.5	66.6
1.0×10^{19}	90.5	61.0
2.0×10^{19}	86.9	55.0
4.0×10^{19}	83.4	53.7
6.0×10^{19}	78.8	52.9
8.0×10^{19}	71.6	52.4
1.0×10^{20}	67.8	52.0
2.0×10^{20}	52.0	50.8
4.0×10^{20}	35.5	49.6

Table 3-37 Mobility of Electrons and Holes in Silicon at T=300K		
Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
6.0×10^{20}	23.6	48.9
8.0×10^{20}	19.0	48.4
1.0×10^{21}	17.8	48.0

Analytic Low-Field Mobility Model

The following analytic function based upon the work of Caughey and Thomas [48, 287] can be used to specify doping- and temperature-dependent low-field mobilities.

$$\mu_{n0} = \mathrm{MU1N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{ALPHAN.CAUG}} + \frac{\mathrm{MU2N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{BETAN.CAUG}} - \mathrm{MU1N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{ALPHAN.CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\mathrm{GAMMAN.CAUG}} \cdot \left(\frac{N}{\mathrm{NCRITN.CAUG}}\right)^{\mathrm{DELTAN.CAUG}}} \quad \text{3-207}$$

$$\mu_{p0} = \mathrm{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{ALPHAP \cdot CAUG}} + \frac{\mathrm{MU2P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{BETAP \cdot CAUG}} - \mathrm{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathrm{ALPHAP \cdot CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\mathrm{GAMMAP \cdot CAUG}} \cdot \left(\frac{N}{\mathrm{NCRITP.CAUG}}\right)^{\mathrm{DELTAP \cdot CAUG}}} \quad \text{3-208}$$

where N is the local (total) impurity concentration in cm^{-3} and T_L is the temperature in degrees Kelvin.

This model is activated by specifying ANALYTIC in the **MODELS** statement. The parameters of this model are specified in the **MOBILITY** statement. The default parameters are for silicon at T_L = 300K.

Table 3-38 User-Specifiable Parameters for Equations 3-207 and 3-208			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MU1N.CAUG	55.24	$cm^2/(V \cdot s)$
MOBILITY	MU1P.CAUG	49.7	$cm^2/(V \cdot s)$
MOBILITY	MU2N.CAUG	1429.23	$cm^2/(V \cdot s)$
MOBILITY	MU2P.CAUG	479.37	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.CAUG	0.0	arbitrary
MOBILITY	ALPHAP.CAUG	0.0	arbitrary
MOBILITY	BETAN. CAUG	-2.3	arbitrary
MOBILITY	BETAP. CAUG	-2.2	arbitrary
MOBILITY	GAMMAN.CAUG	-3.8	arbitrary
MOBILITY	GAMMAP.CAUG	-3.7	arbitrary
MOBILITY	DELTAN.CAUG	0.73	arbitrary
MOBILITY	DELTAP.CAUG	0.70	arbitrary
MOBILITY	NCRITN.CAUG	1.072×10^{17}	cm^{-3}
MOBILITY	NCRITP.CAUG	1.606×10^{17}	cm^{-3}

Arora Model for Low-Field Mobility

Another analytic model for the doping and temperature dependence of the low-field mobility is available in Atlas. This model, which is due to Arora [16], has the following form:

$$\mu_{n0} = \mathrm{MU1N.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAN.ARORA}} + \frac{\mathrm{MU2N.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{BETAN.ARORA}}}{1 + \frac{N}{\mathrm{NCRITN.ARORA}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{GAMMAN.ARORA}}}} \quad 3\text{-}209$$

$$\mu_{p0} = \mathrm{MU1P.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAP.ARORA}} + \frac{\mathrm{MU2P.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{BETAP.ARORA}}}{1 + \frac{N}{\mathrm{NCRITP.ARORA}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{GAMMAP.ARORA}}}} \quad 3\text{-}210$$

where N is the total local dopant concentration.

This model is activated by specifying ARORA in the **MODELS** statement. The parameters of the model are specified in the **MOBILITY** statement. The default parameters are for silicon at T_L=300K.

Table 3-39 User-Specifiable Parameters for Equations 3-209 and 3-210

Statement	Parameter	Default	Units
MOBILITY	MU1N.ARORA	88.0	$cm^2/(V \cdot s)$
MOBILITY	MU1P.ARORA	54.3	$cm^2/(V \cdot s)$
MOBILITY	MU2N.ARORA	1252.0	$cm^2/(V \cdot s)$
MOBILITY	MU2P.ARORA	407.0	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.ARORA	-0.57	
MOBILITY	ALPHAP.ARORA	-0.57	
MOBILITY	BETAN.ARORA	-2.33	
MOBILITY	BETAP.ARORA	-2.33	
MOBILITY	GAMMAN.ARORA	2.546	
MOBILITY	GAMMAP.ARORA	2.546	
MOBILITY	NCRITN.ARORA	1.432×10^{17}	cm^{-3}
MOBILITY	NCRITP.ARORA	2.67×10^{17}	cm^{-3}

Masetti Model For Low-Field Mobility

Masetti et al. [200] modeled the dependence of mobility on carrier concentration over a range of 8 orders of magnitude in carrier concentration from approximately 10^{13} cm^{-3} to 10^{21} cm^{-3}. They found that their model required different parameter sets for the electron mobility in Arsenic and Phosphorous n-doped Silicon. The model is optimized for room temperature, although it does model temperature dependence to some extent.

The functional form of the electron mobility is

$$\mu_n = \mathrm{MTN.MIN1}\, exp\left(-\frac{\mathrm{MTN.PC}}{N}\right) + \frac{\mathrm{MTN.MAX} - \mathrm{MTN.MIN2}}{1 + \left(\frac{N}{\mathrm{MTN.CR}}\right)^{\mathrm{MTN.ALPHA}}} - \frac{\mathrm{MTN.MU1}}{1 + \left(\frac{\mathrm{MTN.CS}}{N}\right)^{\mathrm{MTN.BETA}}} \quad 3\text{-}211$$

where N is the total or ionized doping level. The functional form of the hole mobility is

$$\mu_p = \mathrm{MTP.MIN1}\, exp\left(-\frac{\mathrm{MTP.PC}}{N}\right) + \frac{\mathrm{MTP.MAX} - \mathrm{MTP.MIN2}}{1 + \left(\frac{N}{\mathrm{MTP.CR}}\right)^{\mathrm{MTP.ALPHA}}} - \frac{\mathrm{MTP.MU1}}{1 + \left(\frac{\mathrm{MTP.CS}}{N}\right)^{\mathrm{MTP.BETA}}} \quad 3\text{-}212$$

where N is the total dopant concentration. MTN.MAX and MTP.MAX are given a lattice temperature dependence as in Equations 3-205 and 3-206 respectively. These equations are functionally equivalent to the bulk mobility term in the Lombardi CVT model (Equations 3-258 and 3-259). To use it as a stand alone mobility model, use the MASETTI parameter on the **MODELS** statement for both electron and hole mobility.

N.MASETTI on the **MOBILITY** statement enables it for electrons. P.MASETTI on the **MOBILITY** statement enables it for holes.

For electron mobility, there is a choice of two default parameter sets. The set for arsenic doping are used unless you set the MSTI.PHOS parameter on the **MOBILITY** statement to use the set corresponding to Phosphorous. The following tables show default sets for electron mobility and for Boron doping (p-type).

Table 3-40 Parameter Set for Electron Mobility with Arsenic Doping in Silicon.

Statement	Parameter	Default	Units
MOBILITY	MTN.MIN1	52.2	$cm^2/(Vs)$
MOBILITY	MTN.MIN2	52.2	$cm^2/(Vs)$
MOBILITY	MTN.MAX	1417	$cm^2/(Vs)$
MOBILITY	MTN.MU1	43.4	$cm^2/(Vs)$
MOBILITY	MTN.PC	0.0	cm^{-3}
MOBILITY	MTN.CR	9.68×10^{16}	cm^{-3}
MOBILITY	MTN.CS	3.34×10^{20}	cm^{-3}
MOBILITY	MTN.ALPHA	0.68	
MOBILITY	MTN.BETA	2.0	

Table 3-41 Parameter Set for Electron Mobility with Phosphorous Doping in Silicon (MSTI.PHOS set)

Statement	Parameter	Default	Units
MOBILITY	MTN.MIN1	68.5	$cm^2/(Vs)$
MOBILITY	MTN.MIN2	68.5	$cm^2/(Vs)$
MOBILITY	MTN.MAX	1414	$cm^2/(Vs)$
MOBILITY	MTN.MU1	56.1	$cm^2/(Vs)$
MOBILITY	MTN.PC	0.0	cm^{-3}
MOBILITY	MTN.CR	9.2×10^{16}	cm^{-3}

Table 3-41 Parameter Set for Electron Mobility with Phosphorous Doping in Silicon (MSTI.PHOS set)			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MTN.CS	3.41×10^{20}	cm^{-3}
MOBILITY	MTN.ALPHA	0. 711	
MOBILITY	MTN.BETA	1.98	

Table 3-42 Parameter Set for Hole Mobility with Boron doping in Silicon.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MTP.MIN1	44.9	$cm^2/(Vs)$
MOBILITY	MTP.MIN2	0.0	$cm^2/(Vs)$
MOBILITY	MTP.MAX	470.5	$cm^2/(Vs)$
MOBILITY	MTP.MU1	29.0	$cm^2/(Vs)$
MOBILITY	MTP.PC	9.23×10^{16}	cm^{-3}
MOBILITY	MTP.CR	2.23×10^{17}	cm^{-3}
MOBILITY	MTP.CS	6.1×10^{20}	cm^{-3}
MOBILITY	MTP.ALPHA	0.719	
MOBILITY	MTP.BETA	2.0	

Carrier-Carrier Scattering Models For Low-Field Mobility

Dorkel and Leturcq Models

The Dorkel and Leturcq Model [77] for low-field mobility includes the dependence on temperature, doping, and carrier-carrier scattering. This model is activated by specifying the CCSMOB parameter of the **MODELS** statement. The parameters of the model are specified in the **MOBILITY** statement. This model has the form:

$$\mu_{n0,p0} = \mu^L_{n,p}\left[\left(\frac{1.025}{1+\left[2.126\left(\frac{\mu^L_{n,p}}{\mu_{n,p}}\right)_{IC}\right]^{0.715}}\right)-0.025\right] \quad \text{3-213}$$

where L is the lattice scattering, I is the ionized impurity scattering, and C is the carrier-carrier scattering. Here, $\mu^{IC}_{n,p}$ is defined as:

$$\mu^{IC}_{n,p} = \left[\frac{1}{\mu^C}+\frac{1}{\mu^I_{n,p}}\right]^{-1} \quad \text{3-214}$$

where:

$$\mu^{C} = \frac{1.04 \cdot 10^{21}\left(\frac{T_L}{300}\right)^{3/2}}{\sqrt{np}\ln\left[1 + 7.45 \cdot 10^{13}\left(\frac{T_L}{300}\right)^{2}(np)^{-1/3}\right]} \quad 3\text{-}215$$

$$\mu_n^I = \frac{\mathtt{AN.CCS}(T_L)^{3/2}}{N_T} f\left[\frac{\mathtt{BN.CCS}(T_L)^2}{n+p}\right] \quad 3\text{-}216$$

$$\mu_p^I = \frac{\mathtt{AP.CCS}(T_L)^{3/2}}{N_T} f\left[\frac{\mathtt{BP.CCS}(T_L)^2}{n+p}\right] \quad 3\text{-}217$$

Here, N_T is the total concentration, T_L is the lattice temperature and n,p are the electron and hole carrier concentrations respectively.

$$f(x) = \left[\ln(1+x) - \frac{x}{1+x}\right]^{-1} \quad 3\text{-}218$$

The values of the lattice scattering terms, $\mu_{N,P}{}^{L}$ are defined by Equations 3-205 and 3-206.

Table 3-43 User-Specifiable Parameters for Equations 3-216 and 3-217

Statement	Parameter	Default	Units
MOBILITY	AN.CCS	4.61×10^{17}	cm^{-3}
MOBILITY	AP.CCS	1.0×10^{17}	cm^{-3}
MOBILITY	BN.CCS	1.52×10^{15}	cm^{-3}
MOBILITY	BP.CCS	6.25×10^{14}	cm^{-3}

Conwell-Weisskopf Model

This model adapts Conwell-Weisskopf theory to carrier-carrier scattering [55]. According to this model, the carrier-carrier contribution to mobility is

$$\mu_{ccs} = \frac{\mathtt{D.CONWELL}\left(\frac{T}{300}\right)^{3/2}}{\sqrt{pn}\left(\ln\left(1 + \frac{\mathtt{F.CONWELL}}{(pn)^{(1/3)}}\left(\frac{T}{300}\right)^{2}\right)\right)} \quad 3\text{-}219$$

where T is the Lattice temperature in Kelvin, n is the electron concentration, and p is the hole concentration. This is then combined with other enabled low-field mobility models (if any) using Matthiessens rule, giving overall low-field mobilities as follows

$$\frac{1}{\mu_n} = \frac{1}{\mu_{n0}} + \frac{1}{\mu_{ccs}} \quad 3\text{-}220$$

$$\frac{1}{\mu_p} = \frac{1}{\mu_{p0}} + \frac{1}{\mu_{ccs}} \quad \text{3-221}$$

To enable the model for both electrons and holes, specify CONWELL on the **MODELS** statement. Alternatively, N.CONWELL on the **MOBILITY** statement enables it for electron mobility. P.CONWELL on the **MOBILITY** statement enables it for hole mobility.

Table 3-44 MOBILITY Statement Parameters

Parameter	Type	Default	Units
D.CONWELL	Real	1.04×10^{21}	$(cmV\cdot s)^{-1}$
F.CONWELL	Real	7.452×10^{13}	cm^{-2}
N.CONWELL	Logical	False	
P.CONWELL	Logical	False	

Brooks-Herring Model [190]

Like the Conwell-Weisskopf model, this adds in the effects of carrier-carrier scattering to the low-field mobility using Matthiesen's rule. The carrier-carrier contribution to mobility is

$$\mu_{ccs} = \frac{\text{A.BROOKS}\left(\frac{T}{300}\right)^{\frac{3}{2}}}{\sqrt{pn}\ \ \phi(\eta)} \quad \text{3-222}$$

where

$$\phi(\eta) = \log(1+\eta) - \frac{\eta}{1+\eta} \quad \text{3-223}$$

and

$$\eta(T) = \frac{\text{B.BROOKS}\left(\frac{T}{300}\right)^{2}}{N_c F_{-1/2}\left(\frac{n}{N_c}\right) + N_v F_{-1/2}\left(\frac{p}{N_v}\right)} \quad \text{3-224}$$

$F_{-1/2}$ is the Fermi-Dirac function of order -1/2, N_c and N_v are the conduction band and valence band effective densities of states, n is the electron concentration, p is the hole concentration, and T is the Lattice temperature.

To enable the model for both electrons and holes, specify BROOKS on the **MODELS** statement. Alternatively, N.BROOKS on the **MOBILITY** statement enables it for electron mobility. P.BROOKS on the **MOBILITY** statement enables it for hole mobility.

Table 3-45 MOBILITY Statement Parameters			
Parameter	**Type**	**Default**	**Units**
A.BROOKS	Real	1.56×10^{21}	$(cmV\cdot s)^{-1}$
B.BROOKS	Real	7.63×10^{19}	cm^{-3}
N.BROOKS	Logical	False	
P.BROOKS	Logical	False	

Incomplete Ionization and Doping Dependent Mobility

The default in Atlas is to use the total doping (N_A+N_D) in the formulae for the doping concentration dependence of low-field mobility (e.g., ARORA model). As an alternative, Atlas can use the ionized dopant concentration.

The ionized doping concentration is obtained automatically and used in calculating the charge density when you set the INCOMPLETE parameter on the **MODELS** statement. See Section 3.3.1 "Incomplete Ionization of Impurities" for more information.

Atlas has a MOB.INCOMPL parameter on the **MODELS** statement. This allows you to use the ionized dopant concentration to calculate doping dependent mobilities. Specifying this parameter automatically sets the INCOMPLETE parameter. This ensures that the solution is consistent with an ionized rather than total doping concentration.

For high doping levels, the difference between total doping and ionized doping concentrations can be large and so this can result in quite different mobility values.

If it is required to use the ionized dopant concentration for mobility calculations and the total dopant concentration for calculating the charge density in Poisson's equation, then specify MOB.INCOMPL ^INCOMPLETE on the **MODELS** statement to explicitly clear this flag. This combination of parameters is not recommended.

MOB.INCOMPL affects the ANALYTIC, ARORA, MASETTI, YAMAGUCHI, CVT, KLAASSEN, ALBRECHT and tabulated Low-Field Mobility Models.

Klaassen's Unified Low-Field Mobility Model

The model by D. B. M. Klaassen [158, 159], provides a unified description of majority and minority carrier mobilities. In so doing, it includes the effects of lattice scattering, impurity scattering (with screening from charged carriers), carrier-carrier scattering, and impurity clustering effects at high concentration. The model shows excellent agreement between the modeled and empirical data for:

- majority electron mobility as a function of donor concentration over the range of 10^{14} cm^{-3} to 10^{22} cm^{-3}
- minority electron mobility as a function of acceptor concentration over the range of 10^{17} cm^3 to 10^{20} cm^{-3}
- minority hole mobility as a function of donor concentration from 10^{17} cm^{-3} to 10^{20} cm^{-3}
- temperature dependence over the range of 70 K to 500 K

The Klaassen Model accounts for a broader set of effects and has been calibrated over a wider range of conditions than any other of the low-field bulk mobility models. This is the recommended model for both MOS and bipolar simulation and is the default model for silicon when you set MOS2 or BIPOLAR2 in the **MODELS** statement. You can enable or disable the model by using the KLA parameter in the **MODELS** statement, or independently for electrons and holes by the KLA.N and KLA.P parameters of the **MOBILITY** statement.

The total mobility can be described by its components using Matthiessen's rule as:

$$\mu_{n0}^{-1} = \mu_{nL}^{-1} + \mu_{nDAP}^{-1} \quad \text{3-225}$$

$$\mu_{p0}^{-1} = \mu_{pL}^{-1} + \mu_{pDAP}^{-1} \quad \text{3-226}$$

μ_n and μ_p are the total low-field electron and hole mobilities, μ_{nL} and μ_{pL} are the electron and hole mobilities due to lattice scattering, μ_{nDAP} and μ_{pDAP} are the electron and hole mobilities due to donor (D), acceptor (A), screening (P) and carrier-carrier scattering.

The lattice scattering components, μ_{nL} and μ_{pL} are given as:

$$\mu_{nL} = \mathtt{MUMAXN.KLA}\left(\frac{300}{T_L}\right)^{\mathtt{THETAN.KLA}} \quad \text{3-227}$$

$$\mu_{pL} = \mathtt{MUMAXP.KLA}\left(\frac{300}{T_L}\right)^{\mathtt{THETAP.KLA}} \quad \text{3-228}$$

where T_L is the temperature in degrees Kelvin. MUMAXN.KLA, MUMAXP.KLA, THETAN.KLA, and THETAP.KLA are user-definable model parameters that can be specified as shown in Table 3-46.

Table 3-46 User-Specifiable Parameters for Equations 3-227 and 3-228

Statement	Parameter	Default	Units
MOBILITY	MUMAXN.KLA	1417.0	$cm^2/(V \cdot s)$
MOBILITY	MUMAXP.KLA	470.5	$cm^2/(V \cdot s)$
MOBILITY	THETAN.KLA	2.285	
MOBILITY	THETAP.KLA	2.247	

The impurity-carrier scattering components of the total mobility are given by:

$$\mu_{nDAP} = \mu_{N,n}\frac{N_{nsc}}{N_{nsc,eff}}\left(\frac{\mathtt{NREF1N.KLA}}{N_{nsc}}\right)^{\mathtt{ALPHA1N.KLA}} + \mu_{nc}\left(\frac{n+p}{N_{nsc,eff}}\right) \quad \text{3-229}$$

$$\mu_{pDAP} = \mu_{N,p}\frac{N_{psc}}{N_{psc,eff}}\left(\frac{\mathtt{NREF1P.KLA}}{N_{psc}}\right)^{\mathtt{ALPHA1P.KLA}} + \mu_{pc}\left(\frac{n+p}{N_{psc,eff}}\right) \quad \text{3-230}$$

Table 3-47 User-Specifiable Parameters for Equations 3-229 and 3-230			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALPHA1N.KLA	0.68	
MOBILITY	ALPHA1P.KLA	0.719	
MOBILITY	NREF1N.KLA	9.68×10^{16}	cm^3
MOBILITY	NREF1P.KLA	2.23×10^{17}	cm^3

The impurity scattering components, $\mu_{N,n}$ and $\mu_{N,p}$, are given by:

$$\mu_{N,n} = \frac{\mathrm{MUMAXN.KLA}^2}{\mathrm{MUMAXN.KLA} - \mathrm{MUMINN.KLA}}\left(\frac{T_L}{300}\right)^{(3\times\mathrm{ALPHA1N.KLA}-1.5)} \quad 3\text{-}231$$

$$\mu_{N,p} = \frac{\mathrm{MUMAXP.KLA}^2}{\mathrm{MUMAXP.KLA} - \mathrm{MUMINP.KLA}}\left(\frac{T_L}{300}\right)^{(3\times\mathrm{ALPHA1P.KLA}-1.5)} \quad 3\text{-}232$$

where T_L is the temperature in degrees Kelvin. MUMINN.KLA and MUMINP.KLA are user-defined parameters shown in Table 3-48, and the other parameters are as described in Tables 3-46 and 3-47.

Table 3-48 User-Specifiable Parameters for Equations 3-231 and 3-232			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MUMINN.KLA	52.2	$cm^2/(V\cdot s)$
MOBILITY	MUMINP.KLA	44.9	$cm^2/(V\cdot s)$

The carrier-carrier scattering components, μ_{nc} and μ_{pc}, are given by:

$$\mu_{nc} = \frac{\mathrm{MUMINN.KLA}\times\mathrm{MUMAXN.KLA}}{\mathrm{MUMAXN.KLA}-\mathrm{MUMINN.KLA}}\left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}233$$

$$\mu_{pc} = \frac{\mathrm{MUMINP.KLA}\times\mathrm{MUMAXP.KLA}}{\mathrm{MUMAXP.KLA}-\mathrm{MUMINP.KLA}}\left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}234$$

The N_{nsc} and N_{psc} parameters of Equations 3-229 and 3-230 are given by:

$$N_{nsc} = N_D + N_A + p \quad 3\text{-}235$$

$$N_{psc} = N_D + N_A + n \quad 3\text{-}236$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3}, n is the electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}.

The parameters of Equations 3-229 and 3-230 are given by:

$$N_{nsc,\,eff} = N_D + G(P_n)N_A + \left(\frac{p}{F(P_n)}\right) \quad 3\text{-}237$$

$$N_{psc,\,eff} = N_A + G(P_p)N_D + \left(\frac{n}{F(P_p)}\right) \quad 3\text{-}238$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3} and n is the electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}. The two functions, $G(P)$ and $F(P)$, are functions of the screening factors, P_n and P_p, for electrons and holes. The function, $G(P)$, in Equations 3-237 and 3-238 are given by:

$$G(P_n) = 1 - \frac{\texttt{S1.KLA}}{\left[\texttt{S2.KLA} + P_n\left(\frac{(T_L/300)}{\texttt{ME.KLA}}\right)^{\texttt{S4.KLA}}\right]^{\texttt{S3.KLA}}} + \frac{\texttt{S5.KLA}}{\left[P_n\left(\frac{\texttt{ME.KLA}}{(T_L/300)}\right)^{\texttt{S7.KLA}}\right]^{\texttt{S6.KLA}}} \quad 3\text{-}239$$

$$G(P_p) = 1 - \frac{\texttt{S1.KLA}}{\left[\texttt{S2.KLA} + P_p\left(\frac{(T_L/300)}{\texttt{MH.KLA}}\right)^{\texttt{S4.KLA}}\right]^{\texttt{S3.KLA}}} + \frac{\texttt{S5.KLA}}{\left[P_p\left(\frac{\texttt{MH.KLA}}{(T_L/300)}\right)^{\texttt{S7.KLA}}\right]^{\texttt{S6.KLA}}} \quad 3\text{-}240$$

Here, T_L is the temperature in degrees Kelvin, m_e and m_h are the electron and hole masses and the parameters S1.KLA through S7.KLA are user-specifiable model parameters as shown in Table 3-49..

Table 3-49 User-Specifiable Parameters for Equations 3-239 and 3-240

Statement	Parameter	Default	Units
MOBILITY	S1.KLA	0.89233	
MOBILITY	S2.KLA	0.41372	
MOBILITY	S3.KLA	0.19778	
MOBILITY	S4.KLA	0.28227	
MOBILITY	S5.KLA	0.005978	
MOBILITY	S6.KLA	1.80618	
MOBILITY	S7.KLA	0.72169	

The functions, $F(P_n)$ and $F(P_p)$, in Equations 3-237 and 3-238 are given by:

$$F(P_n) = \frac{\texttt{R1.KLA}\,P_n^{\texttt{R6.KLA}} + \texttt{R2.KLA} + \texttt{R3.KLA}\,\frac{\texttt{ME.KLA}}{\texttt{MH.KLA}}}{P_n^{\texttt{R6.KLA}} + \texttt{R4.KLA} + \texttt{R5.KLA}\,\frac{\texttt{ME.KLA}}{\texttt{MH.KLA}}} \quad 3\text{-}241$$

$$F(P_p) = \frac{\text{R1.KLA}\, P_p^{\text{R6.KLA}} + \text{R2.KLA} + \text{R3.KLA}\, \frac{\text{MH.KLA}}{\text{ME.KLA}}}{P_p^{\text{R6.KLA}} + \text{R4.KLA} + \text{R5.KLA}\, \frac{\text{MH.KLA}}{\text{ME.KLA}}} \quad 3\text{-}242$$

where the parameters, R1.KLA through R6.KLA, are user-specifiable as shown in Table 3-50.

Table 3-50 User-Specifiable Parameters for Equations 3-241 and 3-242

Statement	Parameter	Default	Units
MOBILITY	ME.KLA	1.0	
MOBILITY	MH.KLA	1.258	
MOBILITY	R1.KLA	0.7643	
MOBILITY	R2.KLA	2.2999	
MOBILITY	R3.KLA	6.5502	
MOBILITY	R4.KLA	2.3670	
MOBILITY	R5.KLA	-0.8552	
MOBILITY	R6.KLA	0.6478	

The screening parameters, P_n and P_p, used in Equations 3-241 and 3-242 are given by:

$$P_n = \left[\frac{\text{FCW.KLA}}{P_{CW,n}} + \frac{\text{FBH.KLA}}{P_{BH,n}}\right]^{-1} \quad 3\text{-}243$$

$$P_p = \left[\frac{\text{FCW.KLA}}{P_{CW,p}} + \frac{\text{FBH.KLA}}{P_{BH,p}}\right]^{-1} \quad 3\text{-}244$$

Here, the FCW.KLA and FBH.KLA parameters are user-specifiable model parameters as shown in Table 3-51.

Table 3-51 User-Specifiable Parameters for Equations 3-243 and 3-244

Statement	Parameter	Default	Units
MOBILITY	FCW.KLA	2.459	
MOBILITY	FBH.KLA	3.828	

The functions, $P_{BH,n}$ and $P_{BH,p}$, $P_{CW,n}$, and $P_{CW,p}$ are given by the following equations.

$$P_{BH,n} = \frac{1.36\times10^{20}}{n}(\text{ME.KLA})\left(\frac{T_L}{300}\right)^2 \quad 3\text{-}245$$

$$P_{BH,p} = \frac{1.36\times10^{20}}{p}(\mathtt{MH.KLA})\left(\frac{T_L}{300}\right)^2 \quad 3\text{-}246$$

$$P_{CW,n} = 3.97\times10^{13}\left\{\frac{1}{Z_n^3 N_D}\left(\frac{T_L}{300}\right)^3\right\}^{\frac{2}{3}} \quad 3\text{-}247$$

$$P_{CW,p} = 3.97\times10^{13}\left\{\frac{1}{Z_p^3 N_A}\left(\frac{T_L}{300}\right)^3\right\}^{\frac{2}{3}} \quad 3\text{-}248$$

where T_L is the temperature in degrees Kelvin, m_e/m_0 and m_h/m_0 are the normalized carrier effective masses, and n and p are the electron and hole concentrations in cm^{-3}.

Also here, N_D and N_A are the donor and acceptor concentrations in cm^{-3}, T_L is the temperature in degrees Kelvin, and Z_n and Z_p are clustering functions given by:

$$Z_n = 1 + \frac{1}{\mathtt{CD.KLA} + \left(\frac{\mathtt{NREFD.KLA}}{N_D}\right)^2} \quad 3\text{-}249$$

$$Z_p = 1 + \frac{1}{\mathtt{CA.KLA} + \left(\frac{\mathtt{NREFA.KLA}}{N_A}\right)^2} \quad 3\text{-}250$$

where N_D and N_A are the donor and acceptor concentrations in cm^{-3} and CD.KLA, CA.KLA, NREFD.KLA, and NREFA.KLA are user-definable parameters as given in Table 3-52.

Table 3-52 User-Specifiable Parameters for Equations 3-249 and 3-250

Statement	Parameter	Default	Units
MOBILITY	CD.KLA	0.21	
MOBILITY	CA.KLA	0.50	
MOBILITY	NREFD.KLA	4.0×10^{20}	cm^3
MOBILITY	NREFA.KLA	7.2×10^{20}	cm^3

Note: When the Klaassen low-field mobility is used, remember that it has been calibrated to work with Klaassen's models for bandgap narrowing, KLAAUG recombination, and KLASRH recombination. These models are described in the Section 3.6.3 "Carrier Generation-Recombination Models".

Uchida's Low-Field Model for Ultrathin SOI

The model by Uchida et.al. [317], provides a mobility limit in ultrathin-body MOSFET transistors with SOI thickness less than 4 nm. This mobility limit is due to thickness fluctuations in the nano scale SOI film. The model for electrons and holes is given by Equations 3-251 and 3-252.

$$\mu_{nu} = \texttt{CN.UCHIDA} \cdot \texttt{TN.UCHIDA}^6 \quad \text{3-251}$$

$$\mu_{pu} = \texttt{CP.UCHIDA} \cdot \texttt{TP.UCHIDA}^6 \quad \text{3-252}$$

The parameters CN.UCHIDA, CP.UCHIDA are calibrated from the reference to a default value of 0.78125 $cm^2/(V{\cdot}s{\cdot}nm^6)$. The parameters TN.UCHIDA and TP.UCHIDA represent the thickness of the SOI in µm. This value should as close as possible to match the physical thickness of the SOI layer in the simulated structure file. To enable this model for electrons and holes, specify values for TN.UCHIDA and TP.UCHIDA, and specify the logical parameters UCHIDA.N or UCHIDA.P or both on the **MOBILITY** statement. Table 3-53 lists the user specifiable parameters.

Table 3-53 User-Specifiable Parameters for Equations 3-187 and 3-188

Statement	Parameter	Default	Units
MOBILITY	CN.UCHIDA	0.78125	$cm^2/(V{\cdot}s{\cdot}nm^6)$
MOBILITY	CP.UCHIDA	0.78125	$cm^2/(V{\cdot}s{\cdot}nm^6)$
MOBILITY	TN.UCHIDA	4.0	nm
MOBILITY	TP.UCHIDA	4.0	nm

Inversion Layer Mobility Models

To obtain accurate results for MOSFET simulations, you need to account for the mobility degradation that occurs inside inversion layers. The degradation normally occurs as a result of the substantially higher surface scattering near the semiconductor to insulator interface.

Lombardi CVT Model

The inversion layer model from Lombardi [188] is selected by setting CVT on the **MODELS** statement. This model overrides any other mobility models which may be specified on the **MODELS** statement. In the CVT model, the transverse field, doping dependent and temperature dependent parts of the mobility are given by three components that are combined using Matthiessen's rule. These components are μ_{AC}, μ_{sr} and μ_b and are combined using Matthiessen's rule as follows:

$$\mu_T^{-1} = \mu_{AC}^{-1} + \mu_b^{-1} + \mu_{sr}^{-1} \quad \text{3-253}$$

The first component, μ_{AC}, is the surface mobility limited by scattering with acoustic phonons:

$$\mu_{AC,n} = \frac{\mathtt{BN.CVT}}{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{EN.CVT}}} + \left(\frac{\mathtt{CN.CVT}\left(\frac{N}{N_1}\right)^{\mathtt{TAUN.CVT}}}{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{DN.CVT}}} \right) \frac{T_1}{T_L} \quad 3\text{-}254$$

$$\mu_{AC,p} = \frac{\mathtt{BP.CVT}}{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{EP.CVT}}} + \left(\frac{\mathtt{CP.CVT}\left(\frac{N}{N_1}\right)^{\mathtt{TAUP.CVT}}}{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{DP.CVT}}} \right) \frac{T_1}{T_L} \quad 3\text{-}255$$

where T_L is the temperature, $E_\perp$ is the perpendicular electric field, and N is the total doping concentration. In E_1 is 1V/cm, N_1 is 1cm^{-3}, and T_1 is 1K. You can define the parameters BN.CVT, BP.CVT, CN.CVT, CP.CVT, DN.CVT, DP.CVT, TAUN.CVT, and TAUP.CVT in the **MOBILITY** statement (see Table 3-54 for their defaults).

The second component, μ_{sr}, is the surface roughness factor and is given by:

$$\frac{1}{\mu_{sr,n}} = \frac{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{KN.CVT}}}{\mathtt{DELN.CVT}} + \frac{\left(\frac{E_\perp}{E_1}\right)^3}{\mathtt{FELN.CVT}} \quad 3\text{-}256$$

for electrons and

$$\frac{1}{\mu_{sr,p}} = \frac{\left(\frac{E_\perp}{E_1}\right)^{\mathtt{KP.CVT}}}{\mathtt{DELP.CVT}} + \frac{\left(\frac{E_\perp}{E_1}\right)^3}{\mathtt{FELP.CVT}} \quad 3\text{-}257$$

for holes.

The default values of FELN.CVT and FELP.CVT are set so high that the second terms are negligible. The KN.CVT, KP.CVT, DELN.CVT, DELP.CVT, FELN.CVT and FELP.CVT parameters are user-definable (see Table 3-54 for their defaults).

The third mobility component, μ_b, is the mobility limited by scattering with optical intervalley phonons. This component is given by:

$$\mu_{b,n} = \mathtt{MU0N.CVT}\, exp\left(\frac{-\mathtt{PCN.CVT}}{N}\right) + \frac{\left[\mathtt{MUMAXN.CVT}\left(\frac{T_L}{300}\right)^{-\mathtt{GAMN.CVT}} - \mathtt{MU0N.CVT}\right]}{1 + \left(\frac{N}{\mathtt{CRN.CVT}}\right)^{\mathtt{ALPHN.CVT}}} \quad 3\text{-}258$$

$$- \frac{\mathtt{MU1N.CVT}}{1 + \left(\frac{\mathtt{CSN.CVT}}{N}\right)^{\mathtt{BETAN.CVT}}}$$

$$\mu_{b,p} = \mathtt{MU0P.CVT}\exp\left(\frac{-\mathtt{PCP.CVT}}{N}\right) + \frac{\left[\mathtt{MUMAXP.CVT}\left(\frac{T_L}{300}\right)^{-\mathtt{GAMP.CVT}} - \mathtt{MU0P.CVT}\right]}{1+\left(\frac{N}{\mathtt{CRP.CVT}}\right)^{\mathtt{ALPHP.CVT}}} \quad 3\text{-}259$$

$$- \frac{\mathtt{MU1P.CVT}}{1+\left(\frac{\mathtt{CSP.CVT}}{N}\right)^{\mathtt{BETAP.CVT}}}$$

Here, N is the total density of impurities and T_L is the temperature in degrees Kelvin.

Because μ_{AC} and μ_{sr} are related to interaction with an interface, you can specify a typical distance from the interface over which these components are significant [259]. If you specify the N.LCRIT or P.LCRIT parameters or both then factors\

$$FE = exp(-l/\mathtt{N.LCRIT}) \quad 3\text{-}260$$

and

$$FH = exp(-l/\mathtt{P.LCRIT}) \quad 3\text{-}261$$

are obtained, where l is the shortest distance to the nearest interface from the point at which the mobility will be calculated. These factors are then used to modify the Matthiesen's rule of Equation 3-253 to

$$\frac{1}{\mu_T} = \frac{FE}{\mu_{AC}} + \frac{FE}{\mu_{sr}} + \frac{1}{\mu_b} \quad 3\text{-}262$$

for electrons and

$$\frac{1}{\mu_T} = \frac{FH}{\mu_{AC}} + \frac{FH}{\mu_{sr}} + \frac{1}{\mu_b} \quad 3\text{-}263$$

for holes. Far from the interface, the mobility is the same as the bulk mobility.

Table 3-54 User-Specifiable Parameters for Equations to 3-259

Statement	Parameter	Default	Units
MOBILITY	ALPHN.CVT	0.680	
MOBILITY	ALPHP.CVT	0.71	
MOBILITY	BETAN.CVT	2.00	
MOBILITY	BETAP.CVT	2.00	
MOBILITY	BN.CVT	4.75×10^{7}	$cm^2/(V\cdot s)$
MOBILITY	BP.CVT	9.925×10^{6}	$cm^2/(V\cdot s)$
MOBILITY	CN.CVT	1.74×10^{5}	$cm^2/(V\cdot s)$

Table 3-54 User-Specifiable Parameters for Equations to 3-259			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	CP.CVT	8.842×10^{5}	$cm^2/(V\cdot s)$
MOBILITY	CRN.CVT	9.68×10^{16}	cm^{-3}
MOBILITY	CRP.CVT	2.23×10^{17}	cm^{-3}
MOBILITY	CSN.CVT	3.43×10^{20}	cm^{-3}
MOBILITY	CSP.CVT	6.10×10^{20}	cm^{-3}
MOBILITY	DELN.CVT	5.82×10^{14}	$cm^2/(V\cdot s)$
MOBILITY	DELP.CVT	2.0546×10^{14}	$cm^2/(V\cdot s)$
MOBILITY	DN.CVT	0.333	
MOBILITY	DP.CVT	0.333	
MOBILITY	EN.CVT	1.0	
MOBILITY	EP.CVT	1.0	
MOBILITY	FELN.CVT	1.0×10^{50}	$cm^2/(V\cdot s)$
MOBILITY	FELP.CVT	1.0×10^{50}	$cm^2/(V\cdot s)$
MOBILITY	KN.CVT	2.0	
MOBILITY	KP.CVT	2.0	
MOBILITY	GAMN.CVT	2.5	
MOBILITY	GAMP.CVT	2.2	
MOBILITY	MU0N.CVT	52.2	$cm^2/(V\cdot s)$
MOBILITY	MU0P.CVT	44.9	$cm^2/(V\cdot s)$
MOBILITY	MU1N.CVT	43.4	$cm^2/(V\cdot s)$
MOBILITY	MU1P.CVT	29.0	$cm2/(V\cdot s)$
MOBILITY	MUMAXN.CVT	1417.0	$cm^2/(V\cdot s)$
MOBILITY	MUMAXP.CVT	470.5	$cm^2/(V\cdot s)$
MOBILITY	N.LCRIT	1.0	cm
MOBILITY	P.LCRIT	1.0	cm
MOBILITY	PCN.CVT	0.0	cm^{-3}

Table 3-54 User-Specifiable Parameters for Equations to 3-259			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	PCP.CVT	9.23×10^{16}	cm^{-3}
MOBILITY	TAUN.CVT	0.125	
MOBILITY	TAUP.CVT	0.0317	

Note: The CVT model when activated will also, by default, apply the Parallel Electric Field Mobility Model which is described in the "Parallel Electric Field-Dependent Mobility" on page 205. In this model, the low-field mobility is supplied from the CVT model.

Darwish CVT Model

The CVT model already described has been successfully used in many device simulations. Lately, an improved version of this model has been proposed by Darwish et al. [71] where several modifications have been implemented. The first modification is that the bulk mobility is calculated using Klaassen's model (see "Klaassen's Unified Low-Field Mobility Model" on page 176) to take into account coulomb screening effects. The second modification is a new expression for surface roughness is used.

The new model for surface roughness replaces Equations 3-256 and 3-257 with:

$$\frac{1}{\mu_{sr,n}} = \frac{\left(\frac{E_\perp}{E_2}\right)^{\gamma_n}}{\mathrm{DELN.CVT}} + \frac{\left(\frac{E_\perp}{E_2}\right)^3}{\mathrm{FELN.CVT}} \quad 3\text{-}264$$

$$\frac{1}{\mu_{sr,p}} = \frac{\left(\frac{E_\perp}{E_2}\right)^{\gamma_p}}{\mathrm{DELP.CVT}} + \frac{\left(\frac{E_\perp}{E_2}\right)^3}{\mathrm{FELP.CVT}} \quad 3\text{-}265$$

where:

$$\gamma_n = \mathrm{AN.CVT} + \frac{\mathrm{ALN.CVT}\times(n+p)}{\left(\frac{N}{N_2}\right)^{\mathrm{ETAN.CVT}}} \quad 3\text{-}266$$

$$\gamma_p = \mathrm{AP.CVT} + \frac{\mathrm{ALP.CVT}\times(n+p)}{\left(\frac{N}{N_2}\right)^{\mathrm{ETAP.CVT}}} \quad 3\text{-}267$$

Here, N is the total doping density ($N_D + N_A$), N_2 is $1cm^{-3}$, and E_2 is 1 V/cm.

Table 3-55 shows the default parameters for the new equations.

Table 3-55 Default Parameters for the Surface Roughness Components of New CVT Model			
Parameter	**Statement**	**Default**	**Units**
AN.CVT	**MOBILITY**	2.58	
AP.CVT	**MOBILITY**	2.18	
ALN.CVT	**MOBILITY**	6.85×10^{-21}	cm^3
ALP.CVT	**MOBILITY**	7.82×10^{-21}	cm^3
ETAN.CVT	**MOBILITY**	0.0767	
ETAP.CVT	**MOBILITY**	0.123	

Next, in the Darwish model, the expressions for acoustic phonon scattering are modified as shown in Equations 3-268 and 3-269.

$$\mu_{AC,n} = \frac{BN.CVT}{\left(\frac{E_\perp}{E_2}\right)^{EN.CVT}} + \left(\frac{CN.CVT\left(\frac{N}{N_2}\right)^{TAUN.CVT}}{\left(\frac{E_\perp}{E_2}\right)^{DN.CVT}}\right)\frac{1}{T_n'} \quad 3\text{-}268$$

$$\mu_{AC,p} = \frac{BP.CVT}{\left(\frac{E_\perp}{E_2}\right)^{EP.CVT}} + \left(\frac{CP.CVT\left(\frac{N}{N_2}\right)^{TAUP.CVT}}{\left(\frac{E_\perp}{E_2}\right)^{DP.CVT}}\right)\frac{1}{T_p'} \quad 3\text{-}269$$

where

$$T_n' = (T_L/300)^{KAPPAN.CVT} \quad 3\text{-}270$$

$$T_p' = (T_L/300)^{KAPPAP.CVT} \quad 3\text{-}271$$

Table 3-56 shows the default values for KAPPAN.CVT and KAPPAP.CVT.

Table 3-56 Default Parameter Values for Equations 3-270 and 3-271			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	KAPPAN.CVT	1.7	
MOBILITY	KAPPAP.CVT	0.9	

Finally, the default values for many of the parameters are changed from those listed Table 3-54. Table 3-57 lists the defaults used for the Darwish model.

Table 3-57 Default Parameters for the Darwish model.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	BN.CVT	3.61×10^{7}	cm/s
MOBILITY	BP.CVT	1.51×10^{7}	cm/s
MOBILITY	CN.CVT	1.7×10^{4}	
MOBILITY	CP.CVT	4.18×10^{3}	
MOBILITY	TAUN.CVT	0.0233	
MOBILITY	TAUP.CVT	0.0119	
MOBILITY	DELN.CVT	3.58×10^{18}	$cm^2/(V\cdot s)$
MOBILITY	DELP.CVT	4.1×10^{15}	$cm^2/(V\cdot s)$

You can enable the Darwish model for electrons and holes by specifying NEWCVT.N or NEWCVT.P on the **MOBILITY** statement.

Remote Coulomb Scattering term for Inversion Layer Mobility

A high density of interface charge at the semiconductor-insulator interface in a lateral MOSFET can significantly reduce the channel mobility. This effect is caused by Coulomb scattering of the mobile carriers by the interface charge, and it can be a relatively long range effect. Atlas allows this Coulomb Scattering term to be included in the Lombardi, Darwish and Alternative CVT models. It has up to three components of the form.

$$\mu_{c,n} = \frac{COULOMB.NMOB\left(\frac{Tl}{300}\right)^{COULOMB.NKT}\left(1+\left[\frac{NCONC}{COULOMB.NCT\ (CHG)^{COULOMB.NP1}}\right]^{COULOMB.NNU}\right)}{CHG^{COULOMB.NP2}\exp\left(-\frac{D}{N.LCRIT}\right)\left(1-\exp\left(-\frac{FPERP}{COULOMB.E2N}^{COULOMB.NGAM}\right)\right)} \quad 3\text{-}272$$

for electrons, where CHG represents one of three scaled charge contributions to the mobility.

For electrons these are

1. CHG = $\frac{N_a}{COULOMB.NTD}$, where is N_a is channel acceptor density.

2. CHG = $\frac{N_t}{COULOMB.NTT}$, where is N_t is -ve interface density.

3. CHG = $\frac{P_t}{COULOMB.NTT}$, where is P_t is +ve interface charge.

And for holes

$$\mu_{c,p} = \frac{COULOMB.PMOB\left(\frac{Tl}{300}\right)^{COULOMB.PKT}\left(1+\left[\frac{PCONC}{COULOMB.NCT\ (CHG)^{COULOMB.PP1}}\right]^{COULOMB.PNU}\right)}{CHG^{COULOMB.PP2}\exp\left(-\frac{D}{P.LCRIT}\right)\left(1-\exp\left(-\frac{FPERP}{COULOMB.E0P}^{COULOMB.PGAM}\right)\right)} \quad 3\text{-}273$$

1. CHG $= \frac{N_a}{COULOMB.PTD}$, where is N_a is channel donor density.

2. CHG $= \frac{N_t}{COULOMB.PTT}$, where is N_t is -ve interface density.

3. CHG $= \frac{P_t}{COULOMB.PTT}$, where is P_t is +ve interface charge.

The Coulomb scattering contribution to the mobility at any point in the channel is calculated by first finding the shortest distance, D, to the gate insulator. The scaling factor governing the proximity dependence of the coulomb mobility contribution is N.LCRIT for electrons and P.LCRIT for holes. This means that the mobility evaluated at distances many multiples of N.LCRIT or P.LCRIT away from the interface is only weakly affected by interface scattering. The electron carrier density, NCONC, and the hole carrier density, PCONC, are obtained at the nearest point of the gate insulator interface from the point at which mobility is required. The interface charge densities are evaluated at the same point as NCONC and PCONC. Fixed charges which are specified by either the QF or CHARGE parameters on the INTERFACE statement are included in the interface charge density. So also are charges arising from interface traps defined by the **INTTRAP** and INTDEFECTS statements. The quantity FPERP is the field normal to the interface, and is evaluated at each point for which the mobility is calculated, and Tl is lattice temperature.

The model is enabled by using the flags COULOMB.N for electrons, and COULOMB.P for holes on the **MOBILITY** statement. These flags are not set by default. When the coulomb scattering model is enabled, each component is also enabled by default. Each component can be individually disabled by clearing the flags COULOMB.DOP, COULOMB.ACCT and COULOMB.DONT on the **MOBILITY** statement, for components 1,2 and 3 respectively. The components which are enabled, are combined with the bulk, surface roughness and surface phonon low field mobilities by using Matthiessen's model.

$$\frac{1}{\mu_{cvt,n}} = \frac{1}{\mu_{b,n}} + \frac{1}{\mu_{sr,n}} + \frac{1}{\mu_{ac,n}} + \frac{1}{\mu_{1,n}} + \frac{1}{\mu_{2,n}} + \frac{1}{\mu_{3,n}} \quad 3\text{-}274$$

for electrons, and

$$\frac{1}{\mu_{cvt,p}} = \frac{1}{\mu_{b,p}} + \frac{1}{\mu_{sr,p}} + \frac{1}{\mu_{ac,p}} + \frac{1}{\mu_{1,p}} + \frac{1}{\mu_{2,p}} + \frac{1}{\mu_{3,p}} \quad 3\text{-}275$$

for holes. The parameters for the Coulomb scattering model, along with their default values are given in Table 3-58.

Table 3-58 Default Values of Coulomb Scattering Mobility Model Parameters			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	COULOBM.N	False	
MOBILITY	COULOBM.P	False	
MOBILITY	COULOBM.DOP	False	
MOBILITY	COULOBM.ACCT	False	
MOBILITY	COULOBM.DONT	False	
MOBILITY	COULOBM.NMOB	40.0	cm^2/Vs
MOBILITY	COULOBM.PMOB	40.0	cm^2/Vs
MOBILITY	COULOBM.NCT	10^{18}	cm^{-3}
MOBILITY	COULOBM.PCT	10^{18}	cm^{-3}
MOBILITY	COULOBM.NTT	10^{11}	cm^2
MOBILITY	COULOBM.PTT	10^{11}	cm^2
MOBILITY	COULOBM.NTD	10^{18}	cm^{-3}
MOBILITY	COULOBM.PTD	10^{18}	cm^{-3}
MOBILITY	COULOBM.NP1	1.0	
MOBILITY	COULOBM.PP1	1.0	
MOBILITY	COULOBM.NP2	0.5	
MOBILITY	COULOBM.PP2	0.5	
MOBILITY	COULOBM.NNU	1.5	
MOBILITY	COULOBM.PNU	1.5	
MOBILITY	COULOBM.E0N	2.0×10^5	V/cm
MOBILITY	COULOBM.E0P	2.0×10^5	V/cm
MOBILITY	COULOBM.NGAM	2.0	
MOBILITY	COULOBM.PGAM	2.0	
MOBILITY	COULOBM.NKT	1.0	
MOBILITY	COULOBM.PKT	1.0	

Alternative CVT mobility model

An alternative inversion mobility model is available in Atlas. It has been used successfully in the simulation of MOSFET's with 4H-SiC channels, [334]. It comprises four terms, bulk mobility, surface roughness mobility , surface phonon mobility and the coulomb scattering mobility (which is described in the previous section). These terms are combined using Matthiessen's rule. The first term, the bulk mobility is given by

$$\mu_{b,n} = ALT.N.MUMIN\left(\frac{Tl}{300}\right)^{ALT.N.ALPHA} + \frac{ALT.N.MUBP1\left(\frac{Tl}{300}\right)^{ALT.N.EXP1} + ALT.N.MUBP2\left(\frac{Tl}{300}\right)^{ALT.N.EXP2} - ALT.N.MUMIN\left(\frac{Tl}{300}\right)^{ALT.N.ALPHA}}{1 + \left(\frac{Tl}{300}\right)^{ALT.N.EXP3}\left(\frac{Ntot}{ALT.N.REF}\right)^{ALT.N.EXP4}} \quad 3\text{-}276$$

and for holes

$$\mu_{b,p} = ALT.P.MUMIN\left(\frac{Tl}{300}\right)^{ALT.P.ALPHA} + \frac{ALT.P.MUBP1\left(\frac{Tl}{300}\right)^{ALT.P.EXP1} + ALT.P.MUBP2\left(\frac{Tl}{300}\right)^{ALT.P.EXP2} - ALT.P.MUMIN\left(\frac{Tl}{300}\right)^{ALT.P.ALPHA}}{1 + \left(\frac{Tl}{300}\right)^{ALT.P.EXP3}\left(\frac{Ntot}{ALT.P.REF}\right)^{ALT.P.EXP4}} \quad 3\text{-}277$$

This is very similar to the Caughey-Thomas expression for bulk mobility, although it has an extra temperature dependent term. The default values of the parameters are given in Table 3-59.

Table 3-59 Default Values of Bulk Mobility Parameters

Statement	Parameter	Default	Units
MOBILITY	ALT.N.MUMIN	40.0	cm/s
MOBILITY	ALT.P.MUMIN	40.0	cm/s
MOBILITY	ALT.N.MUBP1	500.0	
MOBILITY	ALT.P.MUBP1	500.0	
MOBILITY	ALT.N.MUBP2	450.0	
MOBILITY	ALT.P.MUBP2	450.0	
MOBILITY	ALT.N.ALPHA	-0.5	$cm^2/(V \cdot s)$
MOBILITY	ALT.P.ALPHA	-0.5	$cm^2/(V \cdot s)$
MOBILITY	ALT.N.EXP1	-11.6	
MOBILITY	ALT.P.EXP1	-11.6	

Table 3-59 Default Values of Bulk Mobility Parameters			
MOBILITY	ALT.N.EXP2	-2.74	
MOBILITY	ALT.P.EXP2	-2.74	
MOBILITY	ALT.N.EXP3	-12.5	
MOBILITY	ALT.P.EXP3	-12.5	
MOBILITY	ALT.N.EXP4	0.76	
MOBILITY	ALT.P.EXP4	0.76	
MOBILITY	ALT.N.NREF	1.7×10^{4}	
MOBILITY	ALT.N.NREF	4.18×10^{3}	

The next term is the surface roughness term.

$$\mu_{sr.n}^{-1} = \left[\frac{(FPERP)^2}{F0} + \frac{(FPERP)^3}{F0}\right] \exp(-D/N.LCRIT) \quad \text{3-278}$$

$$\frac{ALT.N.DELTA}{ALT.ETA}$$

for electrons, and

$$\mu_{sr.p}^{-1} = \left[\frac{(FPERP)^2}{F0} + \frac{(FPERP)^3}{F0}\right] \exp(-D/P.LCRIT) \quad \text{3-279}$$

$$\frac{ALT.P.DELTA}{ALT.P.ETA}$$

for holes. F0 is a scaling factor, having an immutable value of 1V/cm, and FPERP is the perpendicular component of the electric field. These expressions are very similar to equations 3-256 and 3-257, except for the fixed exponent on the first term. The default values of the parameters are given in Table 3-60.

Table 3-60 Default Values of Surface Roughness Mobility Model Parameters			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALT.N.DELTA	1.6×10^{14}	cm^2/Vs
MOBILITY	ALT.P.DELTA	1.6×10^{14}	cm^2/Vs
MOBILITY	ALT.N.ETA	1.6×10^{30}	cm^2/Vs
MOBILITY	ALT.P.ETA	1.6×10^{30}	cm^2/Vs
MOBILITY	ALT.SR.N	True	
MOBILITY	ALT.SR.P	True	

The surface roughness term is included by default, but can be excluded by clearing the flags ALT.SR.N and ALT.SR.P for electrons and holes respectively.

The third term is the surface phonon (also known as acoustic phonon) scattering term. The surface phonon component is given by

$$\mu_{sp.n} = \left[\frac{ALT.N.SPB}{FPERP} + \frac{ALT.N.SPC\left(\frac{Ntot}{ALT.N.SPN0}\right)^{ALT.N.BETA}}{FPERP^{(1/3)}\left(\frac{Tl}{300}\right)^{ALT.N.KTEMP}}\right]\exp(-D/N.LCRIT) \quad \text{3-280}$$

$$\mu_{sp.p} = \left[\frac{ALT.P.SPB}{FPERP} + \frac{ALT.P.SPC\left(\frac{Ntot}{ALT.P.SPN0}\right)^{ALT.P.BETA}}{FPERP^{(1/3)}\left(\frac{Tl}{300}\right)^{ALT.P.KTEMP}}\right]\exp(-D/P.LCRIT) \quad \text{3-281}$$

for holes. The default values are given inTable 3-61

Table 3-61 Default Values of Surface Phonon Mobility Model Parameters

Statement	Parameter	Default	Units
MOBILITY	ALT.N.SPB	1.0×10^{6}	cm/s
MOBILITY	ALT.P.SPB	1.0×10^{6}	cm/s
MOBILITY	ALT.N.SPC	1.077×10^{4}	$cm^{5/3}V^{-2/3}s^{-1}$
MOBILITY	ALT.P.SPC	1.077×10^{4}	$cm^{5/3}V^{-2/3}s^{-1}$
MOBILITY	ALT.N.SPN0	1.0	cm^{-3}
MOBILITY	ALT.P.SPN0	1.0	cm^{-3}
MOBILITY	ALT.N.BETA	0.0284	
MOBILITY	ALT.P.BETA	0.0284	
MOBILITY	ALT.N.KTEMP	1.0	
MOBILITY	ALT.P.KTEMP	1.0	
MOBILITY	ALT.SR.N	True	
MOBILITY	ALT.SR.P	True	

The surface phonon term is included by default, but can be excluded by clearing the flags ALT.SP.N and ALT.SP.P for electrons and holes respectively.

The final component is the Coulomb scattering term, as described in the previous subsection. This is set to true by default for the Alternative CVT model (for Lombardi and Darwish models its default is false). You can exclude the Coulomb scattering terms by clearing the flags COULOMB.N and COULOMB.P on the **MOBILITY** statement. The active components are combined using Matthiessen's rule, giving

$$\frac{1}{\mu_{alt,n}} = \frac{1}{\mu_{b,n}} + \frac{1}{\mu_{sr,n}} + \frac{1}{\mu_{sp,n}} + \frac{1}{\mu_{coul,n}} \quad (3\text{-}282)$$

for electrons, and

$$\frac{1}{\mu_{alt,p}} = \frac{1}{\mu_{b,p}} + \frac{1}{\mu_{sr,p}} + \frac{1}{\mu_{sp,p}} + \frac{1}{\mu_{coul,p}} \quad (3\text{-}283)$$

for holes. To use this model for electron mobility you set ALTCVT.N on the **MOBILITY** statement, and for hole mobility you set ALTCVT.P on the **MOBILITY** statement. In this inversion layer mobility model, unlike in the Lombardi model, the parallel field dependent mobility is not automatically enabled. Specify FLDMOB on the **MODELS** statement to set this.

Yamaguchi Model

The Yamaguchi Model [362] is selected by setting YAMAGUCHI in the **MODELS** statement. This model overrides any mobility model specifications other than the CVT model. The model consists of calculating the low-field, doping dependent mobility. Surface degradation is then accounted for based upon the transverse electric field before including the parallel electric field dependence.

The low-field part of the Yamaguchi Model is given as follows:

$$\mu_{n0} = \mathrm{MULN.YAMA}\left[1 + \frac{N}{\frac{N}{\mathrm{SN.YAMA}} + \mathrm{NREFN.YAMA}}\right]^{-\frac{1}{2}} \quad (3\text{-}284)$$

$$\mu_{p0} = \mathrm{MULP.YAMA}\left[1 + \frac{N}{\frac{N}{\mathrm{SP.YAMA}} + \mathrm{NREFP.YAMA}}\right]^{-\frac{1}{2}} \quad (3\text{-}285)$$

where N_i is the total impurity concentration. The equation parameters: MULN.YAMA, MULP.YAMA, SN.YAMA, SP.YAMA, NREFP.YAMA, and NREFP.YAMA are user-definable in the **MOBILITY** statement (see Table 3-62 for their defaults).

The transverse electric field dependence is accounted for as follows:

$$\mu_{s,\,n} = \mu_{n0}(1 + \mathrm{ASN.YAMA}\ E_{\perp})^{-\frac{1}{2}} \quad (3\text{-}286)$$

$$\mu_{s,\,p} = \mu_{p0}(1 + \mathrm{ASP.YAMA}\ E_{\perp})^{-\frac{1}{2}} \quad (3\text{-}287)$$

where $E_{\perp}$ is the perpendicular electric field and the equation parameters, ASN.YAMA and ASP.YAMA, are user-definable in the **MOBILITY** statement (see Table 3-62 for their defaults).

The final calculation of mobility takes into account the parallel electric field dependence which takes the form:

$$\mu_n = \mu_{s,n}\left[1 + \left(\frac{\mu_{s,n}E}{\texttt{ULN.YAMA}}\right)^2\left(\texttt{GN.YAMA} + \frac{\mu_{s,n}E}{\texttt{ULN.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,n}E}{\texttt{VSN.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad \text{3-288}$$

$$\mu_p = \mu_{s,p}\left[1 + \left(\frac{\mu_{s,p}E}{\texttt{ULP.YAMA}}\right)^2\left(\texttt{GP.YAMA} + \frac{\mu_{s,p}E}{\texttt{ULP.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,p}E}{\texttt{VSP.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad \text{3-289}$$

where E is the parallel electric field and the equation parameters: ULN.YAMA, ULP.YAMA, VSN.YAMA, VSP.YAMA, GN.YAMA, and GP.YAMA are user-definable in the **MOBILITY** statement (see Table 3-62 for their defaults).

Table 3-62 User-Specifiable Parameters for Equations 3-384 to 3-289

Statement	Parameter	Default	Units
MOBILITY	SN.YAMA	350.0	
MOBILITY	SP.YAMA	81.0	
MOBILITY	NREFP.YAMA	3.0×10^{16}	cm^{-3}
MOBILITY	NREFP.YAMA	4.0×10^{16}	cm^{-3}
MOBILITY	MULN.YAMA	1400.0	$cm^2/(V\cdot s)$
MOBILITY	MULP.YAMA	480.0	$cm^2/(V\cdot s)$
MOBILITY	ASN.YAMA	1.54×10^{-5}	cm/V
MOBILITY	ASP.YAMA	5.35×10^{-5}	cm/V
MOBILITY	VSN.YAMA	1.036×10^{7}	cm/s
MOBILITY	VSP.YAMA	1.2×10^{7}	cm/s
MOBILITY	ULN.YAMA	4.9×10^{6}	cm/s
MOBILITY	ULP.YAMA	2.928×10^{6}	cm/s
MOBILITY	GN.YAMA	8.8	
MOBILITY	GP.YAMA	1.6	

The Tasch Model

S-Pisces includes an improved local field-dependent mobility model. This model, which was originally derived and published by Tasch et. al [291,292] has been designed explicitly for MOSFETs. It defines the mobility as a function of the perpendicular and parallel electric fields, the interface charge, the lattice temperature and the doping concentration. To activate this model, is activated use the TASCH parameter on the **MODELS** statement. This mobility model is given by the following expressions:

$$\mu_n = \Gamma_n + (E_{perp} - E_0)\frac{d\Gamma_n}{dE_{perp}} \quad 3\text{-}290$$

$$\mu_p = \Gamma_p + (E_{perp} - E_0)\frac{d\Gamma_p}{dE_{perp}} \quad 3\text{-}291$$

where E_{perp} is the transverse electric field and E_0 is the transverse electric field at the edge of the inversion layer. The functions $\Gamma_{n,p}$ are defined as:

$$\Gamma_n = \frac{\mu_{eff,n}}{\left(1 + \left(\frac{\mu_{eff,n} E_{\parallel}}{\mathrm{VSATN}}\right)^{\mathrm{BETAN}}\right)^{1/\mathrm{BETAN}}} \quad 3\text{-}292$$

$$\Gamma_p = \frac{\mu_{eff,p}}{\left(1 + \left(\frac{\mu_{eff,p} E_{\parallel}}{\mathrm{VSATP}}\right)^{\mathrm{BETAP}}\right)^{1/\mathrm{BETAP}}} \quad 3\text{-}293$$

The carrier mobilities $\mu_{eff,n}$ and $\mu_{eff,p}$ are defined by three components μ_{ph}, μ_{sr},and μ_c that are combined by Mathiessen's rule according to:

$$\mu_{eff} = \left[\frac{1}{\mu_{ph}} + \frac{1}{\mu_{sr}} + \frac{1}{\mu_c}\right]^{-1} \quad 3\text{-}294$$

The term μ_{ph} takes account of the degradation in mobility due to acoustic phonon scattering through the expressions:

$$\mu_{ph,n}^{-1} = \left(\mathrm{MUBN.TAS}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUBN.TAS}}\right)^{-1} + \left(Z_n/\mathrm{DN.TAS}\ Y_n\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}295$$

$$\mu_{ph,p}^{-1} = \left(\mathrm{MUBP.TAS}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUBP.TAS}}\right)^{-1} + \left(Z_p/\mathrm{DP.TAS}\ Y_p\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}296$$

The function, $Z_{n,p}$, is defined as:

$$Z_n = \mathrm{Z11N.TAS}\left(\frac{T_L}{300}\right) E_{eff,n}^{-1} + \mathrm{Z22N.TAS}\ E_{eff,n}^{-\frac{1}{3}} \quad 3\text{-}297$$

$$Z_p = \mathrm{Z11P.TAS}\left(\frac{T_L}{300}\right) E_{eff,p}^{-1} + \mathrm{Z22P.TAS}\ E_{eff,p}^{-\frac{1}{3}} \quad 3\text{-}298$$

where:

$$E_{eff,\,n} = \frac{(E_{perp} + (\mathrm{RN.TAS} - 1)\cdot E_0)}{\mathrm{RP.TAS}} \quad 3\text{-}299$$

$$E_{eff,\,p} = \frac{(E_{perp} + (\mathrm{RN.TAS} - 1)\cdot E_0)}{\mathrm{RP.TAS}} \quad 3\text{-}300$$

Also, the function $Y_{n,p}$, is defined as:

$$Y_n = \mathrm{P1N.TAS}\left(\frac{T_L}{300}\right)^{\mathrm{B1N.TAS}} + \mathrm{P2N.TAS}\; n \;+ \mathrm{B2N.TAS}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}301$$

$$Y_p = \mathrm{P1P.TAS}\left(\frac{T_L}{300}\right)^{\mathrm{B1P.TAS}} + \mathrm{P2P.TAS}\; p \;+ \mathrm{B2P.TAS}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}302$$

Mobility degradation due to surface roughness is accounted for by the term μ_{sr} that is calculated according to:

$$\mu_{sr,\,n} = \left(\frac{\mathrm{ESRN.TAS}}{E_{eff,\,n}}\right)^{\mathrm{BETAN.TAS}} \quad 3\text{-}303$$

$$\mu_{sr,\,p} = \left(\frac{\mathrm{ESRP.TAS}}{E_{eff,\,p}}\right)^{\mathrm{BETAP.TAS}} \quad 3\text{-}304$$

The final term, μ_C, models Coulombic scattering with the expressions:

$$\mu_{C,\,n} = \frac{\mathrm{N2N.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_A\; ln(1+\gamma_{BH,\,n}) - \frac{\gamma_{BH,\,n}}{(1+\gamma_{BH,\,n})}} \quad 3\text{-}305$$

$$\mu_{C,\,p} = \frac{\mathrm{N2P.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_D\; ln(1+\gamma_{BH,\,p}) - \frac{\gamma_{BH,\,p}}{(1+\gamma_{BH,\,p})}} \quad 3\text{-}306$$

Here:

$$\gamma_{BH,\,n} = \frac{\mathrm{N1N.TAS}}{n}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAN.TAS}} \quad 3\text{-}307$$

$$\gamma_{BH,\,p} = \frac{\mathrm{N1P.TAS}}{p}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAP.TAS}} \quad 3\text{-}308$$

T_L is the lattice temperature in degrees Kelvin, N_f is the fixed interface charge at the gate dielectric-silicon interface (cm^{-2}), N_A is the channel acceptor doping concentration in cm^{-3}, N_D is the channel donor doping concentration in cm^{-3}, n and p are the electron and hole concentrations per unit volume in the inversion layer (cm^{-3}). The default parameters within each equation is defined in the **MOBILITY** statement. The default parameters are shown in Table 3-62.

Table 3-63 Parameters for Equations 3-290 through 3-308

Statement	Parameter	Default	Units
MOBILITY	RN.TAS	2	
MOBILITY	RP.TAS	3	
MOBILITY	BETAN	2	
MOBILITY	BETAP	1	
MOBILITY	MUBN.TAS	1150	
MOBILITY	MUBP.TAS	270	
MOBILITY	TMUBN.TAS	2.5	
MOBILITY	TMUBP.TAS	1.4	
MOBILITY	DN.TAS	3.2×10^{-9}	
MOBILITY	DP.TAS	2.35×10^{-9}	
MOBILITY	P1N.TAS	0.09	
MOBILITY	P1P.TAS	0.334	
MOBILITY	B1N.TAS	1.75	
MOBILITY	B1P.TAS	1.5	
MOBILITY	P2N.TAS	4.53×10^{-8}	
MOBILITY	PEP.TAS	3.14×10^{-7}	
MOBILITY	B2N.TAS	-0.25	
MOBILITY	B2P.TAS	-0.3	
MOBILITY	Z11N.TAS	0.0388	
MOBILITY	Z11P.TAS	0.039	
MOBILITY	Z22N.TAS	1.73×10^{-5}	
MOBILITY	Z22P.TAS	1.51×10^{-5}	

Table 3-63 Parameters for Equations 3-290 through 3-308			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ESRN.TAS	2.449×10^{7}	
MOBILITY	ESRP.TAS	1.0×10^{8}	
MOBILITY	BETAN.TAS	2	
MOBILITY	BETAP.TAS	1	
MOBILITY	N2N.TAS	1.1×10^{21}	
MOBILITY	N2P.TAS	1.4×10^{18}	
MOBILITY	N1N.TAS	2.0×10^{19}	
MOBILITY	N1P.TAS	8.4×10^{16}	
MOBILITY	ALPHAN.TAS	2	
MOBILITY	ALPHAP.TAS	3.4	

Shirahata's Mobility Model

The Shirahata Mobility Model [293] is a general purpose MOS mobility model that accounts for screening effects in the inversion layer and uses an improved perpendicular field dependence for thin gate oxides. In the original paper, the authors present the model as a combination of portions of Klaassen's model for low-field mobility contributions and an empirically fit expression for the perpendicular field dependent mobility in the inversion layer. In this implementation, if the Klaassen Low-Field Model is used with the Shirahata model, the lattice scattering term in the Klaassen model is omitted and Matthiesen's rule is used to combine the Klaassen and Shirahata mobilities. But for other low-field models, at any given location, the lesser of the low-field and the Shirahata mobilities are used.

To enable the Shirahata Mobility Model, use the SHI parameter of the **MODELS** statement. You can also enable it individually for electrons and holes using the SHI.N and SHI.P parameters of the **MOBILITY** statement.

The Shirahata models for electrons and holes are given by:

$$\mu_n = \frac{\mathtt{MU0N.SHI}\left(\frac{T_L}{300}\right)^{-\mathtt{THETAN.SHI}}}{\left[1+\frac{|E_\perp|}{\mathtt{E1N.SHI}}\right]^{\mathtt{P1N.SHI}}+\left[\frac{|E_\perp|}{\mathtt{E2N.SHI}}\right]^{\mathtt{P2N.SHI}}} \quad 3\text{-}309$$

$$\mu_p = \frac{\mathtt{MU0P.SHI}\left(\frac{T_L}{300}\right)^{-\mathtt{THETAP.SHI}}}{\left[1+\frac{|E_\perp|}{\mathtt{E1P.SHI}}\right]^{\mathtt{P1P.SHI}}+\left[\frac{|E_\perp|}{\mathtt{E2P.SHI}}\right]^{\mathtt{P2P.SHI}}} \quad 3\text{-}310$$

where $E_{\perp}$ is the perpendicular electric and the equation parameters: MU0N.SHI, MU0P.SHI, E1N.SHI, E1P.SHI, E2N.SHI, E2P.SHI, P1N.SHI, P1P.SHI, P2N.SHI, P2P.SHI, THETAN.SHI, and THETAP.SHI are user-definable in the **MOBILITY** statement (see Table 3-64 for their defaults).

Table 3-64 User-Specifiable Parameters for Equations 3-309 and 3-310

Statement	Parameter	Default	Units
MOBILITY	MU0N.SHI	1430.0	$cm^2/(V \cdot s)$
MOBILITY	MU0P.SHI	500.0	$cm^2/(V \cdot s)$
MOBILITY	E1N.SHI	6.3×10^3	V/cm
MOBILITY	E1P.SHI	8.0×10^3	V/cm
MOBILITY	E2N.SHI	0.77×10^6	V/cm
MOBILITY	E2P.SHI	3.9×10^5	V/cm
MOBILITY	P1N.SHI	0.28	
MOBILITY	P1P.SHI	0.3	
MOBILITY	P2N.SHI	2.9	
MOBILITY	P2P.SHI	1.0	
MOBILITY	THETAN.SHI	2.285	
MOBILITY	THETAP.SHI	2.247	

Note: If the maximum low-field mobility has been user-defined, then it's important to also define this value inside the Shirahata model with the MU0N.SHI and MU0P.SHI parameters in the **MOBILITY** statement.

Remote Coulomb Scattering Mobility Model

One problem with simulation of FET devices constructed with high-k gate dielectrics is the observed mobility reduction in such devices. The principal contributions to this reduction in mobility are believed to be due to remote Coulomb scattering [312] and remote phonon scattering (discussed in the next section) [381].

The model is enabled by specifying RCS.N for elections and RCS.P for holes.

The analytic forms of the remote coulomb scattering for electrons and holes are given in Equations 3-311 and 3-312.

$$\mu_{rcsn} = \mathrm{MUN\_RCS}\left(\frac{N}{3\times 10^{16}}\right)^{\mathrm{ALPHAN\_RCS}}\left(\frac{T}{300}\right)^{\mathrm{BETAN\_RCS}}\left(\frac{N_s}{10^{11}}\right)^{\left[\mathrm{GAMMAN\_RCS}+\mathrm{DELTAN\_RCS}\,ln\left(\frac{N}{3\times 10^{16}}\right)\right]} \quad 3\text{-}311$$

$$\mu_{rcsp} = \mathrm{MUP.RCS}\left(\frac{N}{3\times10^{16}}\right)^{\mathrm{ALPHAP.RCS}}\left(\frac{T}{300}\right)^{\mathrm{BETAP.RCS}}\left(\frac{N_s}{10^{11}}\right)^{\left[\mathrm{GAMMAP.RCS}+\mathrm{DELTAP.RCS}\,ln\left(\frac{N}{3\times10^{16}}\right)\right]} \quad 3\text{-}312$$

where N is the doping concentration, T is the temperature in K, and N_s is the inversion layer carrier density. The user-definable parameters are given in Table 3-65.

Table 3-65 User Definable Parameters of the Remote Coulomb Scattering Mobility Model

Parameter	Statement	Default	Units
MUN.RCS	**MOBILITY**	240.0	$cm^2/(V*s)$
MUP.RCS	**MOBILITY**	240.0	$cm^2/(V*s)$
ALPHAN.RCS	**MOBILITY**	-0.3	
ALPHAP.RCS	**MOBILITY**	-0.3	
BETAN.RCS	**MOBILITY**	2.1	
BETAP.RCS	**MOBILITY**	2.1	
GAMMAN.RCS	**MOBILITY**	0.4	
GAMMAP.RCS	**MOBILITY**	0.4	
DELTAN.RCS	**MOBILITY**	0.035	
DELTAP.RCS	**MOBILITY**	0.035	

Remote Phonon Scattering Mobility Model

Along with the remote Coulomb scattering mobility model, the remote phonon scattering mobility model is suggested to help describe the mobility reduction observed in high-k gate dielectric FET devices [381].

This model is enabled by specifying RPS.N for electrons and RPS.P for holes on the **MOBILITY** statement. The analytic forms of the remote phonon scattering model are given in Equations 3-313 and 3-314.

$$\mu_{rpsn} = \mathrm{MUN.RPS}\left(\frac{E_\perp}{10^6 V/cm}\right)^{\mathrm{ALPHAN.RPS}}\left(\frac{T}{300K}\right)^{\mathrm{BETAN.RPS}} \quad 3\text{-}313$$

$$\mu_{rpsp} = \mathrm{MUP.RPS}\left(\frac{E_\perp}{10^6 V/cm}\right)^{\mathrm{ALPHAP.RPS}}\left(\frac{T}{300K}\right)^{\mathrm{BETAP.RPS}} \quad 3\text{-}314$$

where N is the doping concentration, $E_\perp$ is the perpendicular field, and T is the temperature in K. The user-definable parameters are given in Table 3-64.

Table 3-66 User Definable Parameters of the Remote Coulomb Scattering Mobility Model

Parameter	Statement	Default (HfO2)	Alternate (SiO2)	Units
MUN.RPS	**MOBILITY**	216.9	323.8	$cm^2/(V*s)$
MUP.RPS	**MOBILITY**	216.9	323.8	$cm^2/(V*s)$
ALPHAN.RPS	**MOBILITY**	-0.415	-0.505	
ALPHAP.RPS	**MOBILITY**	-0.415	-0.505	
BETAN.RPS	**MOBILITY**	-0.898	-1.98	
BETAP.RPS	**MOBILITY**	-0.898	-1.98	

This table includes some alternate values for SiO2 that are suggested by the reference. In general, the model parameters are material dependent and may require calibration.

Perpendicular Electric Field-Dependent Mobility

Perpendicular Field Mobility Model

Mobility degradation effects due to perpendicular fields can be accounted for using a simple perpendicular field dependent mobility model. To enable this model described by Equations 3-315 and 3-316, specify PRPMOB on the **MODELS** statement.

$$\mu_n = \mathrm{GSURFN} \cdot \frac{\mu_{0n}}{\sqrt{1 + \dfrac{E_\perp}{\mathrm{ECN.MU}}}} \qquad 3\text{-}315$$

$$\mu_p = \mathrm{GSURFP} \cdot \frac{\mu_{0n}}{\sqrt{1 + \dfrac{E_\perp}{\mathrm{ECN.MP}}}} \qquad 3\text{-}316$$

The default values for the user specifiable parameters of Equations 3-315 and 3-316 are described in Table 3-67.

Table 3-67 User Specifiable Parameters of the Perpendicular Field Mobility Model

Statement	Parameter	Default	Units
MOBILITY	GSURFN	1.0	

Table 3-67 User Specifiable Parameters of the Perpendicular Field Mobility Model			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	GSURFP	1.0	
MOBILITY	ECN.MU	6.48×10^4	V/cm
MOBILITY	ECP.MU	1.87×10^4	V/cm

The Watt Model

A surface mobility model derived by J.T.Watt [342] is available in Atlas. This mobility model is activated when the parameter SURFMOB is specified on the **MODELS** statement. The default model parameters are tuned to describe the measured mobilities in silicon at 300K. You can modify model parameters by using the **MOBILITY** statement.

The Watt model takes into consideration the following primary scattering mechanisms in the inversion layer:

1. Phonon scattering which results primarily from the interaction between two-dimensional inversion layer carriers and bulk phonons.
2. Surface roughness scattering caused by the interaction between inversion layer carriers and deviations from ideal planarity at the interface.
3. Charged impurity scattering caused by the interaction between inversion layer carriers and ions located in the oxide, at the interface, or in the bulk

The phonon and surface roughness components are functions of effective electric field. The charged impurity component is a function of the channel doping density.

The effective mobilities for electrons and holes are given by Equations 3-317 and 3-318.

$$\frac{1}{\mu_{eff,\,n}} = \frac{1}{\mathtt{MREF1N.WATT}}\left(\frac{10^6}{E_{eff,\,n}}\right)^{\mathtt{AL1N.WATT}} + \frac{1}{\mathtt{MREF2N.WATT}}\left(\frac{10^6}{E_{eff,\,n}}\right)^{\mathtt{AL2N.WATT}} + \frac{1}{\mathtt{MREF3N.WATT}}\left(\frac{10^{18}}{N_B}\right)^{-1}\left(\frac{10^{12}}{N_i}\right)^{\mathtt{AL3N.WATT}} \quad \text{3-317}$$

$$\frac{1}{\mu_{eff,\,p}} = \frac{1}{\mathtt{MREF1P.WATT}}\left(\frac{10^6}{E_{eff,\,p}}\right)^{\mathtt{AL1P.WATT}} + \frac{1}{\mathtt{MREF2P.WATT}}\left(\frac{10^6}{E_{eff,\,p}}\right)^{\mathtt{AL2P.WATT}} + \frac{1}{\mathtt{MREF3P.WATT}}\left(\frac{10^{18}}{N_B}\right)^{-1}\left(\frac{10^{12}}{N_i}\right)^{\mathtt{AL3N.WATT}} \quad \text{3-318}$$

Here, N_B is the surface trapped charge density, N_i is the inversion layer charge density and E_{eff} is the effective electric field given by:

$$E_{eff,\,n} = E_{\perp} + \texttt{ETAN.WATT}(E_0 - E_{\perp}) \quad \text{3-319}$$

$$E_{eff,\,p} = E_{\perp} + \texttt{ETAP.WATT}(E_0 - E_{\perp}) \quad \text{3-320}$$

Here, $E_{\perp}$ is the electric field perpendicular to the current flow and E_0 is the perpendicular electric field at the insulator-semiconductor interface. The equation parameters and their defaults are listed in Table 3-68.

The terms on the right side of Equations 3-317 and 3-318, describe (in order) the three scattering mechanisms previously discussed. Each component contains two constants: a pre-exponential factor and the exponent of the principal independent parameter. The charge impurity scattering component is assumed to be inversely proportional to doping density.

The expression for effective mobility contains a number of normalizing constants. These normalizing constants are included to allow easy comparison of constants. The first two terms in this mobility model are dependent on E_{eff} and represent the universal mobility-field relationship. The third term accounts for deviation from the universal relationship resulting from charged impurity scattering.

Table 3-68 User-Specifiable Parameters for Equations 3-317 - 3-320

Statement	Parameter	Default	Units
MOBILITY	ETAN.WATT	0.50	
MOBILITY	ETAP.WATT	0.33	
MOBILITY	MREF1N.WATT	481.0	cm2/(V·s)
MOBILITY	MREF1P.WATT	92.8	cm2/(V·s)
MOBILITY	MREF2N.WATT	591.0	cm2/(V·s)
MOBILITY	MREF2P.WATT	124.0	cm2/(V·s)
MOBILITY	MREF3N.WATT	1270.0	cm2/(V·s)
MOBILITY	MREF3P.WATT	534.0	cm2/(V·s)
MOBILITY	AL1N.WATT	-0.16	
MOBILITY	AL1P.WATT	-0.296	
MOBILITY	AL2N.WATT	-2.17	
MOBILITY	AL2P.WATT	-1.62	
MOBILITY	AL3N.WATT	1.07	
MOBILITY	AL3P.WATT	1.02	

Modifications to the Watt's Model

By default the Watt mobility model is a surface model that applies, only to those grid points on the silicon/ oxide interface. A modification has been added that now applies the Watt

model to points below the interface. This extension to the Watt model is enabled using the MOD.WATT.N and MOD.WATT.P parameters of the **MOBILITY** statement.

The distance over which the model is applied can be controlled by using the YMAXN.WATT and the YMAXP.WATT parameters of the **MOBILITY** statement. These parameters specify the maximum value of the Y coordinate over which the model is applies below the interface for electrons and holes respectively.

The XMINN.WATT, XMINP.WATT, XMAXN.WATT and XMAXP.WATT of the **MOBILITY** statement can be used to limit the range of the model in the X direction, to prevent the model from applying to the source and drain regions. The MIN.SURF parameter of the **MODELS** statement can also be used for this purpose. When enabled, the MIN.SURF parameter will ensure that the Watt model will only apply to minority regions.

The logical parameters EXP.WATT.N and EXP.WATT.P of the **MOBILITY** statement can also be used to enable an additional modification to the Watt model. When these parameters are enabled the effective normal electric field becomes a function of the depth beneath the silicon/oxide interface according to:

$$E_{\perp,n} = E_y exp\frac{-(y - y_{int})}{\text{YCHARN.WATT}} \quad \text{3-321}$$

$$E_{\perp,p} = E_y exp\frac{-(y - y_{int})}{\text{YCHARP.WATT}} \quad \text{3-322}$$

where $E_\perp$ is the perpendicular electric field, E_y is the perpendicular electric field at the interface, y is the local Y coordinate and y_{int} is the Y coordinate of the silicon/oxide interface. The YCHARN.WATT and YCHARP.WATT parameters are user-definable in the **MOBILITY** statement.

Table 3-69 User-Specifiable Parameters for Equations 3-194 and 3-195

MOBILITY	Parameter	Default	Units
MOBILITY	XMINN.WATT	-1.0×10^{32}	microns
MOBILITY	XMAXN.WATT	1.0×10^{32}	microns
MOBILITY	YMAXN.WATT	-1.0×10^{32}	microns
MOBILITY	XMINP.WATT	-1.0×10^{32}	microns
MOBILITY	XMAXP.WATT	1.0×10^{32}	microns
MOBILITY	YMAXP.WATT	-1.0×10^{32}	microns
MOBILITY	YCHARN.WATT	1.0×10^{32}	microns
MOBILITY	YCHARP.WATT	1.0×10^{32}	microns

Parallel Electric Field-Dependent Mobility

Saturation Velocity Model

As carriers are accelerated in an electric field their velocity will begin to saturate when the electric field magnitude becomes significant. This effect has to be accounted for by a reduction of the effective mobility since the magnitude of the drift velocity is the product of the mobility and the electric field component in the direction of the current flow. The following Caughey and Thomas Expression [48] is used to implement a field-dependent mobility. This provides a smooth transition between low-field and high field behavior where:

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}\,E}{\mathrm{VSATN}}\right)^{\mathrm{BETAN}}}\right]^{\frac{1}{\mathrm{BETAN}}} \quad \text{3-323}$$

$$\mu_p(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}\,E}{\mathrm{VSATP}}\right)^{\mathrm{BETAP}}}\right]^{\frac{1}{\mathrm{BETAP}}} \quad \text{3-324}$$

Here, E is the parallel electric field and μ_{n0} and μ_{p0} are the low-field electron and hole mobilities respectively. The low-field mobilities are either set explicitly in the **MOBILITY** statement or calculated by one of the low-field mobility models. The BETAN and BETAP parameters are user-definable in the **MOBILITY** statement (see Table 3-70 for their defaults).

The saturation velocities are calculated by default from the temperature-dependent models [281]:

$$\mathrm{VSATN} = \frac{\mathrm{ALPHAN.FLD}}{1+\mathrm{THETAN.FLD}\,exp\left(\frac{T_L}{\mathrm{TNOMN.FLD}}\right)} \quad \text{3-325}$$

$$\mathrm{VSATP} = \frac{\mathrm{ALPHAP.FLD}}{1+\mathrm{THETAP.FLD}\,exp\left(\frac{T_L}{\mathrm{TNOMP.FLD}}\right)} \quad \text{3-326}$$

But, you can set them to constant values on the **MOBILITY** statement using the VSATN and VSATP parameters.

In this case, no temperature dependence is implemented. Specifying the FLDMOB parameter on the **MODELS** statement invokes the field-dependent mobility. FLDMOB should always be specified unless one of the inversion layer mobility models (which incorporate their own dependence on the parallel field) are specified.

You can invoke a C-Interpreter function for the saturation velocities. The F.VSATN and F.VSATP parameters in the **MATERIAL** statement can be set to provide the filenames of two text files containing the particular functions. These functions allow you to include the temperature dependence. See Appendix A "C-Interpreter Functions" for more details.

Canali Modification

The Canali model [46] has been implemented as an alternative to using fixed values of BETAN and BETAP in the Caughey-Thomas model. This uses the exponent BETA, which depends on lattice temperature (TL), and will calculate the values of BETAN and BETAP as

$$\text{BETAN} = \text{N.BETA0}*(TL/300)^{(\text{N.BETAEXP})} \quad 3\text{-}327$$

$$\text{BETAP} = \text{P.BETA0}*(TL/300)^{(\text{P.BETAEXP})} \quad 3\text{-}328$$

The Canali model can be used for Silicon up to 430 K.

To enable the model, use the N.CANALI and P.CANALI parameters on the **MOBILITY** statement. You should also set the FLDMOB flag on the **MODELS** statement. The EVSATMOD and HVSATMOD should also have their default values of 0.

Driving Force for FLDMOB Model

It has been demonstrated that using the electric field as the driving force for field dependent mobility models results in some inaccuracy in semiconductors with strongly inhomogeneous fields. It has been suggested that using the gradient of quasi-fermi level in place of field can lead to improved results. Additionally, it has been shown that using the square root of the product of electric field and gradient of quasi-fermi level as driving force is an even better approximation [374]. You have the option to select different quantities to replace E in Equations 3-323 and 3-324 for the FLDMOB model and for the Canali extension of the FLDMOB model. You specify VSAT.QFN on the **MOBILITY** statement to use a driver

$$\left|\nabla\phi_n\right| \quad 3\text{-}329$$

instead of E for the electron mobility, where ϕ_n is the Electron quasi-Fermi potential. You specify VSAT.QFP on the **MOBILITY** statement to use a driver

$$\left|\nabla\phi_p\right| \quad 3\text{-}330$$

for the hole mobility, where ϕ_p is the Hole quasi-Fermi potential. You specify VSAT.ZAKN on the **MOBILITY** statement to use a driver

$$F = \begin{cases} \sqrt{\nabla\phi_n \cdot \vec{E}}; & \nabla\phi_n \cdot \vec{E} > 0 \\ 0; & \nabla\phi_n \cdot \vec{E} \le 0 \end{cases} \quad 3\text{-}331$$

and VSAT.ZAKP on the **MOBILITY** statement to use a driver

$$F = \begin{cases} \sqrt{\nabla\phi_p \cdot \vec{E}}; & \nabla\phi_p \cdot \vec{E} > 0 \\ 0; & \nabla\phi_p \cdot \vec{E} \le 0 \end{cases} \quad 3\text{-}332$$

This means that if the carrier motion is dominated by drift, then the mobility is reduced because the driving force in non-zero. If the carrier motion is dominated by diffusion, then the driving force is zero and the low field mobility is used.

Table 3-70 User-Definable Parameters in the Field-Dependent Mobility Model

Statement	Parameter	Default	Units
MOBILITY	ALPHAN.FLD	2.4×10^7	cm/s
MOBILITY	ALPHAP.FLD	2.4×10^7	cm/s
MOBILITY	BETAN	2.0	
MOBILITY	BETAP	1.0	
MOBILITY	N.BETAEXP	0.66	
MOBILITY	N.BETA0	1.109	
MOBILITY	N.CANALI	False	
MOBILITY	P.BETAEXP	0.17	
MOBILITY	P.BETA0	1.213	
MOBILITY	P.CANALI	False	
MOBILITY	THETAN.FLD	0.8	
MOBILITY	THETAP.FLD	0.8	
MOBILITY	TNOMN.FLD	600.0	K
MOBILITY	VSATN		cm/s
MOBILITY	VSATP		cm/s
MOBILITY	VSAT.QFN	False	
MOBILITY	VSAT.QFP	False	
MOBILITY	VSAT.ZAKN	False	
MOBILITY	VSAT.ZAKP	False	
MOBILITY	TNOMP.FLD	600.0	K

Note: Equation 3-326, which was derived for the drift-diffusion approximation, ensures that velocity overshoot cannot occur. To model velocity overshoot in silicon, apply the Energy Balance Transport Model. This model follows the above implementation but with the electric field term replaced by a new “effective” field calculated from the carrier temperature see the following section for more details.

Note: Blaze includes a different field dependent mobility model that does simulate velocity overshoot in GaAs. See Chapter 6 “Blaze: Compound Material 2D Simulator” for more information.

Carrier Temperature Dependent Mobility

Energy Balance Model

The Energy Balance Transport Model allows the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated, which is the uniform electric field value that causes the carriers in a homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields, $E_{eff,n}$ and $E_{eff,p}$, are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\texttt{TAUMOB.EL}} \quad \text{3-333}$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\texttt{TAUMOB.HO}} \quad \text{3-334}$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived from the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introduced into the relevant field dependent mobility model. The TAUMOB.EL and TAUMOB.HO parameters can be set on the **MODELS** statement and have the default values shown in Table 3-71.

The values of TAUMOB.EL and TAUMOB.HO are permitted to be different from the corresponding values of TAUREL.EL and TAUREL.HO, so as to enable a better fit to experimental data. This is necessary because it is usually a poor approximation to model the carrier relaxation times as constants. The carrier energy relaxation times may be modeled as being temperature dependent, as detailed in Section 3.4.4 "Temperature Dependence of Relaxation Times". In this case, the same electron energy relaxation time model is used in Equations 3-151 and 3-333, and the same hole energy relaxation time model is used in Equations 3-152 and 3-334.

Specify E.TAUR.GONZALEZ or E.TAUR.VAR on the **MODELS** statement to enable a temperature dependent model of electron energy relaxation time. Specify H.TAUR.GONZALEZ or H.TAUR.VAR on the **MODELS** statement to enable a temperature dependent model of hole energy relaxation time.

Table 3-71 User-Specifiable Parameters for Equations 3-333 and 3-334

Statement	Parameter	Default	Units
MODELS	TAUMOB.EL	2.5×10^{-13}	s
MODELS	TAUMOB.HO	2.5×10^{-13}	s

Four different models have been implemented into the Atlas Energy Balance Transport Model which can be chosen by the parameter EVSATMOD on the **MODELS** statement. These models shall be described next.

Setting EVSATMOD=0 implements, the default model for silicon based upon the Caughey-Thomas field-dependent mobility model in Equation 3-323. The resultant relationship between the carrier mobility and the carrier temperature is in the forms:

$$\mu_n = \frac{\mu_{n0}}{\left(1 + X_n^{\mathrm{BETAN}}\right)^{\frac{1}{\mathrm{BETAN}}}} \quad 3\text{-}335$$

$$\mu_p = \frac{\mu_{p0}}{\left(1 + X_p^{\mathrm{BETAP}}\right)^{\frac{1}{\mathrm{BETAP}}}} \quad 3\text{-}336$$

$$X_n^{\mathrm{BETAN}} = \frac{1}{2}\left(\alpha_n^{\mathrm{BETAN}}(T_n - T_L)^{\mathrm{BETAN}} + \sqrt{\alpha_n^{2\mathrm{BETAN}}(T_n - T_L)^{2\mathrm{BETAN}} - 4\alpha_n^{\mathrm{BETAN}}(T_n - T_L)^{\mathrm{BETAN}}}\right) \quad 3\text{-}337$$

$$X_p^{\mathrm{BETAP}} = \frac{1}{2}\left(\alpha_p \wedge \mathrm{BETAP}(T_n - T_L)^{\mathrm{BETAP}} + \sqrt{\alpha_p^{2\mathrm{BETAP}}(T_p - T_L)^{2\mathrm{BETAP}} - 4\alpha_p^{\mathrm{BETAP}}(T_p - T_L)^{\mathrm{BETAP}}}\right) \quad 3\text{-}338$$

$$\alpha_n = \frac{3}{2}\frac{k_B\mu_{n0}}{q\mathrm{VSATN}^2(\ \mathrm{TAUREL.EL})} \quad 3\text{-}339$$

$$\alpha_p = \frac{3}{2}\frac{k_B\mu_{p0}}{q\mathrm{VSATP}^2(\ \mathrm{TAUREL.HO})} \quad 3\text{-}340$$

where μ_{n0} and μ_{p0} are the low-field carrier mobilities and VSATN and VSATP are the saturated velocities for electrons and holes. The VSATN, VSATP, BETAN, and BETAP parameters are user-definable in the **MOBILITY** statement. The terms, TAUREL.EL and TAUREL.HO, are the energy relaxation times for electrons and holes and can be defined in the **MATERIAL** statement.

Setting EVSATMOD=0 with the additional parameter, MOBTEM.SIMPL, allows you to apply a simplified form of the above model. This form is:

$$\mu_n = \frac{\mu_{n0}}{\sqrt{1 + \alpha_n^{\ 2}(T_n - T_L)^2}} \quad 3\text{-}341$$

$$\mu_p = \frac{\mu_{p0}}{\sqrt{1 + \alpha_p^{\ 2}(T_p - T_L)^2}} \quad 3\text{-}342$$

where μ_{n0} and μ_{p0} are again the low-field carrier mobilities and $a_{n,p}$ are as defined above.

Setting EVSATMOD=1 implements the GaAs carrier temperature dependent mobility model. See Chapter 6 “Blaze: Compound Material 2D Simulator” for more information about this model.

Setting EVSATMOD=2 will apply the simple velocity limiting model based upon the electric field. In other words, the temperature dependent mobility is turned off and the standard electric field based mobility model is applied.

Note: If the YAMAGUCHI or TASCH mobility models are chosen in the **MODELS** statement, then no energy dependence is applied. No energy dependence is included in any perpendicular electric field model, such as SHIRAHATA or SURFMOB.

Table 3-72 User-Specifiable Parameters for Equations 3-335– 3-342

Statement	Parameter	Units
MOBILITY	MUN	$cm^2/(V \cdot s)$
MOBILITY	MUP	$cm^2/(V \cdot s)$
MATERIAL	VSATN	cm/s
MATERIAL	VSATP	cm/s

Meinerzhagen-Engl Model

The Meinerzhagen-Engl model is another velocity saturation model in which the mobility depends on the carrier temperature [210].

$$\mu_n = \frac{\mu_{n0}}{(1 + \alpha_n^{\beta}(T_n - T_L)^{\beta})^{1/\beta}} \quad 3\text{-}343$$

$$\mu_p = \frac{\mu_{n0}}{(1 + \alpha_p^{\beta}(T_p - T_L)^{\beta})^{1/\beta}} \quad 3\text{-}344$$

where α_n and α_p are the same as in Equations 3-339 and 3-340. If β=2, then you have the same result as in Equations 3-341 and 3-342. The exponent β, however, may now depend on lattice temperature (T_L).

As in the Canali model, the values of BETAN and BETAP are calculated as

$$\mathtt{BETAN} = \mathtt{N.BETA0} * (TL/300)^{(\mathtt{N.BETAEXP})} \quad 3\text{-}345$$

$$\mathtt{BETAP} = \mathtt{P.BETA0} * (TL/300)^{(\mathtt{P.BETAEXP})} \quad 3\text{-}346$$

although the default values differ from the Canali model (see Table 3-73).

To activate the model, specify FLDMOB on the **MODELS** statement and additionally N.MEINR or P.MEINR or both on the **MOBILITY** statement. You must at least solve for one of the electron or hole temperatures. The exponent β is constant unless you enable Lattice temperature too.

Table 3-73 User Specifiable Parameters for Meinerzhagen-Engl Model		
Statement	**Parameter**	**Default**
MOBILITY	N.MEINR	False
MOBILITY	P.MEINR	False
MOBILITY	N.BETA0	0.6
MOBILITY	P.BETA0	0.6
MOBILITY	N.BETAEXP	0.01
MOBILITY	P.BETAEXP	0.01

Complete C-interpreter Functions for Mobility

Atlas provides some C-Interpreter functions that allows you to completely specify the mobility. These can be considered as stand-alone mobility specifications as they are not combined with other mobility parameters.

First, there is a C-Interpreter function that allows the input of a mobility function, which depends explicitly on position. The parameters supplied to the function include the (X, Y, Z) coordinates of each node point, acceptor and donor concentrations, electric field, electrical potential, lattice temperature and carrier temperature.

For electron mobility, use the F.LOCALMUN parameter on the **MATERIAL** or **MOBILITY** statement. For hole mobility, use the F.LOCALMUP parameter on the **MATERIAL** or **MOBILITY** statement.

There is also a C-Interpreter function that allows you to supply a mobility, which depends explicitly on perpendicular electric field (as in a surface mobility model) and on parallel field (as in a velocity saturation model). Other parameters that are supplied are dopant densities, carrier densities, lattice temperature and composition fractions. For electron mobility, use the F.TOFIMUN parameter on the **MATERIAL** or **MOBILITY** statement. For hole mobility, use the F.TOFIMUP parameter on the **MATERIAL** or **MOBILITY** statement.

For details about C-Interpreter functions, see Appendix A “C-Interpreter Functions” and the C-Interpreter templates, which come with your Atlas distribution.

3.6.2 Mobility Model Summary

Tables 3-74 and 3-75 shows a brief description of the mobility models.

Table 3-74 Mobility Models Summary

Model	Syntax	Notes
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora's Model	ARORA	Alternative to ANALYTIC for Si.
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity saturation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.
Klaassen Model	KLA	Includes N, T, and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices
Shirahata Model	SHI	Includes N, $E_{\perp}$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_{\perp}$effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E// and $E_{\perp}$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_{\perp}$ effects. Only for 300K.

Table 3-75 Mobility Models Summary

	CONMOB	FLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	OK		—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	[2]

Key to Table Entries:

MODEL ABBREVIATION = The model that supersedes when a combination is specified. In some cases, but not all, a warning message is issued when a model is ignored.

OK = This combination is allowed.

NO = This combination isn't allowed.

NOTES:

1. Uses internal model similar to FLDMOB.
2. When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

3.6.3 Carrier Generation-Recombination Models

Carrier generation-recombination is the process through which the semiconductor material attempts to return to equilibrium after being disturbed from it. If we consider a homogeneously doped semiconductor with carrier concentrations n and p to the equilibrium concentrations n_0 an p_0 then at equilibrium a steady state balance exists according to:

$$n_0 p_0 = n_i^2 \qquad 3\text{-}347$$

Semiconductors, however, are under continual excitation whereby n and p are disturbed from their equilibrium states: n_0 and p_0. For instance, light shining on the surface of a p-type semiconductor causes generation of electron-hole pairs, disturbing greatly the minority carrier concentration. A net recombination results which attempts to return the semiconductor to equilibrium. The processes responsible for generation-recombination are known to fall into six main categories:

- phonon transitions
- photon transitions
- Auger transitions
- surface recombination
- impact ionization
- tunneling

The following sections describes the models implemented into Atlas that attempts the simulation of these six types of generation-recombination mechanisms.

Shockley-Read-Hall (SRH) Recombination

Phonon transitions occur in the presence of a trap (or defect) within the forbidden gap of the semiconductor. This is essentially a two step process, the theory of which was first derived by Shockley and Read [294] and then by Hall [113]. The Shockley-Read-Hall recombination is modeled as follows:

$$R_{SRH} = \frac{p\,n - n_{ie}^2}{\mathtt{TAUP0}\left[n + n_{ie}\,exp\left(\frac{\mathtt{ETRAP}}{kT_L}\right)\right] + \mathtt{TAUN0}\left[p + n_{ie}\,exp\left(\frac{-\mathtt{ETRAP}}{kT_L}\right)\right]} \qquad 3\text{-}348$$

where ETRAP is the difference between the trap energy level and the intrinsic Fermi level, T_L is the lattice temperature in degrees Kelvin and TAUN0 and TAUP0 are the electron and hole lifetimes. This model is activated by using the SRH parameter of the **MODELS** statement. The electron and hole lifetime parameters, TAUN0 and TAUP0, are user-definable in the **MATERIAL** statement. The default values for carrier lifetimes are shown in Table 3-76. Materials other than silicon will have different defaults. A full description of these parameters are given in Appendix B “Material Systems”.

Table 3-76 User-Specifiable Parameters for Equation 3-348			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	ETRAP	0	eV
MATERIAL	TAUN0	1×10^{-7}	s
MATERIAL	TAUP0	1×10^{-7}	s

Note: This model only presumes one trap level which, by default, is ETRAP=0 and it corresponds to the most efficient recombination centre. If the **TRAP** statement is used to define specific trap physics then separate SRH statistics are implemented as described earlier in "Trap Implementation into Recombination Models" on page 123.

SRH Concentration-Dependent Lifetime Model

The constant carrier lifetimes that are used in the SRH recombination model above can be made a function of impurity concentration [271,178,92] using the following equation:

$$R_{SRH} = \frac{pn - n_{ie}^2}{\tau_p\left[n + n_{ie}exp\left(\frac{ETRAP}{kT_L}\right)\right] + \tau_n\left[p + n_{ie}exp\left(\frac{-ETRAP}{kT_L}\right)\right]} \quad 3\text{-}349$$

where:

$$\tau_n = \frac{TAUN0}{AN + BN\left(\frac{Ntotal}{NSRHN}\right) + CN\left(\frac{Ntotal}{NSRHN}\right)^{EN}} \quad 3\text{-}350$$

$$\tau_p = \frac{TAUP0}{AP + BP\left(\frac{Ntotal}{NSRHP}\right) + CP\left(\frac{Ntotal}{NSRHP}\right)^{EP}} \quad 3\text{-}351$$

Here, N is the local (total) impurity concentration. The TAUN0, TAUP0, NSRHN, and NSRHP parameters can be defined on the **MATERIAL** statement (see Table 3-77 for their default values). This model is activated with the CONSRH parameter of the **MODELS** statement.

Table 3-77 User-Specifiable Parameters for Equations 3-349 to 3-351			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	TAUN0	1.0×10^{-7}	s
MATERIAL	NSRHN	5.0×10^{16}	cm^{-3}
MATERIAL	TAUP0	1.0×10^{-7}	s
MATERIAL	NSRHP	5.0×10^{16}	cm^{-3}
MATERIAL	AN	1.0	
MATERIAL	AP	1.0	
MATERIAL	BN	1.0	
MATERIAL	BP	1.0	
MATERIAL	CN	0.0	
MATERIAL	CP	0.0	
MATERIAL	EN	0.0	
MATERIAL	EP	0.0	

You can choose an alternate set of default values for the concentration dependent SRH model as suggested by Law et.al. [178]. Specifying CONSRH.LAW on the **MODELS** statement will select the use of the alternate default values shown in Table 3-78.

Table 3-78 Alternate default values for Equations 3-349 to 3-351			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	TAUN0	30×10^{-6}	s
MATERIAL	TAUP0	10×10^{-6}	s
MATERIAL	NSRHN	1×10^{17}	cm^{-3}
MATERIAL	NSRHP	1×10^{17}	cm^{-3}

Klaassen's Concentration Dependent Lifetime Model

The Klaassen Concentration and Temperature-dependent SRH Lifetime Model [159] is enabled by setting the KLASRH logical parameter in the **MODELS** statement. The lifetimes for electrons and holes in this model are given by the equations:

$$\mathrm{TAUN0}^{-1} = (\mathrm{KSRHTN}^{-1} + \mathrm{KSRHCN} \times N)\left(\frac{300}{T_L}\right)^{\mathrm{KSRHGN}} \quad 3\text{-}352$$

$$\mathrm{TAUP0}^{-1} = (\mathrm{KSRHTP}^{-1} + \mathrm{KSRHCP} \times \mathrm{N})\left(\frac{300}{\mathrm{T}_L}\right)^{\mathrm{KSRHGP}} \quad 3\text{-}353$$

Here, N is the local (total) impurity concentration. You can define the KSRHTN, KSRHTP, KSRHCN, KSRHCP, KSRHGN, and KSRHGP parameters in the **MATERIAL** statement. Their default values are given in Table 3-79.

Table 3-79 User-Specifiable Parameters for Equations 3-352 to 3-353

Statement	Parameter	Default	Units
MATERIAL	KSRHTN	2.5×10^{-3}	s
MATERIAL	KSRHTP	2.5×10^{-3}	s
MATERIAL	KSRHCN	3.0×10^{-13}	cm^3/s
MATERIAL	KSRHCP	11.76×10^{-13}	cm^3/s
MATERIAL	KSRHGN	1.77	
MATERIAL	KSRHGP	0.57	

Scharfetter Concentration Dependent Lifetime Model

This is an another model for the dependence of recombination lifetimes on total doping level. To enable this model, use the flag SCHSRH on the **MODELS** statement. The lifetimes are given by

$$\tau_n(N) = \mathrm{N.SCH.MIN} + \frac{(\mathrm{N.SCH.MAX} - \mathrm{N.SCH.MIN})}{\left[1 + (N/\mathrm{N\cdot SCH\cdot NREF})^{\mathrm{N.SCH\cdot GAMMA}}\right]} \quad 3\text{-}354$$

and

$$\tau_p(N) = \mathrm{P.SCH.MIN} + \frac{(\mathrm{P.SCH.MAX} - \mathrm{P.SCH.MIN})}{\left[1 + (N/\mathrm{P\cdot SCH\cdot NREF})^{\mathrm{P.SCH\cdot GAMMA}}\right]} \quad 3\text{-}355$$

where N is the Total doping density. Table 3-80 shows the **MATERIAL** statement parameters.

Table 3-80 User Specifiable Parameters for Equations 3-354 and 3-355			
Statement	**Parameter**	**Units**	**Default**
MATERIAL	N.SCH.MIN	s	0.0
MATERIAL	P.SCH.MIN	s	0.0
MATERIAL	N.SCH.MAX	s	1.0×10^{-5}
MATERIAL	P.SCH.MAX	s	3.0×10^{-6}
MATERIAL	N.SCH.GAMMA	s	1.0
MATERIAL	P.SCH.GAMMA	s	1.0
MATERIAL	N.SCH.NREF	cm^{-3}	1.0×10^{16}
MATERIAL	P.SCH.NREF	cm^{-3}	1.0×10^{16}

Coupled Defect Level Recombination Model

This model is a modification of the SRH model to the situation where there is charge transfer between two defect levels. This can lead to large excess currents in devices. The model has explained some anomalous diode characteristics [276].

Figure 3-6 shows the assumptions about the trap levels.

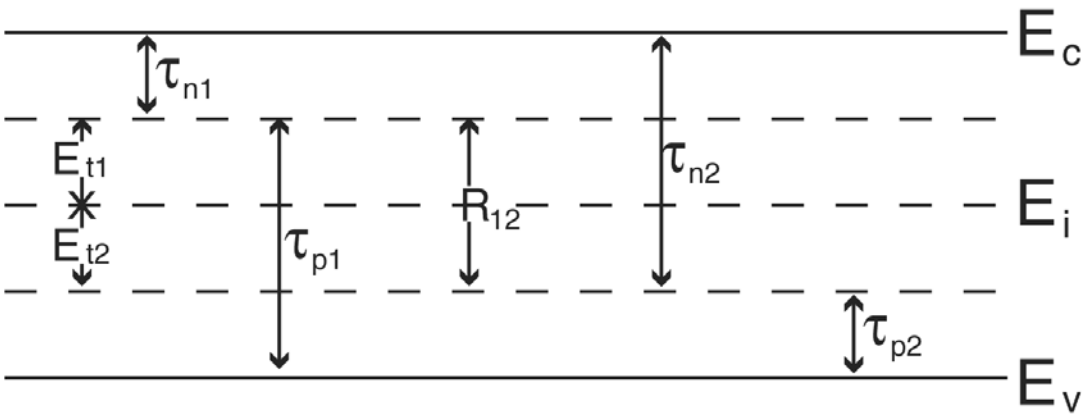


Figure 3-6: Trap Level Assumptions

If there is no coupling between the two defect levels, then the recombination total recombination rate is just the sum of two SRH terms, $R_1 + R_2$, where

$$R_1 = \frac{np - n_i^2}{r_1} \qquad 3\text{-}356$$

and

$$R_2 = \frac{np - n_i^2}{r_2} \quad \text{3-357}$$

where the denominators are

$$r_k = \tau_{nk}(p + p_k) + \tau_{pk}(n + n_k) : k = 1, 2 \quad \text{3-358}$$

Assuming Boltzmann statistics apply, the quantities n_1, n_2, p_1 and p_2 are given by

$$\left.\begin{aligned} n_1 &= n_i exp\left(\frac{E_{t1}}{kT}\right) \\ p_1 &= n_i exp\left(\frac{-E_{t1}}{kT}\right) \\ n_2 &= n_i exp\left(\frac{E_{t2}}{kT}\right) \\ p_2 &= n_i exp\left(\frac{-E_{t2}}{kT}\right) \end{aligned}\right\} \quad \text{3-359}$$

If r_{12} is non-zero, then the total recombination rate is given by

$$R = R_1 + R_2 + (\sqrt{R_{12}^2 - S_{12}} - R_{12}) \quad \text{3-360}$$
$$\times \frac{\mathtt{CDL.TN1} \cdot \mathtt{CDL.TP2}(n + n_2)(p + p_1) - \mathtt{CDL.TN2} \cdot \mathtt{CDL.TP1}(n + n_1)(p + p_2)}{r_1 r_2}$$

where

$$R_{12} = \frac{r_1 r_2}{2 \times \mathtt{CDL.COUPLING} \cdot \mathtt{CDL.TN1} \cdot \mathtt{CDL.TN2} \cdot \mathtt{CDL.TP1} \cdot \mathtt{CDL.TP2}(1 - \varepsilon)} \quad \text{3-361}$$
$$+ \frac{\mathtt{CDL.TN1}(p + p_1) + \mathtt{CDL.TP2}(n + n_2)}{2\mathtt{CDL} \cdot \mathtt{TN1} \cdot \mathtt{CDL.TP2}(1 - \varepsilon)}$$
$$+ \frac{\varepsilon[\mathtt{CDL} \cdot \mathtt{TN2}(p + p_2) + \mathtt{CDL.TP1}(n + n_1)]}{2\mathtt{CDL.TN2} \cdot \mathtt{CDL.TP1}(1 - \varepsilon)}$$

and

$$S_{12} = \frac{1}{\mathtt{CDL.TN1} \cdot \mathtt{CDL.TP2}(1 - \varepsilon)}\left(1 - \frac{\mathtt{CDL.TN1} \cdot \mathtt{CDL.TP2}\varepsilon}{\mathtt{CDL.TN2} \cdot \mathtt{CDL.TP1}}\right)(np - n_i^2) \quad \text{3-362}$$

The factor ε is given by

$$\varepsilon = exp\left(\frac{-(|E_{t2} - E_{t1}|)}{K_B T}\right) \quad \text{3-363}$$

and will generally be less than 1. It can be shown that as r_{12} becomes small, the total recombination rate simplifies to $R_1 + R_2$ as required.

You can set all the relevant parameters in Atlas. To set the rate r_{12}, use CDL.COUPLING on the **MATERIAL** statement. To set the carrier lifetimes, use CDL.TN1, CDL.TN2, CDL.TP1 and CDL.TP2 on the **MATERIAL** statement. To set the energies of the defect levels relative to the intrinsic level, use CDL.ETR1 and CDL.ETR2 on the **MATERIAL** statement.

With reference to the energies shown in Figure 3-6, CDL.ETR1 will be positive and CDL.ETR2 will be negative. Table 3-81 shows these quantities and the units and default values.

To enable the model, use either the CDL parameter or the CONCDL parameter on the **MODELS** statement. If you set CDL, then the values used for the lifetimes are those set by CDL.TN1, CDL.TN2, CDL.TP1 and CDL.TP2. If you set CONCDL, then the lifetimes used are doping dependent and are derived from Equations 3-350 and 3-351 with the lifetimes being CDL.TN1, CDL.TN2, CDL.TP1 and CDL.TP2, instead of TAUN0 and TAUP0. The default values for the other parameters are shown in Table 3-77.

SRH and CDL are mutually exclusive parameters. If you enable CDL, the CDL recombination rate is output to a structure file in the SRH slot and are added to the output of total recombination rate.

Table 3-81 Material Parameters for Coupled Defect Level Model

Statement	Parameter	Type	Units	Default
MATERIAL	CDL.ETR1	Real	eV	0.0
MATERIAL	CDL.ETR2	Real	eV	0.0
MATERIAL	CDL.TN1	Real	s	1.0
MATERIAL	CDL.TN2	Real	s	1.0
MATERIAL	CDL.TP1	Real	s	1.0
MATERIAL	CDL.TP2	Real	s	1.0
MATERIAL	CDL.COUPLING	Real	s^{-1} cm^{-3}	1.0

The individual lifetimes cannot be output to a structure file.

Trap Assisted Auger Recombination

This model adds in a dependence of recombination lifetime on carrier density, and will only be significant at fairly high carrier densities [93].

The carrier lifetimes are reduced according to the following formula

$$\tau_n = \frac{\tau_n}{(1 + \mathrm{TAA.CN}(n + p)\tau_n)} \quad \text{3-364}$$

$$\tau_p = \frac{\tau_p}{(1 + \mathrm{TAA.CP}(n + p)\tau_p)} \quad \text{3-365}$$

where n is the electron density and p the hole density. To enable the model, specify TRAP.AUGER on the **MODELS** statement. It will then apply to the SRH model if enabled or to the CDL model if enabled.

You can set the parameters on the **MATERIAL** statement with the defaults as shown in Table 3-82. .

Table 3-82 Parameters for Trap assisted Auger model

Statement	Parameter	Units	Default
MATERIAL	TAA.CN	cm^3/s	1.0×10^{-12}
MATERIAL	TAA.CP	cm^3/s	1.0×10^{-12}

Trap-Assisted Tunneling

In a strong electric field, electrons can tunnel through the bandgap via trap states. This trap-assisted tunneling mechanism is enabled by specifying TRAP.TUNNEL on the **MODELS** statement and is accounted for by modifying the Schockley-Read-Hall recombination model.

$$R_{SRH} = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUP0}}{1 + \Gamma_p^{DIRAC}}\left[n + n_{ie}exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right)\right] + \frac{\mathrm{TAUN0}}{1 + \Gamma_n^{DIRAC}}\left[p + n_{ie}exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]} \quad \text{3-366}$$

Here, $\Gamma_n^{\mathrm{DIRAC}}$ is the electron field-effect enhancement term for Dirac wells, and $\Gamma_p^{\mathrm{DIRAC}}$ is the hole field-effect enhancement term for Dirac wells. $\Gamma_n^{\mathrm{DIRAC}}$ and $\Gamma_p^{\mathrm{DIRAC}}$ are defined in Equations 3-101 and 3-102.

Schenk Model for Trap Assisted Tunneling [278]

The Schenk model gives the field-effect enhancement factors as an analytic function. Before using this analytic function, however, Atlas works out several intermediate quantities. The first is an effective trap level E_t given by

$$E_t = \frac{1}{2}Eg(T) + \frac{3}{4}kT\ln\left(\frac{m_c}{m_v}\right) + \mathtt{ETRAP} - \left(32R_c h^3\Theta^3\right)^{1/4} \quad \text{3-367}$$

for electrons

$$E_t = \frac{1}{2}Eg(T) - \frac{3}{4}kT\ln\left(\frac{m_c}{m_v}\right) - \mathtt{ETRAP} - \left(32R_v h^3\Theta^3\right)^{1/4} \quad \text{3-368}$$

for holes. The quantities R_c and R_v are the effective Rydberg energies given by

$$R_c = m_c\left(\frac{\mathtt{Z.SCHENK}}{\mathtt{PERMITTIVITY}}\right)^2 \times 13.6eV \quad \text{3-369}$$

$$R_v = m_v\left(\frac{\mathtt{Z.SCHENK}}{\mathtt{PERMITTIVITY}}\right)^2 \times 13.6eV \quad \text{3-370}$$

The quantity Θ is known as the electro-optical frequency and is given as

$$\Theta = \left(\frac{q^2F^2}{2\hbar m_o \mathtt{ME.TUNNEL}}\right)^{1/3}$$
$$\Theta = \left(\frac{q^2F^2}{2\hbar m_o \mathtt{MH.TUNNEL}}\right)^{1/3} \quad \text{3-371}$$

where F is the electric field. If you do not specify `ME.TUNNEL`, then a value of `0.5*(MT1 + MT2)` will be used for silicon and the density of states effective electron mass for other materials. If you do not specify `MH.TUNNEL`, then the density of states effective hole mass will be used.

The next quantity is the lattice relaxation energy given by

$$\varepsilon_R = \mathtt{HUANG.RHYS} \times \mathtt{PHONON.ENERGY} \quad \text{3-372}$$

From this is derived the quantity ε_F given by

$$\varepsilon_F = \frac{(2\varepsilon_R kT)^2}{(\hbar\Theta)^3} \quad \text{3-373}$$

Atlas then calculates the optimum horizontal transition energy, E_0. This is the energy at which the tunneling probability for carriers to tunnel from the trap level to the band edge has a maximum. It is given by

$$E_o = 2\sqrt{\varepsilon_F}\sqrt{\varepsilon_F + E_t + \varepsilon_R} - 2\varepsilon_F - \varepsilon_R \quad \text{3-374}$$

with the maximum permitted value of E_o being

$$E_0 = E_t - \frac{(E_T + \varepsilon_R)^2}{4\varepsilon_F}\left(1 - \frac{(E_T + \varepsilon_R)}{2\varepsilon_F}\right) \quad \text{3-375}$$

when $E_T + \varepsilon_R \ll \varepsilon_F$ and the minimum permitted value of E_o, when ε_F is small, being

$$E_0 = \varepsilon_R \exp\left(-\frac{\text{PHONON.ENERGY}}{KT}\right) \quad \text{3-376}$$

Equation 3-375 represents the low field limit and Equation 3-376 represents the high field limit.

The field enhancement factor for electrons is then given by

$$\Gamma_n = \left[1 + \frac{(\hbar\theta)^{3/2}\sqrt{E_t - E_o}}{E_o \text{PHONON.ENERGY}}\right]^{-1/2} \frac{(\hbar\theta)^{3/4}(E_t - E_o)^{1/4}}{2\sqrt{E_t E_o}}\left(\frac{h\Theta}{kT}\right)^{3/2} \quad \text{3-377}$$

$$\times exp\left[\frac{-(E_t - E_o)}{\text{PHONON.ENERGY}} + \frac{\text{PHONON.ENERGY} - kT}{2\text{PHONON.ENERGY}}\right.$$

$$\left. + \frac{E_t + kT/2}{\text{PHONON.ENERGY}} ln(E_t/\varepsilon_R) - \frac{E_o ln(E_o/\varepsilon_R)}{\text{PHONON.ENERGY}}\right]$$

$$\times exp\left(\frac{E_t - E_o}{kT}\right) exp\left[-\frac{4}{3}\left(\frac{E_t - E_o}{kT}\right)^{3/2}\right]$$

where E_T is calculated by using Equation 3-367. The field enhancement factor for holes is obtained using Equation 3-377 but with E_t obtained from Equation 3-368. The default effective masses used in the above calculation are correct for Silicon with the electric field in the <100> direction. They are also accurate for the <110> direction, but for the <111> direction a value of

$$m_c = 3\left(\frac{1}{m_{t1}} + \frac{1}{m_{t2}} + \frac{1}{m_l}\right)^{-1} \quad \text{3-378}$$

should be used for electrons. To enable the model, specify SCHENK.TUN on the **MODELS** statement and the parameters shown in Table 3-83.

Table 3-83 User Definable Parameters for Schenk Trap Assisted Tunneling Model

Statement	Parameter	Units	Default
MATERIAL	PHONON.ENERGY	eV	0.068
MATERIAL	Z.SCHENK		0.0
MATERIAL	HUANG.RHYS		3.5

Poole-Frenkel Barrier Lowering for Coulombic Wells

The Poole-Frenkel effect can enhance the emission rate for Coulombic wells. I the TRAP.COULOMBIC parameter is specified in the **MODELS** statement, the Schockley-Read-Hall electron and hole recombination model becomes:

$$R_{n,SRH} = \frac{p_n - n_{ie}^2}{\left(\frac{\text{TAUP0}}{1+\Gamma_p^{DIRAC}}\right)\left[n + n_{ie}\exp\left(\frac{\text{ETRAP}}{kT_L}\right)\right] + \left(\frac{\text{TAUN0}}{\chi_F+\Gamma_n^{COUL}}\right)\left[p + n_{ie}\exp\left(\frac{-\text{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}379$$

$$R_{p,SRH} = \frac{p_n - n_{ie}^2}{\left(\frac{\text{TAUP0}}{\chi_F+\Gamma_n^{COUL}}\right)\left[n + n_{ie}\exp\left(\frac{\text{ETRAP}}{kT_L}\right)\right] + \left(\frac{\text{TAUN0}}{1+\Gamma_p^{DIRAC}}\right)\left[p + n_{ie}\exp\left(\frac{-\text{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}380$$

The Poole-Frenkel thermal emission factor, χ_F, is defined in Equation 3-109. The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_P^{COUL} are defined in Equations 3-111 and 3-112.

JTAT Model

A Trap assisted Tunneling model, based on [50], has been implemented in Atlas. It is an alternative to the TRAP.TUNNEL model and cannot be used concurrently with the TRAP.TUNNEL model. Instead of modifying the SRH lifetimes, as does the TRAP.TUNNEL model, it calculates a Generation-Recombination rate dependent on the local value of electric field. This is given by

$$\frac{Q^3 m_o m^* F M^2 N_t G_t}{8\pi\hbar^3(E_g - E_t)} exp\left(-\frac{4\sqrt{2m_o m^*(E_g - E_t)^3}}{3Q\hbar F}\right) \quad 3\text{-}381$$

where

- F is the magnitude of the local electric field.
- m^* is the effective mass.
- E_g is the semiconductor energy gap.
- N_t is the trap density.
- E_t is the trap energy.
- G_t is the trap degeneracy.

The factor M^2 is the square of the matrix element associated with the trap potential. The quantity E_g-E_t is assumed to correspond to the rate-limiting step of the two-stage tunnelling process and will have a minimum possible value of 0.5 E_g. The model can be used with or without traps being defined by the **TRAP** or **DEFECTS** statements. To use it without **TRAP**, you specify the flag JTAT on the **MODELS** statement. The value of E_g-E_t is obtained as $0.5E_g + |\text{JTAT.LEVEL}|$ because the trap level will be relatively close in energy to the middle of the energy gap. JTAT.LEVEL is the trap energy relative to the midgap energy. To set the trap density, you use the JTAT.DENSITY parameter on the **MODELS** statement. The square of

the matrix element is set by using the JTAT.M2 parameter on the **MODELS** statement. The value of JTAT.M2 can also be set for specific materials using the JTAT.M2 parameter on the **MATERIAL** statement. The factor G_t is assumed to be 1 for the JTAT model.

The units and default values of these quantities are shown in Table 3-84.

Table 3-84 Parameters for the JTAT Model

Statement	Parameter	Type	Default	Units
MODELS	JTAT	Logical	False	
MODELS	JTAT.LEVEL	Real	0.0	eV
MODELS	JTAT.DENSITY	Real	0.0	cm^{-3}
MODELS	JTAT.M2	Real	10^{-20}	V^2cm^{-3}
MATERIAL	JTAT.M2	Real	10^{-20}	V^2cm^{-3}

TRAP.JTAT Model

The JTAT model can only be used for one tunneling level, and does not put the trap charge density into the simulation. The TRAP.JTAT model is designed to be used along with traps as specified on the **TRAP** statement (Section 3.3.3 “Traps and Defects”) or using the **DEFECTS** statement (Chapter 15 “TFT: Thin-Film Transistor Simulator”). For traps defined by the **TRAP** (or **DOPING**) statements, the quantity E_g-E_t is interpreted as being the maximum value of E_g-E.LEVEL and E.LEVEL. The trap density is given by the DENSITY parameter and the trap degeneracy G_t, by DEGEN.FAC. The square of the interaction matrix element, JTAT.M2, is set on a material by material basis on the **MATERIAL** statement or globally on the **MODELS** statement. With the **DEFECTS** statement, the density used for the TRAP.JTAT calculation is the value obtained from Equations 15-2 and 15-5, evaluated at the energy JTAT.LEVEL. The tunneling is assumed to occur through the defects at an energy level JTAT.LEVEL with E_g-E_t being the maximum value of E_g-JTAT.LEVEL and JTAT.LEVEL. The value of the interaction matrix element is the one specified on the **MATERIAL** statement. The trap degeneracy G_t, is assumed to be 1, and the parameter JTAT.M2 is used as before.

The extra parameters for the TRAP.JTAT model are shown in Table 3-85 .

Table 3-85 Parameters for the TRAP.JTAT Model

Statement	Parameter	Type	Default	Units
MODELS	TRAP.JTAT	Logical	False	
MATERIAL	JTAT.M2	Real	10^{-20}	V^2cm^{-3}

The generation rate given by Equation 3-381 is output to Silvaco Structure Files in the overall recombination rate. You set the U.TRAP flag on the **OUTPUT** statement to additionally output it as a trap recombination rate.

Optical Generation/Radiative Recombination

The next physical mechanisms we have to consider for generation/recombination are photon transition. This mechanism occurs primarily in one step and is therefore a direct generation/ recombination mechanism. There are two partial processes involved. For radiative recombination, an electron loses energy on the order of the band gap and moves from the conduction band to the valence band. For optical generation, an electron moves from the valence band to the conduction. In silicon, band to band generation/recombination is insignificant. This effect, however, is important for narrow gap semiconductors and semiconductors whose specific band structure allows direct transitions. By assuming a capture rate C_c^{OPT} and an emission rate C_e^{OTP}, the involved partial processes can be written as

$$R_{np}^{OPT} = C_c^{OPT} np, \tag{3-382}$$

for recombination and

$$G_{np}^{\mathtt{OPT}} = C_e^{\mathtt{OPT}} \tag{3-383}$$

for generation.

These rates must be equal in thermal equilibrium so that

$$C_{np}^{OPT} = C_c^{OPT} n_{ie}^2 \tag{3-384}$$

The total band to band generation/recombination is the difference of the partial rates, which equates to

$$R_{np}^{OPT} = C_c^{OPT} (np - n_{ie}^2). \tag{3-385}$$

In Atlas, C_c^{OPT} and can be defined by COPT on the **MATERIAL** statement or implemented using a C-Interpreter routine. To turn on the optical recombination/ generation model, define the OPTR keyword on the **MODELS** statement.

Auger Recombination

Auger recombination occurs through a three particle transition whereby a mobile carrier is either captured or emitted. The underlying physics for such processes is unclear and normally a more qualitative understanding is sufficient [286].

Standard Auger Model

Auger Recombination is commonly modeled using the expression [78]:

$$R_{Auger} = \mathtt{AUGN}\,(pn^2 - nn_{ie}^2) + \mathtt{AUGP}\,(np^2 - pn_{ie}^2) \tag{3-386}$$

where the model parameters AUGN and AUGP are user-definable in the **MATERIAL** statement (see Table 3-86 for its default value). You can activate this model with the AUGER parameter from the **MODELS** statement.

Table 3-86 User-Specifiable Parameters for Equation 3-386			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	AUGN	2.8×10^{-31}	cm^6/s
MATERIAL	AUGP	9.9×10^{-32}	cm^6/s

Klaassen's Temperature-Dependent Auger Model

The Klaassen Auger Recombination Model [160] is activated by specifying the KLAAUG parameter of the **MODELS** statement. The form of this model is

$$R_{Auger} = C_n(pn^2 - nn_{ie}^2) + C_p(np^2 - pn_{ie}^2) \quad 3\text{-}387$$

where the Auger coefficients are temperature dependent according to:

$$C_n = \text{KAUGCN}\left(\frac{T_L}{300}\right)^{\text{KAUGDN}} \quad 3\text{-}388$$

$$C_p = \text{KAUGCP}\left(\frac{T_L}{300}\right)^{\text{KAUGDP}} \quad 3\text{-}389$$

Here, the KAUGCN, KAUGCP, KAUGDN, and KAUGDP parameters are user-definable in the **MATERIAL** statement and have the defaults shown in Table 3-87.

Table 3-87 User-Specifiable Parameters for Equation 3-388 and 3-389			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	KAUGCN	1.83×10^{-31}	cm^6/s
MATERIAL	KAUGCP	2.78×10^{-31}	cm^6/s
MATERIAL	KAUGDN	1.18	
MATERIAL	KAUGDP	0.72	

Narrow Bandgap Auger Model

An alternative model for the Auger recombination coefficients that is more suitable for modeling Auger processes in narrow bandgap semiconductors can be enabled by setting the parameters AUGKN and AUGKP on the **MATERIAL** statement and by specifying AUGER on the **MODELS** statement. The model in Atlas is a simplification of that by Beattie [24] and takes the form:

$$R_{Auger} = C_n(pn^2 - nn_{ie}^2) + C_p(np^2 - pn_{ie}^2) \quad \text{3-390}$$

where the Auger coefficients are concentration dependent according to:

$$C_n = \frac{\text{AUGN}}{1 + \text{AUGKN}\ n} \quad \text{3-391}$$

$$C_p = \frac{\text{AUGP}}{1 + \text{AUGKP}\ p} \quad \text{3-392}$$

Here, n and p are the electron and hole carrier concentrations and the new parameters, AUGKN and AUGKP, are user-definable on the **MATERIAL** statement.

Auger Recombination Model for High and Low Injection Conditions [64,153]

The Auger recombination rate given in Equation 3-390 can be modified to account for high and low injection conditions. In this model, the Auger rate coefficients, C_n and C_p, are carrier and doping concentration dependent and are given by:

$$C_n = \text{AUG.CNL}\left(\frac{N_D}{N_D + p}\right) + \frac{\text{AUG.CHI}}{2}\left(\frac{p}{N_D + p}\right) \quad \text{3-393}$$

$$C_p = \text{AUG.CPL}\left(\frac{N_A}{N_A + n}\right) + \frac{\text{AUG.CHI}}{2}\left(\frac{n}{N_A + n}\right) \quad \text{3-394}$$

Here, you can specify the AUG.CNL, AUG.CPL, and AUG.CHI parameters on the **MATERIAL** statement and have defaults described in Table 3-66.

Table 3-88 User Specifiable Parameters for Equations 3-330 and 3-331.

Statement	Parameter	Default	Units
MATERIAL	AUG.CNL	2.2×10^{-31}	cm^6/s
MATERIAL	AUG.CPL	9.2×10^{-32}	cm^6/s
MATERIAL	AUG.CHI	1.66×10^{-30}	cm^6/s

To enable this model, specify PIC.AUG on the **MODELS** statement.

Kerr Model for Auger Recombination

In addition to equations 3-393 and 3-394, reference [153] presents a further model for Auger recombination, applicable to both low and high injection situations. In this model, the auger recombination rate is given by

$$R_{auger} = (n\,p - n_i^2)(C1.KERR\ N0^{P1.KERR} + C2.KERR\ N0^{P2.KERR} + C3.KERR\ \Delta N^{P3.KERR}) \quad 3\text{-}395$$

where

$$N0 = \begin{cases} N_d - N_a & \text{if} \quad N_d > N_a \\ \dfrac{n_i^2}{P0} & \text{if} \quad N_d < N_a \end{cases} \quad 3\text{-}396$$

$$P0 = \begin{cases} N_a - N_d & \text{if} \quad N_a > N_d \\ \dfrac{n_i^2}{N0} & \text{if} \quad N_a < N_d \end{cases} \quad 3\text{-}397$$

where N_d is the local donor density, N_a is the local acceptor density and n_i^2 is the square of the intrinsic carrier density. The quantity ΔN is the minimum of either the local electron or hole concentration.

This recombination model is enabled by specifying the parameter KERR.AUG on the **MODELS** statement. The default values of the parameters, as shown in Table 3-89, are fitted to experimental values for Silicon at 300K. They can also be set on the **MATERIAL** statement for other materials and other temperatures. In the case where $n\,p < n_i^2$, equation will only act as a generation term if the flag AUGGEN is enabled on the **MODELS** statement. The units of the parameters C1.KERR, C2.KERR and C3.KERR, depend on the values of the parameters P1.KERR, P2.KERR and P3.KERR so as to give overall units for R_{auger} as $cm^{-3}s^{-1}$.

Table 3-89 Default Values of Parameters for Kerr's Auger Recombination Model

Statement	Parameter	Default	Units
MODELS	KERR.AUG	False	
MATERIAL	C1.KERR	1.8×10^{-24}	$cm^{3(1+P1.KERR)}s^{-1}$
MATERIAL	C2.KERR	6.0×10^{-25}	$cm^{3(1+P2.KERR)}s^{-1}$
MATERIAL	C3.KERR	3.0×10^{-27}	$cm^{3(1+P3.KERR)}s^{-1}$
MATERIAL	P1.KERR	0.65	
MATERIAL	P2.KERR	0.65	
MATERIAL	P3.KERR	0.8	

Richter Model for Auger Recombination

This model is based on measurements of effective lifetimes of carriers injected into silicon with various doping densities and injection levels. It is an advanced parameterization of the results at 300K which takes into account coulomb enhancements [266]. The auger recombination rate is given by

$$R_{auger} = (n\,p - n_i^2)(C1.RICHTER\,G_{eeh}\,N0 + C2.RICHTER\,G_{ehh}\,P0 + C3.RICHTER\,\Delta N^{P3.RICHTER}) \quad 3\text{-}398$$

where

$$N0 = \begin{cases} N_d - N_a & \text{if} \quad N_d > N_a \\ \dfrac{n_i^2}{P0} & \text{if} \quad N_d < N_a \end{cases} \quad 3\text{-}399$$

$$P0 = \begin{cases} N_a - N_d & \text{if} \quad N_a > N_d \\ \dfrac{n_i^2}{N0} & \text{if} \quad N_a < N_d \end{cases} \quad 3\text{-}400$$

where N_d is the local donor density, N_a is the local acceptor density and n_i^2 is the square of the intrinsic carrier density. The quantity ΔN is the minimum of either the local electron or hole concentration.

The quantities G_{eeh} and G_{ehh} are defined as

$$G_{eeh} = 1.0 + 13.0\left\{1.0 - \tanh\left[\left(\frac{N0}{NE.RICHTER}\right)^{P1.RICHTER}\right]\right\} \quad 3\text{-}401$$

$$G_{ehh} = 1.0 + 7.5\left\{1.0 - \tanh\left[\left(\frac{P0}{NH.RICHTER}\right)^{P2.RICHTER}\right]\right\} \quad 3\text{-}402$$

The model is enabled by specifying RICHTER.AUG on the **MODELS** statement. The values of the default parameters are fitted to Silicon at 300K and are given in Table 3-90.

Table 3-90 Default Values of Parameters for Richter's Auger Recombination Model

Statement	Parameter	Default	Units
MODELS	RICHTER.AUG	False	
MATERIAL	C1.RICHTER	2.5×10^{-31}	cm^6s^{-1}
MATERIAL	C2.RICHTER	8.5×10^{-32}	cm^6s^{-1}
MATERIAL	C3.RICHTER	3.0×10^{-29}	$cm^{3\,(1+P3.KERR)}s^{-1}$
MATERIAL	P1.RICHTER	0.66	
MATERIAL	P2.RICHTER	0.63	

Table 3-90 Default Values of Parameters for Richter's Auger Recombination Model			
MATERIAL	P3.RICHTER	0.92	
MATERIAL	NE.RICHTER	3.3×10^{17}	cm^{-3}
MATERIAL	NH.RICHTER	7.0×10^{17}	cm^{-3}

Auger Recombination Model With Temperature and Concentration Dependent Coefficients

The rate of Auger recombination is usually given by

$$R_{Auger} = C_n(n^2p - nn_{ie}^2) + C_p(p^2n - pn_{ie}^2) \quad \text{3-403}$$

The coefficients are dependent on both lattice temperature and carrier density [131]. The carrier density dependence is such that the coefficients are reduced in size as carrier density increases. The physical explanation of this dependence is that carrier-carrier interactions become more significant at higher values of carrier density and reduce the Auger recombination rate [112].

The coefficients are given by

$$C_n(T,n) = \begin{aligned}&[\mathtt{HNS.AE} + \mathtt{HNS.BE}(T/300) + \mathtt{HNS.CE}(T/300)^2] \\ &\times[1 + \mathtt{HNS.HE}[exp(-n/\mathtt{HNS.N0E})]]\end{aligned} \quad \text{3-404}$$

$$C_p(T,p) = \begin{aligned}&[\mathtt{HNS.AH} + \mathtt{HNS.BH}(T/300) + \mathtt{HNS.CH}(T/300)^2] \\ &\times[1 + \mathtt{HNS.HH}[exp(-p/\mathtt{HNS.N0H})]]\end{aligned} \quad \text{3-405}$$

where n is electron density, p is hole density, and T is lattice temperature.

To enable the model, set the flag HNSAUG on the **MODELS** statement. You can change all the parameters by specifying them on the **MATERIAL** statement. Table 3-92 shows the default values. Setting HNS.HE=0 and HNS.HH=0 will remove the carrier density dependence if required.

Table 3-91 User-Specifiable Parameters for Equations 3-405 to 3-406			
MATERIAL	HNS.AE	6.7×10^{-32}	cm^6/s
MATERIAL	HNS.AH	7.2×10^{-32}	cm^6/s
MATERIAL	HNS.BE	2.45×10^{-31}	cm^6/s
MATERIAL	HNS.BH	4.5×10^{-33}	cm^6/s
MATERIAL	HNS.CE	-2.2×10^{-32}	cm^6/s
MATERIAL	HNS.CH	2.63×10^{-32}	cm^6/s
MATERIAL	HNS.HE	3.4667	

MATERIAL	HNS.HE	8.25688	
MATERIAL	HNS.N0E	1.0×10^{18}	cm^{-3}
MATERIAL	HNS.N0H	1.0×10^{18}	cm^{-3}

Surface Recombination

In addition to generation-recombination within the bulk of the semiconductor, electrons or holes may recombine or be generated at interfaces. The rate of surface recombination may be even greater than within the bulk. The standard method is to model interface recombination in a similar manner as the bulk generation-recombination rate [108] where:

$$R_{surf} = \frac{pn - n_{ie}^2}{\tau_p^{eff}\left[n + n_{ie}exp\left(\frac{ETRAP}{kT_L}\right)\right] + \tau_n^{eff}\left[p + n_{ie}exp\left(\frac{-ETRAP}{kT_L}\right)\right]} \quad 3\text{-}406$$

Here:

$$\frac{1}{\tau_n^{eff}} = \frac{1}{\tau_n^i} + \frac{d_i}{A_i}S.N \quad 3\text{-}407$$

and

$$\frac{1}{\tau_p^{eff}} = \frac{1}{\tau_p^i} + \frac{d_i}{A_i}S.P \quad 3\text{-}408$$

τ_n^i is the bulk lifetime calculated at node i along the interface and which may be a function of the impurity concentration as well. The d_i and A_i parameters are the length and area of the interface for node i. The S.N and S.P parameters are the recombination velocities for electrons and holes respectively, which are user-definable in the **INTERFACE** statement. The X.MIN, X.MAX, Y.MIN, and Y.MAX parameters can also be set in the **INTERFACE** statement to define the region, where the specified values of the surface recombination velocities apply. This model is activated by the presence of the recombination velocities in the **INTERFACE** statement.

Table 3-92 User-Specifiable Parameters for Equations 3-407 to 3-408

Statement	Parameter	Default	Units
INTERFACE	S.N	0	cm/s
INTERFACE	S.P	0	cm/s

Zamdmer Model for LT-GaAs

Low temperature grown GaAs (LT-GaAs) is GaAs grown at a temperature of 200 - 300°, usually with a slight excess of Arsenic. The properties of LT-GaAs depend on the growth temperature and depend on the degree of post-growth annealing [193]. It can be produced with high resistivity and with sub-picosecond carrier lifetime, which makes it ideal for

ultrafast photodetectors that are based on displacement current. In that category of device, the response time is limited by the recombination lifetime of photogenerated carriers. It has been observed that operating these devices under high bias conditions tends to increase the response time [375].

One possible explanation of this effect is the effective decrease in the capture cross-section with field of the recombination centers as also observed in SiO_2 [222]. According to [375], this causes an increase in the electron lifetime with electric field, F, which can be modeled as

$$\tau(F) = \tau(0)\left(\frac{T_e(F)}{T_L}\right)^{1.5}\left(\frac{r(0)}{r(F)}\right)^3 \quad \text{3-409}$$

where $\tau(0)$ is the electron lifetime with no field present. The variable r is the radius of the coulombic potential well caused by the trapping centers. With zero applied field, this radius is isotropic but as a field is applied, it becomes anisotropic and you can define a miniumum and maximum radius in the direction of the field. $r(F)$ is the minimum radius defined as the minimum radius at which the potential between two trapping centers is equal to the maximum potential between them minus 2 K_BT/q. The ratio $\frac{T_e(F)}{T_L}$ is the ratio of electron temperature to lattice temperature. The electron temperature is approximated as

$$T_e = T_L + \frac{\lambda Fe}{K_B} \quad \text{3-410}$$

where λ is the optical phonon mean free path and K_B is Boltzmanns constant.

The model is designed for discrete traps. These are created by either the **TRAP** statement or the **TRAP** parameter on the **DOPING** statement. You enable the model by setting the ZAMDMER parameter on the **MODELS** statement. You set the distance between the recombination centers using the ZAMDMER.Z0 field on the **MATERIAL** statement, and set the value of λ with the IG.LRELE parameter on the **MODELS** statement.

The model evaluates the recombination lifetime at every point in the LT-GaAs according to the value of electric field there. An example of the amount by which the electron lifetime is enhanced as a function of field is given in Figure 3-7.

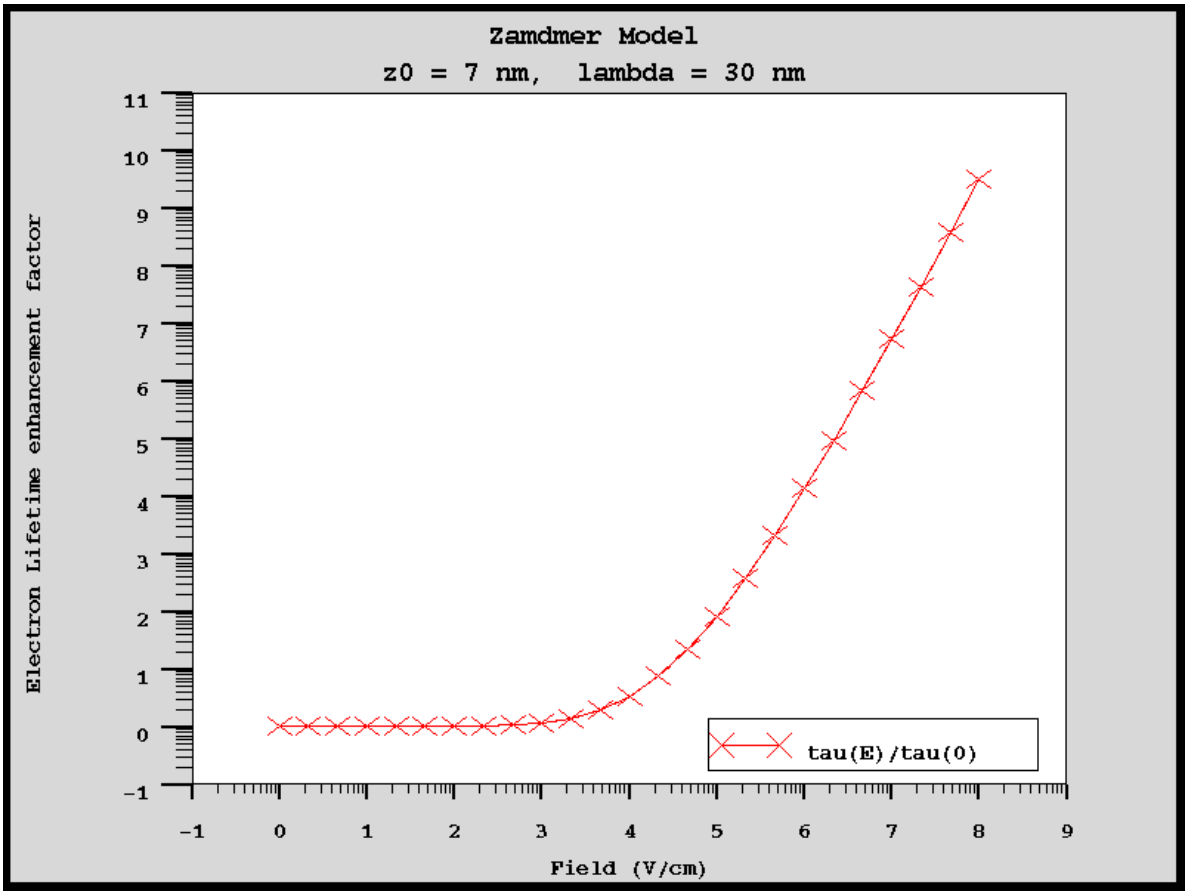


Figure 3-7: Log-Log plot of lifetime enhancement factor as a function of electric field. The trap separation is 7 nm and the value of λ is 30 nm for this example.

A summary of the parameters used for the ZAMDMER model are given in Table 3-93.

Table 3-93 Parameters for the ZAMDMER Model

Statement	Parameter	Type	Default	Units
MODELS	ZAMDMER	Logical	False	
MODELS	IG.LRELE	Real	2.0×10^{-6}	cm
MATERIAL	ZAMDMER.Z0	Real	7.0×10^{-7}	cm

3.6.4 Impact Ionization Models

In any space charge region with a sufficiently high reverse bias, the electric field will be high enough to accelerate free carriers up to a point where they will have acquired sufficient energy to generate more free carriers when in collision with the atoms of the crystal. In order to acquire sufficient energy, two principle conditions must be met.

First, the electric field must be sufficiently high. Then, the distance between the collisions of the free carrier must be enough to allow acceleration to a sufficiently high velocity

In other words, the carrier must gain the ionization energy E_i between collisions. If the generation rate of these free carriers is sufficiently high this process will eventually lead to avalanche breakdown.

The general impact ionization process is described by the Equation 3-411.

$$G = \alpha_n \left|\vec{J}\right|_n + \alpha_p \left|\vec{J}\right|_p \quad \text{3-411}$$

Here, G is the local generation rate of electron-hole pairs, $\alpha_{n,p}$ are the ionization coefficient for electrons and holes and $J_{n,p}$ are their current densities. The ionization coefficient represents the number of electron-hole pairs generated by a carrier per unit distance travelled. The accurate calculation of this parameter has been researched because it is vital if the effects related to impact ionization, such as substrate current and device breakdown, are to be simulated. These models can be classified into two main types: local and non-local models.

The former assume that ionization at any particular point within the device is a function only of the electric field at that position. Non-local models, however, perform a more rigorous approach by taking into account the energy that the carrier gains.

Geometrical Considerations for Impact Ionization Models

In all the available models discussed in the following sections, the ionization coefficients are dependent on electric field. There are several different ways to consider the interaction between electric field and current implied by Equation 3-411.

During simulation, currents and fields are calculated both as scalar values on the edges of triangles and as vector quantities on the triangles themselves. Atlas allows three different ways of looking at Equation 3-411.

The first model uses Equation 3-412.

$$G = \alpha_n(\left|E_{tri}\right|)J_{ntri} + \alpha_p(\left|E_{tri}\right|)J_{ptri} \quad \text{3-412}$$

Here, E_{tri} is the vector field on the triangle, J_{ntri} is the electron current vector on the triangle, and J_{ptri} is the hole current vector on the triangle. In Atlas3D, electric field is combined with the z-component of the field at each corner of the prism to give an overall field modules at each node. To select this model, specify `E.VECTOR` of the **`IMPACT`** statement.

A simpler model can be selected by specifying `E.SIDE` on the **`IMPACT`** statement (see Equation 3-413).

$$G = \alpha_n(\left|E_{side}\right|)J_{nside} + \alpha_p(\left|E_{side}\right|)J_{pside} \quad \text{3-413}$$

Here, E_{side} is the scalar field along the side of a triangle, J_{nside} is the electron current along the side and J_{pside} is the hole current along the side. This model is the most non-physical but has the advantages of better robustness and calculation speed and is compatible with older device simulators.

The most complex and physically sound model is selected by specifying `E.DIR` on the `IMPACT` statement (see Equation 3-414).

$$G = \alpha_n\left(\left|\frac{|\bar{E}_{tri}| \cdot |\bar{J}_{ntri}|}{|\bar{J}_{ntri}|}\right|\right)|\bar{J}_{ntri}| + \alpha_p\left(\left|\frac{|\bar{E}_{tri}| \cdot |\bar{J}_{ptri}|}{|\bar{J}_{ptri}|}\right|\right)|\bar{J}_{ptri}| \quad 3\text{-}414$$

In this model, the ionization coefficients are a function of the field in the direction of the current. If the dot product of $\bar{E}$ and $\bar{J}$ is negative, then the field component is taken as 0. Consequently, impact ionization may only occur when a current is dominated by the drift term. This model is the most physically sound and is the default model for the field dependence of the impact ionization coefficients.

Another option is to abandon the use of the Electric field and adopt the gradient of the Quasi-Fermi levels to use when calculating the ionization coefficients.

$$G = \alpha_n(|\nabla\phi_n|)J_n + \alpha_p(|\nabla\phi_p|)J_p \quad 3\text{-}415$$

The modulus of the quasi-Fermi level gradients across each triangle are used, which is similar to the E.VECTOR electric field model. Using the gradient of quasi-Fermi level has the advantage that built-in electric fields (such as those existing across highly doped *n-p* junctions) do not result in an artificially high ionization rate at low contact biases. In the situation where the current is dominated by drift rather than diffusion, the gradient of quasi-fermi level will be essentially equal to the electric field. The impact ionization coefficients calculated from gradient of quasi-fermi level are in that case the same as the E.DIR or E.VECTOR options. To enable this model, specify GRADQFL on the **IMPACT** statement.

Local Electric Field Models for Impact Ionization

Selberherr's Impact Ionization Model

The ionization rate model proposed by Selberherr [286] is a variation of the classical Chynoweth model [62]. Activate this model by using the SELB parameter of the **IMPACT** statement, which is based upon the following expressions [325]:

$$\alpha_n = \mathrm{AN}\,exp\left[-\left(\frac{\mathrm{BN}}{E}\right)^{\mathrm{BETAN}}\right] \quad 3\text{-}416$$

$$\alpha_p = \mathrm{AP}\,exp\left[-\left(\frac{\mathrm{BP}}{E}\right)^{\mathrm{BETAP}}\right] \quad 3\text{-}417$$

Here, E is the electric field in the direction of current flow at a particular position in the structure and the parameters AN, AP, BN, BP, BETAN, and BETAP are defined on the **IMPACT** statement and have the default values shown in Table 3-94. In the case of AN, AP, BN, and BP you can define a value of electric field, EGRAN V/cm, where for electric fields, >EGRAN V/cm, the parameters are: AN1, AP1, BN1, BP1, while for electric fields, <EGRAN V/cm, the parameters become AN2, AP2, BN2, and BP2.

The AN and BN parameters are also a function of the lattice temperature in this model [198]. The temperature dependence of these coefficients is defined as follows:

$$\mathrm{AN} = AN_{1,2}\left(1 + \mathrm{A.NT}\left[\left(\frac{T_L}{300}\right)^{\mathrm{M.ANT}} - 1\right]\right) \quad 3\text{-}418$$

$$\mathtt{AP} = AP_{1,2}\left(1 + \mathtt{A.PT}\left[\left(\frac{T_L}{300}\right)^{\mathtt{M.APT}} - 1\right]\right) \quad 3\text{-}419$$

$$\mathtt{BN} = BN_{1,2}\left(1 + \mathtt{B.NT}\left[\left(\frac{T_L}{300}\right)^{\mathtt{M.BNT}} - 1\right]\right) \quad 3\text{-}420$$

$$\mathtt{BP} = BP_{1,2}\left(1 + \mathtt{B.PT}\left[\left(\frac{T_L}{300}\right)^{\mathtt{M.BPT}} - 1\right]\right) \quad 3\text{-}421$$

The parameters associated with these equations are shown in Table 3-95.

An alternative model for temperature dependence of AN and AP is given by the following expressions:

$$\mathtt{AN} = AN_{1,2} + \mathtt{CN2} \cdot \mathtt{T} + \mathtt{DN2} \cdot \mathtt{T}^2 \quad 3\text{-}422$$

$$\mathtt{AP} = AP_{1,2} + \mathtt{CP2} \cdot \mathtt{T} + \mathtt{DP2} \cdot \mathtt{T}^2 \quad 3\text{-}423$$

where T is temperature and CN2, CP2, DN2, and DP2 are user-specifiable parameters on the **IMPACT** statement. By default, the temperature model of Equations 3-418 and 3-419 are used and the values of CN2, CP2, DN2 and DP2 are all zero. You can use the temperature dependence models described in Equations 3-422 or 3-423 or both by specifying non-zero values for CN2, CP2, DN2 and DP2.

The critical fields given by BN and BP may be modeled based on band gap and optical phonon mean free paths using the following expressions:

$$\mathtt{BN} = \frac{E_g}{q\lambda_n^0} \quad 3\text{-}424$$

$$\mathtt{BP} = \frac{E_g}{q\lambda_p^0} \quad 3\text{-}425$$

where $q\lambda_n^0$ and $q\lambda_p^0$ are the optical phonon mean free paths for electrons and holes and E_g is the local temperature dependent band gap. The free paths are modeled using the following expressions:

$$\lambda_n^o = \mathtt{LAMDAH}\frac{tanh[q\mathtt{OPPHE}/2kT_L]}{tanh[q\mathtt{OPPHE}/2k300]} \quad 3\text{-}426$$

$$\lambda_n^o = \mathtt{LAMDAE}\frac{tanh[q\mathtt{OPPHE}/2kT_L]}{tanh[q\mathtt{OPPHE}/2k300]} \quad 3\text{-}427$$

where T is the lattice temperature and LAMDAE, LAMDAH, OPPHE are user-specifiable parameters listed in Table 3-96. To enable the models described by Equations 3-420, 3-421, 3-422, and 3-423, either specify $BN_{1,2}$ or $BP_{1,2}$ or both as zero.

Table 3-94 User-Definable Parameters in the Selberherr Impact Ionization Model

Statement	Parameter	Default
IMPACT	AN1	7.03×10^5 cm^{-1}
IMPACT	AN2	7.03×10^5 cm^{-1}
IMPACT	AP1	6.71×10^5 cm^{-1}
IMPACT	AP2	1.58×10^6 cm^{-1}
IMPACT	BN1	1.231×10^6V/cm
IMPACT	BN2	1.231×10^6 V/cm
IMPACT	BP1	1.693×10^6 V/cm
IMPACT	BP2	2.036×10^6 V/cm
IMPACT	BETAN	1.0
IMPACT	BETAP	1.0
IMPACT	EGRAN	4×10^5 V/cm

Table 3-95 Temperature Coefficient Parameters of the Selberherr Impact Ionization Model for Silicon in Equations 3-418 to 3-421

Statement	Parameter	Default
IMPACT	A.NT	0.588
IMPACT	B.NT	0.248
IMPACT	A.PT	0.588
IMPACT	B.PT	0.248
IMPACT	M.ANT	1.0
IMPACT	M.BNT	1.0
IMPACT	M.APT	1.0
IMPACT	M.BPT	1.0

Table 3-96 User specifiable parameters for the optical phonon mean free path model in Equations 3-426 and 3-427.			
Statement	**Parameter**	**Default**	**Units**
IMPACT	LAMDAE	6.2×10^{-7}	cm
IMPACT	LAMDAH	3.8×10^{-7}	cm
IMPACT	OPPHE	0.063	eV

Selberherr Tabular Model

Given a tabulation of ionization rate versus field or reciprocal field, as it more commonly appears, you can extract values of the A and B coefficients in Equations 3-415 and 3-416 assuming a value for BETA (in this case 1.0) for each consectutive pair of samples.

Atlas provides an interface to automate this process whereby you will provide a table of ionization rates versus fields or reciprocal fields and the Selberherr parameters are automatically derived and used to reproduce a representation of continuous ionization rates versus field to be used in the Selberherr model.

To use this feature, you must assign the values of IINOFF or IIPOFF or both to the name of a file containing an ASCII table of the ionization rates as a function of field for electrons or holes or both.

The table itself must first contain the integer number of samples (pairs) in the table. This is followed by that number of pairs of either field - ionization rate or reciprocal field - ionization rate values.

If reciprocal field is used, the designator flag "oe" must preceed the samples. If the "oe" designator flag is missing the ordinate values will be interpreted as field values.

The units of fields are V/cm. Ionization rates units are 1/cm. Reciprocal field units are cm/V.

Comments may be embedded in the file following a "#" character in the first column any row in the file.

The following is an example ionization file:

```
# Oguzman, I., Bellotti, E. and Brennan, K., "Theory of hole
initiated impact

# ionization in bulk zincblende and wurtzite GaN", J. Appl. Phys,
B. 81,

# 15 June 1997, pp. 7827-7834.

oe

5

2.465e-07 4.711e+04

2.840e-07 1.350e+04

3.335e-07 3.670e+03

4.011e-07 3.658e+02

5.012e-07 1.098e+01
```

In this case, we have some comments that are useful for tracing down the origin of the data followed by the reciprocal field designator flag.

Next, the number of samples is specified as 5. Finally, the five samples of reciprocal field ionization rate pairs are specified.

For this case, you might introduce this impact ionization characteristic for a region using the following statement:

```
MATERIAL REGION=3 IINOFF=mcclintockn.tbl
```

Using tabular ionization coefficients presents an issue when evaluating ionization coefficients outside of the range of the table. We provide two extrapolation schemes. The default specified by the parameter TBL.EXTRAP on the **IMPACT** statement implies that fields outside the range of the table uses the Selberherr coefficients from the nearest pair of samples.

Extrapolation is disabled by specifying "^TBL.EXTRAP" on the **IMPACT** statement. Once extrapolation is disabled, ionization coefficient values for fields larger than the peak field in the table will be set to the value of the ionization coefficient at the peak field. If the fields are less than the lowest field in the table, the ionization coefficient will be linearly interpolated with field between zero at zero field and the value of the tabulated ionization rate at the lowest tabulated field.

The difference between the extrapolation and non-extrapolation of tabulated ionization rates is illustrated using the example table above in Figure 3-8.

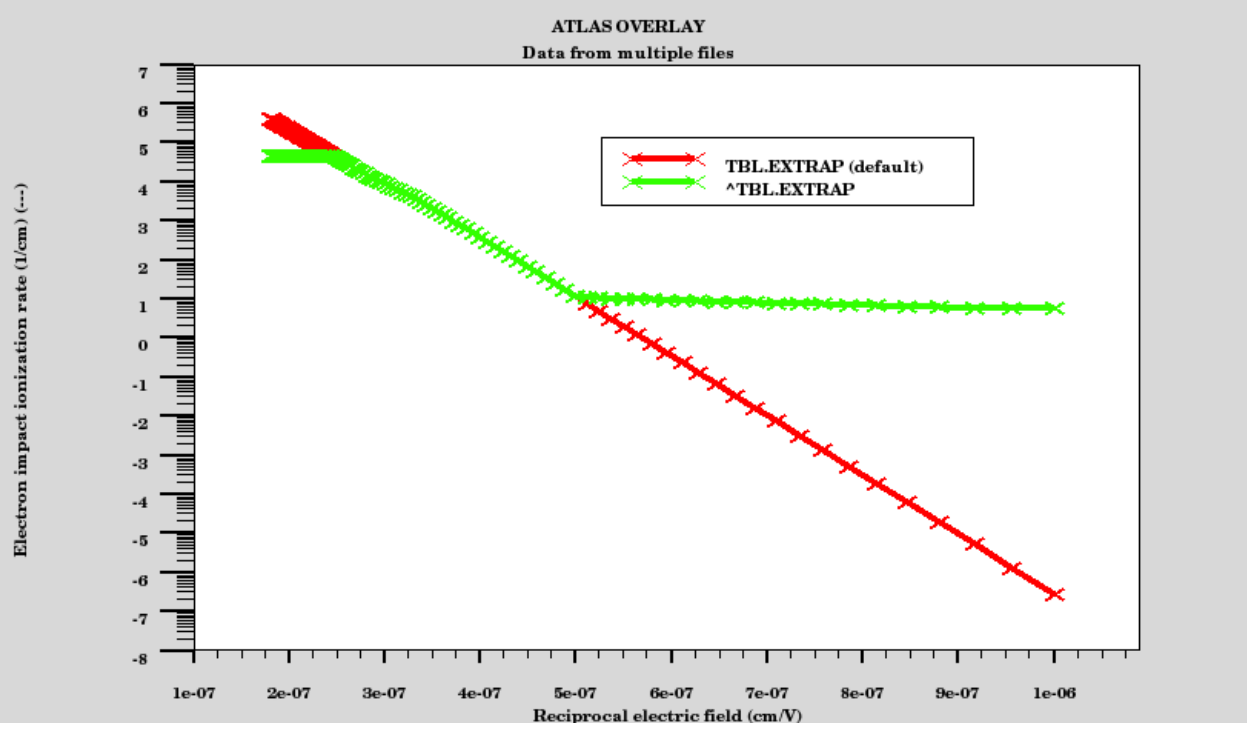


Figure 3-8: Calculated ionization rates for the tabulated Selberherr model with and without extrapolation

Van Overstraeten–de Man Impact Ionization Model

Based on the Chynoweth law [62], this model is very similar to the Selberherr model. The differences, however, are that the exponents BETAN and BETAP are set equal to unity. The lattice temperature dependence of the coefficients is also different. The functional forms of the ionization rate are

$$\alpha_n = \gamma_n \mathrm{AN} exp\left[\frac{-\gamma_n \mathrm{BN}}{E}\right] \quad 3\text{-}428$$

$$\alpha_p = \gamma_p \mathrm{AP} exp\left[\frac{-\gamma_p \mathrm{BP}}{E}\right] \quad 3\text{-}429$$

where the γ factors depend on Lattice temperature, or device temperature if GIGA is not enabled. They are calculated as

$$\gamma_n = \frac{tanh\left(\frac{\mathrm{N.VANHW}}{2k(300)}\right)}{tanh\left(\frac{\mathrm{N.VANHW}}{2kT}\right)} \quad 3\text{-}430$$

$$\gamma_p = \frac{tanh\left(\frac{\mathrm{P.VANHW}}{2k300}\right)}{tanh\left(\frac{\mathrm{P.VANHW}}{2kT}\right)} \quad 3\text{-}431$$

and will be unity if the device is at a uniform 300 K. The same parameters are used as in the Selberherr model but the BETAN and BETAP are fixed at unity. The other parameters are in Table 3-97. To activate the model, use the VANOVERS parameter on the **IMPACT** statement.

Table 3-97 van Overstraeten–de Man model Parameters

Statement	Parameter	Default	Units
IMPACT	N.VANHW	0.063	eV
IMPACT	P.VANHW	0.063	eV
IMPACT	VANOVERS	False	--

Valdinoci Impact Ionization Model

Valdinoci et al. [321] reported on a measurement technique to calibrate the temperature dependence of impact ionization models and proposed a new model based on their study. Their model of both electron and hole ionization coefficients was calibrated for silicon for temperature ranging from 25 to 400°C. This model is based on the following:

$$\alpha_{n,p} = \frac{E}{a_{n,p}(T_L) + b_{n,p}(T_L) exp[d_{n,p}(T_L)/(E + c_{n,p}(T_L))]} \quad 3\text{-}432$$

where E is the electric field along the current flow lines. The parameters in Equation 3-432 depend on temperature as follows:

$$a_n(T_L) = \mathtt{VAL.AN0} + \left(\mathtt{VAL.AN1} \cdot T_L^{\mathtt{VAL.AN2}}\right) \quad 3\text{-}433$$

$$b_n(T_L) = \mathtt{VAL.BN0}\,exp\left(\mathtt{VAL.BN1} \cdot T_L\right) \quad 3\text{-}434$$

$$c_n(T_L) = \mathtt{VAL.CN0} + \left(\mathtt{VAL.CN1} \cdot T_L^{\mathtt{VAL.CN2}}\right) + \mathtt{VAL.CN3} \cdot T_L^2 \quad 3\text{-}435$$

$$d_n(T_L) = \mathtt{VAL.DN0} + \mathtt{VAL.DN1} \cdot T_L + \mathtt{VAL.DN2} \cdot T_L^2 \quad 3\text{-}436$$

$$a_p(T_L) = \mathtt{VAL.AP0} + \left(\mathtt{VAL.AP1} \cdot T_L^{\mathtt{VAL.AP2}}\right) \quad 3\text{-}437$$

$$b_p(T_L) = \mathtt{VAL.BP0}\,exp\left(\mathtt{VAL.BP1} \cdot T_L\right) \quad 3\text{-}438$$

$$c_p(T_L) = \mathtt{VAL.CP0} + \left(\mathtt{VAL.CP1} \cdot T_L^{\mathtt{VAL.CP2}}\right) + \mathtt{VAL.CP3} \cdot T_L^2 \quad 3\text{-}439$$

$$d_p(T_L) = \mathtt{VAL.DP0} + \mathtt{VAL.DP1} \cdot T_L + \mathtt{VAL.DP2} \cdot T_L^2 \quad 3\text{-}440$$

Our default parameters for silicon are shown in Table 3-98. These parameters are also taken from Valdinoci et al. [321]. To enable this model, specify the logical parameter, VALDINOCI, in the **IMPACT** statement.

Table 3-98 Default Parameters for Valdinoci Impact Ionization Model

Statement	Parameter	Type	Default	Units
IMPACT	VAL.AN0	Real	4.3383	
IMPACT	VAL.AN1	Real	-2.42×10^{-12}	
IMPACT	VAL.AN2	Real	4.1233	
IMPACT	VAL.BN0	Real	0.235	
IMPACT	VAL.BN1	Real	0.0	
IMPACT	VAL.CN0	Real	1.6831×10^{4}	
IMPACT	VAL.CN1	Real	4.3796	

Table 3-98 Default Parameters for Valdinoci Impact Ionization Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
IMPACT	VAL.CN2	Real	1.0	
IMPACT	VAL.CN3	Real	0.13005	
IMPACT	VAL.DN0	Real	1.233735×10^{6}	
IMPACT	VAL.DN1	Real	1.2039×10^{3}	
IMPACT	VAL.DN2	Real	0.56703	
IMPACT	VAL.AP0	Real	2.376	
IMPACT	VAL.AP1	Real	0.01033	
IMPACT	VAL.AP2	Real	1.0	
IMPACT	VAL.BP0	Real	0.17714	
IMPACT	VAL.BP1	Real	-0.002178	
IMPACT	VAL.CP0	Real	0.0	
IMPACT	VAL.CP1	Real	0.00947	
IMPACT	VAL.CP2	Real	2.4924	
IMPACT	VAL.CP3	Real	0.0	
IMPACT	VAL.DP0	Real	1.4043×10^{6}	
IMPACT	VAL.DP1	Real	2.9744×10^{3}	
IMPACT	VAL.DP2	Real	1.4829	

Zappa's Model for Ionization Rates in InP

A model for temperature and field dependent ionization rates in InP was proposed by Zappa et. al. [376]. To enable this model, specify ZAPPA in the **IMPACT** statement. The model is described for electrons and holes by Equations 3-441 through 3-446:

$$\alpha_n = \frac{qE}{\mathrm{ZAP.AN}} exp\left\{0.217\left(\frac{\mathrm{ZAP.AN}}{E_n}\right)^{1.14} - \left[\left(0.217\left(\frac{\mathrm{ZAP.AN}}{E_n}\right)^{1.14}\right)^2 + \left(\frac{\mathrm{ZAP.AN}}{qE\lambda_n}\right)^2\right]^{0.5}\right\} \quad 3\text{-}441$$

$$\lambda_n = \mathrm{ZAP.BN}\, tanh\left(\frac{\mathrm{ZAP.CN}}{2kT}\right) \quad 3\text{-}442$$

$$E_n = \mathrm{ZAP.DN}\, tanh\left(\frac{\mathrm{ZAP.EN}}{2kT}\right) \quad 3\text{-}443$$

$$\alpha_p = \frac{qE}{\mathrm{ZAP.AP}} exp\left\{0.217\left(\frac{\mathrm{ZAP.AP}}{E_p}\right)^{1.14} - \left[\left(0.217\left(\frac{\mathrm{ZAP.AP}}{E_p}\right)^{1.14}\right)^2 + \left(\frac{\mathrm{ZAP.AP}}{qE\lambda_p}\right)^2\right]^{0.5}\right\} \quad 3\text{-}444$$

$$\lambda_p = \mathrm{ZAP.BP}\, tanh\left(\frac{\mathrm{ZAP.CP}}{2kT}\right) \quad 3\text{-}445$$

$$E_p = \mathrm{ZAP.DP}\, tanh\left(\frac{\mathrm{ZAP.EP}}{2kT}\right) \quad 3\text{-}446$$

where E is the local electric field and T is the local temperature. Table 3-99 describes the user-specifiable parameters for Zappa's model.

Table 3-99 User-Specifiable Parameters for Zappa's Ionization Rate Model

Statement	Parameter	Type	Default	Units
IMPACT	ZAP.AN	Real	1.9	eV
IMPACT	ZAP.BN	Real	41.7	angstroms
IMPACT	ZAP.CN	Real	46.0	meV
IMPACT	ZAP.DN	Real	46.0	meV
IMPACT	ZAP.EN	Real	46.0	meV
IMPACT	ZAP.AP	Real	1.4	eV
IMPACT	ZAP.BP	Real	41.3	angstroms
IMPACT	ZAP.CP	Real	36.0	meV
IMPACT	ZAP.DP	Real	36.0	meV
IMPACT	ZAP.EP	Real	36.0	meV

Grant's Impact Ionization Model

The second ionization model has the same form as the Selberherr model but a simpler implementation where:

$$\alpha_n = \mathrm{AN}\, exp\left[-\left(\frac{\mathrm{BN}}{E}\right)\right] \quad 3\text{-}447$$

$$\alpha_p = \mathrm{AP}\, exp\left[-\left(\frac{\mathrm{BP}}{E}\right)\right] \quad 3\text{-}448$$

This implementation has three key differences:

- The model has a low-field, an intermediate field and a high field region.
- The coefficients for silicon are different.
- There is no temperature dependence.

This model was developed after investigations by Baraff [22] suggested the existence of a low, intermediate and high field response region for electron and hole ionization rates. The coefficients implemented into this model match the experimental data of Grant [106], which suggested that the three different regions existed.

This model is activated with the GRANT parameter of the **IMPACT** statement. The model parameters: AN, AP, BN, and BP aren't user-definable. Instead, the three electric field regions have in-built values as follows:

1) **Low Electric Field** $E < 2.4\times10^5$ V/cm 3-449

$AN = 2.6\times10^6$ $AP = 2.0\times10^6$

$BN = 1.43\times10^6$ $BN = 1.97\times10^6$

2) **Intermediate Electric Field** $2.4\times10^5 > E > 5.3\times10^5$ *V/cm* 3-450

$AN = 6.2\times10^5$ $AP = 2.0\times10^6$

$BN = 1.08\times10^6$ $BP = 1.97\times10^6$

3) **High Electric Field** $E > 5.3\times10^5$ *V/cm* 3-451

$AN = 5.0\times10^5$ $AP = 5.6\times10^5$

$BN = 9.9\times10^6$ $BP = 1.32\times10^6$

Crowell-Sze Impact Ionization Model

Crowell and Sze [69] have proposed a more physical relationship between the electric field and the ionization rates. This model represents ionization coefficients as follows:

$$\alpha_{n,p} = \frac{1}{\lambda_{n,p}}\exp[C_o(r) + C_1(r)x + C_2(r)x^2] \quad 3\text{-}452$$

$$C_0 = -1.92 + 75.5r - 75.7r^2 \quad 3\text{-}453$$

$$C_1(r) = -1.75\times10^{-2} - 11.9r + 46r^2 \quad 3\text{-}454$$

$$C_2(r) = 3.9\times10^{-4} - 1.17r + 11.5r^2 \quad 3\text{-}455$$

where:

$$r = \frac{\mathrm{OPPHE}}{E_i};\quad x = \frac{E_i}{q\lambda_{n,p}E} \quad 3\text{-}456$$

$$E_i = \begin{cases} 1.1eV & for\ electrons \\ 1.8eV & for\ holes \end{cases} \quad 3\text{-}457$$

$$\lambda_n^o = \mathrm{LAMDAE}\frac{tanh[q\mathrm{OPPHE}/2kT_L]}{tanh[q\mathrm{OPPHE}/2k300]} \quad 3\text{-}458$$

$$\lambda_n^o = \text{LAMDAH}\frac{tanh[qE_r/2kT_L]}{tanh[qE_r/2k300]} \quad \text{3-459}$$

The Crowell-Sze Model for impact ionization is selected by setting the CROWELL parameter of the **IMPACT** statement.

Table 3-100 Crowell-Sze Impact Ionization Model Parameters.

Statement	Parameter	Default
IMPACT	LAMDAE	6.2×10^{-7}cm
IMPACT	LAMDAH	3.8×10^{-7} cm

Okuto-Crowell Impact Ionization Model

This is another empirical model with temperature dependent coefficients where the ionization coefficients take the following forms:

$$\alpha_n = \text{ANOKUTO}(1 + \text{CNOKUTO}(T - 300)) \cdot F^{\text{NGAMOKUTO}} \quad \text{3-460}$$

$$\times exp\left(-\left(\frac{\text{BNOKUTO}(1 + \text{DNOKUTO}(T - 300))}{F}\right)^{\text{NDELOKUTO}}\right)$$

$$\alpha_p = \text{APOKUTO}(1 + \text{CPOKUTO}(T - 300)) \cdot F^{\text{PGAMOKUTO}} \quad \text{3-461}$$

$$\times exp\left(-\left(\frac{\text{BPOKUTO}(1 + \text{DPOKUTO}(T - 300))}{F}\right)^{\text{PDELOKUTO}}\right)$$

where F is the effective electric field.

The parameters are optimized to fit in the electric field range 10^5 to 10^6 V/cm but the same ones are used for the whole range of field. It uses a relatively large set of parameters and so you can easily adjust the default values (applicable to silicon) to fit other materials [226].

To enable the model, use the OKUTO parameter on the **IMPACT** statement. The adjustable model parameters are given in Table 3-101.

Table 3-101 Okuto-Crowell Model Parameters

Statement	Parameter	Default	Units
IMPACT	ANOKUTO	0.426	/V
IMPACT	APOKUTO	0.243	/V
IMPACT	BNOKUTO	4.81×10^5	V/cm
IMPACT	BPOKUTO	6.53×10^5	V/cm
IMPACT	CNOKUTO	3.05×10^{-4}	/K
IMPACT	CPOKUTO	5.35×10^{-4}	/K

Table 3-101 Okuto-Crowell Model Parameters			
Statement	**Parameter**	**Default**	**Units**
IMPACT	DNOKUTO	6.86×10^{-4}	/K
IMPACT	DPOKUTO	5.67×10^{-4}	/K
IMPACT	NGAMOKUTO	1.0	
IMPACT	PGAMOKUTO	1.0	
IMPACT	NDELOKUTO	2.0	
IMPACT	PDELOKUTO	2.0	

Note: If NGAMOKUTO or PGAMOKUTO is different from 1.0, the units of ANOKUTO or APOKUTO will be different from /V.

Lackner Impact Ionization Model

This is another, more recent, impact ionization model based on Chynoweth's law [169]. It is similar to the van Overstraeten - de Man model but with an extra factor Z, dividing the ionization rate. The factor Z and the ionization coefficients are

$$Z = 1 + \frac{\mathrm{GAMMAN.BNLACKNER}}{F}\exp\left(-\frac{\mathrm{GAMMAN.BNLACKNER}}{F}\right) + \frac{\mathrm{GAMMAP.BPLACKNER}}{F}\exp\left(-\frac{\mathrm{GAMMAP.BPLACKNER}}{F}\right) \quad (3\text{-}462)$$

$$\alpha_n = \frac{\mathrm{GAMMAN.ANLACKNER}}{Z}\exp\left(-\frac{\mathrm{GAMMAN.BNLACKNER}}{F}\right) \quad (3\text{-}463)$$

$$\alpha_p = \frac{\mathrm{GAMMAP.APLACKNER}}{Z}\exp\left(-\frac{\mathrm{GAMMAP.BPLACKNER}}{F}\right) \quad (3\text{-}464)$$

$$\mathrm{GAMMAN} = \frac{\tanh\left(\frac{\mathrm{N.LACKHW}}{2k300}\right)}{\tanh\left(\frac{\mathrm{N.LACKHW}}{2kT}\right)},\ \mathrm{GAMMAP} = \frac{\tanh\left(\frac{\mathrm{P.LACKHW}}{2k300}\right)}{\tanh\left(\frac{\mathrm{P.LACKHW}}{2kT}\right)} \quad (3\text{-}465)$$

To enable the model, use the LACKNER parameter on the **IMPACT** statement. The adjustable model parameters are given in Table 3-102.

Table 3-102 LACKNER Model Parameters

Statement	Parameter	Default	Units
IMPACT	ANLACKNER	1.316×10^6	/cm
IMPACT	APLACKNER	1.818×10^6	/cm
IMPACT	BNLACKNER	1.474×10^6	V/cm
IMPACT	BPLACKNER	2.036×10^6	V/cm
IMPACT	N.LACKHW	0.063	eV
IMPACT	P.LACKHW	0.063	eV

Non-Local Carrier Energy Models for Impact Ionization

All local electric field based models will normally overestimate the rate of impact ionization. This occurs because lucky electron theory inherently assumes that a carrier is traveling through a constant electric field E. As a result it will predict a distance $\Delta x=E_i/qE$ over which the carrier will gain the ionization energy E_i. In real devices, however, the electric field is never constant but is normally sharply peaked at metallurgical junctions. Therefore, as a carrier passes through the peaked electric field the lucky electron model will predict the ionization distance Δx to be too small. As a result the ionization rate is overestimated. The effect of this is that all the simulated breakdown voltages will be underestimated and substrate currents overestimated.

The Energy Balance Transport Model can be used to improve the simulation of impact ionization by implementing ionization models based upon the carrier temperature instead of the electric field. The carrier temperature is a more meaningful basis as the velocity-field relationship is more closely modeled. This allows a non-local dependence on the electric field within the impact ionization model. Energy Balance Transport Models will therefore result in more accurate simulations of breakdown voltage and substrate current. Two different impact ionization models have been implemented into Atlas, the first is based upon the classical Chynoweth relationship, modified to include carrier temperature, but the second is a more advanced non-Maxwellian approach based upon carrier temperatures.

When the energy balance transport model is applied, only two impact ionization models are available. These are the Toyabe model and the Concannon model.

Toyabe Impact Ionization Model

The temperature dependent impact ionization model is founded on the Selberherr model and is similar to that suggested by Toyabe [149]. The carrier temperature is used to calculate an effective electric field based upon the homogeneous temperature-field relation. To maintain self-consistency within the energy balance transport model this is the same relationship used for the effective electric field within the carrier temperature dependent mobility. This model is the default model for impact ionization with energy balance transport and is activated with the TOYABE parameters on the **IMPACT** statement. The ionization rates now have the form:

$$\alpha_n = \mathrm{AN} exp\left(\frac{-\mathrm{BN}}{E_{eff,\,n}}\right) \quad 3\text{-}466$$

$$\alpha_p = \mathrm{AP} exp\left(\frac{-\mathrm{BP}}{E_{eff,\,p}}\right) \quad 3\text{-}467$$

where the model parameters: AN, AP, BN, and BP are user-definable in the **IMPACT** statement. The AN, AP, BN, and BP parameters are set by the AN1, AP1, BP1, and BN1 parameters that have the same default values as listed in Table 3-94. The parameters EGRAN, AN2, AP2, BN2, and BP2 are not used in the Toyabe model.

The effective electric field is calculated according to:

$$E_{eff,\,n} = \frac{3}{2}\frac{kT_n}{q\mathrm{LREL.EL}} \quad 3\text{-}468$$

$$E_{eff,\,p} = \frac{3}{2}\frac{kT_p}{q\mathrm{LREL.HO}} \quad 3\text{-}469$$

where the energy relaxation lengths, LREL.EL and LREL.HO, can be explicitly defined in the IMPACT statement, or can be calculated according to:

```
LREL.EL = VSATN* TAUSN
```
3-470

```
LREL.HO = VSATP* TAUSP
```
3-471

VSATN and VSATP are the saturation velocities for electrons and holes, and TAUSN and TAUSP correspond to the electron energy relaxation times (TAUREL.EL and TAUREL.HO) in Equations 3-339 and 3-340. You must set the LENGTH.REL flag to use the values of LREL.EL and LREL.HO specified in the **IMPACT** statement.

Table 3-103 User-Specifiable Parameters for Equations 3-468-3-471

Statement	Parameter	Units
IMPACT	LREL.EL	μm
IMPACT	LREL.HO	μm
MATERIAL	VSAT	cm/s
MATERIAL	VSATN	cm/s
MATERIAL	VSATP	cm/s
IMPACT	TAUSN	s
IMPACT	TAUSP	s

Note: As an added level of flexibility, the relaxation times used for the energy balance equation and those used in the impact ionization model are separated into two user-definable parameters. In contrast to TAUREL.EL

and TAUREL.HO, which are used in different formulae, the TAUSN and TAUSP parameters are only applicable in the impact ionization expression in Equations 3-468 and 3-469. By default, TAUREL.EL=TAUSN and TAUREL.HO=TAUSP.

It can also be argued that the AN, AP, BN, and BP parameters should also be a function of the carrier temperature. But, no clear theoretical basis for this has been proposed and accepted. Instead, the C-Interpreter within Atlas has been extended to include two C-Interpreter functions. These functions are specified via the F.EDIIN and F.EDIIP parameters of the **IMPACT** statement. These parameters specify the filename of a text file containing a C-Interpreter function that describes the dependence of the model parameters AN, AP, BN and BP as a function of the carrier temperatures. These values will then be used within Toyabe's energy dependent impact ionization model.

Concannon's Impact Ionization Model

The previous non-local impact ionization model inherently assumes a Maxwellian shape to the distribution of hot carriers. Recent work by Fiegna et. al. [87] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the energy distribution function. To accurately model these effects, a non-Maxwellian based model from Concannon [66] has been implemented. Based upon this energy distribution the model calculates the probability of a carrier having sufficient energy to cause impact ionization. The model results show good agreement with measured results for a 0.9µm flash EPROM device [66].

To enable the Concannon substrate current for the electron and hole continuity equations, specify the N.CONCANNON and P.CONCANNON parameters in the **IMPACT** statement.

The generation rate is a function of the carrier temperature and concentration is given by:

$$G_n(x,y) = \text{CSUB.N} \times n \int_{\text{ETH.N}}^{\infty} F(\varepsilon, T_n(x,y)))d\varepsilon \quad \text{3-472}$$

$$G_p(x,y) = \text{CSUB.P} \times p \int_{\text{ETH.P}}^{\infty} F(\varepsilon, T_p(x,y)))d\varepsilon \quad \text{3-473}$$

where $n(x,y)$ and $p(x,y)$ are the electron and hole carrier concentrations within the semiconductor, ε is energy, $T_n(x,y)$ and $T_p(x,y)$ are the electron and hole carrier temperatures in the semiconductor, F is given in Equation 3-474, CSUB.N, CSUB.P, ETH.N, and ETH.P are user-specifiable parameters as given in Table 3-104.

Table 3-104 User-Specifiable Parameters for Equations 3-472 and 3-473

Statement	Parameter	Default	Units
IMPACT	CSUB.N	2.0×10^{14}	

Table 3-104 User-Specifiable Parameters for Equations 3-472 and 3-473			
Statement	**Parameter**	**Default**	**Units**
IMPACT	CSUB.P	4.0×10^{14}	
IMPACT	ETH.N	1.8	eV
IMPACT	ETH.P	3.5	eV

The function, $F(\varepsilon,T)$, in Equations 3-472 and 3-473 is given by the product of the density of states function, g(ε), and the energy distribution function $f(\varepsilon)$ as:

$$F = \frac{g(\varepsilon)f(\varepsilon)}{\int_{\circ}^{\infty} g(\varepsilon)f(\varepsilon)} \quad 3\text{-}474$$

The density of states function is given by:

$$g(\varepsilon) = \varepsilon^{1.25} \quad 3\text{-}475$$

The energy distribution functions for electrons and holes are:

$$f_n(\varepsilon) = \left[exp\left(\frac{-\mathtt{CHIA}\,\varepsilon^3}{T_n^{1.5}}\right) + \mathtt{C0}\,exp\left(\frac{-\mathtt{CHIB}\,\varepsilon^3}{T_n^{1.5}}\right)\right] \quad 3\text{-}476$$

$$f_p(\varepsilon) = exp\left(\frac{-\mathtt{CHI.HOLES}\,\varepsilon^3}{T_p^{1.5}}\right) \quad 3\text{-}477$$

Here, ε is energy, $T_{n,p}$ are the carrier temperatures, CHIA, CHIB, and C0 are user-specifiable parameters (see Table 3-105).

Table 3-105 User-Definable Parameters for the Energy Distribution Functions			
Statement	**Parameter**	**Default**	**Units**
IMPACT	CHIA	3.0×10^{5}	
IMPACT	CHIB	5.0×10^{4}	
IMPACT	C0	2.5×10^{-10}	
IMPACT	CHI.HOLES	4.6×10^{4}	

Two other parameters in the **IMPACT** statement are user-definable, which may affect the result of the numeric integration. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter

sets the upper limit of the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

Note: To maintain self-consistent results, implement this model if the Concannon model is being used for the simulation of gate current.

Other Hydrodynamic Models

You can also use the Lackner, Okuto-Crowell, and van Overstraeten - de Man models with the hydrodynamic model. To use this model, set the command line flag -ISE instead of the Atlas default model (Equations 3-468 and 3-469). The Atlas default model assumes the effective field used to calculate impact ionization rate is directly proportional to carrier Temperature.

A more physical relationship between effective field and carrier temperature is given by

$$E_{eff,n} = \frac{1.5k_B(T_n - T_L)}{\texttt{TAUREL.EL NISELAMBDA VSATN}} + \alpha_n(E_{eff,n})\texttt{N.UPSILON}(E_g(T_L) + \texttt{NISEDELTA}k_BT_n) \quad \text{3-478}$$

$$E_{eff,p} = \frac{1.5k_B(T_p - T_L)}{\texttt{TAUREL.HO PISELAMBDA VSATP}} + \alpha_p(E_{eff,p})\texttt{P.UPSILON}(E_g(T_L) + \texttt{PISEDELTA}k_BT_p) \quad \text{3-479}$$

The parameters N.UPSILON and P.UPSILON can be either 0(unset) or 1(set). The default for Silicon is 1. It is 0 for other materials. In the case where UPSILON is unity, the equation becomes a transcendental one because it contains α as a function of E_{eff}. In this case, the equation is solved iteratively in order to obtain α. If no solution is found, then an error message will output and you must set N.UPSILON=0 or P.UPSILON=0. The user adjustable parameters are given in Table 3-106.

Table 3-106 Hydrodynamic Model Parameters

Statement	Parameter	Default	Units
IMPACT	N.UPSILON	True (in silicon)	False (otherwise)
IMPACT	P.UPSILON	True (in silicon)	False (otherwise)
IMPACT	NISELAMBDA	1.0	
IMPACT	PISELAMBDA	1.0	
IMPACT	NISEDELTA	1.5	
IMPACT	PISEDELTA	1.5	

Solutions for Wide Bandgap Semiconductors

In models of wide bandgap semiconductors, the drift-diffusion current may be so small that the simulated impact ionization carrier generation rates remain so low that breakdown does not occur at the expected bias state. In reality, impact ionization is a non-local process and is caused by hot carriers in the non-Maxwellian tail of the energy distribution function.

The drift-diffusion model fails to correctly model these effects, although you can simulate the impact ionization breakdown. This is done by putting a minimum value on the current densities used in Equation 3-413 to calculate the impact generation rate. You set a minimum current density by specifying the values of its directional components with the JNX.MIN, JNY.MIN, JNZ.MIN, JPX.MIN, JPY.MIN, and JPZ.MIN parameters on the **IMPACT** statement. The drift-diffusion current density along any side has the current density calculated from the above parameters added to it before being used in Equation 3-413.

This minimum current density causes the impact ionization breakdown to be bootstrapped when the ionization coefficients become sufficiently large. The simulated currents below the breakdown will depend on the particular values chosen for the minimum current and will be unphysical. As breakdown occurs, however, the generated current densities will significantly exceed the imposed minimum currents, and the simulation of the breakdown should be physically realistic.

This model is only available with the E.SIDE option at present. It is targeted at wide bandgap semiconductors but can also be used in devices where the current densities are too noisy or too low to reliably simulate breakdown by impact ionization.

Table 3-107 Parameters for Specifying Minimum Current Values

Statement	Parameter	Default	Units
IMPACT	JNX.MIN	0.0	A/cm^2
IMPACT	JNY.MIN	0.0	A/cm^2
IMPACT	JNZ.MIN	0.0	A/cm^2
IMPACT	JPX.MIN	0.0	A/cm^2
IMPACT	JPY.MIN	0.0	A/cm^2
IMPACT	JPZ.MIN	0.0	A/cm^2

3.6.5 Geiger Mode Simulation

Geiger mode simulation is the calculation of the single photon, single electron, or single hole probability of avalanche breakdown usually in devices that are cooled sufficiently to neglect thermal generation of carriers. These calculations are done using line integrals suggested by McIntyre [206] of ionization rates along paths of steepest potential gradient. These calculations are done as a post-processing step after convergence is obtained without any impact ionization or thermal generation models being turned on.

To enable the post processing, you should specify GEIGER on the **MODELS** statement. The next three different calculations can be done. First, you can probe the value of the probability of avalanche due to introduction of an electron, hole, or electron-hole pair by specifying PE.AVALANCE, PH.AVALANCHE, or PP.AVALANCE on the **PROBE** statement. This should be

accompanied by the location of the probe using the X and Y parameters of the **PROBE** statement. Second, if you specify a value for the FILENAME parameter of the **PROBE** statement, the line integral through the specified point will also be outputted to the specified file. The quantities which you can view in this file are

- The path coordinates, x position as a function of y or y position as a function of x. (The path passes through the PROBE point)
- The ionization probabilities for all starting points along the path
- The ionization coefficients at all points along the path
- The electric field and velocity at all points along the path
- The transit time for electrons and holes to go between the PROBE position and every other point on the path.

Finally, once GEIGER is specified on the **MODELS** statement, all structure files saved thereafter will contain the spatial distribution of the probabilities of electron, hole, and pair generated avalanche breakdown.

In reference [206], two solution methods are presented. One of them makes the simplifying assumption that the ratio $\alpha_n(|\vec{E}|)/\alpha_p(|\vec{E}|)$ is field independent. This is the default and you can also explicitly choose it by setting the **GEI.METH** parameter to 0 on the **METHOD** statement. The other solution method is more general because it makes no assumptions about the field dependence of the ratio $\alpha_n(|\vec{E}|)/\alpha_p(|\vec{E}|)$. You can select it by setting **GEI.METH** to 1 on the **METHOD** statement.

The paths on which the line integrals are performed are found by following field lines from the positions at which the carriers are introduced.The calculation of the carrier trajectories terminates when the carrier either reaches a contact or the electric field falls below a certain threshold value. You can set the value of this threshold field using the GEIGER.MINFIELD parameter on the **MODELS** statement. The default value of this parameter is 1 V/cm.

Table 3-108 User Specified Parameters and Flags for the GEIGER Model

Statement	Parameter	Default	Units
MODELS	GEIGER	False	
MODELS	GEIGER.MINFIELD	1.0	V/cm
PROBE	PE.AVALANCHE	False	
PROBE	PH.AVALANCHE	False	
PROBE	PP.AVALANCHE	False	
PROBE	FILENAME	NULL	
METHOD	GEI.METH	0	

3.6.6 Band-to-Band Tunneling

Standard Models

If a sufficiently high electric field exists within a device local band bending may be sufficient to allow electrons to tunnel, by internal field emission, from the valence band into the conduction band. An additional electron is therefore generated in the conduction band and a hole in the valence band. This generation mechanism is implemented into the right-hand side of the continuity equations. The tunneling generation rate is [132,133,134,157] as:

$$G_{BBT} = D \ \mathtt{BB.A} \ E^{\mathtt{BB.GAMMA}} exp\left(-\frac{\mathtt{BB.B}}{E}\right) \quad \text{3-480}$$

where E is the magnitude of the electric field, D is a statistical factor, and BB.A, BB.B, and BB.GAMMA are user-definable parameters. In Atlas, there are three different sets of values that may be applied to the model parameters.

The model parameters can be set to the standard model [132] by specifying BBT.STD on the **MODELS** statement. The parameter defaults for the standard model are as follows:

BB.A = 9.6615e18 $cm^{-1} V^{-2} s^{-1}$ BB.B= 3.0e7 V/cm BB.GAMMA= 2.0

The model parameters may also be set to the Klaassen Model [132,157,133] by specifying BBT.KL on the **MODELS** statement. The parameter defaults for the Klaassen model are as follows:

BB.A = 4.00e14 $cm^{-1/2} V^{-5/2} s^{-1}$ BB.B = 1.9e7 V/cm BB.GAMMA= 2.5

In application, use the standard model with direct transitions while using the Klaassen model with indirect transitions.

Another modification allows these model parameters to be calculated from the first principles by specifying the AUTOBBT parameter in the **MODELS** statement. In this case, the parameters are calculated according to

$$\mathtt{BB.A} = \frac{q^2\sqrt{(2 \times \ \mathtt{MASS.TUNNEL}\ m_0)}}{h^2\sqrt{\mathtt{EG300}}} \quad \text{3-481}$$

$$\mathtt{BB.B} = \frac{\pi^2\ \mathtt{EG300}^{\frac{3}{2}}\sqrt{\frac{\mathtt{MASS.TUNNEL}\ m_0}{2}}}{qh} \quad \text{3-482}$$

$$\mathtt{BB.GAMMA} = 2 \quad \text{3-483}$$

where q is the electronic charge, h is Planck's constant, E_g is the energy bandgap, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. The parameter MASS.TUNNEL may be set on the **MODELS** statement and the bandgap at 300K, EG300, is defined on the **MATERIAL** statement.

An alternative expression to Equation 3-554 for BB.B is used if the flag AUTOBBT2 is set on the **MODELS** statement. This is

$$\mathtt{BB.B} = \frac{8\pi\sqrt{2m_o\mathtt{MASS.TUNNEL}}\ \mathtt{EG300}^{3/2}}{3qh} \quad \text{3-484}$$

The value of BB.A is the same for the AUTOBBT parameter as with the AUTOBBT2 parameter.

The default value of the statistical factor D is 1. This factor can be calculated as suggested by Hurkx et al [132]:

$$D = \frac{n_{ie}^2 - np}{(n + n_{ie})(p + n_{ie})} \quad \text{3-485}$$

To enable this modification, specify BBT.HURKX in the **MODELS** statement. If BBT.DEHURKX is specified in the **MODELS** statement, then D will be set to 0 if

$$D = \begin{cases} 0 & \text{if } \nabla\phi n \cdot E > 0 \text{ and } \nabla\phi p \cdot E > 0 \\ \dfrac{n_{ie}^2 - np}{(n + n_{ie})(p + n_{ie})} & \text{otherwise} \end{cases} \quad \text{3-486}$$

If BBT.DJHURKX is specified in the **MODELS** statement, then D will be set to 1 if

$$D = \begin{cases} 1 & j_n > 1 \times 10^{-3} q n_{ie} \text{VSATN and} \\ & j_p > 1 \times 10^{-3} q n_{ie} \text{VSATP} \\ \dfrac{n_{ie}^2 - np}{(n + n_{ie})(p + n_{ie})} & \text{otherwise} \end{cases} \quad \text{3-487}$$

If the BBT.ALPHA parameter is specified in the **MATERIAL** statement, then the statistical factor will be given by

$$D = \frac{n_{ie} - np}{(n + n_{ie})(p + n_{ie})} \quad (1 - |\text{BBT.ALPHA}|) - \text{BBT.ALPHA} \quad \text{3-488}$$

where

- BBT.ALPHA=0 corresponds to the original Hurkx model,
- BBT.ALPHA=1 corresponds to recombination only, and
- BBT.ALPHA=-1 corresponds to generation only.

If BBT.SHURKX is specified in the **MODELS** statement, then the statistical factor will be taken from Schenk et al [280].

Table 3-109 User-Definable Parameters in the Band-to-Band Tunneling Model

Statement	Parameter	Default	Units
MODELS	BBT.ALPHA	0	
MODELS	BB.A	4.0×10^{14}	$cm^{-1/2}\ V^{-5/2}\ s^{-1}$
MODELS	BB.B	1.9×10^{7}	V/cm
MODELS	BB.GAMMA	2.5	

Schenk Band to Band Tunneling Model

A comprehensive study of band-to-band tunneling was carried out by Schenk [280]. A rigorous theory was developed and then an approximate result suitable for use in device simulations was derived. The result shows that phonon assisted band-to-band tunneling rate is generally dominant compared to the band-to-band tunneling that doesn't involve a phonon scattering event. The direct band-to-band tunneling is thus neglected. The model also assumes that the electric field is constant over the tunneling length. Therefore, it is a local model.

The recombination-generation rate is given by

$$G_{BBT}^{SCHENK} = A.BBT.SCHENK F^{7/2} S \left(\frac{(A^{\mp})^{-3/2} exp\left(\frac{A^{\mp}}{F}\right)}{exp\frac{(HW.BBT.SCHENK)}{kT} - 1} + \frac{(A^{\pm})^{-3/2} exp\left(\frac{A^{\pm}}{F}\right)}{1 - exp\frac{(-HW.BBT.SCHENK)}{kT}} \right) \quad 3\text{-}489$$

where $A^{\pm}$ =B.BBT.SCHENK($\hbar\omega \pm$ HW.BBT.SCHENK)$^{3/2}$, S is a statistical factor dependent on carrier concentrations, and $\hbar\omega$ is the energy of the transverse acoustic phonon. The upper sign refers to tunneling generation of electron-hole pairs (reverse biased junction). The lower sign refers to tunneling recombination of electron-hole pairs (forward bias).

You can set parameters from Table 3-110 on the **MODELS** or **MATERIAL** statement. To enable the model, specify SCHENK.BBT on the **MODELS** statement.

Table 3-110 Schenk Band to Band Tunneling Model Parameters

Statement	Parameter	Type	Default	Units
MATERIAL/**MODELS**	A.BBT.SCHENK	Real	8.977×10^{20}	$cm^2s^{-1}V^{-2}$
MATERIAL/**MODELS**	B.BBT.SCHENK	Real	2.14667×10^{7}	$eV^{(-3/2)}Vcm^{-1}$
MATERIAL/**MODELS**	HW.BBT.SCHENK	Real	0.0186	eV

Kane Band-To-Band Tunneling Model

Another local field band-to-band tunneling model is based on the work of Kane [148]. In this model, the tunneling generation rate is given by

$$G_{BBT} = \frac{D\ BBT.A\_KANE}{\sqrt{E_g}} F^{BBT.GAMMA} \exp\left(-BBT.B\_KANE \frac{E_g^{\frac{3}{2}}}{F}\right) \quad 3\text{-}490$$

where E_g is the position dependent bandgap and F is the magnitude of the electric field. It is similar to the standard models, except that it automatically includes a dependence on bandgap.

To enable this model specify BBT.KANE on the **MODELS** statement. The parameters are also aliased as indicated in Table 3-111.

Table 3-111 Parameters (with aliases) for the BBT.KANE Model

Statement	Parameter	Alias	Type	Default	Units
MODELS	BBT.KANE	BTBT	Logical	False	
MODELS	BBT.A_KANE	A.BTBT	Real	3.5×10^{21}	$eV^{1/2}/cm\text{-}s\text{-}V^2$
MODELS	BBT.B_KANE	B.BTBT	Real	2.25×10^{7}	$V/cm\text{-}eV^{3/2}$
MODELS	BBT.GAMMA		Real	2.5	

Non-local Band-to-Band Tunneling

The band-to-band tunneling models described so far in this section calculate a recombination-generation rate at each point based solely on the field value local to that point. For this reason, we refer to them as local models. To model the tunneling process more accurately, you need to take into account the spatial variation of the energy bands. You also need to take into account that the generation/recombination of opposite carrier types is not spatially coincident. Figure 3-11 illustrates this for a reverse biased p-n junction where it is assumed that the tunneling process is elastic. For degenerately doped p-n junctions, you can obtain tunneling current at forward bias and consequently can obtain negative differential resistance in the forward I-V curve.

The Atlas Non-local Band-To-Band tunneling model, BBT.NONLOCAL, allows modeling of the forward and reverse tunneling currents of degenerately doped p-n junctions.

It is less suitable for lightly doped p-n junctions. It is not suitable for application to the unipolar, high-field regions of a device. It is currently only available in Atlas2D.

The first step in using BBT.NONLOCAL is in defining the areas where it will be applied.

These areas must each contain a single p-n junction, which may be either planar or non-planar. They have a mesh that interpolates data from the underlying device mesh and performs the band-to-band tunneling calculations on the interpolated data.

BBT.NONLOCAL thus assumes that the tunneling takes place on a series of 1D slices through the junction, each slice being locally perpendicular to the junction. The slices themselves will be approximately parallel to their neighbors.

Atlas has two methods of setting up these areas of slices. The first method is only applicable to planar junctions parallel to the x or y-axes. It allows you to set up a rectangular area using the QTX.MESH and QTY.MESH statements. You also specify the number of tunneling slices and the number of mesh points along the slices.

For example:

```
QTX.MESH LOCATION=0.0 SPACING=0.25
QTX.MESH LOCATION=1.0 SPACING=0.25

QTY.MESH LOCATION=0.99 SPACING=0.0005
QTY.MESH LOCATION=1.01 SPACING=0.0005
```

will set up an area bounded by the sides X=0.0, X=1.0, Y=0.99, and Y=1.01. By additionally setting the QTUNN.DIR parameter on the **MODELS** statement to be 0, you define the tunneling direction to be in the y-direction. Therefore, we have 5 parallel tunneling slices at a uniform separation of 0.25 microns, each with 0.02/0.0005 + 1 = 41 mesh points along them. This region is consistent with a p-n junction parallel to the x-direction, and having a y-position of close to 1.0 microns. Notice that the mesh is very fine in the direction of tunneling.

It is recommended to have as fine mesh as possible in the tunneling direction, both for the physical mesh and this tunneling mesh, so that values can be accurately interpolated between the two meshes. If the tunneling direction is the x-direction, then set QTUNN.DIR to 1 on the **MODELS** statement.

The second method of setting up a mesh for tunneling can be applied to either planar or non-planar junctions. It can be used instead of, or in addition to, the first method.

To use the second method, create one or more **QTREGION** statements. These must be located after any **X.MESH**, **Y.MESH**, **ELECTRODE**, and **REGION** statements and before any **MODELS** statements that specify BBT.NONLOCAL. The parameters for **QTREGION** include the position coordinates of the four corners of a general quadrilateral. These parameters are X1, Y1, X2, Y2, X3, Y3, X4, Y4, and should describe the quadrilateral in a counter-clockwise sense as illustrated in Figure 3-9. The pair (X1, Y1) describe the coordinates of point 1. The other pairs (e.g., (X2, Y2)) describe the coordinates of the other points (e.g., 2).

The tunneling is calculated between points 1 and 4, and between points 2 and 3, and along all the slices in between. The tunneling slices do not need to be parallel to one another as shown in Figure 3-9 for the special case where the spacing between tunneling slices is regular along side 3-4 and also along side 1-2. The tunneling direction changes gradually as the quadrilateral is traversed (see Figure 3-9).

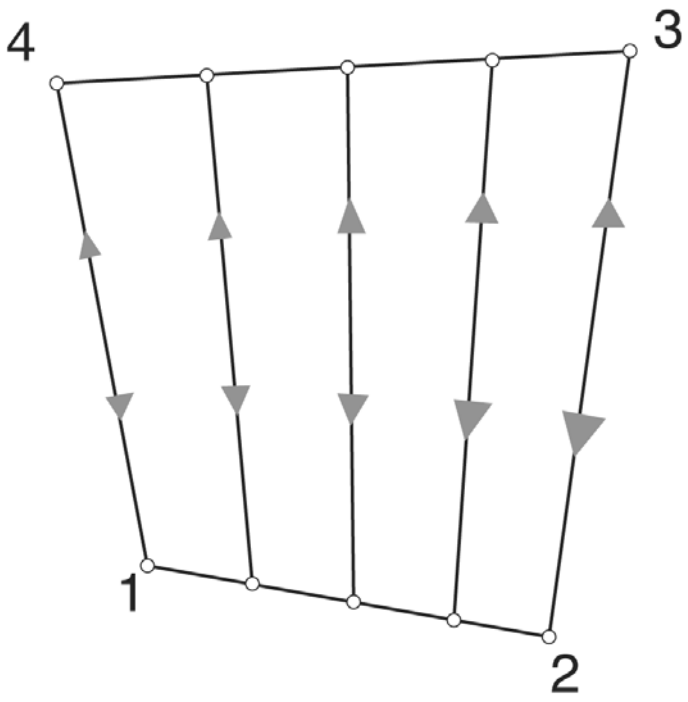


Figure 3-9: Quadrilateral region example arising from a QTREGION statement. PTS.NORMAL= 5 has been specified, modeling tunneling along 5 rays, which are not quite parallel. The spacing between the lines is regular along edge 1->2 and along edge 4->3. The tunneling direction is indicated by the arrows.

For tunneling through non-planar junctions, it is preferable to construct a complicated polygonal region to use the BBT.NONLOCAL method.

You can achieve this by using several joined **QTREGION**s: all with a common value of the NUMBER parameter. The simple rule that must be followed to join the quadrilaterals is to ensure that point (X2, Y2) on the first matches (X1, Y1) on the other, and that point (X3, Y3) on the first matches point (X4, Y4) on the next. If (X1, Y1) or (X4, Y4) or both are not specified, then they default to the values of (X2, Y2) and (X3, Y3) respectively from the previous **QTREGION** statement. The first **QTREGION** must have all 4 corner points described. In this way, you can build up a tunneling mesh to cover quite complicated p-n junction shapes. An example is given in Figure 3-10.

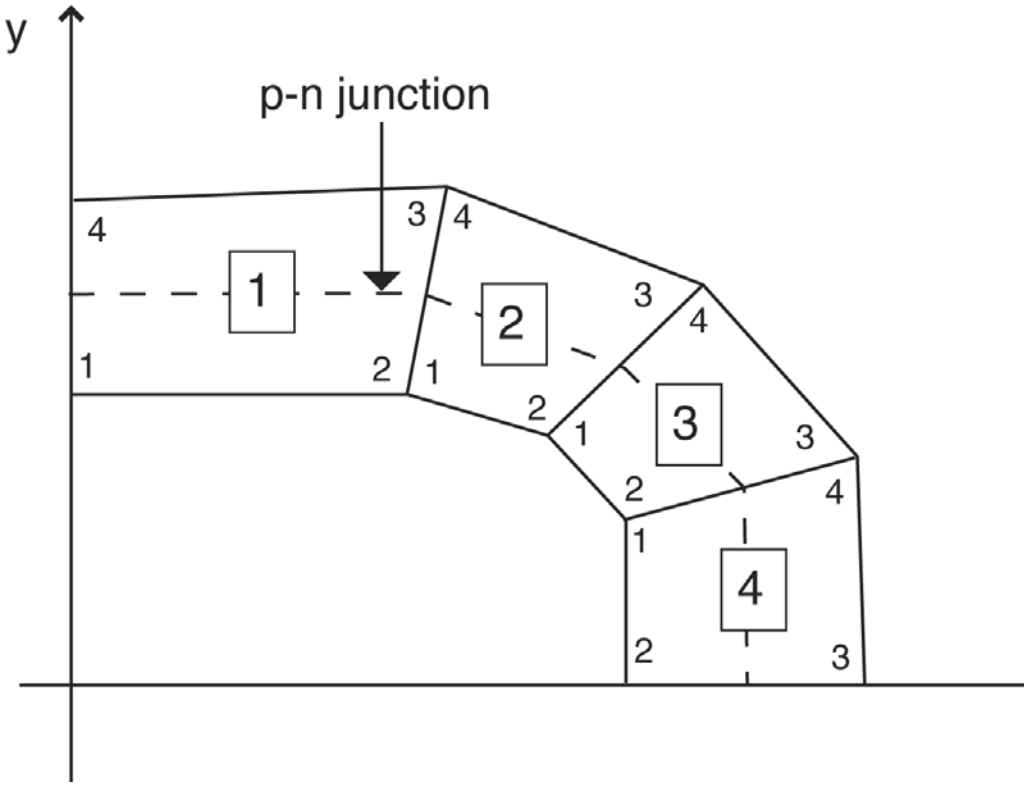


Figure 3-10: Example of a polygonal region made by joining 4 quadrilaterals. The region encloses the p-n junction, which is geometrically curved and ensures the BBT current calculation occurs along lines that are approximately locally normal to the junction. Quadrilateral 1 must have (X1,Y1), (X2,Y2), (X3,Y3), and (X4,Y4) specified. For quadrilaterals 2,3 and 4, specifying (X1,Y1) and (X4, Y4) is optional. If they are specified, they must be equal to the values of (X2,Y2) and (X3,Y3) on the adjacent quadrilateral.

You must set up the number of mesh points you require in the tunneling direction and the simplest way to do this is to use the PTS.TUNNEL parameter. The value of PTS.TUNNEL applies to all slices in a region of one or more joined quadrilaterals. You can also set the number of slices in each quadrilateral using the PTS.NORMAL parameter. You may have a different value of PTS.NORMAL for each quadrilateral in the same region. If there is more than one value of PTS.TUNNEL specified for a particular region, then the one specified by the last QTREGION statement will be used.

To allow some flexibility, there are two alternatives for specifying non-uniform mesh spacing in the tunneling direction. You can use the parameters STNL.BEG and STNL.END to give the mesh spacing at the beginning and end of the slice. Otherwise, you can use the F.RESOLUTION parameter to specify the name of an ASCII-format datafile. This data file must contain a single column of ascending values, in the range [0,1]. The density of mesh points in the tunneling direction is obtained from linearly mapping from the [0,1] range in the datafile to the start and end position of each tunneling slice for each quadrilateral having the same NUMBER parameter.

If you wish to specify non-uniform separations of the tunnel slices in a quadrilateral, then you can achieve this using the SNRM.BEG and SNRM.END parameters. The spacing at the left hand side of the quadrilateral is given by SNRM.BEG and that at the end by SNRM.END. This functionality is useful because it allows you to match the interpolated tunneling mesh to the underlying device mesh. Each quadrilateral may have its own unique values of SNRM.BEG and SNRM.END.

In summary, you can set up one or more quantum band-to-band tunneling regions, each comprised of one or more **QTREGION**s. All **QTREGION**s having the same value of the NUMBER parameter are treated as part of the same region, and they must be joined together. Additionally, one region set up by QTX.MESH and QTY.MESH can also be set up. The number of slices in each region, and the density of points along those slices can be set up in several ways. Quantities from the underlying device mesh are obtained from interpolation, and it is imperative that the device mesh resolves the rapid changes in potential near to a highly doped p-n junction.

The resolution of the tunneling slices in the tunneling direction should also be fine enough to capture these rapid changes. Each tunneling slice must include exactly one p-n junction for the model to work. The start and end points of the tunnel slices should ideally be in the flat band regions on either side of the junction.

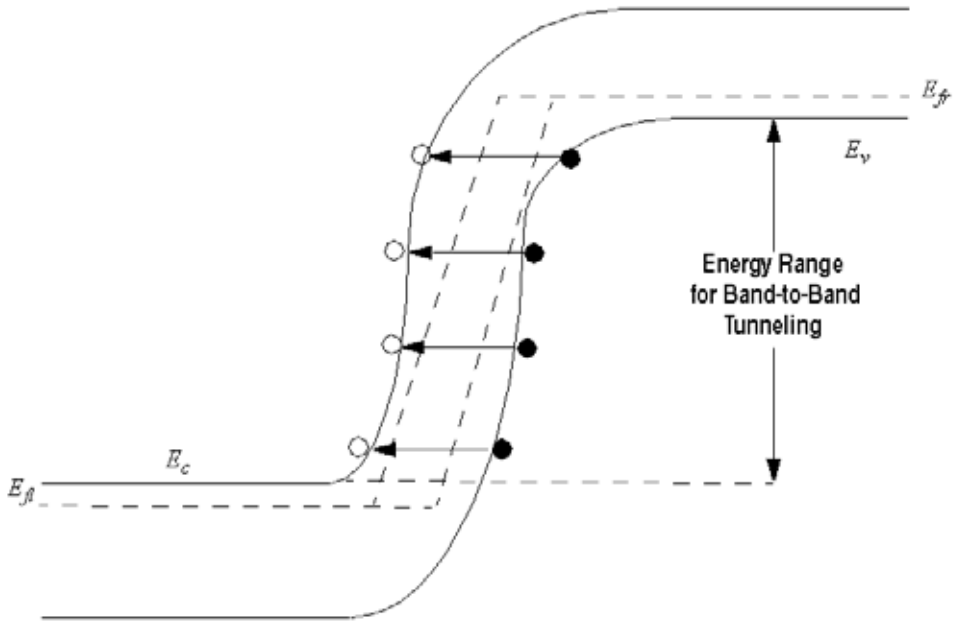


Figure 3-11: Schematic of non-local band to band tunneling in reverse bias

In order to explain how the tunneling current is calculated, let us consider the energy band profile along each tunneling slice with reverse bias applied across the junction. The range of valence band electron energies for which tunneling is permitted is shown in the schematic of the energy band profile in Figure 3-11. The highest energy at which an electron can tunnel is E_{upper} and the lowest is E_{lower}. The tunneling can be thought of being either the transfer of electrons or the transfer of holes across the junction. The rates for electrons and holes are equal and opposite because the tunneling results in the generation or recombination of electron-hole pairs.

Considering the tunneling process as a transfer of an electron across the junction the net current per unit area for an electron with longitudinal energy E and transverse energy E_T is

$$J(E) = \frac{q}{\pi\hbar}\int\int T(E)[f_l(E+E_T) - f_r(E+E_T)]\rho(E_T)dE\,dE_T \qquad 3\text{-}491$$

where $T(E)$ is the tunneling probability for an electron with longitudinal energy E. $\rho(E_T)$ is the 2-dimensional density of states corresponding to the 2 transverse wavevector components and

$$f_l = (1+\exp[(E+E_T-E_{Fl})/KT])^{-1} \qquad 3\text{-}492$$

is the Fermi-Dirac function using the quasi Fermi-level on the left hand side of the junction. Similarly

$$f_r = (1+\exp[(E+E_T-E_{Fr})/KT])^{-1} \qquad 3\text{-}493$$

uses the quasi-Fermi level on the right hand side of the junction. This assumes that the transverse energy is conserved in the tunneling transition. Because we are using a 2-band model to give the evanescent wavevector, the transverse electron effective mass and the transverse hole effective mass are combined in the 2D density of states

$$\rho(E_T) = \frac{\sqrt{(m_e m_h)}}{2\pi\hbar^2} \qquad 3\text{-}494$$

By integrating over transverse carrier energies, we obtain the contribution to the current from the longitudinal energy range E - $\Delta E/2$ to E + $\Delta E/2$ as follows:

$$J(E)\Delta E = \frac{qKT\sqrt{m_e m_h}}{2\pi^2\hbar^3}T(E)\Big|\log(1+\exp[(E_{Fl}-E-E_T)/KT]) - \log(1+\exp[(E_{Fr}-E-E_T)/KT])\Big|_0^{E_{max}}\Delta E \qquad 3\text{-}495$$

where the upper limit of integration, E_{max} is smaller of E - E_{lower} and E_{upper} - E. This is consistent with the 2-band model of the tunneling and the restriction on total carrier energy for tunneling $E_{lower} \le E + E_T \le E_{upper}$. Putting the above integration limits into Equation 3-495 we obtain

$$J(E)\Delta E = \frac{qKT\sqrt{m_e m_h}}{2\pi^2\hbar^3}T(E)\log\left(\frac{(1+\exp[(E_{Fr}-E)/KT])(1+\exp[(E_{Fl}-E-E_{max})/KT])}{(1+\exp[(E_{Fl}-E)/KT])(1+\exp[(E_{Fr}-E-E_{max})/KT])}\right)\Delta E \qquad 3\text{-}496$$

The Fermi levels used in Equation 3-496 are the quasi-Fermi levels belonging to the majority carrier at the relevant side of the junction. In Figure 3-11 for example, E_{Fl} would be the Electron quasi-Fermi level and E_{Fr} would be the hole quasi-Fermi level. In equilibrium, $E_{Fl} = E_{Fr}$ and the current contributions are all zero. As also seen in Figure 3-11, the start and end points of the tunneling paths, x_{start} and x_{end}, depend on Energy. Atlas calculates these start and end points for each value of E and calculates the evanescent wavevector at points in between as

$$k(x) = \frac{k_e k_h}{\sqrt{k_e^2 + k_h^2}} \quad \text{3-497}$$

where

$$k_e(x) = \frac{1}{i\hbar}\sqrt{2 m_o m_e(x)(E - E_c(x))} \quad \text{3-498}$$

and

$$k_h(x) = \frac{1}{i\hbar}\sqrt{2 m_o m_h(x)(E_v(x) - E)} \quad \text{3-499}$$

This ensures that the energy dispersion relationship is electron-like near the conduction band and hole-like near the valence band, and approximately mixed in between. The tunneling probability, *T(E)*, is then calculated using the WKB approximation

$$T(E) = \exp\left(-2\int_{x_{start}}^{x_{end}} k(x)\,dx\right) \quad \text{3-500}$$

This value is put into Equation 3-496 to give the tunneling current density at a given perpendicular energy, E, and the resulting current is injected into the simulation at x_{start} and x_{end}. This is repeated for all values of E between E_{lower} and E_{upper} and is done for every tunneling slice in the tunneling regions.

To calibrate the BBT.NONLOCAL model, you can specify the values of effective mass used in Equations 3-496, 3-498, and 3-499. Set the values of effective mass used in Equations 3-498 and 3-499 using ME.TUNNEL and MH.TUNNEL on the **MATERIAL** statement respectively. If ME.TUNNEL (MH.TUNNEL) is not specified, then the value specified for MC (MV) on the **MATERIAL** statement is used. For the transverse effective masses as used in Equation 3-496, a suitable value will be calculated by Atlas, unless MC or MV or both have been specified on the **MATERIAL** statement, in which case these are used. The tunneling current is most sensitive to the effective masses used in Equations 3-498 and 3-499 because the tunneling probability depends exponentially on them.

Across very highly doped junctions, the Band-To-Band tunneling current can be very high and depends very strongly on the band edge profile throughout the junction. In turn, the injected tunnel current creates a charge dipole and strongly affects the potential, and consequently the band energies, at and near the junction. This strong coupling can cause convergence problems in some situations. Atlas has two features to try to mitigate this problem. First, you can specify BBT.NLDERIVS on the **MODELS** statement to include non-local coupling. This puts the derivatives of *J(E)* with respect to all potential values along its path

into the Jacobian matrix. Generally, this improves the convergence properties. But, this increases the solution time per iteration because of the consequent increase in the size of the Jacobian matrix.

Second, the position at which the tunnel current is injected into the simulation can affect convergence. The default described above is the most physically accurate. As an alternative, use the parameter BBT.FORWARD on the **MODELS** statement. This causes the tunnel current to be injected at the start and end positions of each tunnel slice, further from the junction than with the default model. The parameter BBT.REVERSE on the **MODELS** statement causes the tunnel current to be injected in a locally charge neutral manner across the width of the junction, thus avoiding the creation of a charge dipole. The injected currents can be viewed as recombination rates in the structure file. They are automatically output whenever the BBT.NONLOCAL model is used. They will be negative for a reverse biased junction and positive for a forward biased junction.

Because the BBT.NONLOCAL model is unsuitable for all band-to-band tunneling, you can use the local Band-To-Band tunneling models (e.g., BBT.STD and BBT.KLA) simultaneously with BBT.NONLOCAL. These local models are not applied inside any of the Quantum Tunneling regions set up by the **QTREGION** statement or **QTX.MESH**/**QTY.MESH** statements.

Table 3-112 Non-Local Band-to-Band Tunneling Parameters for the MODELS Statement

Parameter	Type	Default
BBT.NONLOCAL	Logical	False
BBT.NLDERIVS	Logical	False
BBT.FORWARD	Logical	False
BBT.REVERSE	Logical	False

Table 3-113 Non-Local Band-to-Band Tunneling Parameters for the MATERIAL Statement

Parameter	Type	Units
ME	Real	None
MV	Real	None
ME.TUNNEL	Real	None
MH.TUNNEL	Real	None

Table 3-114 Parameters for the QTREGION Statement

Statement	Parameter	Type	Units
QTREGION	F.RESOLUTION	Character	
QTREGION	NUMBER	Real	
QTREGION	PTS.NORMAL	Real	
QTREGION	PTS.TUNNEL	Real	

Table 3-114 Parameters for the QTREGION Statement			
Statement	**Parameter**	**Type**	**Units**
QTREGION	SNRM.BEG	Real	Microns
QTREGION	SNRM.END	Real	Microns
QTREGION	STNL.BEG	Real	Microns
QTREGION	STNL.END	Real	Microns
QTREGION	X1	Real	Microns
QTREGION	X2	Real	Microns
QTREGION	X3	Real	Microns
QTREGION	X4	Real	Microns
QTREGION	Y1	Real	Microns
QTREGION	Y2	Real	Microns
QTREGION	Y3	Real	Microns
QTREGION	Y4	Real	Microns

Note: QTREGION must be specified after any X.MESH, Y.MESH, ELECTRODE, and REGION statements, but before any DOPING statements.

Note: If you want to fine tune the Band to Band tunneling, we recommend you use ME.TUNNEL and MH.TUNNEL parameters because they will only modify the effective masses used in tunneling.

3.6.7 Gate Current Models

In devices that have a metal-insulator-semiconductor (MIS) formation, the conductance of the insulating film would ideally be considered as zero. But, for the sub 0.5um generation of MOS devices there is now considerable conductance being measured on the gate contacts. This gate current has resulted in two major consequences, one negative and one positive.

On the negative side, the gate current is responsible for the degradation in device operating characteristics with time. This reliability issue is important because the lifetime of electronic parts has to be guaranteed. You can simulate reliability within the Silvaco suite of tools for device level reliability which are described in a later chapter.

On the positive side, the existence of this gate current has caused the proliferation of the non-volatile memory market. These devices use the existence of gate current to program and erase the charge on a “floating” contact. This concept has resulted in a variety of different devices such as FLASH, FLOTOX, and EEPROM. All such devices rely on the physics of the gate current process for their existence.

There are a variety of different conduction mechanisms within an insulating layer [307], but in the case of nonvolatile memory, only two mechanism are relevant: Fowler-Nordheim

tunneling and hot carrier injection. Models for these two injection processes are described in the following sections. In the case of hot electron injection, two models are available: the lucky electron model and the Concannon gate current model.

Fowler-Nordheim Tunneling

If the electric field across an insulator is sufficiently high, then it may cause tunneling of electrons from the semiconductor (or metal) Fermi level into the insulator conduction band. This process is strongly dependent on the applied electric field but is independent of the ambient temperature.

The Fowler-Nordheim Equation [150] expresses tunnel current density through the oxide as:

$$J_{FN} = \mathrm{F.AE}\ E^2 exp\left(-\frac{\mathrm{F.BE}}{E}\right) \quad 3\text{-}501$$

$$J_{FP} = \mathrm{F.AH}\ E^2 exp\left(-\frac{\mathrm{F.BH}}{E}\right) \quad 3\text{-}502$$

where E specifies the magnitude of the electric field in the oxide. The model parameters: F.AE, F.AH, F.BE, and F.BH can be defined on the **MODELS** statement. The default values for these parameters, obtained from Keeney, Piccini, and Morelli [150], are shown in Table 3-115.

Table 3-115 User-Specifiable Parameters for Equations 3-501 and 3-502

Symbol	Statement	Parameter	Default Values
A_{FN}	**MODELS**	F.AE	1.82×10^{-7}
B_{FN}	**MODELS**	F.BE	1.90×10^{8}
A_{FH}	**MODELS**	F.AH	1.82×10^{-7}
B_{FH}	**MODELS**	F.BH	1.90×10^{8}

The Fowler-Nordheim model in Atlas has been implemented in two ways. In the post-processing implementation, the tunneling current is calculated from the solution to the device equations at each bias step. In the self-consistent implementation, the tunneling current is included directly in the current-continuity equations. Thus, it is part of the solution to the equations and gives accurate current continuity for the device.

To enable the post processing version, use the **MODELS** statement parameters FNPP for electron current and FNHPP for hole current. To enable the self-consistent solution, use the **MODELS** statement parameters FNORD for electrons and FNHOLES for holes. Setting the parameter FN.CUR is the same as specifying FNORD and FNHOLES.

For either model, the implementation scheme is the same. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment, the Fowler-Nordheim current is calculated as described above. This current is then added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the Fowler-Nordheim current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, receives the Fowler-Nordheim current.

A second model may be chosen using the NEARFLG parameter of the **MODELS** statement. In this case, the electrode-insulator segment found closest to the semiconductor-insulator segment receives the Fowler-Nordheim current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary. To include the tunneling current in the logfile, specify J.TUN on the LOG statement.

By default, the Fowler-Nordheim model, when enabled, is applied to all relevant interfaces in the device structure. To localize the model, use one or more **INTERFACE** statements with one or more of the parameters X.MIN, X.MAX, Y.MIN, and Y.MAX specified. This restricts the application of Fowler-Nordheim tunneling to interfaces contained in the union of the rectangular areas defined by the **INTERFACE** statements. The parameter S.I. must also be set on the **INTERFACE** statement but is enabled by default.

In the special situation of an insulator layer between two semiconductor regions, the FN.SIS parameter can be specified on the **REGION** statement. This precludes the Fowler-Nordheim current generated on the interfaces of the region from being added to any terminal current. The current is instead added to current continuity equations at one side of the region and subracted on the other side. This simulates tunneling through an insulator placed between two semiconductors.

Note: Since Fowler-Nordheim tunneling current is responsible for EPROM and EEPROM cell erasure, this model should always be specified when performing erasure simulation. We also recommend that you model the band-to-band tunneling model if you model Fowler-Norheim tunneling.

Note: When simulating EPROM erasure in a transient analysis with this model, the floating contact charge becomes a function of the gate current. In this case, the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of the charge is then used as the boundary condition for the next time step.

Lucky Electron Hot Carrier Injection Model

In the Lucky-Electron Hot Carrier Injection Model it is proposed that an electron is emitted into the oxide by first gaining enough energy from the electric field in the channel to surmount the insulator/semiconductor barrier. Once the required energy to surmount the barrier has been obtained the electrons are redirected towards the insulator/semiconductor interface by some form of phonon scattering. When these conditions are met, the carrier travelling towards the interface will then have an additional probability that it will not suffer any additional collision through which energy could be lost.

The model implemented into Atlas is a modified version of the model proposed by Tam [311] and is activated by the parameters HEI and HHI, for electron and hole injection respectively, on the **MODELS** statement. The gate electrode-insulator interface is subdivided into a number of discrete segments which are defined by the mesh. For each segment the lucky electron

model is used to calculate the injected current into that segment. The total gate current is then the sum of all of the discrete values.

If we consider a discrete point on the gate electrode-insulator boundary we can write a mathematical formula for the current injected from the semiconductor. The formula calculates the injected gate current contribution from every node point within the semiconductor according to:

$$Iinj = \iint P_n(x,y)\left|\vec{J}n(x,y)\right| dx\ dy + \iint P_p(x,y)\left|\vec{J}p(x,y)\right| dx\ dy \quad \text{3-503}$$

where $J_{n,p}$ (x,y) are the electron and hole current densities at a point (x,y) within the semiconductor, and $P_{n,p}$ (x,y) are the probabilities that a fraction of this current reaches the gate oxide and is injected across into the gate electrode. The total probability $P_{n,p}$ (x,y) is defined by:

$$P_n(x,y) = P_{\phi B,\,n} P_{1,\,n} P_{2,\,n}{}^{/\texttt{IG.ELINR}} \quad \text{3-504}$$

$$P_p(x,y) = P_{\phi B,\,p} P_{1,\,p} P_{2,\,p}{}^{/\texttt{IG.HLINR}} \quad \text{3-505}$$

where E is the electric field parallel to the current flow, IG.ELINR and IG.HLINR are the electron and hole mean free path lengths between redirecting collisions. The three probability factors will now be described.

The probability $P\phi_B$ is the probability of a carrier gaining the energy ϕ_B by moving in, and parallel to, an electric field E, without suffering energy loss by optical phonon scattering and is given by:

$$P_{\phi_{B,\,n}} = 0.25\left(\frac{E\ \texttt{IG.ELINF}}{\phi_{B,\,n}}\right) exp\left(-\frac{\phi_{B,\,n}}{E\ \texttt{IG.ELINF}}\right) \quad \text{3-506}$$

$$P_{\phi_{B,\,p}} = 0.25\left(\frac{E\ \texttt{IG.HLINF}}{\phi_{B,\,p}}\right) \exp\left(-\frac{\phi_{B,\,p}}{E\ \texttt{IG.HLINF}}\right) \quad \text{3-507}$$

where IG.ELINF and IG.HLINF are the mean free path lengths of electrons and holes for scattering by optical phonons. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B,\,n} = \texttt{IG.EB0} - \texttt{IG.EBETA}\sqrt{E_\perp} - \texttt{IG.EETAE}_\perp^{2/3} - \Delta\psi(x,y) \quad \text{3-508}$$

$$\phi_{B,\,p} = \texttt{IG.HB0} - \texttt{IG.HBETA}\sqrt{E_\perp} - \texttt{IG.HETAE}_\perp^{2/3} - \Delta\psi(x,y) \quad \text{3-509}$$

where $E_\perp$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0, are reduced to take account of three effects. The first effect is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second effect takes account of

tunneling through the gate oxide by reducing the barrier height. The third effect takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default, this last effect is disabled. But you can enable it by specifying the E.BENDING and H.BENDING parameters for electrons and holes respectively.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected, and is given by:

$$P_{1,n} \sim exp\left(-\frac{r}{IG.ELINF}\right) \quad 3\text{-}510$$

$$P_{1,p} \sim exp\left(-\frac{r}{IG.HLINF}\right) \quad 3\text{-}511$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by:

$$P_{2,n} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{PATH.N}\right) \quad \text{for } \theta > THETA.N \quad 3\text{-}512$$

$$P_{2,n} = 0 \quad \text{for } \theta < THETA.N \quad 3\text{-}513$$

$$P_{2,p} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{PATH.P}\right) \quad \text{for } \theta > THETA.P \quad 3\text{-}514$$

$$P_{2,p} = 0 \quad \text{for } \theta < THETA.P \quad 3\text{-}515$$

Here, PATH.N and PATH.P are the electron and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [337]. His experiments indicate a critical rejection angle, THETA.N and THETA.P, between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle then the electrons are repelled back to the substrate.

Table 3-116 lists the user-definable model parameters that can be set in the **MODELS** statement, their default values, and their units.

Table 3-116 User-Definable Parameters in Concannon's Gate Current Model			
Statement	**Parameter**	**Default**	**Units**
MODELS	IG.EB0	3.2	eV
MODELS	IG.ELINR	6.16×10^{-6}	cm
MODELS	IG.HLINR	6.16×10^{-6}	cm
MODELS	IG.ELINF	9.2×10^{-7}	cm
MODELS	IG.HB0	4.0	eV
MODELS	IG.HLINF	9.2×10^{-7}	cm
MODELS	IG.EBETA	2.59×10^{-4}	$(Vcm)^{1/2}$
MODELS	IG.HBETA	2.59×10^{-4}	$(Vcm)^{1/2}$
MODELS	IG.EETA	2.0×10^{-5}	$V^{1/3}cm^{2/3}$
MODELS	IG.HETA	2.0×10^{-5}	$V^{1/3}cm^{2/3}$
MODELS	IG.LRELE	3.0×10^{-6}	[Q=1] cm
MODELS	IG.LRELH	2.0×10^{-6}	[Q=1] cm
MODELS	PATH.N	3.4×10^{-7}	cm
MODELS	PATH.P	2.38×10^{-7}	cm
MODELS	THETA.N	60	degrees
MODELS	THETA.P	60	degrees

The implementation of this model is similar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments, which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the **MODELS** statement. In this case, the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary.

The lucky electron hot carrier injection model can be used to include the electron or hole carrier temperature in the solution because the carrier temperature does not directly enter the equations.

The one exception is in Atlas2d where the electric field parallel to the current flow is calculated as

$$E = 1.5\ K_B\ T_n\ /\ IG.LRELE$$

for electrons if HCTE.EL is specified and

$$E = 1.5\ K_B\ T_p\ /\ IG.LRELH$$

for holes if HCTE.HO is specified. K_B is Boltzmanns constant in units of eV/Kelvin. If IG.LRELE is set to zero then E will be calculated the same way as if HCTE.EL is unspecified. The same is true for IG.LRELH and HCTE.HO. In Atlas3d, this model is unavailable and E is calculated the same way regardless if HCTE is specified.

Note: When simulating EPROM programming with this model, the floating contact charging is simulated in the transient mode. In this case, the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of charge is then used as the boundary condition for the next time step.

The hot electron and hot hole currents can be output separately to a logfile by specifying the parameters J.HEI and J.HHI on the LOG statement. They are also automatically included in the electron and hole currents which are output to a logfile. The value of hot carrier current density associated with each interface node can be output to any standard structure file by specifying the HEI and HHI parameters on the **OUTPUT** statement.

Concannon's Injection Model

The implicit assumption in the lucky electron approach is a Maxwellian shape for the energy distribution of the hot carriers. Recent work by Fiegna [87] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the distribution function. To accurately model these effects, a non-Maxwellian based model from Concannon [66] has been implemented. This model requires the solution to the energy balance equation for the carrier temperatures but has been implemented in a similar manner to the lucky electron model. The Concannon gate injection model may be specified with the parameters N.CONCANNON and P.CONCANNON on the **MODELS** statement. This choice of parameters automatically activates the Energy Balance Transport Model.

The Concannon injection model has a similar form to the lucky electron model. The injected current is calculated according to:

$$Iinj = \int\int P_n(x,y)\ n(x,y)dx\ dy + \int\int P_p(x,y)\ p(x,y)dx\ dy \qquad 3\text{-}516$$

where $n(x,y)$ and $p(x,y)$ are the carrier concentrations within the semiconductor. The probability functions $P_n(x,y)$ and $P_p(x,y)$ are now defined by:

$$P_n(x,y) = -q\ \text{CGATE.N}\ P_{\phi_{B,n}} P_{1,n} P_{2,n} \qquad 3\text{-}517$$

$$P_p(x,y) = q\ \texttt{CGATE.P}\ P_{\phi_{B,p}} P_{1,p} P_{2,p} \quad \text{3-518}$$

where q is the electronic charge and the parameters CGATE.N and CGATE.P are user-definable on the **MODELS** statement. The three probability functions in Equations 3-517 and 3-518 shall now be described.

The probability that a carrier has sufficient energy to surmount the insulator-semiconductor barrier of height ϕ_B is now defined as a function of energy. The probability has the form:

$$P_{\phi_{B,n}} = \int_{\phi_{B,n}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_n(x,y)) d\varepsilon \quad \text{3-519}$$

$$P_{\phi_{B,p}} = \int_{\phi_{B,p}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_p(x,y)) d\varepsilon \quad \text{3-520}$$

Here, $v_{\perp}(\varepsilon)$ is the perpendicular velocity of a hot carrier and defines the probability of a hot carrier with an energy ε travelling in the direction of the insulator-semiconductor. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B,n} = \texttt{IG.EB0} - \texttt{IG.EBETA}\sqrt{E_{\perp}} - \texttt{IG.EETA}\ E_{\perp}^{2/3} - \Delta\psi(x,y) \quad \text{3-521}$$

$$\phi_{B,p} = \texttt{IG.HB0} - \texttt{IG.HBETA}\sqrt{E_{\perp}} - \texttt{IG.HETA}\ E_{\perp}^{2/3} - \Delta\psi(x,y) \quad \text{3-522}$$

where $E_{\perp}$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0, are reduced to take account of three effects. The first effect is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second effect takes account of tunneling through the gate oxide by reducing the barrier height. The third effect takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default, this last effect is disabled. But you can enable it by specifying the E.BENDING and H.BENDING parameters for electrons and holes respectively.

The carrier velocity model follows the approach of Fiegna et. al. [87] where velocity is proportional to energy according to:

$$v_{\perp} \sim \varepsilon^{0.25} \quad \text{3-523}$$

The function, $F(\varepsilon, T_{n,p}(x,y))$, is determined by the density of states and the energy distribution function according to:

$$F(\varepsilon, T_n(x,y)) \sim \frac{g(\varepsilon) f(\varepsilon)}{\int_{\circ}^{\infty} g(\varepsilon) f(\varepsilon) d\varepsilon} \quad \text{3-524}$$

The density of states $g(\varepsilon)$ follows the analysis of Cassi [49] where:

$$g(\varepsilon) \sim \varepsilon^{1.25} \quad \text{3-525}$$

Finally, the energy distribution functions for electrons and holes are defined by:

$$f_n(\varepsilon) \sim \left[exp\left(\frac{-\mathtt{CHIA}\ \varepsilon^3}{T_n^{1.5}} \right) + \mathtt{C0}\, exp\left(\frac{-\mathtt{CHIB}\ \varepsilon^3}{T_n^{1.5}} \right) \right] \quad \text{3-526}$$

$$f_p(\varepsilon) \sim exp\left(\frac{-\mathtt{CHI.HOLES}\ \varepsilon^3}{T_p^{1.5}} \right) \quad \text{3-527}$$

where CHIA, CHIB, CHI.HOLES, and C0 are user-definable constants found from fitting to measured data. The terms: T_n and T_p are the mean carrier temperatures for electrons and holes, which are calculated from the Energy Balance Transport Model.

Normalization in all of the above equations is accounted for in the combined constants of proportionality, CGATE.N and CGATE.P.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected and is given by:

$$P_{1,\,n} \sim exp\left(-\frac{r}{\mathtt{IG.ELINF}} \right) \quad \text{3-528}$$

$$P_{1,\,p} \sim exp\left(-\frac{r}{\mathtt{IG.HLINF}} \right) \quad \text{3-529}$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by:

$$P_{2,\,n} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathtt{PATH.N}} \right) \quad \text{for } \theta > \mathtt{THETA.N} \quad \text{3-530}$$

$$P_{2,\,n} = 0 \quad \text{for } \theta > \mathtt{THETA.N} \quad \text{3-531}$$

$$P_{2,\,p} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathtt{PATH.P}} \right) \quad \text{for } \theta > \mathtt{THETA.P} \quad \text{3-532}$$

$$P_{2,\,p} = 0 \quad \text{for } \theta < \mathtt{THETA.P} \quad \text{3-533}$$

Here, PATH.N and PATH.P are the electron and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [337]. His experiments indicate a critical rejection angle, THETA.N and THETA.P between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle, then the electrons are repelled back to the substrate.

Note: The current implementation of the Concannon model for hot carrier injection is that only carriers along the semiconductor-insulator interface are significant and as a result the probability P_1 is assumed unity. This also means that the integration is only applied to those node points along the semiconductor-insulator interface.

Two other parameters of the **MODELS** statement that may affect the result of the numeric integration are user-definable. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter sets the upper limit of the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

The implementation of this model is similar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the **MODELS** statement. In this case the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary.

Note: To maintain self-consistent results, it's important that this model is implemented if the Concannon model is being used for the simulation of substrate current.

The Concannon hot electron and hole currents may be output separately to a logfile using the parameters J.HEI and J.HHI on the LOG statement respectively. They are also automatically included in the electron and hole currents that are output to a logfile. The value of Concannon hot carrier current density associated with each interface node can be output to any standard structure file by specifying the HEI and HHI parameters on the **OUTPUT** statement.

Direct Quantum Tunneling Model

For deep submicron devices, the thickness of the insulating layers can be very small. For example, gate oxide thicknesses in MOS devices can be as low as several nanometers. In this case, the main assumptions of the Fowler-Nordheim approximation are generally invalid and you need a more accurate expression for tunneling current. The one Atlas uses is based on a formula, which was introduced by Price and Radcliffe [253] and developed by later authors. It formulates the Schrodinger equation in the effective mass approximation and solves it to calculate the transmission probability, *T(E)*, of an electron or hole through the potential barrier formed by the oxide layer. The incident (perpendicular) energy of the charge carrier, *E*, is a parameter. It is assumed that the tunneling process is elastic. After taking into account carrier statistics and integrating over lateral energy, the formula

$$J = \frac{qkT}{2\pi^2 h^3}\sqrt{m_y m_z}\int T(E)\ln\left\{\frac{1+\exp[(E_{Fr}-E)/kT]}{1+\exp[(E_{Fl}-E)/kT]}\right\}dE \qquad 3\text{-}534$$

is obtained, which gives the current density J (A/m^2) though the barrier. The effective masses m_y and m_z are the effective masses in the lateral direction in the semiconductor. For example, for a direct bandgap material, where the Γ valley is isotropic, both m_y and m_z are the same as the density of states effective mass. The logarithmic term includes the carrier statistics and E_{fl} and E_{fr} are the quasi-Fermi levels on either side of the barrier (see Figure 3-12). The range of integration is determined according to the band edge shape at any given contact bias.

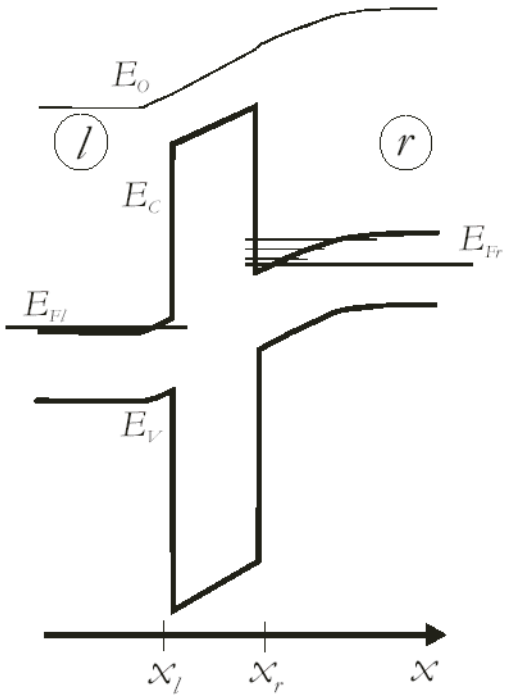


Figure 3-12: Typical conduction and valence band profiles of a MOS capacitor. The right region represents the MOS bulk, the left region represents the gate.

For indirect bandgap materials, Equation 3-535 is applied to each valley and the resulting sum gives the tunneling current. For the conduction band of silicon, for example, summing over the 6 valleys gives

$$J = \frac{qkT}{2\pi^2\hbar^3}(2m_t + 4\sqrt{m_l m_t})\int T(E)\ln\left\{\frac{1+\exp[(E_{Fr}-E)/kT]}{1+\exp[(E_{Fl}-E)/kT]}\right\}dE \quad 3\text{-}535$$

where m_t is the transverse effective mass and m_l the longitudinal mass. For a valence band, you need to sum the light hole and heavy hole contributions separately. For the valence band, the light hole and heavy contributions are summed to give

$$J = \frac{qkT}{2\pi^2\hbar^3}(m_{lh} + m_{hh})\int T(E)\ln\left\{\frac{1+\exp[(E_{Fr}-E)/kT]}{1+\exp[(E_{Fl}-E)/kT]}\right\}dE \quad 3\text{-}536$$

In equilibrium, $E_{Fl} = E_{Fr}$ and the logarithmic term and consequently J is identically zero.

The tunneling probability T(E) is calculated by using a transfer matrix method to solve the Schrodinger equation. The method uses a stepwise approximation of the potential. Consequently, it is recommended to use a fine mesh in the tunnelling direction.

Tunneling Types

Atlas can account for several types of tunneling. For clarity, we assume the material on the left hand side of Figure 3-12 is polysilicon. Electron tunneling occurs when electrons tunnel through the insulator energy barrier from one conduction band to the other. Hole tunneling occurs when holes tunnel from one valence band to the other. If the bias applied to the contact is sufficiently large, then the situation illustrated in Figure 3-13 can occur. In this case, an electron in the silicon valence band tunnels through the insulator to the polysilicon conduction band.

If the bias is reversed, then the opposite occurs in which an electron in the polysilicon valence band tunnels to the Silicon conduction band. Both of these cases are referred to as band-to-band Tunneling. These should not be confused, however, with the other band-to-band tunneling models implemented in Atlas, which apply to regions of high electric field within a semiconductor. All these types of tunneling can be used as post-processing calculations or self-consistent calculations. In the former case, the tunneling currents are calculated after Atlas has found a converged solution at a given bias. Atlas then outputs the calculated tunneling currents to the logfile. In the latter case, the tunneling currents are coupled with the solution of the current continuity equations in the semiconductor. Therefore, the terminal currents are self-consistent.

In situations where there is a strong quantum confinement in the semiconductor, Atlas can include charge quantization effects. To do this, use the Schrodinger-Poisson solver in the Silicon, which causes the tunneling current to be calculated as a sum over eigenstates. The relevant formulae are

$$J = \frac{qkT}{\pi\hbar^2}\sqrt{m_y m_z}\sum_i v_r(E_i)T(E_i)ln\left\{\frac{1+exp[(E_{Fl}-E_i)/kT]}{1+exp[(E_{Fr}-E_i)/kT]}\right\} \quad 3\text{-}537$$

where the sum is over all bound state energies, E_i, for a particular band minimum.

For electrons in silicon, with 6 conduction band minima, we obtain

$$J = \frac{2qkT}{\pi\hbar^2} m_t \sum_i v_{ril} T(E_{il}) ln\left\{\frac{1 + exp[(E_{Fl} - E_{il})/kT]}{1 + exp[(E_{Fr} - E_{il})/kT]}\right\} + \frac{4qkT}{\pi\hbar^2}\sqrt{m_t m_l}\sum_i v_{rit} T(E_{it}) ln\left\{\frac{1 + exp[(E_{Fl} - E_{it})/kT]}{1 + exp[(E_{Fr} - E_{it})/kT]}\right\} \quad 3\text{-}538$$

where the sums over longitudinal bound eigenstates and transverse bound eigenstates are done separately. For holes, the current is calculated using the equation

$$J = \frac{qkT}{\pi\hbar^2} m_{lh} \sum_i v_{ri\_lh} T(E_{i\_lh}) ln\left\{\frac{1 + exp[(E_{i\_lh} - E_{Fl})/kT]}{1 + exp[(E_{i\_lh} - E_{Fr})/kT]}\right\} + \frac{qkT}{\pi\hbar^2} m_{hh} \sum_i v_{ri\_hh} T(E_{i\_hh}) ln\left\{\frac{1 + exp[(E_{i\_hh} - E_{Fl})/kT]}{1 + exp[(E_{i\_hh} - E_{Fr})/kT]}\right\} \quad 3\text{-}539$$

where the sums over light hole and heavy hole bound eigenstates are done separately.

The expression υ is called the attempt frequency and is given by

$$v_r = \frac{\hbar k}{4m}\left[\left|\Psi(x_r)\right|^2 + \left|\frac{d\Psi}{dx}(x_r)\right|^2 \frac{1}{k^2}\right] \quad 3\text{-}540$$

It is evaluated at the semiconductor-oxide interface if quantum confinement occurs there. This quantum confined modification of the tunneling will be applied when you set the SCHRODINGER parameter and set quantum tunneling. This requires you to set CARRIERS=0 on the **METHOD** statement. Therefore, this model cannot be used if it is required to solve the current continuity equations.

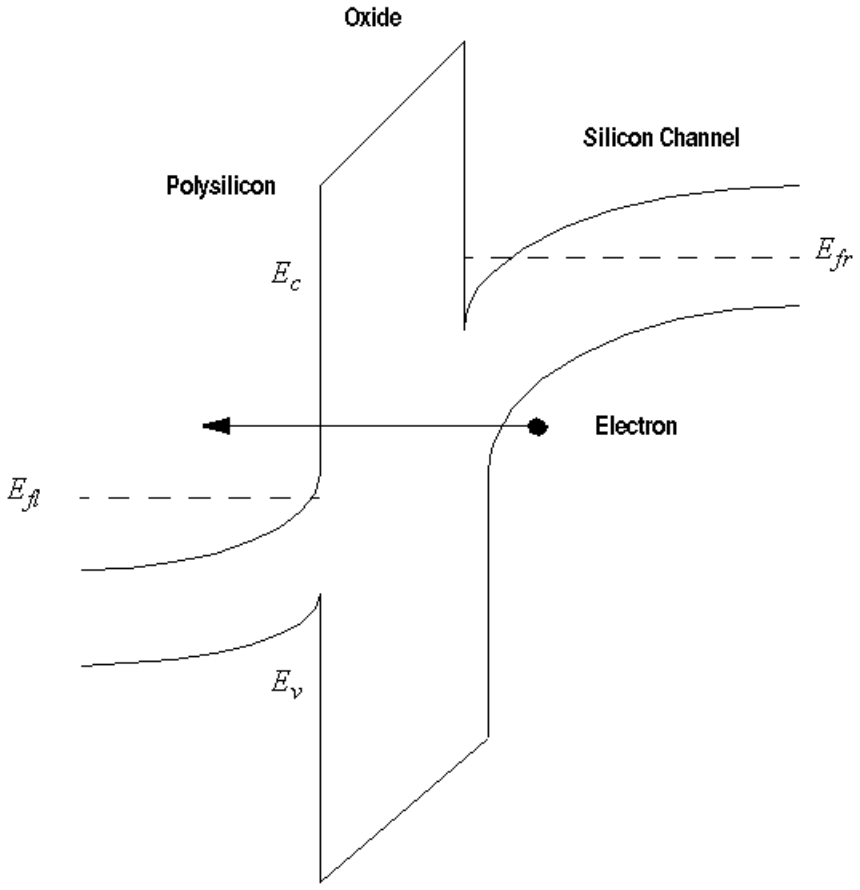


Figure 3-13: Schematic of band-to-band tunneling across a gate insulator with a polysilicon gate

Enabling the Models

To enable the quantum tunneling model for electrons, specify QTUNN.EL in the **MODELS** statement. To enable the quantum tunneling model for holes, specify QTUNN.HO in the **MODELS** statement. To enable Band-to-Band tunneling, specify QTUNN.BBT in the **MODELS** statement. In order to enable all three options, specify QTUNN in the **MODELS** statement.

Table 3-117 MODELS Statement: Quantum Tunneling Parameters

Parameter	Type	Default
QTUNN	Logical	False
QTUNN.BBT	Logical	False
QTUNN.EL	Logical	False
QTUNN.HO	Logical	False

QTUNN.EL and QTUNN.HO are available with the CARRIERS=0, CARRIERS=1, and CARRIERS=2 options on the METHOD statement. With CARRIERS=0, the semi-classical and quantum-confined variants are available. QTUNN.BBT is only available as part of a semi-classical calculation with either CARRIERS=0, CARRIERS=1 or CARRIERS=2.

To enable the self-consistent versions of the quantum tunneling models, specify the QTNLSC.EL and QTNLSC.HO parameters on the **MODELS** statement for electron and hole tunneling respectively. The Band-to-Band tunneling option is enabled using QTNLSC.BBT. You can enable all three models using the QTUNNSC parameter on the **MODELS** statement.

Table 3-118 MODELS Statement: Self-consistent Quantum Tunneling Parameters

Parameter	Type	Default
QTNL.DERIVS	Logical	false
QTNLSC.EL	Logical	false
QTNLSC.HO	Logical	false
QTNLSC.BBT	Logical	false
QTUNNSC	Logical	false

For the self-consistent implementations, convergence can be somewhat poor in some circumstances, particularly for high values of tunneling current. In this case, you can set the QTNL.DERIVS parameter on the **MODELS** statement. This will include extra terms in the Jacobian matrix, which should improve convergence. It will also increase the time needed to solve each iteration.

Mesh Considerations

The tunneling current model is quasi one-dimensional and the tunneling current is evaluated on a series of parallel or nearly parallel slices through the insulator. Implementation details therefore depend on the type of mesh. If the mesh was generated by Atlas meshing commands, then everything including the direction of tunneling is determined automatically. If the mesh was created by another Silvaco product, for example Athena or DevEdit, then there are two options.

The first option is to use the **QTX.MESH** and **QTY.MESH** commands to create a rectangular mesh on which all quantum tunneling is calculated. Interpolation is used to couple between this supplementary mesh and the device mesh. By default, the tunneling will be assumed to be in the Y direction. You can change this by setting the QTUNN.DIR parameter on the **MODELS** statement to 1 instead of its default value of 0. Place the QT mesh to enclose the insulator and to not overextend into the conductor or semiconductor. The second option is to allow Atlas to automatically generate slices through the oxide layer. This is to be preferred if the oxide geometry is non-planar.

There are a choice of two algorithms for determining the slices. The default is to calculate the currents on slices constructed as being locally perpendicular to each semiconductor-oxide segment on the interface. To enable the alternative algorithm, use the SHAPEOX parameter on the **MODELS** statement. In this case, slices giving the shortest distance to the contact are constructed from each node on the semiconductor-insulator interface.

If you specified the Schrodinger equation using the NEW.SCHRODINGER parameter and solved on a rectangular mesh specified by the **SPX.MESH** and **SPY.MESH** commands, then the Quantum tunneling current must also be calculated on the rectangular mesh defined by the **QTX.MESH** and **QTY.MESH** commands. Best results will be obtained if the mesh lines in the direction of tunneling are roughly coincident and if the SP mesh ends at the semiconductor-insulator interface.

Table 3-119 QTX.MESH and QTY.MESH Statements

Parameter	Type	Units	Default
NODE	Int		-999
LOCATION	Real	microns	-999
X	Real	microns	-999
RATIO	Real	microns	1.0
SPACING	Real	microns	-999

Table 3-120 MODELS Statement

Parameter	Type	Default
QTUNN.DIR	Real	0
SHAPEOX	Logical	False

Contact Specifications

Carriers that tunnel through an oxide are added to the current of the electrode into or from which they flow. If they tunnel into a polysilicon region with several contacts attached, then the tunnel current is added to the electrode that is nearest to the segment of the oxide/polysilicon interface across which the current is being calculated. The NEARFLG parameter in the **MODELS** statement is automatically set for quantum tunneling. Therefore, the algorithm used to obtain the nearest electrode is the same as when you set NEARFLG. To exclude any electrode from this algorithm, set the EXCLUDE_NEAR flag in the **CONTACT** statement.

You can also set the effective mass to use inside each contact for the tunneling current calculation using the QTUNN.CMASS (electrons) and QTUNN.VMASS (holes) parameters in the **CONTACT** statement. If the contact is polysilicon, then the default effective mass is either the conduction band or valence band density of states effective mass depending on its dopant specification.

Table 3-121 CONTACTS Statement

Parameter	Type	Units	Default
EXCLUDE_NEAR	Logical		False
QTUNN.CMASS	Real		1.0
QTUNN.VMASS	Real		1.0

To calibrate the tunneling current, use the effective mass in the oxide region. You can set this by using either the MC or ME.TUNNEL parameters on the **MATERIAL** statement for electrons and the MV or MH.TUNNEL parameters on the **MATERIAL** statement for holes.

For example:

```
MATERIAL MATERIAL=OXIDE MC=0.6 MV=0.2
```

In addition to this direct tunneling model and Fowler Nordheim model, there are several quantum tunneling models included in Atlas for compatibility with other products. These are mutually exclusive and the tunneling current outputs to a variable I_{tnl} (A) or J_{tnl} (A/um) regardless of model chosen.

The direct quantum tunneling current density associated with each interface node can be output to any standard structure file using the parameters QTUNN.EL, QTUNN.HO, and QTUNN.BBT on the **OUTPUT** statement. This applies to both post-processing and self-consistent versions of the direct quantum tunneling model.

When using the direct quantum tunnelling model in conjunction with the Bohm Quantum Potential or Density Gradient models, some correction is made for the quantum confinement effects introduced by these models. In Equations 3-430 and 3-431, the lower limit of energy integration is usually the band edge in the semiconductor at the interface with the insulator. It is changed to be the ground state of the confining potential well formed by the band edge. This is modeled as a triangular potential well. The ground state of the potential well is approximated as

$$\left[\frac{9}{8}\pi\right]^{2/3}\left(\frac{(qF\hbar)^2}{2m_0 m^*}\right)^{1/3} \qquad 3\text{-}541$$

relative to the minimum of the well. The quantity F is the local electric field at the interface. If the potential is not confining, then the Field will be zero or negative, and the ground state is set to zero. By integrating over a continuous energy range beginning at the ground state energy, quantization effects in the channel are introduced into the gate tunnel current model for a drift-diffusion solution.

Schenk Oxide Tunneling model

Another approximate tunneling model is based on the Gundlach model [110] and includes the effects of barrier lowering due to the image force potentials [277]. As such, it is especially suited for tunneling through ultra-thin gate oxides. It involves mapping from the energy barrier arising from the actual barrier profile plus the correction due to the image force to an effective trapezoidal barrier. The exact Transmission Coefficient for a trapezoidal barrier can then be obtained from

$$T(E) = \frac{2}{1 + g(E)} \qquad 3\text{-}542$$

where

$$g(E) = \frac{\pi^2}{2}\left[\frac{m_s k_c}{m_c k_s}(Bi'_d Ai_o - Ai'_d Bi_o)^2 + \frac{m_c k_s}{m_s k_c}(Bi_d Ai'_o - Ai_d Bi'_o)^2\right. \qquad 3\text{-}543$$
$$\left. + \frac{m_c m_s}{\lambda_o^2 m_{ox}^2 k_c k_s}(Bi'_d Ai'_o - Ai'_d Bi'_o)^2 + \frac{\lambda_o^2 m_{ox}^2 k_c k_s}{m_c m_s}(Bi_d Ai_o - Ai_d Bi_o)^2\right]$$

where k_c is the wavevector in the contact, k_s is the wavevector in the semiconductor, m_c is the effective mass in the contact, m_{ox} is the effective mass in the oxide and m_s is the effective mass in the semiconductor.

Complication arises because the image potential is not of a simple functional form, it is given by

$$E_{im}(x) = \frac{q^2}{16\pi\varepsilon_{ox}} \sum_{n=0}^{\infty} (k_1 k_2)^n \times \left[\frac{k_1}{nd + x} + \frac{k_2}{d(n+1) - x} + \frac{2k_1 k_2}{d(n+1)}\right] \qquad 3\text{-}544$$

with

$$k_1 = \frac{\varepsilon_{ox} - \varepsilon_M}{\varepsilon_{ox} + \varepsilon_M} = -1, \qquad k_2 = \frac{\varepsilon_{ox} - \varepsilon_s}{\varepsilon_{ox} + \varepsilon_s} \qquad 3\text{-}545$$

where ε is the relative dielectric permittivity (of the material indicated by the subscript), d is the oxide thickness and x is the position in the barrier. The sum can be evaluated numerically and the potential added to the barrier potential. This allows one to evaluate the action of an electron of incident energy E when moving through this barrier by numerically integrating the barrier energy minus the electron energy as a function of distance between the classical turning points of the motion.

This can be equated with the action of the carrier moving through a trapezoidal barrier with the barrier height as a fitting parameter.

$$S_{eff}(E) = S_{im+actual}(E) \quad \text{3-546}$$

This results in an effective trapezoidal barrier height as a function of electron incident energy E. This is evaluated at three different energies and these values are used to calculate the effective barrier height as a function of electron energy. The interpolation formula used is

$$\Phi_B(E) = \Phi_B(E_0) + \frac{\Phi_B(E_2) - \Phi_B(E_0)}{(E_2 - E_0)(E_1 - E_2)}(E - E_0)(E_1 - E) - \frac{\Phi_B(E_1) - \Phi_B(E_0)}{(E_1 - E_0)(E_1 - E_2)}(E - E_0)(E_2 - E) \quad \text{3-547}$$

For a given electron incident energy, you use Equation 3-547 to calculate the effective barrier height and then use Equation 3-543 to calculate the Transmission Probability, *T(E)*. This is placed in an equation like Equation 3-534 and the integration over the range of tunneling energies is carried out to give the tunneling current.

The model has been implemented as both a post-processing step and alternatively, as being solved self-consistently with the current continuity equations. To enable the post-processing option, specify SCHENK on the **MODELS** statement. To enable the self-consistent version, use SCHKSC with the **MODELS** statement. You can enable them separately for electrons and holes if required using the parameters SCHENK.EL, SHENCK.HO (post-processing) and SCHKSC.EL, SCHKSC.HO (self-consistent) on the **MODELS** statement. The values of effective mass and permittivity will affect the quantity of tunneling current as will the electron affinities and work functions of the materials involved.

Table 3-122 Schenk Oxide Tunneling Flags

Parameter	Type	Default	Units
SCHENK.EL	Logical	False	
SCHENK.HO	Logical	False	
SCHENK	Logical	False	
SCHKSC.EL	Logical	False	
SCHKSC.HO	Logical	False	
SCHKSC	Logical	False	
SCHENK.BBT	Logical	False	

Gate Tunneling Models for SONOS Type Structures

One approach to obtaining Non-Volatile Memories with low operating Voltages is to use a SONOS structure. The (S)emiconducting channel has a thin layer of tunnel (O)xide grown on it, followed by a thin layer of silicon (N)itride, and then followed by a thicker blocking or capping layer of (O)xide, and finally a (S)emiconducting polysilicon gate. The nitride has trapping levels located within it and the nitride-oxide band offset allows charge to be accumulated in the nitride layer.

The use of a charge-trapping Silicon nitride layer as a basis for a Non-Volatile memory device goes back to 1967 [339]. There has recently been a lot of interest in SONOS devices because of the continuing trend to smaller device dimensions.

To achieve low-voltage, low-power operation, you need to have an ultrathin tunneling layer so that charging by direct tunneling is possible. The presence of a defect in an ultrathin oxide tunneling layer can cause the complete failure of a floating gate device because it will fully discharge. In a SONOS device, the charge is stored in traps in the Silicon Nitride and consequently a defect will have only a localized effect. Thus, the SONOS structure is potentially more reliable [349].

Atlas has three different SONOS models for attempting to model the behavior of these devices.

The first is the FNONOS model that relies on using embedded floating gates in the Nitride layer but can model capture efficiency and trap saturation.

The second is the SONOS model that assumes that the trapping of charge in the Silicon Nitride occurs at the interface with the tunneling oxide.

The third, and most complete, is the DYNASONOS model. This model includes several tunneling mechanisms, carrier transport and trap dynamics in the Silicon Nitride layer. It assumes that the traps are evenly distributed throughout the Silicon Nitride layer. It is the most complete model for SONOS devices in Atlas.

All three models are described below.

FNONOS Model

To enable this model, specify FNONOS on the **MODELS** statement. To use the FNONOS model, you must either set the whole Silicon Nitride layer as a floating contact or embed floating contacts in the Silicon Nitride layer. Use the FLOATING flag on the **CONTACT** statement to to make an electrode into a floating contact. For each point in the channel-oxide interface, Atlas calculates the distance to the nearest point in the Silicon Nitride layer. The tunneling current for this point is then calculated as'

$$J_n = \mathrm{F.AE}\ \mathrm{E}^2 / factor1\, exp(-\mathrm{F.BE}\ factor2/\mathrm{E}) \qquad 3\text{-}548$$

where

$$factor1 = \left[1-(1-\mathrm{DV}/\mathrm{BH.FNONOS})^{\frac{1}{2}}\right]^2 \qquad 3\text{-}549$$

and

$$factor2 = \left[1-(1-\mathrm{DV}/\mathrm{BH.FNONOS})^{\frac{3}{2}}\right] \qquad 3\text{-}550$$

DV is the potential drop across the tunnel oxide layer and is calculated automatically by Atlas. The value BH.FNONOS is the Barrier height and if not specified directly Atlas will calculate it. This formula is calculated using WKB theory for the tunneling co-efficient through a trapezoidal barrier.

If DV > BH.FNONOS, then *factor1* and *factor2* are both set to be unity. In this case, Equation 3-548 is the same as the Fowler-Nordheim Expression (Equation 3-501).

ETA.FNONOS times the tunneling current is added to the nearest floating electrode, where ETA.FNONOS is the capture efficiency. A factor (1-ETA.FNONOS) times the tunneling current is added to the next nearest (gate) electrode. The efficiency can depend on the Charge state of the floating electrode itself. To enable this, set the NT.FNONOS parameter on the **MODELS** statement to a positive value. The efficiency is modified by an extra term

$$1.0 - Q_{floating}/(q\mathrm{NT\_FNONOS}) \qquad 3\text{-}551$$

where $Q_{floating}$ is the Floating gate charge density in C/micron. NT.FNONOS is the integrated trap density /micron. If this term is negative, then zero is used instead. This allows you to model the phenomenon of trap saturation.

Table 3-123 MODELS Statement

Parameter	Type	Default	Units
FNONOS	Logical	False	
ETA.FNONOS	Real	1.0	
BH.FNONOS	Real	3.07	eV
NT.FNONOS	Real	0.0	um^{-1}

SONOS Model

This is a model of intermediate complexity for modeling SONOS devices. It assumes that the trapped charge in the Silicon Nitride is at the interface of the Silicon Nitride and the surrounding insulator materials. The traps can be charged up and discharged by direct tunneling through the tunnel oxide, or by hot carrier injection (lucky electron or concannon models).

To enable this model, you include an **INTERFACE** statement with the parameter N.I specified. You must also explicitly suppress the DYNASONOS flag by specifying ^DYNASONOS. This is because the DYNASONOS model (see below) is the default. The gate stack materials must all be insulators. If they are changed to wide bandgap semiconductors, then the model will not work correctly.

You can view the trapped carrier density as a sheet charge density under the **Insulator Charge** field in TonyPlot.

The tunneling current is calculated using the direct quantum tunneling Equation 3-534 with the tunneling probability *T(E)* obtained using the WKB approximation. The quasi-Fermi energy E_{fl} in the Silicon Nitride is obtained from the energy level at the gate contact. This value is then clipped to ensure that it lies within the Silicon Nitride bandgap. The value of E_{fr} is the usual value of quasi-Fermi level near the interface of the channel and the oxide. The shortest paths between each point on the interface and the Silicon Nitride are used as the tunneling paths.

The value of *T(E)* is strongly dependent on the values of effective mass in the tunnel oxide. You can use the MC and MV parameters on the **MATERIAL** statement as fitting parameters.

If a hot carrier gate current model is selected (HEI, HEI, N.CONCANNON, or P.CONCANNON on the **MODELS** statement), then the hot carrier current charges points on the Silicon Nitride interface in a way, which is consistent with the motion of the hot carriers in the local electric field.

The limitations of this model are that:

- The trapped charge is constrained to be negative (holes can only erase).
- The trapped charge is localized at the Silicon Nitride boundary.
- Trap dynamics are not included.
- Charge transport in the Silicon Nitride is not included.

Its advantage is that it is a robust model and can model spatial variations in the trapped charge density.

DYNASONOS Model

To address some of the limitations of the SONOS model, the DYNASONOS model was developed by adapting features of various published models [254], [95], [60], [75], [96], [61].

To enable this model, you specify the N.I parameter on the **INTERFACE** statement. You can also specify the DYNASONOS parameter, although this is enabled by default (it has no effect unless you also specify N.I).

The N.I parameter causes every point on the interface between Nitride and other insulators to be treated as a double point, allowing easier implementation of the thermionic and direct tunneling models for these structures. This requires that the **INTERFACE** statement be placed at the end of the structure specification group of statements. The current across this interface is automatically modeled as a thermionic current.

You must also change all of the insulator materials in the gate stack to wide bandgap semiconductors using the SEMICONDUCTOR parameter on the **MATERIAL** statement. You also need to specify effective masses, mobilities, and effective densities of states in the gate stack materials with the **MATERIAL** statement.

The properties of the traps in the Silicon Nitride must be set using the **NITRIDECHARGE** statement. These parameters are now introduced. Atlas models the trap states in the Nitride as being either acceptor-like (for storing electrons) or donor-like (for storing holes), at a single discrete energy level below the Nitride conduction band (acceptor-like) or above the valence band (donor-like). The number of available trap states is assumed to be spatially uniform, with the number density of acceptor-like traps being set by the NT.N parameter in units of cm^{-3} and the number density of donor-like states being set by NT.P in the same units.

By default, the traps are initialized to be empty (uncharged) and are charged using a transient **SOLVE** statement. It is possible, however, to set up an initial distribution of charged traps by using the NIT.N and NIT.P parameters on the **SOLVE** statement. These set the trapped electron and trapped hole densities respectively to the specified value within the limits specified by the optional parameters: NIT.XMIN, NIT.XMAX, NIT.YMIN, and NIT.YMAX. The default values of these parameters are the maximum extents of the Nitride region. The effect of consecutive SOLVE statements with NIT.N or NIT.P is cumulative, allowing you to create a complicated profile of trapped charge density. This is useful for studying known trapped charge distributions.

To simulate the charging of the Nitride layer, you use a transient **SOLVE** statement. The Silicon Nitride layer can be charged by quantum mechanical tunneling or by hot carrier injection. Two types of tunneling are modeled. The first is tunneling to the Silicon Nitride conduction band and valence band. The second is direct tunneling to and from the trap levels. In the former case, some of the free carriers are captured by the trapping centers. For acceptor traps, the rate of capture of electron density is

$$\mathtt{SIGMAT.N} v_{th,n} n(\mathtt{NT.N} - n_t) \quad \text{3-552}$$

where SIGMAT.N is a capture cross-section in cm^2, $v_{th,n}$ is the electron thermal velocity in cm/s, n is the free electron density in cm^{-3} and n_t is the trapped electron density in cm^{-3}. Trapped electrons can be re-emitted to the conduction band, the rate for this process being

$$n_t e_n \quad \text{3-553}$$

where e_n is an emission rate in s^{-1}. There are also rates for hole capture from the valence band

$$\mathtt{SIGMAN.P} v_{th,p} p n_t \quad \text{3-554}$$

where p is the free hole density in cm^{-3} and $v_{th,p}$ is the hole thermal velocity in cm/s. To get an equilibrium solution, you need to include an hole emission term

$$(\mathtt{NT.N} - n_t) e_{eq,p} \quad \text{3-555}$$

where the hole emission rate $e_{eq,p}$ is calculated from the other parameters and the equilibrium carrier densities and cannot be independenly specified. The rate of direct trap charging due to tunneling is given by

$$P(s\mathtt{NT.N} - n_t) \quad \text{3-556}$$

where s is the Fermi-Dirac factor in the conduction band, and the tunneling rate per trap is P in units of per second (see Equation 3-559).

Putting these terms together we obtain an expression for the rate of the change of the local trapped electron density.

$$\frac{dn_t}{dt} = \mathtt{SIGMAT.N} v_{th,n} n(\mathtt{NT.N} - n_t) - \mathtt{SIGMAN.P} v_{th,p} p n_t + (\mathtt{NT.N} - n_t) \quad \text{3-557}$$

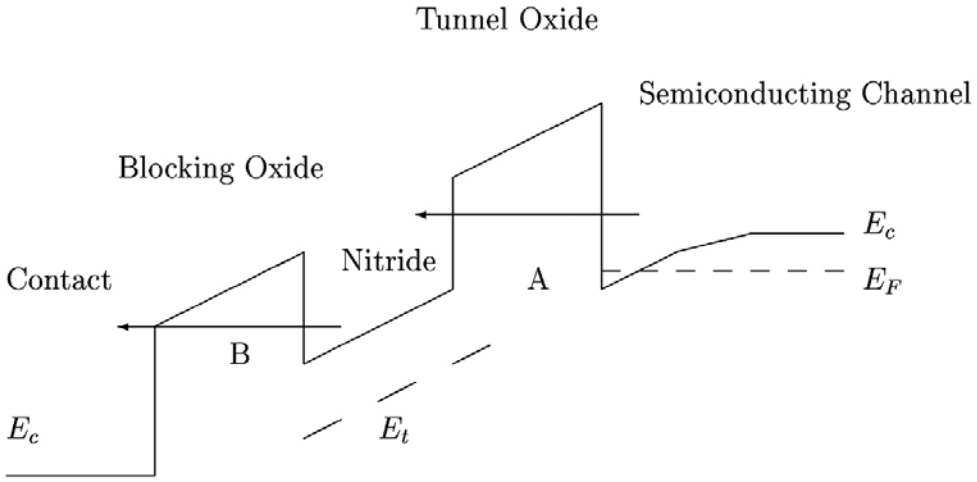
$$e_{eq,p} - n_t e_n + P(s\mathtt{NT.N} - n_t)$$

The equivalent expression for the rate of change of trapped hole density is

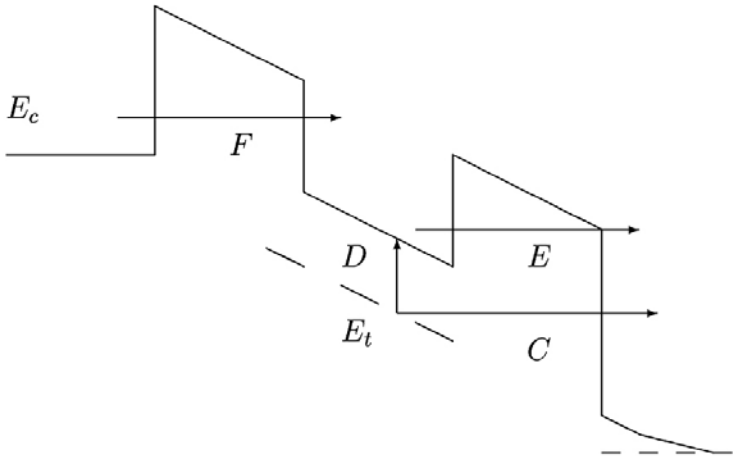
$$\frac{dp_t}{dt} = \mathtt{SIGMAT.P} v_{th,p} p(\mathtt{NT.P} - p_t) - \mathtt{SIGMAP.N} v_{th,n} n p_t + (\mathtt{NT.P} - p_t) \quad \text{3-558}$$

$$e_{eq,n} - p_t e_p + P(s\mathtt{NT.P} - p_t)$$

The tunneling terms are zero for any tunneling processes, which start or end in the band gap of the channel material. We now consider electron trapping. To charge the traps, there must be electrons in the Silicon Nitride and these are generated by direct quantum tunneling, shown schematically as process A in part a) of Figure 3-14, where the tunnelling transitions are allowed at energies greater than the conduction band energy in the channel. Any excess electrons in the Silicon Nitride may tunnel to the contact as shown schematically as process B. Clearly tunneling directly into the Silicon Nitride traps is not possible in this band lineup. For a band lineup during a typical erase operation, as in part b) of Figure 3-14, direct trap-to-channel tunneling is possible (process C). In process D, trapped electrons are emitted to the Silicon Nitride conduction band, drift to the interface with the tunnel oxide and then tunnel out of the Silicon Nitride in process E. Some electrons may be injected from the contact as shown in process F. There are analogous process for hole traps.



a. Conduction band configuration for trap charging



b. Conduction band configuration for trap erasing

Figure 3-14: Schematic of band structure and processes for SONOS model

For trap-to-channel tunneling, the range of states in the channel corresponding to the tunneling to or from a specific trap energy are such that the total energy must add up to the energy at the trap level. This means that we do an integral over transverse wavevector $k_{||}$, with the constraint that the transverse and perpendicular energies sum to the value of the trap energy. The electron band-to-trap tunneling rate in units of s^{-1} is

$$P = \frac{4\hbar m_{nitride}}{m_{oxide}^2}\int_0^\infty \frac{Hk_\perp}{\kappa_1^2 + k_\perp^2} WKB(E, k_{||}) \frac{\kappa_2^2}{(\kappa_2 + \kappa_3)^2} k_{||} dk_{||} \quad 3\text{-}559$$

where $H^2 = \frac{2m_{nitride}}{\hbar^2}$ ELEC.DEPTH, $k_\perp$ is the perpendicular wavevector at the channel side of the Channel/Oxide interface and κ_1 is the evanescent wavevector at the Oxide side of the Channel/Oxide interface. Similarly, κ_2 and κ_3 are evanescent wavevectors evaluated at the Nitride/Oxide interface and *WKB(E, $k_{||}$)* is the product of the WKB tunneling probabilities through the oxide and through part of the Nitride to the trap position. This is derived by extending the result of Lundstörm et al. [191] to a realistic bandstructure.

The electron emission rate (for acceptor traps) and hole emission rate (for donor traps) are important in determining the trapped carrier density after charging, and also the importance of process D in Figure 3-14. These rates are either set using the TAU.N and TAU.P parameters on the **NITRIDECHARGE** statement, or by enabling the Poole-Frenkel detrapping model by specifying the PF.NITRIDE flag on the **MODELS** statement. In the former case, the emission rates are constants e_n = 1/TAU.N and e_p = 1/TAU.P. In the latter case, the emission rates depend on the local value of field in the device. For electrons

$$e_n = \text{PF.B}\exp\left(-\left[\frac{\text{ELEC.DEPTH} - q\sqrt{qF/\pi\varepsilon}}{K_{BT}}\right]\right) \quad 3\text{-}560$$

where ELEC.DEPTH is the depth of the electron traps below the conduction band, F is the local electric field in V/m and ε is the dielectric permittivity of the Silicon Nitride in F/m. The overall rate is proportional to PF.B, which has a default value of 10^{13}/s.

If you want to use the Poole-Frenkel model without modifying ELEC.DEPTH, then you can use PF.BARRIER. If PF.BARRIER is specified, then it is used in place of ELEC.DEPTH in Equation 3-560.

The parameters that are set on the **NITRIDECHARGE** statement are summarized in Table 3-124, along with their default values. Parameters that are relevant to electron traps (acceptor-like) are only used if NT.N is greater than zero, and parameters that are relevant to hole traps (donor-like) are only used if NT.P is greater than zero.

Table 3-124 NITRIDECHARGE Statement

Parameter	Type	Default	Units
NT.N	Real	0.0	cm^3
NT.P	Real	0.0	cm^3
TAU.N	Real	1.0e300	s

Table 3-124 NITRIDECHARGE Statement			
Parameter	**Type**	**Default**	**Units**
TAU.P	Real	1.0e300	s
ELEC.DEPTH	Real	-999.0	eV
HOLE.DEPTH	Real	-999.0	eV
SIGMAN.P	Real	1.0e-15	cm^2
SIGMAP.N	Real	1.0e-14	cm^2
SIGMAT.N	Real	1.0e-16	cm^2
SIGMAT.P	Real	1.0e-14	cm^2
PF.BARRIER	Real		eV
PF.B	Real	1.0e13	Hertz

In the Silicon Nitride, the fully transient current continuity equations are solved, self-consistently along with the Equations 3-557 (if NT.N is non-zero) and 3-558 (if NT.P is non-zero) for the trap occupancies. The Poisson Equation is also solved to self-consistently include the effect of nitride trap charging.

The default tunneling path is determined as being the shortest distance between each node on the Nitride–Tunneling Insulator interface and the channel, or the shortest distance between each node on the Nitride–Blocking Insulator interface and the gate. The path chosen must be outwardly oriented. In other words, it cannot cross any part of the Nitride layer itself.

For some geometries evaluating the tunnelling over only the shortest distance path is too restrictive, as for example in Figure 3-15. Atlas allows you to specify a range of angles for tunneling path. These angles are relative to the shortest distance path. If you specify NOS.ANGLE on the **INTERFACE** statement, then all valid tunneling paths within an angle of ± NOS.ANGLE degrees will be considered for every point on the interface between the Nitride and the tunneling oxide. For points on the interface between the Nitride and the Blocking oxide, the same algorithm is used if the NOM.ANGLE parameter is specified on the **MODELS** statement. In Figure 3-15, point A on the nitride/oxide interface has its shortest tunneling path to point C. If NOS.ANGLE is specified, then it will also consider linear tunneling paths between A and points like B and D, so long as the angle, θ, between these paths and A-C is less than NOS.ANGLE.

To restrict the calculation to paths that have a significant contribution, the parameters NOS.DIST and NOM.DIST are available. These reject any tunneling path longer than NOS.DIST or NOM.DIST microns. Although the contribution of a tunneling path depends on other factors as well as tunneling path length, applying this restriction eliminates much unnecessary calculation. Atlas prints out the average number of tunneling paths from the nitride interface to both semiconductors and contacts. This information can be used to fine tune the values of NOS.ANGLE, NOM.ANGLE, NOS.DIST, and NOM.DIST.

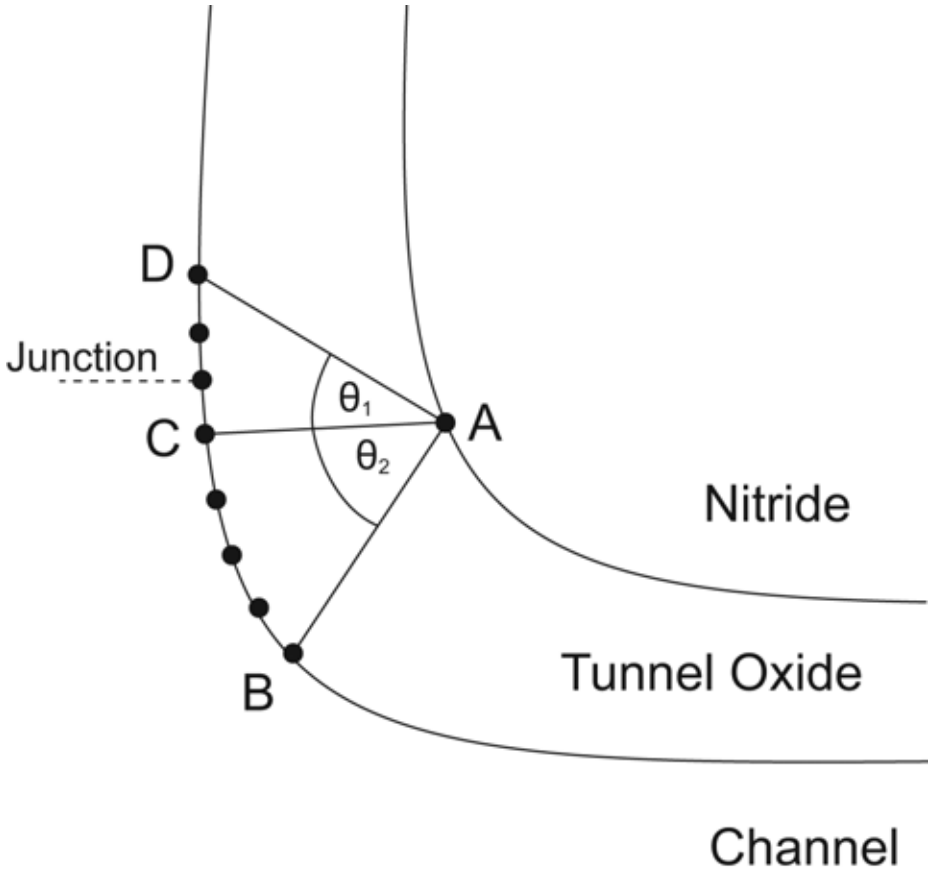


Figure 3-15: Use of NOS.ANGLE parameter. All linear paths so that O_1 <NOS.ANGLE or O_2 <NOS.ANGLE are considered.

In order to simulate hot carrier injection into the Silicon Nitride, you may enable the existing hot carrier models HEI/HHI for drift diffusion simulations or N.CONCANNON/P.CONCANNON with Energy balance simulations. In general, N.CONCANNON and P.CONCANNON should give physically more reasonable results. An additional model is available to use in drift-diffusion simulations. It is enabled by specifying N.HOTSONOS (for electrons) and P.HOTSONOS (for holes) on the **MODELS** statement. It uses the parallel field along the channel/insulator interface to obtain an effective carrier temperature using the formulae:

$$T_n = F \ \texttt{IG.LRELE}/(1.5K_B)$$

$$T_p = F \ \texttt{IG.LRELH}/(1.5K_B) \qquad \text{3-561}$$

where F is the parallel field along the interface.

It then uses these effective carrier temperatures in the Concannon expressions for Hot carrier gate current.

The final distribution of trapped charge depends on the distribution of carrier generation and recombination, the parameters of the trap equation and the simulation time. The trapped carrier concentrations are output automatically to any structure file as Trapped Insulator e- Concentration and Trapped Insulator h+ Concentration. The instantaneous rates of carrier generation in the Silicon Nitride layer can be output to any structure file by specifying SONOS.RATES on the **OUTPUT** statement. Both through barrier direct tunneling and trap-to-channel tunneling are included and are given as a generation term in units of cm^{-3} s^{-1}. The net rate of electron capture from the conduction band and hole capture from the valence band are calculated from Equations 3-557 and 3-558 and output as recombination rates in units of cm^{-3} s^{-1} if SONOS.RATES is specified.

You can obtain the net trapped charge density (electron density - hole density) by specifying the SONOS.CHARGE statement on the **PROBE** statement. Furthermore, you can view the overall tunneling current injected into the Nitride layer from the channel as a function of time by specifying SONOS.CURR on the LOG statement. Choosing this option will also output the tunnel current entering the external contact from the Nitride layer. The are refered to as the SONOS Tunneling Insulator Current and SONOS Blocking Insulator Current respectively and are given in units of A/μm.

In steady state, the trapped charge in the Silicon Nitride layer remains fixed at its charged value, allowing you to do threshold voltage shift calculations. If the gate stack materials remain as wide bandgap semiconductors, the large stored fixed charge combined with weakly coupled quasi-Fermi levels mean that SOLVE INIT may wrongly introduce a compensating free carrier charge. To avoid this, either revert the gate stack materials to insulators or specify the SONOS flag on the **SOLVE** statement when you also specify INITIAL.

The BESONOS model is an extension of the DYNASONOS model to SONOS devices with Band-Engineered tunnel layers. It is enabled by setting the parameter BESONOS on the INTERFACE statement together with the DYNASONOS parameter. The BESONOS model uses the direct quantum tunneling model for calculating the tunneling current through the layered insulator tunnel stack. This model is described in Section 3.6.7 “Gate Current Models” and can calculate tunnel current through a stack of materials with different band offsets and effective masses.

Some Band-Engineered tunnel layers may contain ultra-thin Nitride layers, which do not trap charge. In order to specify which Silicon Nitride regions to charge in the BESONOS model, you must use the TRAPPY parameter on the **REGION** statement. Only Silicon Nitride regions that have the TRAPPY parameter set will be able to have trapped charge stored in them. The TRAPPY parameter is only active for the BESONOS model.

Table 3-125 Parameters Relevant to the SONOS MODEL

Statement	Parameter	Type	Default
INTERFACE	BESONOS	Logical	False
INTERFACE	N.I	Logical	False
INTERFACE	DYNASONOS	Logical	True
LOG	SONOS.CURR	Logical	False
INTERFACE	NOS.ANGLE	Real	0.0

Table 3-125 Parameters Relevant to the SONOS MODEL			
Statement	**Parameter**	**Type**	**Default**
INTERFACE	NOM.ANGLE	Real	0.0
INTERFACE	NOS.DIST	Real	1.0
INTERFACE	NOM.DIST	Real	1.0
MODELS	N.HOTSONOS	Logical	False
MODELS	P.HOTSONOS	Logical	False
OUTPUT	SONOS.RATES	Logical	False
PROBE	SONOS.CHARGE	Logical	False
REGION	TRAPPY	Logical	False
SOLVE	NIT.N	Real	0.0
SOLVE	NIT.P	Real	0.0
SOLVE	NIT.XMAX	Real	0.0
SOLVE	NIT.XMIN	Real	0.0
SOLVE	NIT.YMAX	Real	0.0
SOLVE	NIT.YMIN	Real	0.0
SOLVE	SONOS	Logical	False

Floating Gate to Control Gate (FGCG) Current

In a conventional Flash memory device structure, it is usual to neglect the tunneling current between the control gate and the floating gate. This is because the insulating layer separating the floating gate from the channel (tunnel oxide) is much thinner than that separating the floating gate from the control gate (barrier oxide). This will typically result in the dominance of the channel to floating gate tunneling current, at least under normal operating conditions.

As flash memory technology develops, it may be necessary to quantify the floating gate to control gate tunneling currents [75]. Atlas incorporates this capability with a new model called the FGCG model. This model calculates the tunneling currents between the floating gate and the semiconducting channel and between the floating gate and the control gate. Both of these currents can be taken into account when calculating the charge state of the floating gate. The model also applies to more complicated geometries where there are several floating gates and/or several control gates.

The tunneling currents are calculated using the direct tunneling formula, Equation 3-534, with the Transmission co-efficient being obtained using the WKB approximation.

This formula includes the carrier statistics and automatically determines the sign of the currents using the Fermi levels in the gates and quasi-Fermi in the channel.

To enable the model for electron tunneling, use the parameter E.FGCGTUN on the **MODELS** statement. To enable the model for hole tunneling, use the parameter H.FGCGTUN on the **MODELS** statement.

It is also important to specify how the floating gate to control gate current is output and how it affects the charge state of the floating gate. This is controlled by the CGTUNN parameter on the **CONTACT** statement. This only has an effect on a floating gate. If it is set (default), then the tunneling current attributed to the floating gate is obtained by calculating the current from that floating gate to every control gate.

This is then subtracted from the channel to floating gate current to give a net current. This net current is the one output. It is also the one used in the calculation of the amount of charging during a transient simulation.

If CGTUNN is explicitly cleared for a floating gate, then only the channel to floating gate current is used for the charging calculation and output.

Note: The tunneling current at a control gate is the sum of channel to control gate current, plus the floating gate to control gate current from all floating gates in the device. This is not affected by the CGTUNN parameter because the CGTUNN parameter is only effective for floating contacts.

A specific example is shown in Figure 3-16, where there are 3 control gates and 2 floating gates. The following CONTACT statements are issued:

```
CONTACT NAME=fgate1 FLOATING WORKF=4.28 CGTUNN
CONTACT NAME=fgate2 FLOATING WORKF=4.28 ^CGTUNN
```

so that CGTUNN is set for fgate1 and explicitly cleared for fgate2. The currents attributed to each electrode are calculated by Atlas as follows:

```
jfgate1 = jchannel1 - jcg1fg1 - jcg2fg1 - jcg3fg1
jfgate2 = jchannel3
jcgate1 = jcg1fg1 + jcg1fg2
jcgate2 = jchannel2 + jcg2fg1 + jcg2fg2
jcgate3 = jcg3fg1 + jcg3fg2
```

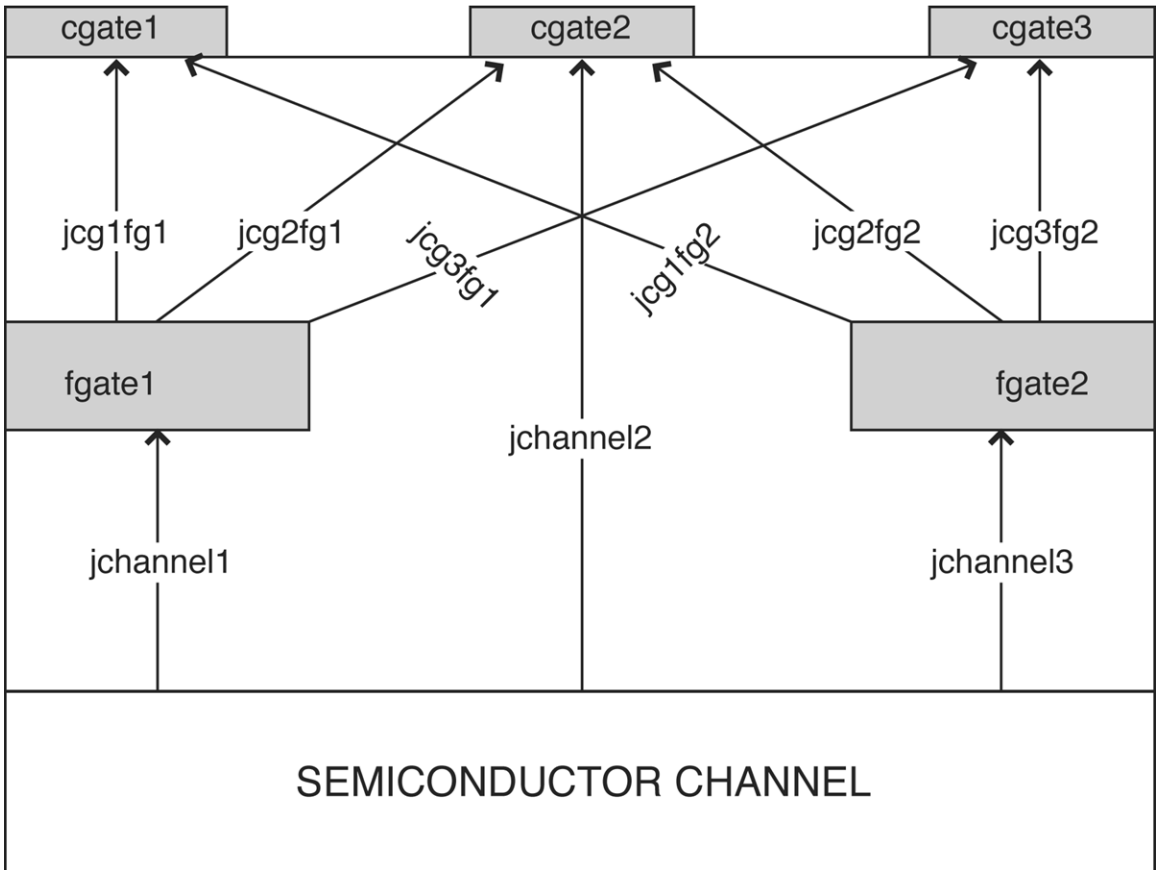


Figure 3-16: Example Structure that can be Modeled with FGCG Model, Showing the Current Components

The intention is that the model is used with CGTUNN set on every floating gate. You are, however, given the ability to see the effect of leaving out the control gate components of the floating gate current by clearing CGTUNN.

When used in transient mode, the floating gate tunneling currents are integrated over time to give the charges on the floating gates. Because the floating gate charge affects the floating gate voltage, which in turn affects the floating gate to control gate current, there is strong coupling between floating gate to control gate current and floating gate charge. For large timesteps this can cause numerical instability.

One approach to avoid this problem is to control the maximum timestep allowed using the DT.MAX parameter on the **METHOD** statement. The recommended alternative, however, is to specify FGCNTRL on the **METHOD** statement, which invokes an algorithm to control step size based on the relative size of the charging current. There are two parameters that can be used to control the behavior of this stabilization algorithm. One is QMAX.FGCNTRL, which is in units of electronic charge. If the charge on the floating gate is less than QMAX.FGCNTRL, then the algorithm will not be applied. The other parameter is RAT.FGCNTRL. This controls how much the timestep is reduced by the algorithm. Reduce it from the default value of 1.0 to give better control, but longer runtime.

For example, the following statement will enable the stability algorithm for those floating gateshaving an absolute charge of more than 8.01×10^{-17} Coulombs, and give a high level of stability due to the low value of RAT.FGCNTRL.

```
METHOD FGCNTRL RAT.FGCNTRL=0.05 QMAX.FGCNTRL=500
```

The parameters E.FGCGTUN and H.FGCGTUN enable the FGCG model. This will not include the current tunneling from the semiconducting channel into the current continuity equations in the semiconductor. The parameters E.SC.FGCGTUN and H.SC.FGCGTUN enable the FGCG model. This will include this current in the current continuity equations. If you set also CGTUNN is for every floating gate in the device, then overall current continuity for the device will be attained when you set the E.SC.FGCGTUN and H.SC.FGCGTUN parameters.

Table 3-126 shows the parameters used in the FGCG model.

Table 3-126 FGCG Model Parameters

Statement	Parameter	Default	Units
MODELS	E.FGCGTUN	False	
MODELS	H.FGCGTUN	False	
MODELS	E.SC.FGCGTUN	False	
MODELS	H.SC.FGCGTUN	False	
CONTACT	CGTUNN	True	
METHOD	FGCNTRL	True	
METHOD	RAT.FGCNTRL	1.0	
METHOD	QMAX.FGCNTRL	100	

Metal-Insulator-Metal Direct Tunneling

The simulation of some classes of device, for example Metal-Insulator-Metal tunnel diodes, requires the calculation of the tunneling current between electrodes. Atlas has the MIMTUN model for this class of device. The MIMTUN model calculates the tunneling paths between electrodes and obtains the tunnel currents using the Direct Quantum Tunneling model of Equation 3-534.

The quasi-Fermi levels are defined by the biases on the electrodes. The tunneling current can be calculated through a single or multi-layer insulator stack. The insulator materials can also be modeled as wide bandgap semiconductors using the SEMICONDUCTOR parameter on the **MATERIAL** statement. It is possible to model electrodes having different workfunctions by using the WORKF parameter on the **CONTACT** statement. To enable this model, specify MIMTUN on the **MODELS** statement.

It is recommended that the effective masses of the insulator materials are explicitly assigned using the MC and MV parameters on the **MATERIAL** statement.

MIMTUN enables tunneling for both holes and electrons, including the band-to-band component. To enable tunneling for electrons above the Fermi energy, specify MIMTUN.EL on the **MODELS** statement. To enable tunneling for holes below the Fermi energy, specify MIMTUN.HO on the **MODELS** statement. To model the tunneling of electrons from below the Fermi-level on the more negatively biased electrode to hole states above the Fermi energy on the more positively biased electrode, specify MIMTUN.BBT. This is usually the dominant component.

Band structure data is interpolated onto the tunneling paths. The number of regularly spaced points per tunneling path can be specified. Use the MIMSLICEPTS parameter of the **MODELS** statement to change the number of points.

Table 3-127 Parameters for the MIMTUN Model

Statement	Parameter	Type	Default
MODELS	MIMTUN	Logical	False
MODELS	MIMTUN.EL	Logical	False
MODELS	MIMTUN.HO	Logical	False
MODELS	MIMSLICEPTS	Real	51

Ielmini Inelastic Trap Assisted Tunneling Model

A common problem with non-volatile memory devices is the increase in tunneling current after a number of write-erase cycles. This Stress-Induced Leakage Current (SILC) is present even at low biases and causes degradation of the device retention time. It is believed to be mediated by traps that are created in the gate insulator by the repeated electrical stressing. The current is therefore assumed to be a two-step tunneling process, where the trap levels form an intermediate state for the tunneling process. Ielmini et al developed a model that can take into account both transient [137] and steady state simulations [138]. This model does not simulate the creation of oxide traps under stressing, rather it allows the simulation of the SILC current for a predefined trap distribution. We consider a cross-section through a typical gate stack with the gate at $x = 0$ and the channel at $x = L$. The gate stack may consist of one or more regions of insulator or wide bandgap semiconductor.

For a trap with a discrete energy level $E_t(x)$ and a spatial distribution $N_t(x)$ [cm^{-3}], where x is its spatial position within the gate insulator stack, four current contributions per carrier type can be defined. These are capture from the gate, emission to the gate, capture from the channel, and emission to the channel. The relationship between capture and emission currents, allowing for inelastic tunneling is derived by Ielmini et al [137]. Using this relationship for electrons, the current from the left hand side into an infinitesimal width Δx of the traps at position x can be written as

$$J_L(x) = QN_t b_L(E_t, E_{Fl}, x)(f_L(E_t, E_{Fl}, x) - f_t(E_t, x))\Delta x \quad \text{3-562}$$

The effective capture rate from the left, b_L, is obtained as

$$b_L = \frac{4\pi m^* K_B T\sigma}{h^3 f_L(E_t, E_{Fl}, x)} \int_{E_{min}}^{E_{max}} \log\left(1 + \exp\left[\frac{-(E - E_{Fl})}{K_B T}\right]\right) W(E, x)dE \quad \text{3-563}$$

where σ is the trap cross-section, m* is the effective mass, K_B, h are Boltzmann's constant and Planck's constant respectively, and T is the device temperaure. The integrand consists of a statistical factor depending on the left hand side quasi-Fermi level, E_{fl}, and a tunneling probability factor $W(E, x)$ for tunneling to point x. Similarly, the expression for the current from the right hand side into the same infinitesimal width Δx of the traps at position x is

$$J_R(x) = QN_t b_R(E_t, E_{Fr}, x)(f_R(E_t, E_{Fr}, x) - f_t(E_t, x))\Delta x \quad \text{3-564}$$

where b_R is the effective capture rate from the right, given by

$$b_R = \frac{4\pi m^* K_B T\sigma}{h^3 f_R(E_t, E_{Fr}, x)}\int_{E_{min}}^{E_{max}} \log\left(1 + \exp\left[\frac{-(E - E_{Fr})}{K_B T}\right]\right) W(E, L - x)dE \quad \text{3-565}$$

where $W(E, L - x)$ is the tunneling probability factor from position L to position x. The probabilities f_L and f_R are the probabilities of electrons being at energy E_t at the left and right edges of the oxide respectively. The probability of the trap being occupied by an electron is f_t. The minimum value of energy E_{min} in the evaluation of b_L is the maximum of either the trap energy at position x or the electron energy at the left hand side of the oxide. The maximum energy is the greater of the conduction band energy at position x or the conduction band energy at the left hand side of the oxide. The situation is analagous for the evaluation of b_R. The expressions for b_L and b_R differ from those in Ielmini et al [137] by the factors f_L and f_R being included in the denominator in order to avoid underflow errors for trap energies a long way outside the Silicon bandgap.

For the traps of infinitesimal width Δx about x and in the general time dependent case, the charge continuity gives

$$QN_t(x)\Delta x\frac{\partial f_t(E_t, x)}{\partial t} = J_L + J_R \quad \text{3-566}$$

In steady state, the trap occupancy is constant and so the condition reduces to $J_L + J_R = 0$, giving the steady state trap occupancy at each position.

$$f_t = \frac{f_L b_L + f_R b_R}{b_L + b_R} \quad \text{3-567}$$

Thus, the total trap assisted tunneling current can be evaluated by using the value of $f_t(x)$ in Equations 3-562 and 3-564 and summing the contributions from each length Δx at position x_i

$$J_{ITAT} = \sum_i J_L(x_i) = -\sum_i J_R(x_i) \quad \text{3-568}$$

where the sum is across the oxide thickness.

There is an analagous expression for hole currents (working with hole occupation probabilities and hole energy) and the models that consider electron and hole current separately are refered to as the ITAT models. For traps that coincide with the bandgap of the channel material, there is a greater likelihood of the trap level bringing about recombination. Therfore, the RTAT model can also be used. This takes into account both the electron and hole currents from left and right (J_{Le}, J_{Lh}, J_{Re}, J_{Rh}, and so Equation 3-566 becomes

$$QN_t(x)\Delta x\frac{\partial f_t(E_t, x)}{\partial t} = J_{Le} - J_{Lh} + J_{Re} - J_{Rh} \quad \text{3-569}$$

Thus, the probability of a trap containing an electron becomes

$$f_t(x) = \frac{f_{Le}b_{Le}(x) + f_{Re}b_{Re}(x) + b_{Lh}(x)(1 - f_{Lh}) + b_{Rh}(x)(1 - f_{Rh})}{b_{Le}(x) + b_{Re}(x) + b_{Lh}(x) + b_{Rh}(x)} \quad \text{3-570}$$

where f_{Le} is the probability of finding an electron at energy E_t(x) at the left hand contact and f_{Lh} the probability of finding a hole at that energy. In general, it impossible to write f_{Le}= 1 - f_{Lh} because the quasi-Fermi levels for electrons and holes are in general different. The same applies to f_{Rh} and f_{Re}, and after substituting the value of f_t(x) into the current equations we obtain

$$J_{RTAT} = \sum_i (J_{Le}(x_i) - J_{Lh}(x_i)) = -\sum_i (J_{Re}(x_i) - J_{Rh}(x_i)) \quad \text{3-571}$$

The charge of the carrier type is correctly included when evaluating Equation 3-571 (i.e., an equal flux of electrons and holes will result in zero current). The implementation of this model in Atlas requires an electrode on one side of the oxide and a semiconducting channel on the other. It automatically detects the gate oxide and sets up a **QTREGION** automatically. You do not need to use the **QTREGION** statement, but you can control the number of interpolated mesh points on each slice through the oxide by using the TAT.SLICEPTS parameter on the **MODELS** statement. The larger this number, the greater the resolution of the integrand in Equations 3-568 and 3-571. This number should be chosen so that the calculated currents are largely unchanged if one uses a slightly higher or lower value of TAT.SLICEPTS. It is possible to treat the ITAT or RTAT current as a post-processing current, where it is not coupled to the continuity equations. To do this, specify ITAT.PP.EL on the **MODELS** statement for electron current, ITAT.PP.HO for hole current, or RTAT.PP for electron and hole current, where the trap can act as a recombination center. To couple the tunneling current self-consistently with the current continuity equations instead, specify ITAT.SC.EL on the **MODELS** statement for electron current, ITAT.SC.HO for hole current, or RTAT.SC for both electron and hole current with recombination. Selecting the ITAT model for both electrons and holes is the same as selecting the RTAT model. Coupling both electron and hole tunneling currents to a trap state means that recombination of electrons and holes must be taken into account. This is exactly what the RTAT model achieves.

The trap densities and levels are set up in the usual way (see Section 3.3.3 “Traps and Defects”) by using either the **TRAP** statement or the **DOPING** statement, restricting the traps to the required regions using the **REGION** statement. Additionally, the TAT.TRAP parameter must be specified on the **TRAP** or **DOPING** statement.

The charge density associated with the traps is incorporated into the Poisson equation by default. For ACCEPTOR-like traps, the charge density is given by

$$\rho(x) = -Qf_t(x)N_t(x) \quad C \quad cm^{-3} \quad \text{3-572}$$

and for DONOR-like traps, it is

$$\rho(x) = Q(1 - f_t(x))N_t(x) \quad C \quad cm^{-3} \quad \text{3-573}$$

It is possible to disable the inclusion of the charge in the Poisson equation by explicitly clearing the flag TAT.POISSON on the **MODELS** statement (see the examples below). This flag is enabled by default.

For a transient simulation, it is necessary to include the dynamic nature of the trap charging and discharging. This is done by using Equation 3-569 to propagate the value of $f_t(x)$ with time. It is therefore time dependent and consequently so are the currents and the trapped charge. In this case, the particle currents obtained by the integrals in Equations 3-568 and 3-571 are no longer equal. So the gate current and current injected into the channel are generally different. Integrating the Poisson equation through the oxide from x=0 to x=L, gives

$$D(L) = D(0) + \int_0^L \rho(x)dx \qquad 3\text{-}574$$

where $D(L)$ is the electric displacement. Differentiating this with respect to time, and using Equations 3-569, 3-572, or 3-573, gives

$$\frac{\partial D(L)}{\partial t} = \frac{\partial D(0)}{\partial t} - Q\int_0^L N_t(x)\frac{\partial f_t}{\partial t}dx \qquad 3\text{-}575$$

$$= \frac{\partial D(0)}{\partial t} - \sum_i (J_{Le}(x_i) - J_{Lh}(x_i) + J_{Re}(x_i) - J_{Rh}(x_i)) \qquad 3\text{-}576$$

where the sum is over the discrete points in the numerical integration. Therefore, the imbalance in particle current through the oxide is exactly balanced by the displacement current, and current continuity is assured so long as the charge due to the traps is included in the Poisson equation. For this reason, it is not possible to disable the TAT.POISSON flag during a transient simulation.

All of the above applies to a single discrete trap level. Atlas allows an arbitrary number of discrete TAT.TRAP levels for this model. A continuous trap distribution can be approximated by a sum of discrete TAT.TRAP levels with different energies and concentrations. A continuous trap distribution having Gaussian dependence on energy can also be set up using the **DEFECTS** statement. You set the flag TAT.TRAP and specify either all of NGA, EGA, WGA for acceptors or all of NGD, EGD, WGD for holes. The densities are calculated according to Equations 15-4 and 15-5 with energy for acceptors taken relative to the conduction band and energy for donors taken relative to the valence band. You specify the trap capture cross-sections for acceptors by SIGGAE, SIGGAH and for donors by SIGGDE, SIGGDH. The number of energy levels used in the integration is NUMA for acceptors and NUMD for DONORS. The default energy range is the entire bandgap, but this can be reduced by specifying the lower and upper limits using the LIMIT1 and LIMIT2 parameters on the **DEFECTS** statement.

Spatial variation of the TAT.TRAP density can be included by using the functionality of the **DOPING** statement for including analytical or user-defined doping profiles.

The tunneling model assumes by default that tunneling is into a crystalline semiconductor. If the semiconductor has localized energy states in the mobility gap, as is the case with amorphous semiconductors, you can enable the TAT.LOCAL flag on the **MODELS** statement to include tunneling to localized energy states as defined on the **DEFECTS** statement.

The expression for b_L for electrons within the mobility gap is

$$b_L = \frac{4\pi m^* K_B T\sigma}{h^2 qfL(E_t,E_{Fl},x)}\int_{E_{transition}}^{E_{min}} v_{thn}(\sigma_{donor}(1-f_{donor}(E))g_{donor}(E) \quad \text{3-577}$$

$$+\sigma_{acceptor}f_{acceptor}(E)g_{acceptor}(E))W(E,x)dE$$

where the quantities f_{donor} and $f_{acceptor}$ are the occupancy fractions of the localized states, v_{thn} is the thermal velocity in cm/s, $\sigma_{acceptor}$ and σ_{donor} are capture cross-sections in cm^2 and g_{donor} and $g_{acceptor}$ are the overall densities of states distributions of donor states and acceptor states respectively in units of $cm^{-3}eV^{-1}$. The value of trap occupancy in the channel, $f_L(E_t, E_{Fl}, x)$ is evaluated at each tunnelling energy by getting the average of the defect occupation probabilities at that energy. If there are no defects at a particular tunnelling energy, the occupation probability is obtained from the Fermi-Dirac distribution function. For hole tunneling, the same expression applies but with $(1-f_{donor})$ replaced by f_{donor} and $f_{acceptor}$ replaced by $(1-f_{acceptor})$ and with different values of capture cross-section and thermal velocity. Above the mobility gap transition energy, $E_{transition}$, the usual expression

$$b_L = \frac{4\pi m^* K_B T\sigma}{h^3 f_L(E_t, E_{fl}, x)}\int_{E_{transition}}^{E_{int}} \log\left(1+\exp\left[\frac{(E-E_{fl})}{K_B T}\right]\right)W(E,x)dE \quad \text{3-578}$$

is used. The overall value of b_L obtained is the sum of the individual values of b_L from Equations 3-25 and 3-26. If the tunneling from the opposite side of the barrier is from a contact or a crystalline semiconductor, then the expression is unchanged from Equation 3-565. Otherwise, it will be evaluated in a similar way to Equation 3-25 using the localized defect densities and occupation fractions. Once these factors have been obtained, the calculation of current proceeds as before. If the self-consistent versions of the models are chosen, then the current is injected into the continuity equations in the semiconductor. This affects the defect occupations and consequently the tunneling rate, resulting in strong coupling, which can give poorer convergence of the non-linear solver. You can specify the TAT.NLCURRS flag on the **MODELS** statement to improve non-local coupling in the system matrix and convergence. TAT.NLCURRS may help to bring about convergence for tunneling into crystalline semiconductors as well as amorphous ones.

The results of the model can be visualized in the run time output. Optionally, the tunneling current associated with each contact is output to log files if the parameter J.TUN is specified on the LOG statement. For the self-consistent version of the model, the current injected into the channel is automatically output to Silvaco standard structure files. These are visualized under Nonlocal e- tunneling rate and Nonlocal h+ tunnelling rate in TonyPlot. Also, output automatically to the structure files are the trap density and trap ionized density for each trap separately. The trap occupation fraction can also be output for each trap by specifying TRAPS.FT on the **OUTPUT** statement.

The TAT.TRAP levels set up by the **DEFECTS** statement are output individually to a standard structure file. If the CONTINUOUS parameter is also specified, then the energy integrated value of density of states and ionized trap density are also output as an extra trap state. If the TRAPS.FT parameter is specified on the **OUTPUT** statement, then the the average trap density, defined as the ratio of integrated ionized density to integrated density of states, is also output.

For very high trap densities, the coupling between the Tunneling current and potential profile across the insulator can be become so strong as to thwart convergence. One way to avoid this is to remove the coupling by explicitly clearing the flag TAT.POISSON. A better alternative is to use the TAT.NLDERIVS flag on the **MODELS** statement. This puts into the numerical solver the couplings between the trap charge at every point with the electrostatic potential at every other point on the same tunneling path. This can improve convergence properties at the minor expense of taking longer for each iteration. Convergence can also be improved in the case of high trap densities by reducing the value of the DVMAX parameter on the **METHOD** statement from its default value.

The RTAT.SC model, when used to connect two semiconducting regions by tunnelling through a wide bandgap material, will use the electron and hole quasi-Fermi levels on both sides of the barrier. The quasi-Fermi level corresponding to a minority carrier can become very sensitive to changes in carrier concentration. This can have a destabilizing effect on the RTAT.SC model and so a stabilization scheme is available. Specify RTAT.MAJQFL on the **MODELS** statement to cause Atlas to use the majority carriers quasi-Fermi level on each side of the barrier.

You can optionally visualize the detailed tunneling data for each slice. Use the ITAT.FILE parameter on the **SOLVE** statement to give the stem for a series of log files. A log file will be created for each tunneling slice and the following data as a function of position along the slice, can be viewed in TonyPlot. The data are Trap occupation probability, Trap density, Trap energy level, Electron and hole current densities from side 1 to the trap, and Electron and hole current densities from the trap to side 2.

For a **SOLVE** statement creating a bias ramp, only data corresponding to the last bias value will be output. Otherwise, the data output will correspond to the solution at the specified bias point.

For example

```
SOLVE VSTEP=0.1 VFINAL=1.0 name=anode ITAT.FILE=feta
with 3 tunnel slices, will produce 3 files :-
feta_slice1.log, feta_slice2.log feta_slice3.log
corresponding to an anode bias of 1 V
```

Table 3-128 Parameters for the ITAT/RTAT Model

Statement	Parameter	Type	Default
DOPING	TAT.TRAP	Logical	False
DEFECTS	TAT.TRAP	Logical	False
DEFECTS	LIMIT1	Real	

Table 3-128 Parameters for the ITAT/RTAT Model			
Statement	**Parameter**	**Type**	**Default**
DEFECTS	LIMIT2	Real	
MODELS	ITAT.SC.EL	Logical	False
MODELS	ITAT.SC.HO	Logical	False
MODELS	RTAT.SC	Logical	False
MODELS	ITAT.PP.EL	Logical	False
MODELS	ITAT.PP.HO	Logical	False
MODELS	RTAT.PP	Logical	False
MODELS	RTAT.MAJQFL	Logical	False
MODELS	TAT.LOCAL	Logical	False
MODELS	TAT.NLCURRS	Logical	False
MODELS	TAT.NLDERIVS	Logical	False
MODELS	TAT.POISSON	Logical	True
MODELS	TAT.SLICEPTS	Real	12
TRAP	TAT.TRAP	Logical	False
SOLVE	ITAT.FILE	Character	

Examples

```
// Setting up a uniform tat trap density for region number 2 via
doping statement

DOPING REGION=2 UNIFORM TAT.TRAP CONC=1E16 ACCEPTOR SIGN=1E-14
SIGP=1E-14 E.LEVEL=4.2


// Setting up the tat trap density for region number 2 via traps
statement

TRAPS REGION=2 TAT.TRAP DENSITY=1E16 ACCEPTOR SIGN=1E-14 SIGP=1E-14
E.LEVEL=4.2 DEGEN.FAC=1


//Setting up Self-consistent RTAT model, and disabling tat.poisson
flag.

// The number of meshpoints used in the integration of current over
the

// oxide thickness is increased to 21.

MODELS RTAT.SC TAT.SLICEPTS=21 ^TAT.POISSON
```

```
// Setting up an acceptor tat trap gaussian density profile with
DEFECTS statement

// The peak density is at energy of 2.0 eV below conduction band
edge and 10 (NUMA)

// traps are only created between 1.0 and 3.0 eV below the
conduction band.

// The CONTINUOUS flag means that output to structure file will
show an extra trap

// with integrated quantities.

DEFECTS NGA=1.0e22 EGA=2.0 WGA=0.5 NUMA=10 SIGGAE=1.0e-14
SIGGAH=1.0e-14

LIMIT1=1.0 LIMIT2=3.0 TAT.TRAP CONTINUOUS
```

Metal-Insulator-Metal Trap-Assisted-Tunneling

The physics of carrier tunneling through traps, as detailed in the previous section, is also applicable to Metal-Insulator-Metal structures. In this case, there is no distinction between self-consistent and post-processing versions because all current is injected directly into the contacts. To enable the model for electron tunneling specify MIM.ITAT.EL on the **MODELS** statement. To enable the model for hole tunneling, specify MIM.ITAT.HO on the **MODELS** statement. To consider both electrons and holes that can recombine inside the insulator, specify MIM.RTAT on the **MODELS** statement (the contact is considered as a zero bandgap semiconductor). Depending on the dominant conduction species of the contact material, only one of these models may be physically appropriate.

Both steady state and transient implementations of the models are available. The models can be used concurrently with the MIMTUN models. The parameter to set the number of points in the tunneling slices, MIMSLICEPTS, is common to both models. The trap distributions are set up in exactly the same way as for the Ielmini model in the previous sections, using either **TRAP**, **DOPING**, or **DEFECTS** statement. All other parameters have the same meaning as for the Ielmini and MIMTUN models. Poor convergence in the case of high trap density can be improved by using the TAT.NLDERIVS flag on the **MODELS** statement as described in the previous section.

Table 3-129 Parameters for the MIM.TAT Models

Statement	Parameter	Type	Default
MODELS	MIM.ITAT.EL	Logical	False
MODELS	MIM.ITAT.HO	Logical	False
MODELS	MIM.RTAT	Logical	False
MODELS	MIMSLICEPTS	Real	51

3.6.8 The Ferroelectric Permittivity Model

Ferroelectric materials exhibit high dielectric constants, polarization and hysteresis. Such materials are finding more and more applications in integrated memory devices. To simulate these effects, a modified version of the ferroelectric model from Miller [214] has been implemented.

To enable the Ferroelectric Model, set the FERRO parameter in the **MODELS** statement. In this model the permittivity used in Poisson's Equation (Equation 3-1) is given the following functional form:

$$\varepsilon(E) = \text{FERRO.EPSF} + \frac{\text{FERRO.PS}}{2\delta} \bullet sech^2\left[\frac{E - \text{FERRO.EC}}{2\delta}\right] \quad 3\text{-}579$$

where FERRO.EPSF is the permittivity, E is the electric field and δ is given as follows:

$$\delta = \text{FERRO.EC}\left[log\frac{1 + \text{FERRO.PR}/\text{FERRO.PS}}{1 - \text{FERRO.PR}/\text{FERRO.PS}}\right]^{-1} \quad 3\text{-}580$$

The FERRO.EPSF, FERRO.PS, FERRO.PR, and FERRO.EC parameters can be modified in the **MATERIAL** statement (see Table 3-130).

The permittivity in Equation 3-579 can be replaced with a user-defined expression with the C-Interpreter. The F.FERRO parameter of the **MATERIAL** statement defines the file that contains the C-function. This function allows the permittivity to be position and field dependent. It is possible to use the FERRO.EPSF parameter along with the F.FERRO parameter. The value specified by FERRO.EPSF is added to the value returned from the C-Interpreter function. This allows you to put the field dependent part of the permittivity in the C-Interpreter function and specify the constant part in the input deck.

For more information about C-Interpreter, see Appendix A "C-Interpreter Functions".

The derivative of the dipole polarization with respect to electric field is given by:

$$\frac{dP_d}{dE} = \Gamma\frac{dP_{sat}}{dE} \quad 3\text{-}581$$

where P_d is the position dependent dipole polarization. A numeric integration of this function is carried out in Atlas to determine the position dependent dipole polarization.

For saturated loop polarization, the Γ function is equal to unity, which corresponds to the default model. If you specify the UNSAT.FERRO parameter in the **MODELS** statement, the Γ function will take on a more general form suitable for simulation of unsaturated loops. In this case, the Γ function is given by:

$$\Gamma = 1 - tanh\left[\left(\frac{P_d - P_{sat}}{\xi P_s - P_d}\right)^{1/2}\right] \quad 3\text{-}582$$

where ξ =1 for increasing fields and ξ = –1 for decreasing fields.

For decks in which convergence to a solution is difficult to achieve, the FERRODAMP parameter is available on the SOLVE statement. The field dependent part of the permittivity

is multiplied by the value of FERRODAMP. You can find a value of FERRODAMP for which convergence is achieved, and then increase it in subsequent SOLVE statements until convergence is achieved with FERRODAMP equal to 1.0.

For example

```
SOLVE PREV FERRODAMP = 0.01
SOLVE PREV FERRODAMP = 0.1
SOLVE PREV FERRODAMP = 1.0
```

In order to use this feature with the C-Interpreter function, you must specify the constant part of the dielectric permittivity using the FERRO.EPSF parameter, as the value returned by the C-Interpreter function is multiplied by the value of the FERRODAMP parameter.

The value of the Ferroelectric permittivity is output to any structure file written after the PERMITTIVITY option is specified on the OUTPUT statement. This is calculated using the component of electric field which has the largest magnitude.

Table 3-130 User-Specifiable Parameters for Equations 3-579 to 3-580

Statement	Parameter	Default	Units
MATERIAL	FERRO.EC	0.0	V/cm
MATERIAL	FERRO.EPS	1.0	
MATERIAL	FERRO.PS	0.0	C/sqcm
MATERIAL	FERRO.PR	0.0	C/sqcm

3.6.9 Epitaxial Strain Tensor Calculations in Zincblende

The strain tensor in epitaxial layers is used in the calculation of the zincblende gain and spontaneous recombination models discussed in Section 3.9.8 "Strained Two-Band Zincblende Model for Gain and Radiative Recombination". In epitaxial layers, the strain tensor can be represented by ε_{xx}, ε_{yy}, ε_{zz}, ε_{xy}, ε_{yz} and ε_{zx}. The relationship between the various components of the strain tensor are given as follows:

$$\varepsilon_{xx} = \varepsilon_{yy} = \frac{a_s - a_0}{a_0} \quad \text{3-583}$$

$$\varepsilon_{zz} = -2\frac{C_{12}}{C_{11}}\varepsilon_{xx} \quad \text{3-584}$$

$$\varepsilon_{xy} = \varepsilon_{yz} = \varepsilon_{zx} = 0 \quad \text{3-585}$$

where C_{12} and C_{11} are the elastic constants, which can be specified by the parameters C12 and C11 on the MATERIAL statement. The default values for C_{12} and C_{11} are given for various binary zincblende materials in Table B-29. Values for ternary and quaternary materials are linearly interpolated from the binary values.

The principal value of strain, ε_{xx}, can be specified or calculated. To specify the value of ε_{xx}, assign the desired value to the STRAIN parameter of the **REGION** statement.

The parameter a_0 is the lattice constant in the "substrate".

In Equation 3-583, a_s is the lattice constant in the layer in question. You can specify the lattice constant in the given layer by the ALATTICE parameter of the **MATERIAL** statement.

Default values for ALATTICE are taken from Table B-29. Temperature coefficients are included in the default calculation.

There are several ways to un-ambiguously specify the substrate or the substrate lattice constant. You can specify a region as the substrate for all strain calculations by specifying the logical parameter SUBSTRATE on the **REGION** statement of the substrate. Alternatively, you can directly specify the substrate lattice constant "local" **REGION** statement using the ASUB parameter.

If the substrate lattice constant is not otherwise specified through the use of the ASUB or SUBSTRATE parameters, the substrate lattice constant is taken as the average of the lattice constants of the two adjacent epitaxial layers (regions above and below the local region). If there is only one adjacent region, the lattice constant of that region is used as the substrate lattice constant.

3.6.10 Epitaxial Strain Tensor Calculation in Wurtzite

The strain tensor in epitaxial layers is used to calculate piezoelectric polarization (Section 3.6.11 "Polarization in Wurtzite Materials") or in gain modeling (Section 3.9.10 "Strained Wurtzite Three-Band Model for Gain and Radiative Recombination") or both. In epitaxial layers, the strain tensor can be represented by ε_{xx}, ε_{yy}, ε_{zz}, ε_{xy}, ε_{yz} and ε_{zx}. The relationship between the various components of the strain tensor are given as follows:

$$\varepsilon_{xx} = \varepsilon_{yy} = \frac{a_s - a_0}{a_0} \quad \text{3-586}$$

$$\varepsilon_{zz} = -2\frac{C_{13}}{C_{33}}\varepsilon_{xx} \quad \text{3-587}$$

$$\varepsilon_{xy} = \varepsilon_{yz} = \varepsilon_{zx} = 0 \quad \text{3-588}$$

where C_{13} and C_{33} are elastic constants, which can be specified by the parameters C13 and C33 on the **MATERIAL** statement. The default values for C_{13} and C_{33} are given for the GaN system in Section B.8 "Material Defaults for GaN/InN/AlN System".

The principal value of strain, ε_{xx}, can be specified or calculated. To specify ε_{xx}, assign the desired value to the STRAIN parameter of the **REGION** statement.

The parameter a_0 is the lattice constant in the "substrate".

In Equation 3-586, a_s is the lattice constant in the layer in question. You can specify the lattice constant in the given layer by the ALATTICE parameter of the **MATERIAL** statement.

Default values for ALATTICE can be found for the GaN/AlN/InN system in Section B.8 "Material Defaults for GaN/InN/AlN System".

The substrate is more ambiguously defined so there are several ways to specify the substrate or the substrate lattice constant. First, you can specify a region as the substrate for strain calculations by specifying the logical parameter SUBSTRATE on the associated **REGION** statement. You can then specify the lattice constant for that region using the ALATTICE parameter of the corresponding **MATERIAL** statement. Alternatively, you can directly specify the substrate lattice constant in the **REGION** statement, which make the strain calculations, using the ASUB parameter.

If the substrate lattice constant is not otherwise specified through the ASUB or SUBSTRATE parameters, the substrate lattice constant is taken as the average of the lattice constants of the two adjacent epitaxial layers (region above and below the region in question). If there is only one adjacent region, the lattice constant of that region is used as the substrate lattice constant.

3.6.11 Polarization in Wurtzite Materials

Polarization modeling is critical for GaN based devices. Automatic calculation of polarization is enabled by the POLARIZ parameter of the **MODELS** statement. STET in wurtzite materials is characterized by two components, spontaneous polarization, P_{sp}, and piezoelectric polarization, P_{pi}. Therefore, the total polarization, P_t, is given by:

$$P_t = \mathrm{PSP} + P_{pi} \qquad \text{3-589}$$

where PSP is specified on the **MATERIAL** statement and specifies the total spontaneous polarization, P_{sp}, for the given material(s). The piezoelectric polarization, P_{pi}, is given by:

$$P_{pi} = 2\frac{a_s - a_0}{a_0}\left(\mathrm{E31} - \frac{\mathrm{C13}}{\mathrm{C33}}\mathrm{E33}\right) \qquad \text{3-590}$$

where E31 and E33 are piezoelectric constants, and C13 and C33 are elastic constants all specified in the **MATERIAL** statement. The a_0 parameter is the lattice constant of the material layer in question, which can be specified by the ALATTICE parameter of the **MATERIAL** statement. The a_s parameter is the average value of the lattice constants of the layers directly above and below the layer in question.

To enable the polarization model, specify POLARIZATION in the **REGION** or **MATERIAL** statement for the region for which you wish to characterize polarization effects. Typically, this will be a quantum well layer or active layer.

The polarization enters into the simulation as a positive and negative fixed charges appearing at the top (most negative Y coordinate) and bottom (most positive Y coordinate) of the layer in question. By default, the positive charge is added at the bottom and the negative charge is added at the top. You can modify the sign and magnitude of this charge by specifying POLAR.SCALE in the **REGION** or **MATERIAL** statement. This parameter is multiplied by the polarization determined by Equation 3-589 to obtain the applied charge. The default value for POLAR.SCALE is 1.0.

In some cases, the introduction of polarization charges may introduce difficulties with convergence due to problems with initial guess. If these problems arise, you can use the PIEZSCALE parameter of the **SOLVE** statement to gradually introduce the effects of

polarization. This parameter defaults to 1.0 and is multiplied by the net charge given by the product of the results of Equation 3-590 and the POLAR.SCALE parameter.

Table 3-131 shows the parameters of the wurtzite polarization model.

Table 3-131 User Specifiable Parameters of the Wurtzite Polarization Model

Statement	Parameter	Default	Units
MATERIAL	PSP	see Tables B-17-B-23	cm^{-2}
MATERIAL	ALATTICE	see Tables B-17-B-23	Å
MATERIAL	E13	see Tables B-17-B-23	cm^{-2}
MATERIAL	E33	see Tables B-17-B-23	cm^{-2}
MATERIAL	C13	see Tables B-17-B-23	GPa
MATERIAL	C33	see Tables B-17-B-23	GPa

In 2D and 3D simulations, you may choose to apply this uniaxial model along layers arranged in the X axis. To accomplish this, you should specify X.EPI on the **MESH** statement.

In 3D simulation, you may choose to apply this uniaxial model along layers arranged in the X or Z axises. To accomplish this, you should specify X.EPI or Z.EPI on the **MESH** statement.

If the layer directly above or below a layer that is being polarized is composed of an insulator or non-crystaline material or if there is no layer above or below the layer in question by default, then no polarization charges are added to the interface in question. When the PCH.INS parameter of the **MODELS** statement is specified, the polarization charge is added at such interfaces.

The parameter IF.CHAR of the **MATERIAL** statement can be used to spread the distribution of polarizaiton charges spatially in a direction normal to the interface analogous to Equation 22-3.

The parameter IF.CHAR specifies the characteristic distance from the interface for an equivalent Gaussian distribution of polarization charge that is applied instead of the 2D charge produced by polarization. The units of IF.CHAR are microns.

You can also calculate polarization charge exactly as in Equations 3-583 and 3-590 but also includes contributions by external mechanical strain. These strains are loaded from the simulator (e.g., Victory Stress) and contain strain fields induced by mechanical means. The POLARIZ model accounts for axial strain due to lattice mismatch. All of the polarization scale factors POLAR.SCALE and PSP.SCALE apply to external strain but the mechanically induced polarization can be scaled separately by using the TENSOR.SCALE parameter of the **MATERIAL** statement.

3.6.12 Stress Effects on Bandgap in Si

Mechanical stress causes change in the band edges in silicon. These band edge shifts are given by the deformation potential theory [31]. The shifts for the conduction band edges are given by:

$$\Delta E_c^{(i)} = \texttt{D.DEFPOT}(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + \texttt{U.DEFPOT}*\varepsilon_{ii} \quad \text{3-591}$$

where $\Delta E_c^{(i)}$ is the shift in the band edge of the ith ellipsoidal conduction band minima. The parameters U.DEFPOT and D.DEFPOT are the user-definable dialation and shear deformation potentials for the conduction band.

The ε_{xx}, ε_{yy}, and ε_{zz} parameters are the diagonal components of the strain tensor.

The shifts in the valence band edges are calculated by:

$$\Delta E_v^{(hl)} = \texttt{A.DEFPOT}(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \pm \sqrt{\xi} \quad \text{3-592}$$

where $\Delta E_v^{(hl)}$ are the band edge shifts in the light and heavy hole valence band maxima. The ξ parameter is given by:

$$\xi = \frac{\texttt{B.DEFPOT}^2}{2}\left\{(\varepsilon_{xx}-\varepsilon_{yy})^2 + (\varepsilon_{yy}-\varepsilon_{zz})^2 + (\varepsilon_{zz}-\varepsilon_{xx})^2\right\} + \texttt{C.DEFPOT}^2(\varepsilon_{xy}{}^2 + \varepsilon_{yz}{}^2 + \varepsilon_{zx}{}^2) \quad \text{3-593}$$

where ε_{xy}, ε_{yz}, and ε_{zx} are the off diagonal components of the strain tensor.

The strain components are calculated from the stress components stored in the input structure file if included (typically imported from Athena). The conversion between stress and strain is given by Equations 3-594, 3-595, and 3-596.

$$\varepsilon_{xx} = \sigma_{xx}s_{11} + \sigma_{yy}s_{12} \quad \text{3-594}$$

$$\varepsilon_{yy} = \sigma_{xx}s_{12} + \sigma_{yy}s_{11} \quad \text{3-595}$$

$$\varepsilon_{zz} = 2\sigma_{zz}s_{44} \quad \text{3-596}$$

Here, σ_{xx}, σ_{yy}, and σ_{zz} are the diagonal components of the stress tensor and s_{11}, s_{12}, and s_{44} are the material compliance coefficients.

The compliance coefficients are given by:

$$s_{11} = \frac{c_{11} + c_{12}}{c_{11}^2 + c_{11}c_{12} - 2c_{12}^2} \quad \text{3-597}$$

$$s_{12} = \frac{-c_{12}}{c_{11}^2 + c_{11}c_{12} - 2c_{12}^2} \quad \text{3-598}$$

$$s_{44} = \frac{1}{c_{44}} \quad \text{3-599}$$

where c_{11}, c_{12}, and c_{44} are the elastic stiffness coefficients. The stiffness coefficients for silicon and germanium are given by [140]:

$$\left.\begin{aligned} c_{11} &= 163.8 - T \times 0.0128 \\ c_{12} &= 59.2 - T \times 0.0048 \\ c_{44} &= 81.7 - T \times 0.0059 \end{aligned}\right\} Si \quad \text{3-600}$$

$$\left.\begin{aligned} c_{11} &= 126.0 \\ c_{12} &= 44.0 \\ c_{44} &= 67.7 \end{aligned}\right\} Ge \quad \text{3-601}$$

where T is the temperature. The stiffness coefficients for SiGe at composition are given by linear interpolation.

If the stress tensor is not loaded from a structure file, you can specify the values of the strain tensor as described in Table 3-132. If the stress tensor is not loaded from a structure file and the strain tensor is not specified, then the strain tensor is calculated as [196]:

$$\varepsilon_{xx} = \varepsilon_{yy} = \frac{(1+\nu)}{(1-\nu)} \frac{a_{SiGe} - a_{Si}}{a_{SiGe}} \quad \text{3-602}$$

where ν is Poisson's ratio, a_{SiGe} is the lattice constant of SiGe, and a_{Si} is the lattice constant of Ge. The lattice constants and Poisson's ratio for Si and Ge are given by:

$$a_{Si} = 5.43102 + 1.41 \times 10^{-5}(T - 300) \quad \text{3-603}$$

$$a_{Ge} = 5.6579 + 3.34 \times 10^{-5}(T - 300) \quad \text{3-604}$$

$$\nu_{Si} = 0.28 \quad \text{3-605}$$

$$\nu_{Ge} = 0.273 \quad \text{3-606}$$

The lattice constant and Poisson's ratio for $Si_{1-x}Ge_x$ is calculated by linear interpolation.

In Equations 3-591 through 3-592, `A.DEFPOT`, `B.DEFPOT`, and `C.DEFPOT` are user-definable valence band deformation potential constants.

The user-definable parameters are shown in Table 3-132.

Table 3-132 User Definable Parameters for Strained Silicon Band Gap			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	A.DEFPOT	2.1	eV
MATERIAL	B.DEFPOT	-2.33	eV
MATERIAL	C.DEFPOT	-4.75	eV
MATERIAL	D.DEFPOT	1.1	eV
MATERIAL	U.DEFPOT	10.5	eV
MATERIAL	EPS11	*	
MATERIAL	EPS22	*	
MATERIAL	EPS33	*	
MATERIAL	EPS12	*	
MATERIAL	EPS13	*	
MATERIAL	EPS23	*	

Note: * If unspecified, Equations 3-594, 3-595, and 3-596 calculate these parameters.

The net changes in the band edges, under Boltzman's statistics, are given by Equations 3-607 and 3-608.

$$\Delta E_c = kT\ln\left[\sum_{i=1}^{3}\frac{\exp\left(-\frac{\Delta E_c^{(i)}}{kT}\right)}{3}\right] \quad 3\text{-}607$$

$$\Delta E_v = k_T\ln\left[\frac{r}{1+r}\exp\left(-\frac{\Delta E_v^{(1)}}{kT}\right)+\frac{1}{1+r}\exp\left(-\frac{\Delta E_v^{(h)}}{kT}\right)\right] \quad 3\text{-}608$$

Here, the parameter r is given by Equation 3-609.

$$r = (m_l/m_h)^{3/2} \quad 3\text{-}609$$

In Equation 3-609, m_l and m_h are the effective masses of light and heavy holes as described in other places in this manual.

To enable the model for stress dependent band gap in silicon, specify the STRESS parameter of the **MODELS** statement.

3.6.13 Low-Field Mobility in Strained Silicon

For Boltzman's statistics, the following expressions can be used for strain dependent electron and hole low-field mobilities in silicon [80]:

$$\mu_n = \mu_{n0}\left\{1 + \frac{1 - ML/MT1}{1 + 2(ML/MT1)}\left[exp\left(\frac{\Delta E_c - \Delta E_c^i}{kT}\right) - 1\right]\right\} \quad 3\text{-}610$$

$$\mu_p = \mu_{p0}\left\{1 + (\mathtt{EGLEY.R} - 1)\frac{(MLH/MHH)^{1.5}}{1 + (MLH/MHH)^{1.5}}\left[exp\left(\frac{\Delta E_v^l - \Delta E_v^h}{kT}\right) - 1\right]\right\} \quad 3\text{-}611$$

where μ_{n0} and μ_{p0} are the concentration dependent low field mobilities for electrons and holes, and $\Delta E_c^{(i)}$, ΔE_c, $\Delta E_v^{(l)}$, and $\Delta E_v^{(h)}$ are given by Equations 3-591, 3-607, and 3-592 respectively. Table 3-133 shows user-definable parameters for this model.

Table 3-133 User Definable Parameters for the Strained Silicon Low-Field Mobility Model

Statement	Parameter	Default	Units
MATERIAL	ML	0.916	
MATERIAL	MT1	0.191	
MATERIAL	MLH	0.16	
MATERIAL	MHH	0.49	
MOBILITY	EGLEY.R	2.79	

To enable the strained silicon low-field mobility model, specify EGLEY.N for electrons and EGLEY.P for holes on the **MOBILITY** statement.

3.6.14 Light Absorption in Strained Silicon

The effects of strain on the optical absorption in silicon can be modeled by modifications [241] made to a model suggested by Rajkanan [256]. The Rajkanan model can be described by:

$$\alpha(T) = \sum_{\substack{i = 1,2 \\ j = 1,2}} C_i A_j \left[\frac{\{\hbar\omega - E_{gj}(T) + E_{pi}\}^2}{\{\exp(E_{pi}/kT) - 1\}} - \frac{\{\hbar\omega - E_{gj}(T) + E_{pi}\}^2}{\{1 - \exp(E_{pi}/kT)\}} \right] + A_d[\hbar\omega - E_{gd}(T)]^{1/2} \quad \text{3-612}$$

where:

- $\alpha(T)$ is the temperature dependent absorption coefficient.
- ω is the optical frequency.
- $E_{gi}(T)$ are the indirect energy band gaps given by Equations 3-613 and 3-614.
- $E_{gd}(T)$ is the direct energy band gap given by Equation 3-615.
- T is the temperature in Kelvin.
- A_d, A_j, C_i, and E_{pi} are empirical constants given in Table 3-134.

The temperature dependent E_g in Equation 3-612 are given by Equations 3-613 through 3-615.

$$E_{g1}(T) = \texttt{EG1.RAJ} - (\texttt{BETA.RAJ} \cdot T^2/[T + \texttt{GAMMA.RAJ}]) \quad \text{3-613}$$

$$E_{g2}(T) = \texttt{EG2.RAJ} - (\texttt{BETA.RAJ} \cdot T^2/[T + \texttt{GAMMA.RAJ}]) \quad \text{3-614}$$

$$E_{gD}(T) = \texttt{EGD.RAJ} - (\texttt{BETA.RAJ} \cdot T^2/[T + \texttt{GAMMA.RAJ}]) \quad \text{3-615}$$

To enable the model given in Equation 3-612, enable the RAJKANAN parameter of the **MODELS** statement.

Table 3-134 User Modifiable Parameters of Equation 3-612.

Symbol	Parameter	Statement	Default	Units
E_{p1}	EP1.RAJ	**MATERIAL**	1.827×10^{-2}	
E_{p2}	EP2.RAJ	**MATERIAL**	5.773×10^{-2}	
C_1	C1.RAJ	**MATERIAL**	5.5	
C_2	C2.RAJ	**MATERIAL**	4.0	
A_1	A1.RAJ	**MATERIAL**	3.231×10^{2}	
A_2	A2.RAJ	**MATERIAL**	7.237×10^{3}	
A_d	AD.RAJ	**MATERIAL**	1.052×10^{6}	

Table 3-134 User Modifiable Parameters of Equation 3-612.				
Symbol	**Parameter**	**Statement**	**Default**	**Units**
	BETA.RAJ	**MATERIAL**	7.021×10^{-4}	eV/K
	GAMMA.RAJ	**MATERIAL**	1108.0	K
	EG1.RAJ	**MATERIAL**	1.1557	eV
	EG2.RAJ	**MATERIAL**	2.5	eV
	EGD.RAJ	**MATERIAL**	3.2	eV

To account for strain effects, the expressions in Equations 3-613 through 3-615 are modified by the changes in band edges induced by strain as given in Equations 3-591 and 3-592. To enable these modifications, set the PATRIN parameter of the **MODELS** statement.

3.7 Quasistatic Capacitance - Voltage Profiles

Quasistatic Capacitance is calculated by specifying the QSCV parameter in the **SOLVE** statement. The quasistatic capacitance is obtained for the electrode (whose bias is being ramped) by subtracting the electrode charge on the electrode at one bias from that at the adjacent bias and dividing by the Voltage increment. The charge density is calculated by applying Gauss' Flux Theorem to the electrode. This gives the capacitance at the midpoint between the two bias points. Atlas does not correct the output for this because the voltage increment should be sufficient fine. Therefore, this small shift is negligible. A fine voltage increment will also give a good approximation to the continuous derivative.

QSCV will work with CARRIERS=0, 1 or 2 specified in the **METHOD** statement. If you link electrodes using the COMMON parameter of the **CONTACT** statement, then the capacitance for each one and the sum of capacitances will be calculated. If you specify MULT and FACTOR for a linked electrode, then the capacitance will be calculated for the appropriate bias points, but will be stored as a function of the bias applied to the COMMON electrode. In this case, scaling or shifting of the C-V curve may be necessary.

3.8 Conductive Materials

In certain cases, it may be advantageous to simulate metal conductivity directly rather than handling electrodes as boundaries. You can define metal regions as "conductive" by specifying the CONDUCTOR parameter of the **REGION** statement. This might be useful, for example in simulating self-heating in metal regions.

When the metal is treated as a conductor, the conduction equation for all points in the region are solved as follows:

$$J = E / RESISTIVITY \quad \text{3-616}$$

where J is the current density, E is the electric field, and RESISTIVITY is the metal resistivity in metal resistivity in μΩ·cm. To specify the metal resistivity, use the RESISTIVITY parameter from the **MATERIAL** statement. To specify the thermal coefficient of resistivity, use the DRHODT parameter from the **MATERIAL** statement.

3.8.1 Conductors with Interface Resistance

You can add interface contact resistance at interfaces between conductor and semiconductor regions. To do this, assign a positive value to the INT.RESIST parameter of the **INTERFACE** statement. This value corresponds to the interface resistivity in units of Ωcm^2. To achieve the desired effect, also specify S.C on the **INTERFACE** statement. The interface resistance in completely analogous to using contact resistance (the CON.RESIST parameter of the **CONTACT** statement) only for conductor-semiconductor interfaces.

3.8.2 Importing Conductors from Athena

By default, all imported metals are treated as electrodes. To override this functionality, specify the CONDUCTOR parameter on the **MESH** statement. Also, for regions that you would like to simulate at PCM materials, you should make them into conductors by specifying CONDUCTOR and MODIFY on the **REGION** statement.

3.9 Optoelectronic Models

This section discusses various physical models used to simulate optoelectronic devices. These models predict fundamental optoelectronic processes, such as absorption and gain and radiative recombination rates versus material composition, temperature and optical wavelength.

These models are based on various band theories and account for the following:

- the existence of multiple valence bands supporting multiple optical transitions,
- asymmetry in conduction band effective masses,
- the effects of strain on the band parameters,
- the effects of quantum confinement on allowable transitions.

In the following paragraphs, we will discuss four-band structure dependent optoelectronic models.

The first two gain and spontaneous recombination models are for one- and two-band unstrained zincblende materials and are discussed in Section 3.9.7 “Unstrained Zincblende Models for Gain and Radiative Recombination”. The second band structure dependent gain and spontaneous recombination model includes the effects of strain in zincblende is discussed in Section 3.9.8 “Strained Two-Band Zincblende Model for Gain and Radiative Recombination”. Finally, Section 3.9.10 “Strained Wurtzite Three-Band Model for Gain and Radiative Recombination” describes a strained dependent three band gain and recombination model for wurtzite.

In addition to the band structure based models, there are certain, more general but less physical models that are also mentioned. All of these models are used in the following applications:

- General drift-diffusion to account for radiative recombination for any semiconductor device,
- Laser and VCSEL simulation in both stimulated emission gain and spontaneous emission (see Chapters 9 “Laser: Edge Emitting Simulator” and 10 “VCSEL Simulator”),
- LED light emission (see Chapter 12 “LED: Light Emitting Diode Simulator”).

3.9.1 The General Radiative Recombination Model

At the most fundamental level the radiative recombination model described in Equation 3-617 describes all the most salient features of radiative recombination except spectral content. You can, however, use this model in all three applications mentioned above. It has the advantages of being fast, simple and easily calibrated.

To enable the general model, specify `OPTR` in the **`MODELS`** statement. This will enable radiative recombination using the general model in the drift diffusion part of the simulation. To enable this model in Laser or VCSEL, disable the default model for Laser and VCSEL by specifying `^SPONTANEOUS` in the **`LASER`** statement. To use the general model for LED, make sure no other competing mechanisms are enabled. The disadvantage of using this model for any light emission application is that it lacks spectral information.

When used in Laser and VCSEL the spontaneous recombination rate is given by:

$$r(r,z) = \frac{\texttt{EMISSION\_FACTOR} \cdot \texttt{COPT} \cdot (n \cdot p - n_i^2)}{N_l} \quad \text{3-617}$$

where n and p are the electron and hole concentrations, n_i is the intrinsic concentration, N_l is the number of longitudinal modes, EMISSION_FACTOR and COPT are user-defined parameters on the **MATERIAL** statement. COPT accounts for the radiative rate in all directions and energies. EMISSION_FACTOR represents the fraction of energy coupled into the direction of interest and in the energy range of interest. Note that Equation 3-617 contains no spectral information.

3.9.2 The Default Radiative Recombination Model

The default spontaneous radiative recombination model is given by

$$r_{sp_m}(x,y) = \texttt{ESEP} \cdot \texttt{EMISSION\_FACTOR} \frac{c}{\texttt{NEFF}} \cdot D(E) \cdot \texttt{GAIN0} \cdot \sqrt{\frac{\hbar\omega - E_g}{kT}} \quad \text{3-618}$$
$$\cdot f\left(\frac{E_c - E_{fn} + \texttt{GAMMA}(\hbar\omega - E_g)}{kT}\right) \cdot \left[1 - f\left(\frac{E_v - E_{fp} - (1 - \texttt{GAMMA})(\hbar\omega - E_g)}{kT}\right)\right]$$

where:

- c is the speed of light,
- h Planck's constant,
- k Boltzman's constant,
- E_g is the energy bandgap,
- E_c and E_v are the conduction and valence band edge energies,
- T is the lattice temperature,
- E_{fn} and E_{fp} are the electron and hole quasi-Fermi energies,
- ω is the emission frequency that corresponds to the transition energy E,
- D(E) is the optical mode density,
- EMISSION.FACTOR, GAIN0 and GAMMA are user defined parameters from the **MATERIAL** statement,
- ESEP and NEFF are user-defined parameters specified from the **LASER** statement.

The optical mode density *D(E)* is given by

$$D(E) = \frac{n^3 E^2}{\pi^2 \hbar^3 c^3} \quad \text{3-619}$$

where n is the index of refraction.

The function f(x) is given by

$$f(x) = \frac{1}{1 + exp(x)} \quad \text{3-620}$$

3.9.3 The Standard Gain Model

The standard gain model [225] is enabled by specifying G.STANDARD in the **MODELS** statement. The standard gain model is given by

$$g(x,y) = \text{GAIN0}\sqrt{\frac{\hbar\omega - E_g}{kT}}\left[f\left(\frac{E_c - E_{fn} + \text{GAMMA}(\hbar\omega - E_g)}{kT}\right) - f\left(\frac{E_v - E_{fp} - (1-\text{GAMMA})(\hbar\omega - E_g)}{kT}\right)\right] \quad \text{3-621}$$

where:

- h is Planck's constant,
- E_g is the energy bandgap,
- k is Boltzman's constant,
- T is the lattice temperature,
- E_{fn} and E_{fp} are the electron and hole quasi-Fermi energies,
- ω is the emission frequency,
- E_v and E_c are the valence and conduction band edge energies,
- the function f is defined in Equation 3-620,
- GAMMA and GAIN0 user-definable parameters specified on the **MATERIAL** statement.

Table 3-135 describes the user defined parameters of Equation 3-621.

Table 3-135 User-Specifiable Parameters for Equation 3-621

Statement	Parameter	Default	Units
MATERIAL	GAIN0	2000.0	cm^{-1}
MATERIAL	GAMMA		

If GAMMA in Equation 3-621 is not specified, it is automatically calculated from the following expression:

$$\text{GAMMA} = \frac{1}{\left(\frac{N_c}{N_v}\right)^{\frac{2}{3}} + 1} \quad \text{3-622}$$

where N_c and N_v are the conduction and valence band densities of states.

3.9.4 The Empirical Gain Model

The empirical gain model is enabled by specifying G.EMPIRICAL in the **MODELS** statement. The model is described by the following expression [351]:

$$g(x,y) = \mathrm{GAIN00} + \mathrm{GAIN1N} \cdot n + \mathrm{GAIN1P} \cdot p + \mathrm{GAIN2NP} \cdot np + \mathrm{GAIN1MIN} \cdot min(n,p) \quad \text{3-623}$$

where n and p are the electron and hole concentrations, and GAIN00, GAIN1N, GAIN1P, GAIN2NP and GAIN1MIN are user-specified parameters from the **MATERIAL** statement. Note that the empirical model contains no spectral dependency and should not be used for **LASER** or **VCSEL** simulations with multiple logitudinal modes. Table 3-136 shows the user-definable parameters for Equation 3-623.

Table 3-136 User-Specifiable Parameters for Equation 3-623

Statement	Parameter	Default	Units
MATERIAL	GAIN00	-200.0	cm^{-1}
MATERIAL	GAIN1P	0	cm^2
MATERIAL	GAIN1N	0	cm^2
MATERIAL	GAIN2NP	0	cm^5
MATERIAL	GAIN1MIN	3.0×10^{-16}	cm^2

3.9.5 Takayama's Gain Model

Takayama's gain model [310] is enabled by specifying TAYAMAYA in the **MODELS** statement. Takayama's model is described by the following expression:

$$g(x,y) = \mathrm{GN1}(n(x,y) - \mathrm{NTRANSPARENT}) \qquad n > \mathrm{NTRANSPARENT}$$
$$g(x,y) = \mathrm{GN2}(n(x,y) - \mathrm{NTRANSPARENT}) \qquad n <= \mathrm{NTRANSPARENT} \quad \text{3-624}$$

where n is the electron concentration and GN1, GN2 and NTRANSPARENT are user-specified parameters from the **MATERIAL** statement. As in the empirical gain model, this model contains no spectral dependency and should not be used to simulate multiple longitudinal modes in Laser or VCSEL. Table 3-137 shows the user definable parameters for Equation 3-624.

Table 3-137 User-specified values for Equation 3-624

Statement	Parameter	Default	Units
MATERIAL	GN1	3.0×10^{-16}	cm^2
MATERIAL	GN2	4.0×10^{-15}	cm^2
MATERIAL	NTRANSPARENT	2.0×10^{18}	cm^{-3}

3.9.6 Band Structure Dependent Optoelectronic Models

Although the simple models described in Sections 3.9.1 “The General Radiative Recombination Model” through 3.9.5 “Takayama's Gain Model” are useful for modeling the principal effects, they suffer from one or more of the following drawbacks:

- they lack spectral dependencies,
- they are not generally calibrated to all tools,
- they do not account for effects of strain,
- they do not account for the effects of quantum confinement,
- the lack physical basis.

The following sections will describe more physically based models. Figure 3-17 illustrates the simulation flow for these models.

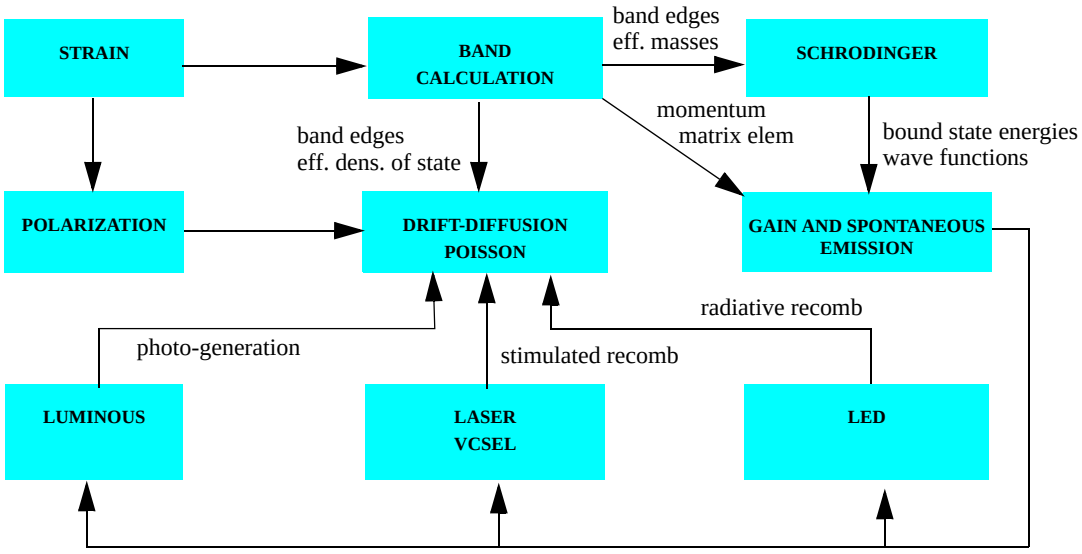


Figure 3-17: Simulation Flow For Physically Based Optoelectronic Models

First, strain is introduced either directly or calculated from lattice mismatch and used to calculate the strain effects on band calculations. The strain is also introduced to polarization calculations that enter directly into the drift diffusion calculations as polarization fields.

Next, the band parameters (band edges and effective masses) are used directly in the drift diffusion simulations as well as feeding into the solutions of Schrodinger's equations to calculate the bound state energies. The band parameters and bound state energies are then used to calculate gain, radiative recombination rate and optical absorption.

3.9.7 Unstrained Zincblende Models for Gain and Radiative Recombination

For zincblende materials, you can select one of two models for optical gain and spontaneous recombination. If the LI parameter is specified on the **MODELS** statement, the model by Li [183] will be used. If the ZB.ONE parameter is specified on the **MODELS** statement, the model by Yan [363] will be used. The difference between the models is that in the ZB.ONE model only one valence band is accounted for, whereas the LI model accounts for both light and heavy holes in the valence band.

Next, the bulk momentum matrix element is calculated as shown in Equation 3-625.

$$\left|M_{avg}\right|^2 = \text{MBULKSQ} = \frac{m_0}{6}\text{EP.MBULK} = \frac{m_0}{6}\left(\frac{m_0}{m^*} - 1 - 2\text{FB.MBULK}\right)\frac{E_g(E_g + \text{SO.DELTA})}{\left(E_g + \frac{2}{3}\text{SO.DELTA}\right)} \quad \text{3-625}$$

In Equation 3-625, you can specify the bulk momentum matrix element directly using the MBULKSQ parameter of the **MATERIAL** statement. You can also calculate it using either the energy parameter EP.MBULK of the **MATERIAL** statement, or the band gap E_g, correction factor FB.MBULK and split-off energy, or SO.DELTA parameters of the **MATERIAL** statement. Table 3-138 shows the default values for the EP.MBULK, FB.MBULK, and SO.DELTA parameters for several materials.

Table 3-138 Default Values for User Specifiable Parameters of Equation 3-625 [246]

Parameter Statement Units	EP.MBULK MATERIAL (eV)	FB.MBULK MATERIAL	SO.DELTA MATERIAL (eV)
AlAs	21.1	-0.48	0.28
AlP	17.7	-0.65	0.07
AlSb	18.7	-0.56	0.676
GaAs	28.8	-1.94	0.341
GaP	31.4	-2.04	0.08
GaSb	27.0	-1.63	0.76
InAs	21.5	-2.9	0.39
InP	20.7	-1.31	0.108
InSb	23.3	-0.23	0.81

In Equation 3-625, m_* is by default equal to the conduction band effective mass, m_c. You may, however, specify the value of m_* using the MSTAR parameter of the **MATERIAL** statement. Yan et. al. [363] recommend a value of 0.053 for GaAs.

For materials not listed in Table 3-138, SO.DELTA has a default value of 0.341 and EP.MBULK and FB.MBULK are 0.

For quantum wells, the matrix elements are asymmetric and for light and heavy holes are given by Equations 3-626 and 3-627.

$$M_{hh} = A_{hh} M_{avg} O_{hh} \quad (3\text{-}626)$$

$$M_{lh} = A_{lh} M_{avg} O_{lh} \quad (3\text{-}627)$$

where *Ahh* and *Alh* are anisotropy factors for heavy and light holes. O_{hh} and O_{lh} are overlap integrals. You can specify the overlap using the WELL.OVERLAP parameter of the **MODELS** statement. If not specified, O_{hh} and O_{lh} are calculated from the wavefunctions.

For TE modes, the values of the anisotropy factors are given by Equation 3-628.

$$A_{hh} = \frac{3 + 3E_{ij}/E}{4}, \quad A_{lh} = \frac{5 - 3E_{ij}/E}{4} \quad for(E > E_{ij})$$
$$A_{hh} = 3/2, \quad A_{lh} = 1/2 \quad for(E \le E_{ij}) \quad (3\text{-}628)$$

For TM modes, the values of the anisotropy factors are given by Equation 3-629.

$$A_{hh} = \frac{3 - 3E_{ij}/E}{2} \quad A_{hh} = \frac{1 + 3E_{ij}/E}{2} \quad for(E > E_{ij})$$
$$A_{hh} = 0 \quad Alh = 2 \quad for(E < E_{ij}) \quad (3\text{-}629)$$

Here, E is the transition energy and E_{ij} is the energy difference between the *ith* valence bound state and the *jth* conduction band state.

The band energies and effective masses are used to calculate the Fermi functions given in Equations 3-630 and 3-631 ($E_{mn} \equiv E_c - E_v$).

$$f_c^n(E) = \frac{1}{1 + exp\{[E_c + (m_r/m_c)(E - E_{mn}) - E_{Fc}]/kT\}} \quad (3\text{-}630)$$

$$f_v^m(E) = \frac{1}{1 + exp\{[E_v - (m_r/m_v)(E - E_{mn}) - E_{Fv}]/kT\}} \quad (3\text{-}631)$$

Here, *Efv* is the hole Fermi level, *Efc* is the electron Fermi level, E_v is the valence band energy, E_c is the conduction band energy, m_v is the valence band effective mass, m_c is the conduction band effective mass, and m_r is the reduced mass given in by Equation 3-633.

$$m_r = \left(\frac{1}{m_c} + \frac{1}{m_v}\right)^{-1} \quad (3\text{-}633)$$

Spontaneous Emission and Gain

For bulk regions, spontaneous emission rate per unit energy and gain are given as functions of photon energy hω and polarization υ=*TE, TM* by the following expressions:

$$r_{spon}^{\upsilon}(h\omega) = \left(\frac{n_r e^2 \omega}{\pi h c^3 \varepsilon_0 m_0^2}\right) \sum_v \rho_r^{3D}(\hbar\omega) f_c \left(1 - f_v\right) \left|M_b^{\upsilon}\right|^2 \quad \text{3-634}$$

$$g^{\upsilon}(h\omega) = \left(\frac{\pi e^2}{n_r c \omega m_0^2 \varepsilon_0}\right) \sum_v \rho_r^{3D}(\hbar\omega)\left(f_c - f_v\right) \left|M_b^{\upsilon}\right|^2 \quad \text{3-635}$$

where

- $\rho_r^{3D}(\hbar\omega) = \frac{1}{2\pi}\left(\frac{2m_r}{\hbar^2}\right)^{3/2} \sqrt{\hbar\omega - E_g}$ is the three-dimensional density of states.
- m_r is a reduced electron-hole effective mass.
- n_r is a material refractive index.
- M_b^{υ} is a polarization- dependent bulk momentum matrix element
- The sum is taken over all valence bands.

Expressions for gain and spontaneous emssion for quantum wells are similar and given in Chapter 14 “Quantum: Quantum Effect Simulator”.

Spontaneous Recombination

In order to compute total radiative recombination rate and include it into drift-diffusion equations, specify the SPONTANEOUS parameter on the **MODELS** statement. Recombination rate is calculated by performing averaging over polarizations and integration over the whole emission spectrum:

$$R_{sp}(x, y, z) = \int_0^{\infty} \frac{\left(2 r_{spon}^{TE}(h\omega) + r_{spon}^{TM}(h\omega)\right)}{3} d(h\omega) \quad \text{3-636}$$

The integration over the spectrum is done using Gauss-Laguere quadrature rule. The number of energy points is set by the WELL.NERSP parameter (default is 32) on the **METHOD** statement. If the parameter is set to zero (not recommended), a trapezoidal integration will be used with a spacing set by WELL.DENERGY parameter on the **MODELS** or **MATERIAL** statement.

Spectrum of spontaneous recombination and gain for each bulk LED region can be stored in a separate file using SPECTRUM='filename' parameter on the **SAVE** statement. The spectra are computed at the geometrical center of the LED region.

3.9.8 Strained Two-Band Zincblende Model for Gain and Radiative Recombination

The zincblende two-band model is derived from the 4x4 k·p model assuming parabolic bands. The model accounts for light and heavy holes and the effects of strain and is enabled by specifying ZB.TWO on the **MODELS** statement.

In the zincblende two band model we first calculate P_ε and Q_ε as given by Equations 3-637 and 3-638.

$$P_\varepsilon = -a_v(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \quad 3\text{-}637$$

$$Q_\varepsilon = -\frac{b}{2}(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \quad 3\text{-}638$$

Here, a_v and b are the valence band hydrostatic deformation potentials and ε_{xx}, ε_{yy}, and ε_{zz} are the strain tensor as calculated in Section 3.6.9 "Epitaxial Strain Tensor Calculations in Zincblende". The default values of a_v and b for various binary materials are given in Table B-B-29. Values for ternary and quaternary materials are linearly interpolated from the binary values.

Next, we can calculate the conduction, heavy hole and light hole band edge energies from Equations 3-639, 3-640, and 3-641.

$$E_c = E_v + E_g + a_c(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \quad 3\text{-}639$$

$$E_{hh} = E_v - P_\varepsilon - \mathrm{sgn}(Q_\varepsilon)\sqrt{Q_\varepsilon^2} \quad 3\text{-}640$$

$$E_{lh} = E_v - P_\varepsilon + \mathrm{sgn}(Q_\varepsilon)\sqrt{Q_\varepsilon^2} \quad 3\text{-}641$$

Here, a_c is the conduction band hydrostatic deformation potential, E_c, E_v, and E_g are the unstrained conduction band edge energy, valence band edge energy, and band gap. These are calculated from the material specific affinity and bandgap.

The default values of a_c for various binary materials are given in Table B-29. Values for ternary and quaternary materials are linearly interpolated from the binary values.

The function sgn() is the "sign" function (i.e., +1 for positive arguments and -1 for negative arguments).

Next, we calculate the effective masses for the various bands using Equations 3-642 through 3-645.

$$m_{hh}^z = m_0/(\gamma_1 - 2\gamma_2) \quad 3\text{-}642$$

$$m_{lh}^z = m_0/(\gamma_1 + 2\gamma_2) \quad 3\text{-}643$$

$$m_{hh}^t = m_0/(\gamma_1 + \gamma_2) \quad 3\text{-}644$$

$$m_{lh}^{t} = m_0/(\gamma_1 - \gamma_2) \quad \text{3-645}$$

Here, m_0 is the rest mass of an electron, and γ_1 and γ_2 are the Luttinger parameters. The default values of the luttinger paramters for various binary materials are given in Table B-30. Values for ternary and quaternary materials are linearly interpolated from the binary values.

Finally, the band edges and effective masses are placed into the gain and spontaneous recombination models described in Section 3.9.7 "Unstrained Zincblende Models for Gain and Radiative Recombination".

3.9.9 Strained Three-Band Zincblende Model for Gain and Radiative Recombination

The zincblende three-band model is derived from the 6x6 k·p model assuming parabolic bands. The model accounts for light, heavy and split-off holes and the effects of strain and is enabled by specifying ZB.THREE on the **MODELS** statement.

The conduction, heavy hole, light hole and split-off hole band edge energies are given by the following expressions.

$$E_c = E_v + E_g + a_c(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \quad \text{3-646}$$

$$E_{hh} = E_v - P_\varepsilon - Q_\varepsilon \quad \text{3-647}$$

$$E_{lh} = E_v - P_\varepsilon + \frac{1}{2}[Q_\varepsilon - \Delta_0 + \sqrt{\Delta_0^2 + 9Q_\varepsilon^2 + 2Q_\varepsilon\Delta_0}] \quad \text{3-648}$$

$$E_{so} = E_v - P_\varepsilon + \frac{1}{2}[Q_\varepsilon - \Delta_0 - \sqrt{\Delta_0^2 + 9Q_\varepsilon^2 + 2Q_\varepsilon\Delta_0}] \quad \text{3-649}$$

where P_ε and Q_ε are given by Equations 3-637 and 3-638 above and Δ_0 is the split-off energy.

The effective masses for the various bands are given by the following expressions:

$$m_{hh}^{z} = m_0/(\gamma_1 - 2\gamma_2) \quad \text{3-650}$$

$$m_{lh}^{z} = m_0/(\gamma_1 + 2\gamma_2 f_+) \quad \text{3-651}$$

$$m_{so}^{z} = m_0/(\gamma_1 + 2\gamma_2 f_-) \quad \text{3-652}$$

$$m_{hh}^{t} = m_0/(\gamma_1 + \gamma_2) \quad \text{3-653}$$

$$m_{lh}^{t} = m_0/(\gamma_1 - \gamma_2 f_+) \quad \text{3-654}$$

$$m_{so}^{t} = m_0/(\gamma_1 - \gamma_2 f_-) \quad \text{3-655}$$

employing the strain factor

$$f_{\pm} = \frac{2s[1 + 1.5(s - 1 \pm \sqrt{1 + 2s + 9s^2})] + 6s^2}{0.75(s - 1 \pm \sqrt{1 + 2s + 9s^2})^2 + s - 1 \pm \sqrt{1 + 2s + 9s^2} - 3s^2} \quad \text{3-656}$$

with the strain parameter $s = Q_\varepsilon/\Delta_0$. Without strain, $s = 0$ and $f_+ = 1$, the effective masses for heavy and light holes are identical to those of the two-band model.

3.9.10 Strained Wurtzite Three-Band Model for Gain and Radiative Recombination

The strained wurtzite three-band model [56, 57, 167] is derived from the k·p method for three valence bands in wurtzite crystalline structure. Given the assumptions of parabolic bands, no valence band mixing and momentum approaching zero, the following approach can be used. First, we calculate the parameters from Equations 3-657 and 3-658.

$$\theta_\varepsilon = D_3\varepsilon_{zz} + D_4(\varepsilon_{xx} + \varepsilon_{yy}) \quad \text{3-657}$$

$$\lambda_\varepsilon = D_1\varepsilon_{zz} + D_2(\varepsilon_{xx} + \varepsilon_{yy}) \quad \text{3-658}$$

Here, D_1, D_2, D_3, and D_4 are shear deformation potentials and ε_{xx}, ε_{yy}, and ε_{zz} are taken from the strain tensor calculations described in Section 3.6.10 “Epitaxial Strain Tensor Calculation in Wurtzite”. Next, we can calculate the valence band energies from Equations 3-659 through 3-661.

$$E_{hh}^0 = E_v^0 + \Delta_1 + \Delta_2 + \theta_\varepsilon + \lambda_\varepsilon \quad \text{3-659}$$

$$E_{lh}^0 = E_v^0 + \frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2} + \lambda_\varepsilon + \sqrt{\left(\frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2}\right)^2 + 2\Delta_3^2} \quad \text{3-660}$$

$$E_{ch}^0 = E_v^0 + \frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2} + \lambda_\varepsilon - \sqrt{\left(\frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2}\right)^2 + 2\Delta_3^2} \quad \text{3-661}$$

Here, Δ_1, Δ_2 and Δ_3 are split energies and E_v is the valence band reference level. Next, we can calculate the hydrostatic energy shift from Equation 3-662.

$$P_{c\varepsilon} = a_{cz}\varepsilon_{zz} + a_{ct}(\varepsilon_{xx} + \varepsilon_{yy}) \quad \text{3-662}$$

Here, a_{cz} and a_{ct} are hydrostatic deformation potentials. From which we can calculate the conduction band energy as given in Equation 3-663.

$$E_c^0 = E_v^0 + \Delta_1 + \Delta_2 + E_g + P_{c\varepsilon} \quad \text{3-663}$$

Here, E_g is the energy bandgap. The default bandgaps for the InGaN system given in Section B.8 "Material Defaults for GaN/InN/AlN System". You can override the default models for bandgap by specifying the EG1300, EG2300, and EG12BOW parameters of the **MATERIAL** statement.

Next, we can calculate the effective masses in the various bands using the expressions in Equations 3-664 through 3-669.

$$m_{hh}^{z} = -m_0(A_1 + A_3)^{-1} \quad \text{3-664}$$

$$m_{hh}^{t} = -m_0(A_2 + A_4)^{-1} \quad \text{3-665}$$

$$m_{lh}^{z} = -m_0\left[A_1 + \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right)A_3\right]^{-1} \quad \text{3-666}$$

$$m_{lh}^{t} = -m_0\left[A_2 + \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right)A_4\right]^{-1} \quad \text{3-667}$$

$$m_{ch}^{z} = -m_0\left[A_1 + \left(\frac{E_{ch}^0 - \lambda_\varepsilon}{E_{ch}^0 - E_{lh}^0}\right)A_3\right]^{-1} \quad \text{3-668}$$

$$m_{lh}^{t} = -m_0\left[A_2 + \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right)A_4\right]^{-1} \quad \text{3-669}$$

Here, m_ϕ is the free space mass of an electron, m_{hh}^{z} , m_{hh}^{t}, m_{lh}^{z}, m_{lh}^{t}, m_{ch}^{z}, and m_{ch}^{t} are the effective masses for heavy holes, light holes and crystal split off holes in the axial and transverse directions and A_1, A_2, A_3 and A_4 are hole effective mass parameters.

The momentum matrix elements for the various transitions can be calculated by Equations 3-670 through 3-675.

$$M_{hh}11 = 0 \quad \text{3-670}$$

$$M_{hh}11 = b^2\left(\frac{m_0}{2}E_{pz}\right) \quad \text{3-671}$$

$$M_{ch}11 = a^2\left(\frac{m_0}{2}E_{pz}\right) \quad \text{3-672}$$

$$M_{hh}\perp = \frac{m_0}{4}E_{px} \quad \text{3-673}$$

$$M_{lh}\perp = a^2\left(\frac{m_0}{4}\right)E_{px} \quad \text{3-674}$$

$$M_{ch}\perp = b^2\left(\frac{m_0}{4}\right)E_{px} \quad \text{3-675}$$

Here, E_{px} and E_{pz} are given by Equations 3-676 and 3-677, a^2 and b^2 are given by Equations 3-680 and 3-681. The values of P_1^2 and P_2^2 are given by Equations 3-678 and 3-679.

$$E_{px} = (2m_0/h^2)P_2^2 \quad \text{3-676}$$

$$E_{pz} = (2m_0/h^2)P_1^2 \quad \text{3-677}$$

$$P_1^2 = \frac{h^2}{2m_0}\left(\frac{m_0}{m_e^z} - 1\right)\frac{(E_g + \Delta_2 + \Delta_2)(E_g + 2\Delta_2) - 2\Delta_3^2}{(E_g + 2\Delta_2)} \quad \text{3-678}$$

$$P_2^2 = \frac{h^2}{2m_0}\left(\frac{m_0}{m_e^t} - 1\right)\frac{E_g[(E_g + \Delta_1 + \Delta_2)(E_g + 2\Delta_2) - 2\Delta_3^2]}{(E_g + \Delta_1 + \Delta_2)(E_g + \Delta_2) - \Delta_3^2} \quad \text{3-679}$$

$$a^2 = \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right) \quad \text{3-680}$$

$$b^2 = \left(\frac{E_{ch}^0 - \lambda_\varepsilon}{E_{ch}^0 - E_{lh}^0}\right) \quad \text{3-681}$$

In Equations 3-676 through 3-681, m_e^t and m_e^z are the transverse and axial conduction band effective masses.

The default values for the parameters of Equations 3-657 through 3-681 are shown in Section B.8 “Material Defaults for GaN/InN/AlN System”.

3.9.11 Lorentzian Gain Broadening

Gain broadening due to intra-band scattering can be introduced by specifying LORENTZ in the **MODELS** statement. The following equation describes gain broadening when it's applied.

$$g(E) = \int_{E_{ij}}^{\infty} g(E')L(E'-E)dE' \qquad 3\text{-}682$$

The Lorentzian shape function is given by Equation 3-683.

$$L(E'-E) = \frac{1}{\pi}\cdot\frac{\mathtt{WELL.GAMMA0}}{(E'-E)^2+\mathtt{WELL.GAMMA0}^2} \qquad 3\text{-}683$$

where WELL.GAMMA0 is user-specifiable in the **MATERIAL** statement.

3.9.12 Ishikawa's Strain Effects Model

The strained layer InGaAsP and InGaAlAs quantum well material models from [141] are implemented in the Atlas simulator. To enable these models, specify the ISHIKAWA parameter in the **MODELS** statement. Strain percentages are specified by the STRAIN parameter in the **MQW** statement. STRAIN between a substrate material with lattice constant, d_s, and an epitaxial material without STRAIN, with lattice constant, d_e, is $(d_e - d_s)|d_s$.

When you enable this model, the band edge parameters for materials in the InGaAsP and InGaAlAs systems are calculated using Equations 3-684, 3-685,and 3-686.

InGaAs

$$\begin{aligned} E_c &= 1.040 - 0.0474\mathtt{STRAIN} + 0.003303\mathtt{STRAIN}^2 \\ E_{v,hh} &= 0.3331 + 0.05503\mathtt{STRAIN} - 0.002212\mathtt{STRAIN}^2 \\ E_{v,lh} &= 0.331 - 0.01503\varepsilon - 0.003695\mathtt{STRAIN}^2 \end{aligned} \qquad 3\text{-}684$$

InGaAsP

$$\begin{aligned} &a)\mathtt{STRAIN} < 0 \\ E_c &= (0.6958 + 0.4836E_g) - 0.03031\mathtt{STRAIN} \\ E_{v,lh} &= (0.5766 - 0.3439E_g) - 0.3031\mathtt{STRAIN} \\ &b)\mathtt{STRAIN} > 0 \\ E_c &= (0.6958 + 0.4836E_g) + 0.003382\mathtt{STRAIN} \\ E_{v,hh} &= (0.6958 - 0.5164E_g) + 0.003382\mathtt{STRAIN} \end{aligned} \qquad 3\text{-}685$$

InGaAlAs

$$a)\,\mathtt{STRAIN} < 0$$
$$E_c = (0.5766 + 0.6561E_g) - 0.02307\mathtt{STRAIN}$$
$$E_{v,\,lh} = (0.5766 - 0.3439E_g) - 0.2307\mathtt{STRAIN}$$
$$b)\,\mathtt{STRAIN} > 0 \qquad \text{3-686}$$
$$E_c = (0.5766 + 0.06561E_g) + 0.01888\mathtt{STRAIN}$$
$$E_{v,\,hh} = (0.5766 - 0.3439E_g) + 0.01888\mathtt{STRAIN}$$

A STRAIN parameter has also been added to the **REGION** statement to account for strain in the bulk materials. Note that for the InGaAs material system, the equations ignore composition fraction and variation in the parameters is accounted strictly through strain (see Equation 3-684).

These band edge parameters are used through all subsequent calculations. Most importantly, the band edges are used in solving the Schrodinger's Equation (See Equations 14-2 and 14-3) to obtain the bound state energies in multiple quantum wells.

If you enable the Ishikawa Model, this would include the effects of strain in the valence and conduction band effective masses as described in [141]. The effects of strain are introduced in Equation 3-687.

$$\frac{1}{m_{lh}} = \mathtt{ASTR} + \mathtt{BSTRSTRAIN} + \mathtt{CSTRSTRAIN}^2$$
$$\frac{1}{m_{hh}} = \mathtt{DSTR} + \mathtt{ESTRSTRAIN} + \mathtt{FSTRSTRAIN}^2 \qquad \text{3-687}$$

The ASTR, BSTR, CSTR, DSTR, ESTR, and FSTR parameters are user-definable in the **MATERIAL** statement. You can also choose the appropriate values for these parameters from Tables 3-139 or 3-140.

Table 3-139 In-Plane Effective Mass of InGaAsP Barrier (1.2µm)/InGaAs(P) Well and InGaAlAs Barrier (1.2µm)/InGa(Al)As Well System for 1.55µm Operation [141]

	–2.0<Strain(%)<–0.5			0<Strain(%)<2.0		
	ASTR	BSTR	CSTR	DSTR	ESTR	FSTR
Lattice-matched InGaAsP barrier/InGaAs well	–4.238	–6.913	–1.687	4.260	2.253	–0.584
InGaAsP (1.6-µm) well	–0.936	–4.825	–1.754	2.986	5.505	–1.736
Strain-compensated InGaAsP barrier/InGaAs well	–7.761	–13.601	–5.100	4.166	1.933	–0.558
Lattice-matched InGaAlAs barrier/InGaAs well	–3.469	–6.133	–1.481	3.9134	2.336	–0.633
InGaAlAs(1.6-µm) well	–1.297	–5.598	–2.104	2.725	6.317	–1.766
Strain-compensated InGaAlAs barrier/InGaAs well	–5.889	–9.467	–2.730	4.193	1.075	–-0.155

Table 3-140 In-Plane Effective Mass of InGaAsP Barrier (1.1µm)/InGaAs(P) Well and InGaAlAs Barrier (1.1µm)/InGa(Al)As Well System for 1.30µm Operation [141]						
–2.0<Strain(%)<–0.5				**–0.5<Strain(%)<2.0**		
	ASTR	BSTR	CSTR	DSTR	ESTR	FSTR
Lattice-matched InGaAsP barrier/InGaAs well	–9.558	–8.634	–1.847	4.631	1.249	–0.313
InGaAsP (1.4-µm) well	–2.453	–4.432	–1.222	3.327	3.425	–1.542
Strain-compensated InGaAsP barrier/InGaAs well				4.720	0.421	–0.014
Lattice-matched InGaAlAs barrier/InGaAs well	–8.332	–8.582	–2.031	3.931	1.760	–0.543
InGaAlAs(1.4-µm) well	–3.269	–5.559	–1.594	3.164	4.065	–1.321
Strain-compensated InGaAlAs barrier/InGaAs well				4.078	0.516	–0.0621

To choose these values, use the STABLE parameter in the **MATERIAL** statement. You can set this parameter to the row number in the tables. The rows are numbered sequentially. For example, the first row in Table 3-139 is selected by specifying STABLE=1. The first row of Table 3-140 is selected by specifying STABLE=7.

You can also choose to specify the effective masses directly. The conduction band effective mass is specified by the MC parameter in the **MATERIAL** statement. There are two ways to specify the valence band. One way, is to use the MV parameter to specify a net effective mass. The other way, is to use the MLH and MHH parameters to specify the light and heavy hole effective masses individually.

If you don't specify the effective masses by using one of these methods, then the masses will be calculated from the default density of states for the given material as described in Section 3.4.2 "Density of States".

The effective masses described are also used to solve the Schrodinger Equation (see Equations 14-2 and 14-3).

3.10 Optical Index Models

The index of refraction of materials are used in several places in the Atlas simulation environment. For Luminous and Luminous 3D, the index is used to calculate reflections during raytrace. For Laser and VCSEL, the index is key to determining optical confinement. For LED, the index is used to calculate reflections for reverse raytrace.

In the Atlas environment there are a variety of ways the optical index of refraction can be introduced as well as reasonable energy dependent defaults for many material systems. For some these defaults are tabular (see Table B-37). For others there exist analytic functions. Here we describe some of those analytic models.

3.10.1 The Sellmeier Dispersion Model

To enable the Sellmeier dispersion model, specify NDX.SELLMEIER on the **MATERIAL** statement. The Sellmeier dispersion model is given in Equation 3-688.

$$n_r(\lambda) = \sqrt{\mathtt{S0SELL} + \frac{\mathtt{S1SELL}\lambda^2}{\lambda^2 - \mathtt{L1SELL}^2} + \frac{\mathtt{S2SELL}\lambda^2}{\lambda^2 - \mathtt{L2SELL}^2}} \qquad 3\text{-}688$$

In this equation, λ is the wavelength in microns, and S0SELL, S1SELL, S2SELL, L1SELL and L2SELL are all user-definable parameters on the **MATERIAL** statement. Default values for these parameters of various materials are given in Table B-38.

3.10.2 Adachi's Dispersion Model

To enable a model by Adachi [1], specify NDX.ADACHI on the **MATERIAL** statement. This model is given by Equations 3-414, 3-415, and 3-416.

$$n_r(w)^2 = \mathtt{AADACHI}\left[f(x_1) + 0.5\left(\frac{E_g}{E_g + \mathtt{DADACHI}}\right)^{1.5} f(x_2)\right] + \mathtt{BADACHI} \qquad 3\text{-}689$$

$$f(x_1) = \frac{1}{x_1^2}(2 - \sqrt{1 + x_1} - \sqrt{1 - x_1}) \quad , x_1 = \frac{\hbar\omega}{E_g} \qquad 3\text{-}690$$

$$f(x_2) = \frac{1}{x_2^2}(2 - \sqrt{1 + x_2} - \sqrt{1 - x_2}) \quad , x_2 = \frac{\hbar\omega}{E_g + \mathtt{DADACHI}} \qquad 3\text{-}691$$

In these equations, E_g is the band gap, h is plank's constant, ω is the optical frequency, and AADACHI, BADACHI and DADACHI are user-specifiable parameters on the **MATERIAL** statement. Default values for these parameters for various binary III-V materials are given in Table B-39. The compositionally dependent default values for the ternary combinations listed in Table B-39 are linearly interpolated from the binary values listed in the same table.

For nitride compounds, you can use a modified version of this model, which is described in Section 6.4.8 “GaN, InN, AlN, Al(x)Ga(1-x)N, In(x)Ga(1-x)N, Al(x)In(1-x)N, and Al(x)In(y)Ga(1-x-y)N”.

3.10.3 Tauc-Lorentz Dielectric Function with Optional Urbach Tail Model for Complex Index of Refraction

A parameterization of the Tauc-Lorentz dielectric function was proposed by Foldyna et.al. [89], which includes Urbach tails. The imaginary part of the dielectric function is given by Equation 3-692.

$$\varepsilon_2(E) = \begin{cases} 1\dfrac{\mathrm{TLU.A\ TLU.E0\ TLU.C}(E-\mathrm{TLU.EG})^2}{E(E^2-\mathrm{TLU.E0}^2)^2+\mathrm{TLU.C}^2E^2} & E \geq \mathrm{TLU.EC} \\ \dfrac{A_u}{E}\exp\left(\dfrac{E}{E_u}\right) & E < \mathrm{TLU.EC} \end{cases} \quad \text{3-692}$$

Here, for energies greater than TLU.EC, the standard Tauc-Lorentz form is used while for energies less than TLU.EC, Urbach tails are parameterized as a function of energy and two matching parameters A_u and E_u that are calculated by Equations 3-693 and 3-694 to give a continuous function and first derivative at E=TLU.EC.

$$E_u = \frac{(\mathrm{TLU.EC}-\mathrm{TLU.EG})}{2-2\mathrm{TLU.EC}(\mathrm{TLU.EC}-\mathrm{TLU.EG})\dfrac{\mathrm{TLU.C}^2+2(\mathrm{TLU.EC}^2-\mathrm{TLU.E0}^2)}{\mathrm{TLU.C}^2\ \mathrm{TLU.EC}^2+(\mathrm{TLU.EC}^2-\mathrm{TLU.E0}^2)^2}} \quad \text{3-693}$$

$$A_u = \exp\left(\frac{-\mathrm{TLU.EC}}{E_u}\right)\frac{\mathrm{TLU.A\ TLU.E0\ TLU.C}(\mathrm{TLU.EC}-\mathrm{TLU.EG})^2}{(\mathrm{TLU.EC}^2-\mathrm{TLU.E0}^2)^2+\mathrm{TLU.C}^2\ \mathrm{TLU.EC}^2} \quad \text{3-694}$$

In our implementation, if you do not specify TLU.EC, then the standard form of the Tauc-Lorentz dielectric function is used. For energies less than TLU.EG, the imaginary part is zero.

The real part of the dielectric function is obtained from the imaginary part using the Kramers-Krönig relation as given in Equation 3-695:

$$\varepsilon_1(E) = \mathrm{TLU.EPS} + \frac{2}{\pi}(\mathrm{C.P.})\int_0^\infty \frac{\xi\varepsilon_2(\xi)}{\xi^2-E^2}d\xi \quad \text{3-695}$$

where (C.P.) denotes the Cauchy principal value of the integral. The integral itself is resolved using some analytic forms the details of which are given in [89].

The complex index of refraction is calculated from the complex dielectric function using Equations 3-696 and 3-697.

$$n = \sqrt{\frac{\sqrt{\varepsilon_1^2+\varepsilon_2^2}+\varepsilon_1}{2}} \quad \text{3-696}$$

$$k = \sqrt{\frac{\sqrt{\varepsilon_1^2+\varepsilon_2^2}-\varepsilon_1}{2}} \quad \text{3-697}$$

You can enable the model by specifying TLU.INDEX on the **MODELS** statement. Table 3-141 gives a summary of the relevant model parameters.

Table 3-141 Parameters for the Tauc-Lorentz-Urbach Dielectric Model

Parameter	Statement	Default	Units
TLU.A	**MATERIAL**		eV
TLU.C	**MATERIAL**		eV
TLU.E0	**MATERIAL**		eV
TLU.EG	**MATERIAL**		eV
TLU.EC	**MATERIAL**		eV
TLU.EPS	**MATERIAL**	1.0	

3.11 Carrier Transport in an Applied Magnetic Field

Magnetic field effects have been used in semiconductor characterization measurements and exploited in semiconductor device applications. Atlas can model the effect of applied magnetic fields on the behavior of a semiconductor device.

The basic property, which is measured is the Hall coefficient R_H. For a uniformly doped device with a current flowing in the X direction and a magnetic field in the Z direction, an electric field (the Hall field) is set up in the Y direction. The Hall coefficient is then obtained by combining these measured quantities in the ratio:

$$R_H = \frac{E_y}{J_x B_z} \quad \text{3-698}$$

The Hall co-efficient also has a theoretical value:

$$R_H = \frac{r(p - s^2 n)}{q(p + sn)^2} \quad \text{3-699}$$

where s is the electron mobility divided by hole mobility. The Hall scattering factor, r, can be obtained from the mean free time between carrier collisions and is typically not very different from unity.

Another quantity that is measured experimentally is the Hall mobility:

$$\mu^* = |R_H \sigma| \quad \text{3-700}$$

where σ is the electrical conductivity. We will consider the electron and hole currents separately. Therefore, taking the limits ($n >> p$) and then ($p >> n$) in Equation 3-700, we obtain the Hall mobilities for electrons and holes respectively as:

$$\mu^*_n = r\mu_n$$
$$\mu^*_p = r\mu_p \quad \text{3-701}$$

where μ_n and μ_p are the carrier mobilities. The Hall scattering factor for electrons can be different to that for holes in general. Published values for silicon are 1.2 and 1.2 [7] and 1.1 and 0.7 [265] for electrons and holes respectively. The default in Atlas is the latter. You can set the values on the **MODELS** statement using the R.ELEC and R.HOLE parameters.

The effect of the magnetic field on a carrier travelling with velocity v is to add a term called the Lorentz force:

$$q(\underline{v} \times \underline{B}) \quad \text{3-702}$$

to the force it feels. The magnetic field density B is a vector (B_x, B_y, B_z) and is in units of Tesla ($V.s/m^2$) in Atlas.

The consequence of this extra force in a semiconductor can be calculated by including it in a relaxation time based transport equation and making a low-field approximation [7]. You can relate the current density vector obtained with a magnetic field applied to that obtained with no magnetic field using a linear equation:

$$\underset{\sim}{J}_B = \underset{\sim}{M}\underset{\sim}{J}_0 \quad \text{3-703}$$

The Matrix M can be written as:

$$\underset{\sim}{M} = \frac{1}{1 + a^2 + b^2 + c^2}\begin{pmatrix} 1 + a^2 & ab - c & ca + b \\ c + ab & 1 + b^2 & bc - a \\ ca - b & a + bc & 1 + c^2 \end{pmatrix} \quad \text{3-704}$$

where $a = \mu^* B_x$, $b = \mu^* B_y$, $c = \mu^* B_z$. The matrix takes this form for both electrons and holes. Using this form, you can include the magnetic field effects in Atlas. The model is derived from an expansion in powers of Magnetic field and is only accurate if the weak field condition:

$$\mu^* |B| \ll 1 \quad \text{3-705}$$

is satisfied. Therefore, for example, a field of 1 Tesla in Silicon with a typical mobility of 0.1 $m^2/(\mathrm{V.s})$, the product is 0.1 so that the condition is satisfied.

There are several published methods that have been considered for discretizing Equation 3-703 on a triangular or prismatic mesh. Some of them are based on the Box method extension of Allegretto et al [7]. It can be proven that, except in the case of uniform carrier density, this method has the considerable drawback that it does not conserve current at the level of individual Voronoi cells. Alternative means are used by Allegretto to obtain continuity of terminal currents, but this is not acceptable for Atlas.

Another method is to obtain a least squares value for the current density vector $\vec{J}_0$, for each triangle [240]. The components of these vectors are then multiplied by M and the Box discretization obtained by projecting the resulting current density vector $\vec{J}_B$ along each side of the triangle. It was found that this method, along with several variations of this method, could result in carrier densities which exhibit unphysical spatial oscillations and also result in negative carrier densities. This instability for methods based on a least squares current has previously been observed [264]. The least-squares based method is accessible in Magnetic for comparative purposes only. You specify `MAG2DLSQ` on the **`METHOD`** statement to enable it.

To avoid the problems associated with the aforementioned methods, we developed two discretizations that are less problematic. The Magnetic 3D module accomodates both of these discretizations. For the first one, we start by expressing the integral of the current flux over a Voronoi cell,

$$\int \vec{J}_B \cdot \vec{dS} = \int M\vec{J}_0 \cdot \vec{dS} \quad \text{3-706}$$

We can rewrite the right hand side as

$$\frac{1}{(1+a^2+b^2+c^2)}\int (N+T)\vec{J_0}\cdot\vec{dS} \qquad 3\text{-}707$$

where the normal matrix N=

$$\begin{pmatrix} 1+a^2 & 0 & 0 \\ 0 & 1+b^2 & 0 \\ 0 & 0 & 1+c^2 \end{pmatrix} \qquad 3\text{-}708$$

and the tangential matrix T =

$$\begin{pmatrix} 0 & ab-c & ca+b \\ c+ab & 0 & bc-a \\ ca-b & a+bc & 0 \end{pmatrix} \qquad 3\text{-}709$$

using the notation for the matrix entries as defined above. So long as the mesh elements are right-angled prisms with non-diagonal faces in the coordinate directions (a rectangular mesh), the normal part results in the usual edge based discretization with only a scalar modification. Also under this restriction, the normals to the Voronoi edges become after multiplication by T^T.

$$\begin{aligned} (1,0,0) &\Rightarrow (0, ab-c, ca+b) \\ (0,1,0) &\Rightarrow (c+ab, 0, bc-a) \\ (0,0,1) &\Rightarrow (c-ab, a+bc, 0) \end{aligned} \qquad 3\text{-}710$$

If we take into account the direction of the outward normals (i.e., $(\pm 1, 0, 0)$, $(0, \pm 1, 0)$, and $(0, 0, \pm 1)$), then you can understand the currents as band integrals around the Voronoi cell in a direction tangential to the magnetic direction. In a prismatic mesh, there are three different configurations to consider. Each requires the current components parallel to the Voronoi cell boundaries to be approximated. This is done by interpolating solution variables to edge and face midpoints, and then using the Scharfetter-Gummel method to obtain the discretised current. By multiplying the interpolated current components by the appropriate magnetic field terms, the contributions to the assembly of the continuity equation over the Voronoi cell may be evaluated. This method apparently gives good convergence behavior and stable solutions. It is enabled for the electron continuity equation by specifying `MAG3DINT.N` on the **METHOD** statement, and for the hole continuity equation, by specifying `MAG3DINT.P` on the **METHOD** statement.

The restriction of a rectangular mesh is unacceptable for many devices, and for this reason The restriction of a rectangular mesh is unacceptable for many devices. For this reason Silvaco developed a new discretization for both the Magnetic and Magnetic 3D modules. The stability of Finite Element discretizations of the current continuity equations was discussed by He [123]. Silvaco indirectly extended this approach to the Box discretization method. The analysis is lengthy, so only the result is given.

The current along any edge in the device is obtained using the Scharfetter-Gummel discretization but with two modifications. The first is that there is a prefactor equal to $(1 + a^2$

$+ b^2 + c^2)^{-1}$, and the second is that the argument to the Bernoulli function has the following term added to it

$$\left(\left(a^2\frac{\partial\phi}{\partial x} + (ab-c)\frac{\partial\phi}{\partial y} + (ca+b)\frac{\partial\phi}{\partial z}\right)\Delta x + \left((c+ab)\frac{\partial\phi}{\partial x} + b^2\frac{\partial\phi}{\partial x} + (bc-a)\frac{\partial\phi}{\partial z}\right)\Delta y\right)/K_BT \quad 3\text{-}711$$
$$+\left((ca-b)\frac{\partial\phi}{\partial x} + (a+bc)\frac{\partial\phi}{\partial y} + c^2\frac{\partial\phi}{\partial z}\right)\Delta z/K_BT$$

where (Δx, Δy, Δz) is the spatial vector describing the edge, and K_B is the Boltzmann constant. The quantity ϕ is the electron quasi-Fermi level for electron current and the hole quasi-Fermi level for hole current. This discretization is the default in Magnetic 3D and the only one available in Magnetic.

One possible weakness of this method is that it uses the quasi-Fermi levels, and for minority carriers these are sometimes only weakly coupled to the boundaries. This can cause instabilities in the solution. To counteract this, you use the parameters MAGMIN.N and MAGMIN.P on the **MODELS** statement. The effect of these parameters is to modify the value of electron concentration and hole concentration used in the calculation of the respective quasi-Fermi levels to

$$n_{eff} = n + \texttt{MAGMIN.N} \times N_c \quad 3\text{-}712$$

and

$$p_{eff} = p + \texttt{MAGMIN.P} \times N_v \quad 3\text{-}713$$

The values of MAGMIN.N and MAGMIN.P are typically in the range 0 to 1. The larger the value, the more insensitive the gradients of the quasi-Fermi levels are to the actual carrier densities. Consequently, they become like the electric field at large values of MAGMIN.N and MAGMIN.P. If MAGMIN.N or MAGMIN.P are set to be a negative value, then the respective addition to the Bernoulli function argument for electrons or holes Equation 3-711 is set to zero.

The boundary conditions on the electrostatic potential are modified in the presence of a magnetic field. On all external surfaces and internal interfaces with insulating materials, the normal gradient of the electrostatic potential is set to be

$$\left.\frac{\partial\psi}{\partial N}\right|_{sc} = (A_n(\vec{B}\wedge\vec{J}_n) + A_p(\vec{B}\wedge\vec{J}_p))\cdot\vec{N} \quad 3\text{-}714$$

where $\vec{N}$ is an outward normal to the surface, $\vec{J}_n$ and $\vec{J}_p$ are the current densities near the surface. Also

$$A_n = \frac{\sigma_n R_n}{\sigma} \quad 3\text{-}715$$

and

$$A_p = \frac{\sigma_p R_p}{\sigma} \quad 3\text{-}716$$

and where $R_{n,p}$ are given explicitly by

$$R_n = \frac{-\mu_n r_n}{\mu_n n + \mu_p p} \quad 3\text{-}717$$

$$R_p = \frac{\mu_p r_p}{\mu_n n + \mu_p p} \quad 3\text{-}718$$

The quantities R_n and R_p simplify to the respective unipolar Hall coefficients in the limits $n >> p$ and $p >> n$. The boundary conditions can be viewed as a local Hall field.

To invoke this model in Atlas2D, simply supply a value for BZ on the **MODELS** statement. Nonzero values of BX and BY are not permitted because they would generally result in current in the Z direction, thereby voiding the assumption of two-dimensionality. In Atlas3D, any combination of BX, BY or BZ can be specified as long as condition (Equation 3-705) is satisfied.

Alternatively, a position dependent magnetic field vector can be specified by using a C-Interpreter function. Use the parameter F.BFIELD on the **MODELS** statement to specify the name of the file containing an implementation of the C-Interpreter function. The magnetic field should satisfy the constraint imposed by Maxwell's equations that the divergence of the magnetic field vector is everywhere zero.

Table 3-142 shows the parameters used for the carrier transport in a magnetic field.

Table 3-142 Parameters for Galvanic Transport Model

Statement	Parameter	Type	Default	Units
MODELS	BX	Real	0.0	Tesla
MODELS	BY	Real	0.0	Tesla
MODELS	BZ	Real	0.0	Tesla
MODELS	F.BFIELD	Character		
METHOD	MAG3DINT.N	Logical	False	
METHOD	MAG3DINT.P	Logical	False	
METHOD	MAG2DLSQ	Logical	False	
MODELS	MAGMIN.N	Real	0.0	
MODELS	MAGMIN.P	Real	0.0	
MODELS	R.ELEC	Real	1.1	
MODELS	R.HOLE	Real	0.7	

3.12 Anisotropic Relative Dielectric Permittivity

In the general case, the relationship between the electric displacement vector and the electric field is given by:

$$\vec{D} = \varepsilon_0 \underset{\sim}{\varepsilon} \vec{E} \quad \text{3-719}$$

where ε, the relative dielectric permittivity, is a symmetric second order tensor represented by a 3×3 matrix. Since it is symmetric, it can be written as:

$$\underset{\sim}{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yx} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{pmatrix} \quad \text{3-720}$$

In the isotropic case, the off-diagonal terms are zero and the diagonal terms all have the same value as each other. In the anisotropic case, you can always set the off-diagonal elements to zero by a suitable transformation of coordinates. This coordinate system may not match the coordinate system used by Atlas for the simulation. Therefore, it is necessary to allow Atlas to model anisotropic permittivity for the relationship Equation 3-720 where the off-diagonal elements are non-zero.

The discretization of the flux term in the Poisson equation:

$$\varepsilon_0 \nabla \cdot \left(\underset{\sim}{\varepsilon} \vec{E} \right) \quad \text{3-721}$$

requires a more general treatment in the anisotropic case. For example, the x-component of the electric displacement vector will now contain a contribution from all the components of the electric field:

$$D_x = \varepsilon_{xx} E_x + \varepsilon_{xy} E_y + \varepsilon_{xz} E_z \quad \text{3-722}$$

and so the discretization is more complicated and uses more CPU time. A simplification can be made when:

- the permittivity matrix is diagonal.
- the mesh used in modeling the device is rectangular.

In this case, discretization will occur only in the directions of the coordinate axes and simply using a scalar permittivity for each direction is correct. This retains the simplicity and speed of the discretization for isotropic materials.

Atlas allows you to specify a value of anisotropic relative dielectric permittivity using the PERM.ANISO parameter on the **MATERIAL** statement. By default, this is the value of (ε_{yy}) if the simulation is in 2D and (ε_{zz}) if it is in 3D. The off-diagonal elements are assumed to be zero. In the case of a 3D simulation, you can apply the value of PERM.ANISO to the Y direction instead of the Z direction by specifying YDIR.ANISO or ^ZDIR.ANISO on the **MATERIAL** statement. You can specify the value of (ε_{xx}) by using the PERMITTIVITY parameter on the

MATERIAL statement. The simpler version of discretization will be used in this case by default.

If the coordinate system in which the permittivity tensor is diagonal and the Atlas coordinate system are non-coincident, then the coordinate transformation can be specified as a set of vectors

```
X = (X.X, X.Y, X.Z)
Y = (Y.X, Y.Y, Y.Z)
Z = (Z.X, Z.Y, Z.Z)
```

where the vector components can be specified on the **MATERIAL** statement.

The default is that

```
X = (1.0, 0.0, 0.0)
Y = (0.0, 1.0, 0.0)
Z = (0.0, 0.0, 1.0)
```

and you should specify all necessary components. You do not need to normalize the vectors to unit length. Atlas will normalize them and transform the relative dielectric tensor according to the usual rules. If you specify any component of X, Y or Z, then the more complete form of discretization will be used. You can also enable the more complete form of electric displacement vector discretization by using the E.FULL.ANISO parameter on the **MATERIAL** statement.

Table 3-143 shows the parameters used in the Anisotropic Relative Dielectric Permittivity.

Table 3-143 MATERIAL Statement Parameters

Parameter	Type	Default
PERM.ANISO	Real	-999.0
YDIR.ANISO	Logical	False
ZDIR.ANISO	Logical	True
E.FULL.ANISO	Logical	False
X.X	Real	1.0
X.Y	Real	0.0
X.Z	Real	0.0
Y.X	Real	0.0
Y.Y	Real	1.0
Y.Z	Real	0.0
Z.X	Real	0.0
Z.Y	Real	0.0
Z.Z	Real	1.0

3.13 Field Emission from Semiconductor Surfaces

Vacuum microelectronic devices often utilize field emission from the sharp tips of emitters made from semiconductor materials [76][219]. The current emitted can be modeled using a tunneling model, usually a variant of the Fowler-Nordheim model. Atlas implements one such model, which we refer to as the Zaidman model [373]. In this model, the tunneling current is calculated at each segment of a semiconductor/vacuum interface. The current density in A/cm^2 is given by

$$J = \frac{\texttt{F.AE}F^2}{\phi\texttt{TSQ.ZAIDMAN}} \exp\left(\frac{-\texttt{F.BE}(0.95 - 1.43641 \times 10^{-7} F/\phi^2)\phi^{3/2}}{F}\right) \quad \text{3-723}$$

Here, ϕ is the semiconductor-vacuum barrier height in eV. F is the electric field strength normal to the surface in units of V/cm and TSQ.ZAIDMAN is a fitting parameter. The parameters F.AE and F.BE are also fitting parameters.

The current produced in each segment then follows the field lines until it either reaches a contact or exits the device boundary. To enable the model, use the ZAIDMAN parameter on the **MODELS** statement. It should not be used concurrently with the FNORD model.

It presently omits self-consistent evaluation with respect to the carrier continuity equations. It also neglects any modification of the current in the vacuum arising from the space charge present there.

Table 3-144 Zaidmann Model Parameters

Statement	Parameter	Default	Units
MODELS	ZAIDMAN	False	
MODELS	TSQ.ZAIDMAN	1.1	
MODELS	F.AE	1.5414×10^{-6}	Q^2/Vs
MODELS	F.BE	6.8308×10^{7}	$cm^{-1}\ V^{-1/2}Q^{-3/2}$

3.14 Conduction in Silicon Nitride

It is known that at high fields and for temperatures above 325 Kelvin, the conduction in Silicon Nitride is dominated by Poole-Frenkel emission. Atlas can model this by adding an electron generation term to the electron continuity equation. To use this model, you define a trap density, NT.N, using the **NITRIDECHARGE** statement. You also change the Silicon Nitride layer into a semiconductor using the SEMICONDUCTOR parameter on the **MATERIAL** statement.

The electron generation term is proportional to

$$\mathtt{PF.A}\exp\left(-\frac{\mathtt{PF.BARRIER}}{K_B T_L}\right) + \mathtt{PF.B}\exp\left(-\frac{\mathtt{PF.BARRIER}}{K_B T_L} + \frac{Beta\sqrt{F}}{K_B T_L}\right) \quad \text{3-724}$$

where PF.A, PF.B, and PF.BARRIER can be specified on the **NITRIDECHARGE** statement. The first term is an ohmic term that will dominate at low fields. The local electric field is F, the lattice temperature is T_L, K_B is the Boltzmann constant, and *Beta* is a factor obtained from the equation:

$$Beta = Q\sqrt{Q/(\pi\varepsilon)} \quad \text{3-725}$$

where Q is the electonic charge, and ε is the dielectric permittivity of Silicon Nitride. This model only applies to steady state simulations. The traps should first be charged using the NIT.N parameter on the **SOLVE** statement. The value MUN on the **MOBILITY** statement can also be used to calibrate the resulting I-V curves.

3.15 Generic Ion Transport Model

The ability to model the transport of ionic species in semiconductor devices is important for studies of performance degradation and for investigating the behavior of novel devices [322]. Atlas is able to simulate the transport of up to three different ionic species within the drift-diffusion framework. Denoting the concentrations of the ionic species as C_1 , C_2, and C_3 , and their fluxes as F_{c1}, F_{c2}, and F_{c3} respectively, the continuity equation for species i is

$$\frac{\partial C_i(r,t)}{\partial t} + \nabla \cdot F_{ci}(r,t) = G_i(r,t) - R_i(r,t) \quad \text{3-726}$$

where $G_i(r,t)$ and $R_i(r,t)$ are the generation and recombination rates of the ions.

All quantities are functions of position r, and time t. The general equation for the flux is

$$F_{Ci} = \upsilon_i C_i(r,t) - D_i \nabla C_i(r,t) \quad \text{3-727}$$

where υ_i is the velocity and D_i the diffusion co-efficient. The velocity due to the electrostatic potential, ψ is given by

$$\upsilon_i = -\frac{Z_i}{|Z_i|}\mu_i \nabla\psi(r,t) \quad \text{3-728}$$

where Z_i is an integer denoting the number of electronic charges on the ion. For a positively charged species, Z_i is positive, and is negative for a negatively charged species. Instead of working with the ionic mobility μ_i, we use the generalised Einstein relationship

$$\mu_i = \frac{D_i|Z_i|}{K_B T} \quad \text{3-729}$$

to give a flux equation

$$F_{Ci} = -D_i\left(\frac{Z_i}{K_B T} C_i(r,t)\nabla\psi(r,t) + \nabla C_i(r,t)\right) \quad \text{3-730}$$

where K_B is the Boltzmann constant and T is the device temperature. For a neutral species $Z_i = 0$, and Equation 3-730 simplifies to Fick's law. For non-zero values of Z_i, the electrostatic field adds a contribution to the flux according to the sign and magnitude of Z_i. The boundary conditions used are that no ions can enter or leave the device, either through its external surfaces or its contacts. In practice some transport out of the contacts may occur. But, a steady state flow of ions through the device is not expected and thus cannot be simulated. These conditions of zero external flux mean that the total ion concentrations are preserved and that the net Generation-Recombination term for each ionic species must be zero. This can be proven by integrating Equation 3-726 over the device and applying Gauss's flux theorem to give

$$\frac{\partial}{\partial t}\int_V C_i(r,t)dV + \int_S F_{ci}\cdot\hat{n}dS = \int_V (G_i - R_i)dV \quad \text{3-731}$$

The integral of the flux over the device surface is identically zero and so for a steady state solution the integral of generation-recombination terms must be zero. This implies that the total number of ions in the device remains constant.

To simulate one or more ionic species you first specify an initial distribution by using the **DOPING** statement. One of the parameters SPECIES1, SPECIES2, or SPECIES3 must be specified on a **DOPING** statement to set an initial ionic species distribution. The species distribution can be set in insulator regions as well as semiconductor regions. In the absence of inter-species reactions, the initial total dose of each simulated species will be preserved throughout the simulation. SOLVE INIT does not calculate equilibrium values of the SPECIES concentrations. This is because the timescale to achieve equilibrium may be much longer than a typical observation time of a device.

You select the number of species to simulate using the NSPECIES parameter on the **MODELS** statement. This can be 1, 2, or 3. If GIGA is enabled this number can only be 1 or 2. NSPECIES=1 causes the simulation of SPECIES1. NSPECIES=2 simulates SPECIES1 and SPECIES2. NSPECIES=3 simulates SPECIES1, SPECIES2, and SPECIES3. You utilize the SPECIES1.Z, SPECIES2.Z, and SPECIES3.Z parameters on the **MODELS** statement to assign charges to the ionic species.

You set the diffusion coefficient D_i for each species on a material by material basis with the **MATERIAL** statement. It is given by

$$D_1 = \text{SPECIES1.HOP}^2 \quad \text{SPECIES1.AF} \quad \text{EXP}(-\text{SPECIES1.EA/KT})$$

$$D_2 = \text{SPECIES2.HOP}^2 \quad \text{SPECIES2.AF} \quad \text{EXP}(-\text{SPECIES2.EA/KT}) \qquad \text{3-732}$$

$$D_3 = \text{SPECIES3.HOP}^2 \quad \text{SPECIES3.AF} \quad \text{EXP}(-\text{SPECIES3.EA/KT})$$

where all the parameters are set on the **MATERIAL** statement. The units of D_i are $cm^2\ s^{-1}$. Different ionic species with the same diffusion co-efficient but different values of Z_i will have different transport properties. This can be seen from consideration of Equation 3-730.

For ions with a very small diffusion coefficient, the steady state solution may not be appropriate as it corresponds to an infinite simulation time. On a practical timescale, the species distribution may not have changed by a significant amount. In this case, you specify the FREEZESPECIES parameter on a steady state **SOLVE**, followed by a transient **SOLVE** without the FREEZESPECIES parameter. The FREEZESPECIES parameter gives the steady state solution with the species concentration held fixed. The transient solve then gives the dynamic development of the species distribution. It is also possible to perform small signal AC simulations in addition to both steady state and transient simulations.

The multistate traps described in Section 3.3.4 “Multistate Traps” can emit and trap ionic species. Because ionic species can be modeled in all materials, the **INTTRAP** [MSCTRAP] statement can be made to apply to interfaces between insulators and electrodes, insulators and conducting regions, and different insulating regions. The respective flags are I.M, I.C, and I.I. This is in addition to the other defined interface types.

By default, the transport of the ionic species is simulated in the whole device. In transient mode, ions can be restricted from entering a particular material by using a very small ionic diffusion co-efficient for that material. This will not work, however, in the case of steady-state simulations and so an alternative method is available. You can restrict the ionic species to insulator regions by using the SPECIES.INS parameter on the **METHOD** statement. You can restrict the ionic species to a numbered region by using the SPECIES.REG parameter on the **METHOD** statement. You can restrict the ionic species to a named material by using the SPECIES.MAT parameter on the **METHOD** statement.

There is no ionic flow across the boundaries between ion containing and ion free areas of the device. Only one of the parameters SPECIES.INS, SPECIES.MAT, and SPECIES.REG can be active at one time.

In common with other variables, the maximum relative update of the species concentrations in a non-linear solver iteration can be limited. This can sometimes help to improve the convergence of the equations. The value of the SPECIES.MAXX parameter on the **METHOD** statement limits the maximum update to $10^{SPECIES.MAXX}$.

This can help improve convergence, as can limiting the maximum update for other variables.

The densities of the ionic species are automatically output to any Standard Structure file that is saved. You can use the parameters SPECIES1, SPECIES2, or SPECIES3 on the **PROBE** statement to save ionic species quantities to a log file.

The input deck parameters that you use for the generic ionic species transport model are summarized in Table 3-145 and some example statements are given below.

How to include reactions between the species is described in Section 3.16 “Generic Ion Reaction Model”.

Table 3-145 Parameters for the Generic Ionic Species Transport Model

Statement	Parameter	Type	Default	Units
DOPING	SPECIES1	Logical	False	
DOPING	SPECIES2	Logical	False	
DOPING	SPECIES3	Logical	False	
MODELS	NSPECIES	Integer	0	
MODELS	SPECIES1.Z	Integer	0	
MODELS	SPECIES2.Z	Integer	0	
MODELS	SPECIES3.Z	Integer	0	
PROBE	SPECIES1	Logical	False	
PROBE	SPECIES2	Logical	False	
PROBE	SPECIES3	Logical	False	
MATERIAL	SPECIES1.HOP	Real	1.0e-6	cm
MATERIAL	SPECIES1.AF	Real	1.0e15	Hz
MATERIAL	SPECIES1.EA	Real	0.25	eV
MATERIAL	SPECIES2.HOP	Real	1.0e-6	cm
MATERIAL	SPECIES2.AF	Real	1.0e15	Hz
MATERIAL	SPECIES2.EA	Real	0.25	eV
MATERIAL	SPECIES3.HOP	Real	1.0e-6	cm

Table 3-145 Parameters for the Generic Ionic Species Transport Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
MATERIAL	SPECIES3.AF	Real	1.0e15	Hz
MATERIAL	SPECIES3.EA	Real	0.25	eV
METHOD	SPECIES.INS	Logical	False	
METHOD	SPECIES.MAXX	Real	1000.0	
METHOD	SPECIES.REG	Real		
METHOD	SPECIES.MAT	Character		
SOLVE	FREEZESPECIES	Logical	False	

Examples

```
//Set up initial profile for SPECIES1
DOPING REGION=1 SPECIES1 GAUSSIAN CONC=1e15 CHAR=0.06 PEAK=0.25

//Enable simulation of 3 species with charges +1,-2 and -1
respectively.
MODELS NSPECIES=3 SPECIES1.Z=1 SPECIES2.Z=-2 SPECIES3.Z=-1

// Get the value of species1 integrated over whole device
PROBE SPECIES1 INTEGRATE

// Restrict simulation to insulator regions only, and limit maximum
relative update of all species in non-linear iteration to be 100.
METHOD SPECIES.MAXX=100 SPECIES.INS
```

3.16 Generic Ion Reaction Model

The generic ion transport model of the previous section is augmented by the capability to simulate general reactions between the ionic species. The examples are

- Capture of free carriers to change the ionic charge state.
- Emission of ionic charges to change the ionic charge state.
- Dimerisation reactions.
- Cation-Anion reactions.
- Vacancy-Interstitial pair generation and recombination. (This is not available yet due to licensing issues).

You specify the reactions required on one or more **REACTION** statements. The syntax to do this requires separating each reaction into reactants and products, or more colloquially into before and after. The reactions are potentially reversible so this is only a semantic distinction. Each equation is written as

$$\texttt{BEFORE.N}\ e^- + \texttt{BEFORE.P}\ h^+ + \texttt{BEFORE.SP1 SPECIES1} + \texttt{BEFORE.SP2 SPECIES2} + \texttt{BEFORE.SP3 SPECIES3} \Leftrightarrow \texttt{AFTER.N}\ e^- + \texttt{AFTER.P}\ h^+ + \texttt{AFTER.SP1 SPECIES1} + \texttt{AFTER.SP2 SPECIES2} + \texttt{AFTER.SP3 SPECIES3} \quad (3\text{-}733)$$

For example, to simulate two charge states of a species, X^+ and X^{++}, you need to take into account the possible reactions with free carriers

$$X^{++} + e^- \Leftrightarrow X^+$$
$$X^+ + h^+ \Leftrightarrow X^{++} \quad (3\text{-}734)$$

Assume that X^+ has been set up as SPECIES1 and X^{++} as SPECIES2, using SPECIES1.Z=1 and SPECIES2.Z=2 on the **MODELS** statement. Then the first equation is set up with the statement

```
REACTION BEFORE.N=1 BEFORE.SP2=1 AFTER.SP1=1
```

and the second reaction by

```
REACTION BEFORE.P=1 BEFORE.SP1=1 AFTER.SP2=1
```

Modeling a double electron capture/emission

$$X^{3+} + e^- + e^- \Leftrightarrow X^+ \quad (3\text{-}735)$$

with X^{3+} as SPECIES1 and X^+ as SPECIES2 requires the statement

```
REACTION BEFORE.N=2 BEFORE.SP1=1 AFTER.SP2=1
```

To model a dimerisation reaction $2\mathrm{H} \Leftrightarrow \mathrm{H2}$ with H as SPECIES1 and H_2 as SPECIES2 requires the statement

```
REACTION BEFORE.SP1=2 AFTER.SP2=1
```

The format allows a wide range of reactions to be specified. Atlas checks that charge conservation is obeyed in each **REACTION** statement and exits with an error if charge conservation is violated.

You also need to model the reaction rates. This is calculated by multiplying together all the densities of the reactant components present in the reaction, raised to the power of their BEFORE coefficients. Thus, a SPECIES1 concentration is raised to the power of BEFORE.SP1 and the result multiplies the forward rate coefficient, similarly for the other SPECIES concentrations and electron and hole concentrations. This is then multipled by $\text{FORWARD.RATE}e^{(-\text{FORWARD.EA/KT})}$ to give the forward reaction rate. The same process is applied to the product species, and this is multiplied by $\text{REVERSE.RATE}e^{(-\text{REVERSE.EA/KT})}$.

If a BEFORE or AFTER coefficient is zero, then the corresponding concentration will not appear in the rate equation.

For the reaction described above in Equation 3-735, the net rate is

$$\text{FORWARD.RATE}e^{(-\text{FORWARD.EA/KT})}[X^{3+}][e]^{-2} - \text{REVERSE.RATE}e^{(-\text{REVERSE.EA/KT})}[X^{+}] \quad 3\text{-}736$$

where the square brackets denote the local concentration of the bracketed quantity. Because the reaction consumes two electrons for every ion, the electron concentration is raised to the power of 2, which is the value of BEFORE.N.

You can add a constant value to the reaction rate using the EQUIRATE parameter on the **REACTION** statement. This is useful if you have a constant product of species in equilibrium, say a thermally generated vacancy-interstitial pair with reaction $VX^0 + IX^0 \Leftrightarrow [NULL]$, then the reaction rate would be

$$\text{FORWARD.RATE}e^{(-\text{FORWARD.EA/KT})}[VX^0][IX^0] - \text{EQUIRATE} \quad 3\text{-}737$$

because there are no terms on the AFTER side of the reaction.

The reaction rates are included in the G_i - R_i terms of the continuity equations of the free carriers and ionic species. Terms on the BEFORE side of the equation are included as recombination rates and those on the AFTER side as generation rates. The rates are multiplied by the relevant BEFORE or AFTER coefficient. This leads to a conservation law for the sums of the volume integrals of the species, so long as one takes into account the composition numbers of the species. This conservation law is

$$\int_V (\text{SPECIES1.A SPECIES1}(r) + \text{SPECIES2.A SPECIES2}(r) + \text{SPECIES3.A SPECIES3}(r))dV = const \quad 3\text{-}738$$

where SPECIES1.A, SPECIES2.A, and SPECIES3.A are the composition numbers. Atlas attempts to work these out automatically, but if this is impossible then you set them on the **MODELS** statement. The following example clarifies this.

We consider two possible reactions between three species. SPECIES1 is an electrically neutral atom, H. SPECIES2 is the neutral molecule, H_2. SPECIES3 is the negatively charged ion, H^-. You set these up on the **MODELS** statement

```
MODELS NSPECIES=3 SPECIES1.Z=0  SPECIES2.Z=0
SPECIES3.Z=-1 [SPECIES1.A=1 SPECIES2.A=2 SPECIES3.A=1]
```

where the composition number specification is optional. The reactions considered are a dimerisation

$$2H \Leftrightarrow H_2 \quad \text{3-739}$$

with a net reaction rate R_{dimer} and an electron capture reaction

$$H + e^- \Leftrightarrow H^- \quad \text{3-740}$$

with a net reaction rate R_{elec} and a hole capture reaction

$$H^- + h^+ \Leftrightarrow H \quad \text{3-741}$$

and a net reaction rate R_{hole}. These reactions are setup using the following three **REACTION** statements (rate parameters are omitted for clarity)

```
REACTION BEFORE.SP1=2 AFTER.SP2=1
REACTION BEFORE.SP1=1 BEFORE.N=1 AFTER.SP3=1
REACTION BEFORE.SP3=1 BEFORE.P=1 AFTER.SP1=1
```

The continuity equations for the three species become

$$\begin{aligned} \frac{dH}{dt} + \nabla \cdot F_H &= -2R_{dimer} - R_{elec} + R_{hole} \\ \frac{dH_2}{dt} + \nabla \cdot F_{H_2} &= R_{dimer} \\ \frac{dH^-}{dt} + \nabla F_{H^-} &= R_{elec} - R_{hole} \end{aligned} \quad \text{3-742}$$

If two times the middle equation is added to the other two equations, then the right hand side of the equation identically vanishes. The resulting equation is integrated over the device volume. The divergence theorem is applied to the flux terms, which will then vanish because of the zero flux boundary condition. The equation is integrated over time to give

$$\int_V (H + 2H_2 + H^-)dV = const \quad \text{3-743}$$

which is the same as obtained from Equation 3-738 after putting in the composition numbers. If Atlas cannot automatically determine the composition numbers, then you must specify them on the **MODELS** statement using the parameters SPECIES1.A, SPECIES2.A, and SPECIES3.A. Atlas will check that your values are correct.

The above set of reactions also result in a recombination rate of R_{elec} going into the electron continuity equation and R_{hole} going into the hole continuity equation. These carriers are supplied in the usual way.

The net reaction rates of the ionic species and electrons and holes are output to any Standard Structure file, which is saved if you set the parameter U.SPECIES on the **OUTPUT** statement. You use the parameters REACTION.ELEC, REACTION.HOLE, REACTION.SP1, REACTION.SP2, or REACTION.SP3 on the PROBE statement to output the net reaction rates for each carrier or species to a log file.

The input deck parameters that you use for the generic species reaction model are summarised in Table 3-146. The associated generic transport model is described in the previous section. All **REACTION** statements must occur after the **MODELS** statement which defines the SPECIES.

Table 3-146 Parameters for the Generic Ionic Species Reaction Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
REACTION	BEFORE.N	Integer	0	
REACTION	BEFORE.P	Integer	0	
REACTION	BEFORE.SP1	Integer	0	
REACTION	BEFORE.SP2	Integer	0	
REACTION	BEFORE.SP3	Integer	0	
REACTION	AFTER.N	Integer	0	
REACTION	AFTER.P	Integer	0	
REACTION	AFTER.SP1	Integer	0	
REACTION	AFTER.SP2	Integer	0	
REACTION	AFTER.SP3	Integer	0	
REACTION	FORWARD.RATE	Real	0.0	Note 1
REACTION	REVERSE.RATE	Real	0.0	Note 2
REACTION	EQUIRATE	Real	0.0	$cm^{-3}s^{-1}$
REACTION	FORWARD.EA	Real	0.0	eV
REACTION	REVERSE.EA	Real	0.0	eV
MODELS	SPECIES1.A	Integer	1	
MODELS	SPECIES2.A	Integer	1	
MODELS	SPECIES3.A	Integer	1	
PROBE	REACTION.ELEC	Logical	False	
PROBE	REACTION.HOLE	Logical	False	
PROBE	REACTION.SP1	Logical	False	
PROBE	REACTION.SP2	Logical	False	
PROBE	REACTION.SP3	Logical	False	
OUTPUT	U.SPECIES	Logical	False	

1. Should give dimensions of forward rate as $cm^{-3}s^{-1}$
2. Should give dimensions of reverse rate as $cm^{-3}s^{-1}$

3.17 The Boltzmann Transport Equation Solver

There are several approaches available for approximately solving the Boltzmann transport equation for electrons and holes in semiconductors. One approach is to expand the carrier distribution function as a series of spherical harmonics in wavevector space [128].. The main application of the approximate solution of the Boltzmann transport equation is in modeling transistor degradation mechanisms.

The General Framework Degradation Model (see Section 4.1.3 "General Framework Degradation Model") requires the zeroth and first order expansions of the carrier distribution function. For this purpose the relatively simple first order Spherical Harmonic method of Ventura et al [331] is adequate and is implemented in Atlas for Silicon.

The basic homogeneous equation for the zeroth order carrier distribution function f_o is

$$q^2F^2\frac{\partial}{\partial E}\left(g(E)\tau(E)u_g^2(E)\frac{\partial f_o}{\partial E}\right) + 3g(E)c_{op}\left[g(E+\hbar\omega)\left(N_{op}^+f_o(E+\hbar\omega)-N_{op}f_o(E)\right)\right.$$
$$\left. - g(E-\hbar\omega)\left(N_{op}^+f_o(E)-N_{op}f_o(E-\hbar\omega)\right)\right] = 0 \quad \text{3-744}$$

where

- E is the energy in eV.
- $g(E)$ is the density of states in $m^{-3}\,ev^{-1}$
- F is the field in V/m.
- τ is a scattering lifetime in seconds.
- u_g is the group velocity in m/s.
- c_{op} is the optical phonon scattering coefficient in m^3J/s.
- N_{op} is the optical phonon occupation number and the optical phonon energy is $\hbar\omega$ in eV .

N_{op}^+ is the optical phonon occupation number plus one, which is simplified as follows:

$$N_{op}^+ = N_{op} + 1 = \exp(q\hbar\omega/K_bT_l)N_{op} \quad \text{3-745}$$

Here, K_b is the Boltzmann constant, and T_l is the lattice temperature.

In the special case of $F = 0$, the solution is

$$f_o(E) = A\exp(-E/K_bT) \quad \text{3-746}$$

which is the Maxwell-Boltzmann distribution with a normalizing constant A. The density of states $g(E)$ is calculated by assuming that the energy valleys are spherically symmetric around their minimum energy and have a non-parabolicity in their dispersion relation. The non-parabolicity is modeled using the following dispersion relationship.

$$E(1+\alpha E) = \frac{\hbar^2}{k^2}2m_om^* \quad \text{3-747}$$

where k is the wavevector, and m^* is the effective mass. You can set the non-parabolicity factor using the BTE.CB.NONPAR and BTE.VB.NONPAR parameters on the **MATERIAL** statement for the conduction band and valence band respectively. Defining $\gamma(E) = E\,(1 + \alpha E)$, the density of states per valley can be written as

$$g(E) = D\left(\frac{2m_o m^*}{\hbar^2}\right)^{3/2} \gamma(E)\sqrt{\gamma'(E)}/(2\pi)^2 \quad 3\text{-}748$$

where $\gamma'(E)$ is the derivative of $\gamma(E)$ with respect to energy. The group velocity is

$$u_g(E) = \frac{1}{\hbar}\frac{\partial E}{\partial k} = \frac{1}{\gamma'(E)}\sqrt{\frac{2\gamma(E)}{m_o m^*}} \quad 3\text{-}749$$

The default band structure model of Atlas is 6 non-parabolic conduction bands and a single valence band combining the heavy hole and light hole bands. A more complicated band structure is available for electrons.

Atlas first solves Equation 3-744 for f_0. It then evaluates the first order carrier distribution function by using the formula

$$f_1 = q\tau(E)u_g(E)F\frac{\partial f_o}{\partial E} \quad 3\text{-}750$$

The lifetime $\tau(E)$ is derived from the carrier scattering mechanisms. Scattering mechanisms that are included by default are optical phonon scattering, acoustic phonon scattering, and ionised impurity scattering. This results in the expression

$$\tau^{-1}(E) = c_{ac}g(E) + c_{op}N_{op}g(E + \hbar\omega) + c_{op}N_{op}^{+}g(E - \hbar\omega) + \tau_{ion}^{-1}(E)$$

where τ_{ion}^{-1} is defined in Equation 3-756. The material coefficients c_{ac} and c_{op} are defined as

$$c_{ac} = \frac{2\pi K_b T_l \varepsilon^2}{\hbar\rho u_l^2} \quad 3\text{-}751$$

and

$$c_{op} = \frac{\pi(D_t K)^2\hbar}{(\rho\hbar\omega)}$$

where

- ρ is the mass density of the material.
- u_l is the longitudinal sound velocity of the material.
- T_l is the lattice temperature.
- ε is the acoustic phonon coupling energy.
- D_tK is the optical phonon coupling coefficient.

These parameters are assignable for both electrons and holes in Atlas, so

$$c_{ac}^{elec} = \frac{2\pi K_b T_l(\texttt{BTE.CB.ACOUST})^2}{\hbar\texttt{BTE.DENSITY}(\texttt{BTE.SOUND})^2} \quad 3\text{-}752$$

and

$$c_{ac}^{hole} = \frac{2\pi K_b T_l (\texttt{BTE.VB.ACOUST})^2}{\hbar \texttt{BTE.DENSITY} (\texttt{BTE.SOUND})^2} \quad \text{3-753}$$

with the parameters defined on the **MATERIAL** statement. Similarly

$$c_{op}^{elec} = \frac{\pi (\texttt{BTE.CB.OPCC})^2 \hbar}{\texttt{BTE.DENSITY BTE.CB.OPHW}} \quad \text{3-754}$$

and

$$c_{op}^{hole} = \frac{\pi (\texttt{BTE.VB.OPCC})^2 \hbar}{\texttt{BTE.DENSITY BTE.VB.OPHW}} \quad \text{3-755}$$

where the optical phonon energies in eV are given by BTE.CB.OPHW for the conduction band and BTE.VB.OPHW for the valence band.

The ionized impurity scattering rate is given by

$$\tau_{ion}^{-1}(E) = \frac{\pi}{2}\left(\frac{Zq^2\gamma'(E)}{\gamma(E)\varepsilon_s}\right)^2 N_{dop} u_g(E)\left[\ln\left(1+\frac{4\gamma(E)}{\gamma_b}\right) - \frac{1}{1+\gamma_b/4\gamma(E)}\right] \quad \text{3-756}$$

with

$$\gamma_b = \frac{\hbar^2\beta^2}{2m_o m^*} \quad \text{3-757}$$

where

- N_{dop} is the total dopant concentration.
- β is the inverse Debye length.
- ε_s is the dielectric permittivity.

The Z parameter can be treated as a fitting parameter by setting the BTE.CB.ZION and BTE.VB.ZION parameters for conduction band and valence band respectively on the **MATERIAL** statement.

Note: You can disable the ionized impurity scattering by specifying ^BTE.IONISED on the **MATERIAL** statement.

You can then use this model when the fields internal to the device are sufficiently small for impact ionization to be negligible. The basic energy step in the device is controlled by the optical phonon energy. The number of major energy planes simulated are set by the NEP.BTE parameter from the **MODELS** statement. The maximum energy simulated is therefore (NEP.BTE×BTE.CB.OPHW) eV for electrons and (NEP.BTE×BTE.VB.OPHW) eV for holes. A finer energy grid can be used by setting the NSUB.BTE parameter on the **MODELS** statement. The basic energy step is divided by NSUB.BTE with the (NEP.BTE×NSUB.BTE - 1) energy planes being simulated in total. To enable the model for electrons, use the BTE.PP.EL flag on the **MODELS** statement. To enable it for holes, use the BTE.PP.HO flag on the **MODELS** statement.

The Boltzmann transport equation solver uses the Atlas potential distribution. Therefore, the solver is invoked once the required biasing has been achieved. Set the DEVDEG.GF.E and/or DEVDEG.GF.H parameters on the required **SOLVE** statement to carry out the Boltzmann transport equation solution for electrons and holes respectively.

The solver works in one of two modes. The first is the homogeneous mode. This requires you to set the BTE.HGEN flag and specify the HGENFLD parameter on the **MODELS** statement. This causes the homogeneous equations as described above (Equations 3-744 and 3-750) to be solved at the first **SOLVE** statement requesting a Boltzmann transport equation solution. The field F is set to the value specified by HGENFLD. The TonyPlot files homogeneousedfs.dat for electrons and homogeneoushdfs.dat for holes are saved to the filesystem. Plotting these files shows the distribution functions f_o and f_1 as a function of energy, along with other relevant quantities.

The second mode is a spatially dependent mode. In this case, Equation 3-744 becomes the following for a two-dimensional device.

$$\sum_{r\,=\,x,\,y} \frac{\partial}{\partial r}\left(g(E)\tau(E)u_g^2(E)\frac{\partial f_o}{\partial r}\right) + 3g(E)c_{op}\left[g(E+\hbar\omega)\left(N_{op}^{+}f_o(E+\hbar\omega)-N_{op}f_o(E)\right)-g(E-\hbar\omega)\left(N_{op}^{+}f_o(E)-N_{op}f_o(E-\hbar\omega)\right)\right] = 0 \qquad 3\text{-}758$$

This equation is then solved in semiconductor regions, with suitable values of f_o(E) applied as boundary conditions. The boundary conditions can be either Maxwell-Boltzmann or homogeneous solutions. In the former case, the energy distribution function is $f_o(E) = A\exp(-E/K_bT_l)$, with A being adjusted to give the same carrier density as in the Atlas simulation. In the latter case, which is the default selection, the total field at the node point is evaluated and the homogeneous solution at this field used as the boundary condition, again after normalization to the nodal carrier density. The size of the resulting matrix system is proportional to the product of the number of semiconductor nodes and energy planes, and can be quite large. You can reduce NSUB.BTE to decrease the run time of the simulation with the compromise of a slight loss in accuracy. After solving the Boltzmann transport equation, the only variables written back to Atlas are the "acceleration integrals" as defined in Section 4.1.3 "General Framework Degradation Model". This reflects the fact that the Boltzmann transport equation solution is only meant for degradation studies. It also allows you to seamlessly continue with your Atlas simulation. You can view more information on the solution by specifying BTE.VERBOSE on the **MODELS** statement, which causes a TonyPlot file to be written to the filesystem for each semiconductor node in the device.

As mentioned above, a slightly more sophisticated bandstructure model is available for the silicon conduction band and considers higher valleys. You can enable this by specifying BTE.GEMODEL=1 on the **MATERIAL** statement when solving for electrons. The solutions have been quite well calibrated using the BTE.GEMODEL=0 case, which is the simple degenerate non-parabolic band model. The default parameter values have been calibrated to give the velocity-field curves for electrons and holes in Figures 3-18 and 3-19 respectively. These figures were obtained with a lattice temperature of 300 K and at three different dopant concentrations. The low field electron and hole mobilities obtained with the default parameter values are shown in Figure 3-20 as a function of lattice temperature with a dopant density of 1×10^{16} cm^{-3}.

For both BTE.GEMODEL options, the energy dependence of density of states and other energy dependent functions can be visualized by using the homogeneous mode of the solver, and plotting the resulting file: homogeneousedfs.dat or homogeneoushdfs.dat.

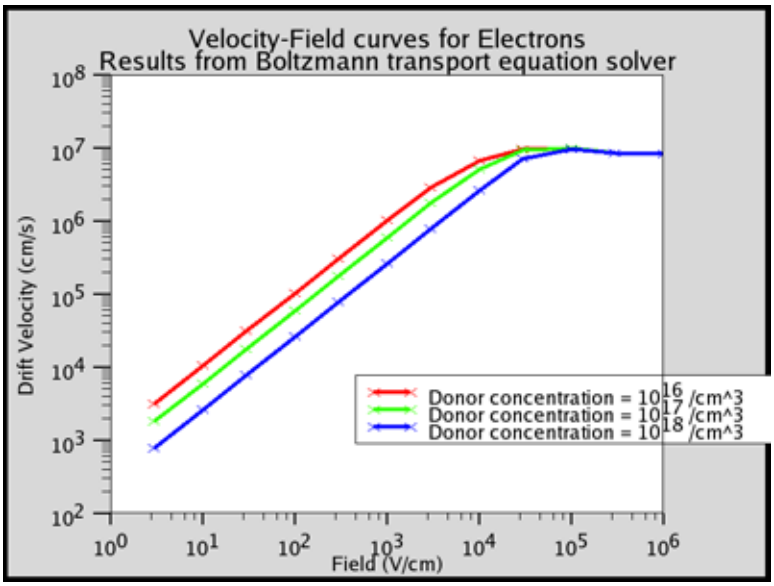


Figure 3-18: Velocity-field curves for electrons from BTE solver at 300K

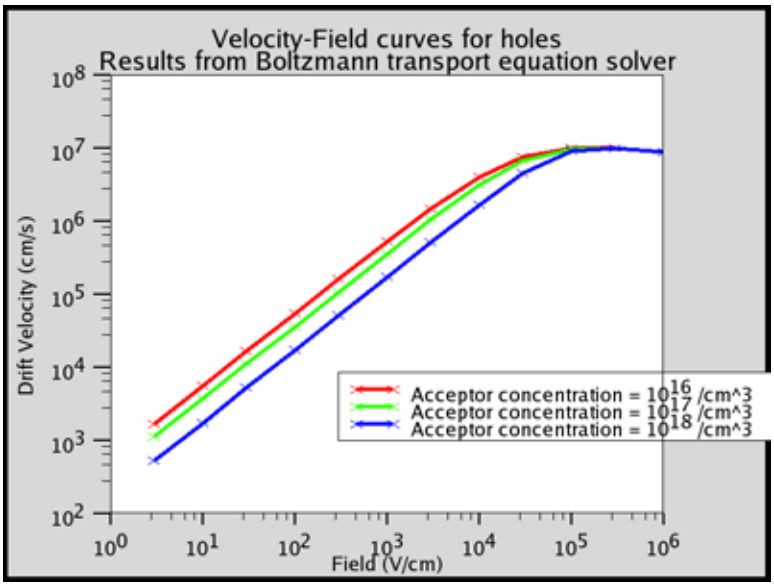


Figure 3-19: Velocity-field curves for holes from BTE solver at 300K

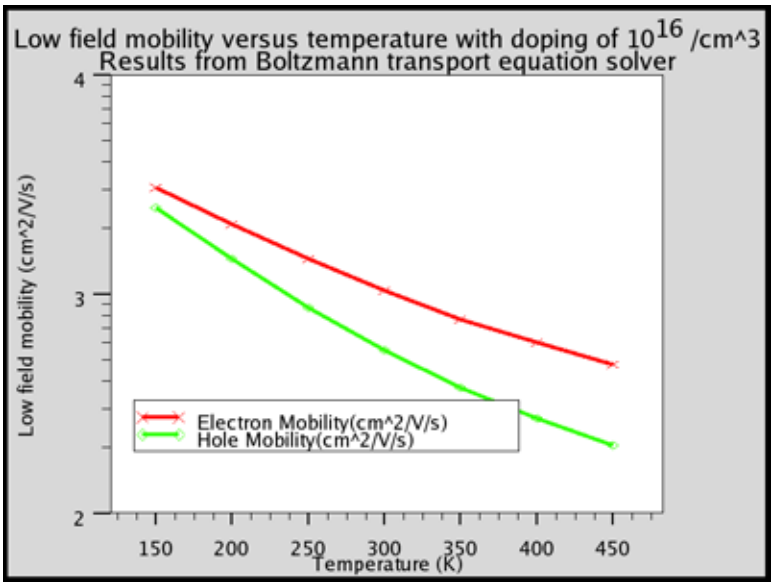


Figure 3-20: Low field electron and hole mobility versus lattice temperature from BTE solver.

The macroscopic quantities are calculated as follows:

$$Concentration = \sum_{i=1}^{D} \int_{0}^{E_{max}} f_o(E) g_i(E) dE \quad \text{3-759}$$

where $g_i(E)$ is the density of states for an individual non-parabolic band and D is the degeneracy factor. D is 6 for the silicon conduction band and 1 for the valence band.

$$Drift\ \ Velocity = \left(\sum_{i=1}^{D} \int_{0}^{E_{max}} f_1 u_g(E) g_i(E) dE \right) / (3 Concentration) \quad \text{3-760}$$

$$CarrierEnergy = \sum_{i=1}^{D} \int_{0}^{E_{max}} E f_o(E) g_i(E) / dE / Concentration \quad \text{3-761}$$

$$Acceleration = \left(\sum_{i=1}^{D} \int_{0}^{E_{max}} f_1 u_g(E) \sigma(E, E_{thresh}) g_i(E) dE \right) / 3 \quad \text{3-762}$$

where $\sigma(E, E_{thresh})$ is the Keldysh factor as given in Section 4.1.3 “General Framework Degradation Model”. The factor 3 in the denomination for drift velocity and acceleration comes from the normalization of the first order Spherical Harmonics expansion.

For large values of electric field, impact ionization becomes important. It affects the carrier distribution functions. The same approach was used as in [331] and involves the addition of the following term to the right hand side of Equations 3-763 and 3-764.

$$\frac{3}{\tau_{ii}} g(E) f_o(E) - 3g(E) \int A(E', E) f_o(E') g(E') dE' \quad \text{3-763}$$

where the outscattering rate is

$$\tau_{ii}^{-1}(E) = \int S_o^{ii}(E, E', E - E' - Eg) g(E') g(E - E' - E_g) dE' \quad \text{3-764}$$

with

$$S_o^{ii}(E, E', E'') = \frac{b_{ii}}{\left[\left(L_{rs}^2 + c\gamma(E) + c\gamma(E') \right)^2 - 4c^2 \gamma(E)\gamma(E') \right]} \quad \text{3-765}$$

In the above, S_o^{ii} is a zeroth-order outscattering term, b_{ii} determines the overall magnitude of the outscattering, L_{rs} is an inverse screening length, and $c = \frac{2m^* m_o}{\hbar^2}$. The quantity E_g can be considered as a fitting parameter.

The term $A(E', E)$ in the inscattering integral

$$\int A(E', E) f_o(E') g(E') dE' \quad \text{3-766}$$

can be reduced to

$$A(E', E) = b_{ii} g(E' - E - E_g) \left(\left[\left(L_{rs}^2 + c\gamma(E) + c\gamma(E') \right)^2 - c^2 \gamma(E)\gamma(E') \right]^{-1} + \left[\left(L_{rs}^2 + c\gamma(E'') + c\gamma(E') \right)^2 - c^2 \gamma(E'')\gamma(E') \right]^{-1} \right) \quad \text{3-767}$$

An iterative process must be used to obtain the solution for f_o. The present estimate for f_o is used to evaluate the integral in Equation 3-763. This is then used to obtain a better estimate for f_o. The process continues until the change in f_o between iterations is less than a certain tolerance.

The calculation of f_1 is also modified by the addition of the term

$$f_1 = q\tau(E) u_g(E) F \frac{\partial f_o}{\partial E} + \frac{\tau_1}{3} \int B(E', E) f_1(E') g(E') dE' \quad \text{3-768}$$

where $B(E', E)$ can be simplified to

$$B(E', E) = g(E' - E - E_g) \frac{3b_{ii}}{8r^2} \left[\frac{4rs}{s^2 - 4r^2} + \log \frac{s - 2r}{s + 2r} \right] \quad \text{3-769}$$

where

$$r = c\sqrt{\gamma(E)\gamma(E')}$$
$$s = L_{rs}^2 + c\gamma(E) + c\gamma(E') \quad \text{3-770}$$

where b_{ii} determines the overall magnitude of the outscattering, L_{rs} is an inverse screening length, and $c = \frac{2m^* m_o}{\hbar^2}$. The solution to this equation is also iterative, and Atlas continues until the solution update is below a certain tolerance. You use the `BTE.CB.II_SCALE` and

BTE.VB.II_SCALE parameters on the **MATERIAL** statement to set b_{ii} for electrons and holes respectively. You use the BTE.CB.INVSL and BTE.VB.INVSL parameters on the **MATERIAL** statement to set L_{rs} for electrons and holes respectively. You use the BTE.CB.II_EG and BTE.VB.II_EG parameters on the **MATERIAL** statement to set E_g for electrons and holes respectively. The impact ionization model is enabled by setting the BTE.IMPACT flag on the **MATERIAL** statement.

In regions of high field, the solution for f_0 may develop oscillations in energy. There are two approaches for mitigating this. The first is to increase the value of the BTE.STABILITY parameter above its default value of 1×10^{-5}. This is on the **METHOD** statement. The second approach is to change the spatial discretisation method by specifying BTE.ALTDIS on the **MODELS** statement.

Table 3-147 Parameters used for the Boltzmann Transport Equation Solver.

Statement	Parameter	Type	Default	Units
MODELS	BTE.ALTDIS	Logical	False	
MODELS	BTE.EQUIBCS	Logical	False	
MODELS	BTE.HGEN	Logical	False	
MODELS	BTE.PP.E	Logical	False	
MODELS	BTE.PP.H	Logical	False	
MODELS	BTE.VERBOSE	Logical	False	
MODELS	HGENFLD	Real	1.0	V/cm
MODELS	NEP.BTE	Real	100	
MODELS	NSUB.BTE	Real	4	
MATERIAL	BTE.AC_PHONON	Logical	True	
MATERIAL	BTE.OP_PHONON	Logical	True	
MATERIAL	BTE.IONIZED	Logical	True	
MATERIAL	BTE.IMPACT	Logical	False	
MATERIAL	BTE.CB.OPHW	Real	0.052	eV
MATERIAL	BTE.VB.OPHW	Real	0.065	eV
MATERIAL	BTE.CB.OPCC	Real	5.5×10^8	eV/cm
MATERIAL	BTE.VB.OPCC	Real	4.45×10^8	eV/cm
MATERIAL	BTE.CB.ACOUST	Real	5.0	eV
MATERIAL	BTE.VB.ACOUST	Real	3.68	eV

Table 3-147 Parameters used for the Boltzmann Transport Equation Solver.				
MATERIAL	BTE.SOUND	Real	9000.0	m/s
MATERIAL	BTE.DENSITY	Real	2329.0	kg/m^3
MATERIAL	BTE.CB.NONPAR	Real	0.35	eV^{-1}
MATERIAL	BTE.VB.NONPAR	Real	0.2	eV^{-1}
MATERIAL	BTE.GEMODEL	Real	0	
MATERIAL	BTE.CB.ZION	Real	0.13	
MATERIAL	BTE.VB.ZION	Real	0.11	
MATERIAL	BTE.CB.INVSL	Real	3.04e7	cm
MATERIAL	BTE.VB.INVSL	Real	5.5e7	cm
MATERIAL	BTE.CB.II_SCALE	Real	5.0	cm^2 eV s^{-1}
MATERIAL	BTE.VB.II_SCALE	Real	5.0	cm^2 eV s^{-1}
MATERIAL	BTE.CB.II_EG	Real	1.2	eV
MATERIAL	BTE.VB.II_EG	Real	1.7	eV
METHOD	BTE.STABILITY	Real	1×10^{-5}	

Chapter 4
Device Reliability

4.1 Operational Reliability

4.1.1 Hansch MOS Reliability Model

The Hansch Reliability Model [115,4,267] can be used to simulate MOS transistor degradation under stress conditions. The causes of device characteristic degradation are the hot electron (hole) injection into gate oxide, and the trapping of electron (hole) charge on the effective interface acceptor (donor) like traps.

The model calculates hot electron (hole) injection current according to the lucky electron model. The arbitrary position-dependent distributions of acceptor and donor-like traps are specified on the oxide-semiconductor interface with corresponding capture cross sections. The device degradation is calculated as a function of stress time by performing transient calculations. The trap rate equation is solved on every time-step and thus the trapped electron (hole) concentration is calculated. The rate of electron trapping can be described by the equations:

$$\frac{dN_n(x,t)}{dt} = \frac{\mathtt{SIGMAE}}{q} \cdot J_{inj,n}(x,t) \cdot (\mathtt{NTA}(x) - N(x,t)) \quad 4\text{-}1$$

$$\frac{dN(x,t)}{dt} = \frac{\mathtt{SIGMAH}}{q} \cdot J_{inj,p}(x,t) \cdot (\mathtt{NTD}(x) - N(x,t)) \quad 4\text{-}2$$

where $N(x,t)$ represents the trapped electron (hole) density, at the interface point x, at time=t during a transient simulation. The NTA and NTD parameters represent the acceptor and donor-like trap densities at time=0. The $J_{inj,n}(x,t)$ and $J_{inj,p}(x,t)$ parameters are the injected electron and hole current densities, SIGMAE and SIGMAH are the capture cross section of electrons and holes.

To activate this model, use the DEVDEG, DEVDEG.E, and DEVDEG.H parameters in the **MODELS** statement (to account for both hot electron and hole injection, hot electron or hot hole injection, respectively). The model parameters are user-definable on the **DEGRADATION** statement.

Table 4-1 User-Definable Parameters for Equations 4-1 and 4-2

Statement	Parameter	Units
DEGRADATION	SIGMAE	cm^2
DEGRADATION	SIGMAH	cm^2
DEGRADATION	NTA/F.NTA	cm^{-2}
DEGRADATION	NTD/F.NTD	cm^{-2}

The results of stress simulation can be used to calculate the characteristics of the degraded device (the shift of the threshold voltage, transconductance degradation, and so on). You can view the distribution of traps, hot electron (hole) current density, and trapped electron (hole) distribution by using TonyPlot.

The model parameters: NTA, NTD, SIGMAE, and SIGMAH can also be defined through the C-Interpreter functions: F.NTA, F.NTD, F.SIGMAE, and F.SIGMAH. This allows you to define these values as functions of their position (x,y) along the insulator-semiconductor interface. These C-function libraries are also defined on the **DEGRADATION** statement. More information on the C-Interpreter functions can be found in Appendix A "C-Interpreter Functions".

4.1.2 Reaction-Diffusion Degradation Model

The problem of the degradation of MOSFET devices with thin oxide layers under bias and temperature stress conditions is generally accepted as being mainly associated with the depassivation of silicon dangling bonds at the Si/SiO_2 interface. These dangling bonds are initially passivated at the end of the fabrication process by heating in an hydrogen or, more rarely, a deuterium environment. The interface trap density is typically reduced by two orders of magnitude by this passivation process, to around $10^{10}cm^{-2}$ or even less [258].

The mechanisms for depassivation under device operating conditions have been the subject of much investigation. It is found that devices passivated with deuterium show much improved degradation properties [53] (and references therein), suggesting that the Si-D bond is much stronger than the Si-H bond. As pointed out by van de Walle and Tuttle [328], the static electronic bonding is the same for Si-H bonds and Si-D bonds. Therefore, the apparent difference in bond strengths must arise from the different dynamic nature of the bonds caused by the larger mass of the deuterium nucleus. It has been suggested that multiple excitation of transverse vibrational states by inelastic carrier scattering can liberate the hydrogen or deuterium atom from the bond. In the case of deuterium, the energy quanta of the excitation are believed to couple strongly to a bulk phonon mode and thus the bending mode excitations have a short relaxation time. In the case of hydrogen, however, the vibrational modes are only weakly coupled to phonon states and a vibrational ladder is climbed until the bond dissociates at a median energy of around 1.5 eV above the ground state [109]. Consequently, charge carriers with relatively low energy may contribute to the depassivation process, and carriers with sufficient energy to be injected into the gate oxide are not primarily responsible for this generation of interface traps [53].

Another suggested reaction for the depassivation of the Si-H bond is that the bond first captures an inversion layer hole that reduces its binding energy [166]. Thereby, increasing the rate of thermal dissocation. The model implemented here allows depassivation by one or a combination of hot channel current, tunneling current, oxide field, or inversion layer hole capture [195]. The mode of stressing determines which is the most significant mechanism. An additional possibility, not considered here, is that a proton and an electron react with the passivated bond to create a trap and an H_2 molecule [234].

The DEVDEG.RD model assumes that atomic, neutral hydrogen is created by the depassivation and that the trap becomes charged to $\pm Q$ *Coulombs* almost instantaneously. This is consistent with the extensive evidence that the dangling bonds, also known as P_b centers, are the origin of the interface charge [248]. The atomic hydrogen can diffuse and dimerize to H_2 molecules.

The interface charge density N_{it} as a function of time is found by integrating the following equation forward in time:

$$\frac{dN_{it}}{dt} = K_f(\mathtt{RD.SIHTOT} - N_{it}) - \mathtt{RD.KR0}\ H\ N_{it} \qquad 4\text{-}3$$

where

- K_f is the forward reaction rate.
- RD.SIHTOT is the total available dangling bond density.
- RD.KR0 is a repassivation rate.
- H is the local density of atomic hydrogen.

This rate is a function of energy and includes contributions to bond-breaking from direct field effects, channel hot carriers, and Fowler-Nordheim tunneling. The channel hot carrier current is not the hot carrier current injected into the oxide, and we model it as the channel current multiplied by the lucky electron expression for the fraction of carriers above the activation energy E. The full expression for the forward reaction rate at a given energy, E, is

$$K_f(E) = \mathtt{RD.KF0}\, exp\left(-\frac{(E + E_{\perp}\mathtt{RD.AESLOPE})}{KT}\right) \times \qquad 4\text{-}4$$

$$\left(1.0 + \mathtt{RD.E.HCCOFF}\, exp\left(-\frac{(E + \mathtt{RD.E.OFFSET})}{(E_{\parallel}\mathtt{IG.ELINF})}\right) J_n\right.$$

$$+ \mathtt{RD.H.HCCOFF}\, exp\left(-\frac{(E + \mathtt{RD.H.OFFSET})}{(E_{\parallel}\mathtt{IG.HLINF})}\right) J_p$$

$$+ \mathtt{RD.E.FNCOEF}\, Jn(FN) + \mathtt{RD \cdot H \cdot FNCOFF}\, J_p(FN)$$

$$\left. + \mathtt{RD.INVHCOFF\ P}\right)$$

The nominal (median) activation energy for the depassivation is given by the parameter RD.AE. $K_f(E)$ can either be evaluated at this discrete energy or integrated over the distribution function

$$G(E) = \frac{1}{2\mathtt{RD \cdot AEVAR}\left(1 + \cosh\left(\frac{E - \mathtt{RD.AE}}{\mathtt{RD \cdot AEVAR}}\right)\right)} \qquad 4\text{-}5$$

where RD.AEVAR is the energy width parameter for the distribution of activation energies. If RD.AEVAR = 0, then the forward reaction rate is evaluated only at RD.AE. In the above expression, $E\perp$ is the field in the oxide perpendicular to the interface and $E\parallel$ is the field along the semiconducting channel. J_n and J_p are the electron and hole channel currents and $J(FN)_n$ and $J(FP)_p$ are the Fowler-Nordheim tunneling currents for electrons and holes respectively. The hole density at the interface is given by P. The overall magnitudes of the different terms can be controlled by setting the values of their prefactor coefficients, as well as other relevant parameters. For example, the parameters IG.ELINF and IG.HLINF are the optical phonon scattering mean free path lengths for electrons and holes respectively, as also used by the HEI and HHI models (Section 3.6.6 “Band-to-Band Tunneling”). The exponential prefactor is given a linear dependence on oxide field as more complicated behavior seems quite contrived [109]. The overall depassivation rate is controlled by the prefactor RD.KF0 and the repassivation rate (by released hydrogen) is controlled by RD.KR0. An initial uniform density of depassivated bonds can be set using the RD.NIT0 parameter. All

of the parameters prefixed with RD are specified on the **DEGRADATION** statement, which must follow the **MODELS** statements and have defaults values as below.

Table 4-2 DEGRADATION Statement

Parameter	Type	Default	Units
RD.SIHTOT	Real	10^{12}	cm^2
RD.NIT0	Real	0.0	cm^2
RD.KF0	Real	10^{-5}	s^{-1}
RD.KR0	Real	3×10^{-9}	cm^3s^{-1}
RD.AE	Real	1.5	eV
RD.AESLOPE	Real	-5.6×10^{-8}	[Q=1]cm
RD.AEVAR	Real	0.1	eV
RD.E.HCCOEF	Real	0.0	cm^2/A
RD.H.HCCOEF	Real	0.0	cm^2/A
RD.E.FNCOEF	Real	0.0	cm^2/A
RD.H.FNCOEF	Real	0.0	cm^2/A
RD.INVHCOEF	Real	0.0	cm^3
RD.E.OFFSET	Real	0.0	eV
RD.H.OFFSET	Real	0.0	eV
RD.COUPLED	Logical	True	

This form of the forward depassivation rate can therefore model the effects of Channel Hot Carrier stress, Fowler-Nordheim stress, and Negative Bias Temperature Instability (NBTI) stress. In this model, each depassivation event releases an atom of hydrogen and creates an amphoteric interface trap. The trap is immediately charged and the released hydrogen does one of three things. It can diffuse away as atomic hydrogen in any direction, it can react with another hydrogen to create molecular hydrogen, or it can be captured by another depassivated bond, thereby repassivating it. The repassivation rate is controlled by the parameter RD.KR0 as seen in Equation 4-3. Atomic Hydrogen (H) and Molecular Hydrogen (H_2) are allowed to react and diffuse according to the equations:

$$\frac{\partial H}{\partial t} = \mathrm{D0.H1}\,exp\left(-\frac{\mathrm{EA.H1}}{KT}\right)\frac{\partial^2 H}{\partial x^2} - \mathrm{H1TOH2RATE}\ H^2 + \mathrm{H2TOH1RATE}\ H_2 \qquad 4\text{-}6$$

$$\frac{\partial H_2}{\partial t} = \mathrm{D0.H2}\,exp\left(-\frac{\mathrm{EA.H2}}{KT}\right)\frac{\partial^2 H_2}{\partial x^2} - 0.5\mathrm{H2TOH1RATE}\ H_2 + 0.5\mathrm{H1TOH2RATE}\ H^2 \qquad 4\text{-}7$$

On the Si/SiO2 interface, Equation 4-6 has an extra hydrogen generation term given by the rate of trap creation and transformed into an volume rather than an area generation rate. The diffusion coefficients and reaction rates can be set on a material by material basis through the **MATERIAL** statement. The material dependence of diffusivity was found to be necessary to explain some observed results of NBTI degradation [121, 165]. The hydrogen diffusivities are given by $D0.H1\,exp\left(-\frac{EA.H1}{KT}\right)$ and $D0.H2\,exp\left(-\frac{EA.H2}{KT}\right)$ for atomic and molecular hydrogen respectively. The reaction rates are H1TOH2RATE H -> H_2 and H2TOH1RATE H_2 -> H, and the surface recombination velocities at the device exteriors are H1SRV and H2SRV respectively.

These Hydrogen parameters are specified on the **MATERIAL** statement and the default values used when material specific ones are absent are given below:

Table 4-3 MATERIAL Statement

Parameter	Type	Default	Units
D0.H1	Real	1.75×10^{-4}	cm^2/s
D0.H2	Real	1.75×10^{-4}	cm^2/s
EA.H1	Real	0.1685	eV
EA.H2	Real	0.1685	eV
H1TOH2RATE	Real	1.0E-2	cm^{-3}/s
H2TOH1RATE	Real	100.0	s^{-1}
H1SRV	Real	0.0	cm/s
H2SRV	Real	0.0	cm/s

and have the following material dependent defaults (same units as in previous table):

Material	D0.H1	D0.H2	EA.H1	EA.H2	H1TOH2RATE	H2TOH1RATE	H1SRV	H2SRV
Silicon [382, 383]	1×10^{-4}	2.0	1.0	0.83	1.4×10^{-3}	95.4	0	0
SiO_2 [316]	8.1×10^{-3}	8.1×10^{-5}	0.2	0.2	1.4×10^{-3}	95.4	0	0
Aluminum [368]	1.75×10^{-4}	1.75×10^{-4}	0.1685	0.1685	1.0e-2	100	0	0
PolySi [130]	1×10^{-4}	9.4×10^{-3}	1.0	0.48	1.4×10^{-3}	95.4	0	0
SiN [122, 369]	1.0	1.0	1.0	1.0	1.4×10^{-3}	95.4	0	0

It is a common assumption in interface depassivation studies to neglect hydrogen diffusion in the Silicon channel [40]. Hydrogen does diffuse in Silicon, however, and so the DEVDEG.RD model includes it [328] with the following caveat. The diffusion of hydrogen in silicon depends strongly on a number of factors, including dopant species, concentration and whether the silicon is crystalline or amorphous. Other charge states H^+, H^- are also present [130],

leading to field dependent transport. The default values for crystalline and polycrystalline silicon should not be regarded as definitive and possible alternatives can be found in [130].

The hydrogen equations are solved in every part of the device. The boundary conditions are applied at all exterior boundaries of the device. The default is to apply dirichlet boundary conditions on a contact (infinite surface recombination velocity) and zero normal gradient boundary conditions elsewhere. This may not be realistic and therefore it is possible to specify a surface recombination velocity for each material. To do this, you use the H1SRV and H2SRV parameters on the **MATERIAL** statement. Then, the normal gradient of hydrogen flux at an exterior surface is given by

$$\frac{\partial H}{\partial x} = \frac{\mathtt{H1SRV}}{\mathtt{D0.H1}} exp\left(-\frac{\mathtt{EA.H1}}{KT}\right) H \qquad 4\text{-}8$$

and

$$\frac{\partial H}{\partial x} = \frac{\mathtt{H2SRV}}{\mathtt{D0.H2}} exp\left(-\frac{\mathtt{EA.H2}}{KT}\right) H_2 \qquad 4\text{-}9$$

for atomic Hydrogen and molecular hydrogen respectively. For an electrode, it is possible to specify these parameters on the **CONTACT** statement.

Table 4-4 CONTACT Statement

Parameter	Type	Default	Units
H1SRV	Real	∞	cm/s
H2SRV	Real	∞	cm/s

where the default of ∞ is equivalent to Dirichlet boundary conditions (H=0, H_2 =0) on the exterior edges of contacts. The values of the hydrogen density are output to a standard structure file. They can also be obtained using the **PROBE** statement. Specify H.ATOM for atomic hydrogen and specify H.MOLE for molecular hydrogen. To enable the model, specify DEVDEG.RD on the **MODELS** statement. To charge created traps with electrons specify DEVDEG.E, and with holes specify DEVDEG.H on the **MODELS** statement. Specify DEVDEG.A for amphoteric traps, where electron current creates negatively charged traps and hole current creates positively charged traps. The default, if none of DEGDEG.H, DEVDEG.A, or DEVDEG.E are specified, is to charge the traps with electrons. The total interface electron and hole charges in units of C/cm as a function of time can be saved to a LOG file by specifying DEVDEG on the **LOG** statement.

Faster run times may be obtained by setting the the flag RD.COUPLED equal to false. This is not recommended when the repassivation coefficient is significant because the solution will be less accurate. It is recommended, however, to use the LTE2STEP flag on the **METHOD** statement when using the DEVDEG.RD model and to use a small value of TOL.TIME when the repassivation rate is significant. In this case, the net repassivation rate is the difference of a large depassivation rate and a large repassivation rate and a small step size is required to accurately solve the equation. The maximum net depassivation rate, $\frac{dN_{it}}{dt}$ in units of $cm^{-2}s^{-1}$, and the position at which it occurs are automatically part of the run time output.

With a large repassivation coefficient, RD.KR0, the net depassivation rate will become small relative to the forward and reverse passivation rates. In this case, a small stepsize is necessary as are higher precision solutions. The optimum parameters for stepsize control should be determined pragmatically. To achieve a small stepsize, you adjust the value of the TOL.TIME parameter on the **METHOD** statement and reduce the value of CX.TOL parameter on the **METHOD** statement. This results in the solution for carrier and hydrogen concentrations being determined more precisely. It also is recommended that you also specify the flag XNORM on the **METHOD** statement. In some cases, it may also be necessary to use a version of Atlas with higher floating point precision.

4.1.3 General Framework Degradation Model

This model considers the creation of interface states when Si-H bonds are broken at the Si/SiO_2 interface [109]. Three different mechanisms are considered:

- field-enhanced thermal degradation
- single-particle (SP) processes
- multi-particle (MP) processes

Simulations of these processes have been carried out and compared to experiment. Starkov et al [299] considered the SP/MP processes and Reggiani et al [260] considered all three. The field-enhanced thermal degradation process can also be studied by using the Reaction-Diffusion model described above with a suitable choice of parameters. First, we describe the single-particle process.

The SP process is the breaking of the Si-H bond via a single hot electron or hole. The reaction rate for this process at position $\boldsymbol{r}$ is given by

$$K_f^{e,h}(SP)(\boldsymbol{r}) = \int_{E_{sp}}^{\infty} f(E,\boldsymbol{r})g(E)u_g(E)\sigma_{sp}^{e,h}(E,E_{sp})dE \qquad 4\text{-}10$$

where $f(E,\boldsymbol{r})$ is the anti-symmetric part of the carrier distribution function, $g(E)$ is the density of states, and u_g is the group velocity.

For electrons, the function $\sigma_{sp}^{e}(E,E_{sp})$ is defined for $E \geq$ ELEC.SP.THRESH, where E_{sp} = ELEC.SP.THRESH as

$$\sigma_{sp}^{e}(E,E_{sp}) = \text{ELEC.SP.SIGMA}\left(\frac{E-\text{ELEC.SP.THRESH}}{K_bT}\right)^{\text{ELEC.SP.POWE}} \qquad 4\text{-}11$$

where the Boltzmann energy K_bT acts as an energy scale. This is known as a soft-threshold, as introduced by Keldysh in the context of impact ionization rate calculations. Therefore, only electrons with an energy of more than ELEC.SP.THRESH contribute to this integral. Similarly, the function is defined for holes as

$$\sigma_{sp}^{h}(E,E_{sp}) = \text{HOLE.SP.SIGMA}\left(\frac{E-\text{HOLE.SP.THRESH}}{K_bT}\right)^{\text{HOLE.SP.POWE}} \qquad 4\text{-}12$$

Equation 4-10 is often referred to as an acceleration integral although its units are s^{-1}. The default value of the parameters are given in Table 4-5.

Equation 4-10 requires the calculation of the carrier distribution function. This is obtained using the Spherical Harmonic solution method developed by the Bologna school [331, 101], which is described in detail in Section 3.17 "The Boltzmann Transport Equation Solver".

The device is biased into the stressing state using the drift-diffusion model. Then, the carrier distribution function is obtained by solving the Boltzmann Transport Equation (BTE) in the semiconductor with suitable boundary conditions. You specify BTE.PP.E on the **MODELS** statement and DEVDEG.GF.E on the **SOLVE** statement where you want the BTE solution for electrons. For holes, you specify BTE.PP.H on the **MODELS** statement and DEVDEG.GF.H on the **SOLVE** statement. You also specify stressing times using the TD1, TD2 ... TD10 parameters on the same **SOLVE** statement and the name of a structure file stem. The degradation charge is calculated at the specified stressing times, TDn, according to the formulae

$$N(\boldsymbol{r}) = \texttt{NTA.SP}(1.0 - \exp(-TDnK_f^e(SP)(\boldsymbol{r}))) \quad \text{4-13}$$

$$P(\boldsymbol{r}) = \texttt{NTD.SP}(1.0 - \exp(-TDnK_f^h(SP)(\boldsymbol{r}))) \quad \text{4-14}$$

For electrons, $N(\boldsymbol{r})$ is negative interface charge density. For holes, $P(\boldsymbol{r})$ is positive interface charge density. NTA.SP and NTD.SP are specified on the **DEGRADATION** statement.

The multi-particle (MP) process is more complicated and requires information about the Si-H bond. The bond is broken by a gradual ascent of quantized vibrational states, followed by thermal emission of the hydrogen from the highest bonded state to the transport state. This thermal emission occurs over a barrier of energy difference, GF.BARREMI eV, with an attempt frequency of GF.NUEMI Hz, giving an emission rate of

$$P_{emi} = \texttt{GF.NUEMI}\exp(-\texttt{GF.BARREMI}/K_bT) \quad \text{4-15}$$

where T is the lattice temperature. There is also the reverse process for repassivation of the bond, where the hydrogen overcomes a barrier of height GF.BARRPASS to become bonded again. The overall repassivation rate is

$$P_{emi} = \texttt{GF.NUPASS}\exp(-\texttt{GF.BARRPASS}/K_bT) \quad \text{4-16}$$

and it is expected that GF.BARREMI≥GF.BARRPASS. These parameters are specified on the **DEGRADATION** statement.

The kinetics of the bond gradually gaining (quantized) vibrational energy through collisions with warm electrons or holes is described by a series of coupled differential equations. Generally, the number of equations is quite large. Therefore, some assumptions are made to give an overall formula

$$N(x) = \texttt{NTA.MP}\left[\frac{P_{emi}}{P_{pass}}\left(\frac{P_u}{P_d}\right)^{Nl}(1.0 - \exp(-TDnP_{emi}))\right]^{\frac{1}{2}} \quad \text{4-17}$$

$$P(x) = \texttt{NTD.MP}\left[\frac{P_{emi}}{P_{pass}}\left(\frac{P_u}{P_d}\right)^{Nl}(1.0 - \exp(-TDnP_{emi}))\right]^{\frac{1}{2}} \quad \text{4-18}$$

where P_u and P_d are phonon excitation and decay rates respectively. These are modeled by the following equations:

$$P_u = \texttt{GF.NUPHONON}\exp(-\texttt{GF.HBAROMEGA}/K_bT) + K_f^{e,h}(MP)(\boldsymbol{r}) \quad 4\text{-}19$$

$$P_d = \texttt{GF.NUPHONON} + K_f^{e,h}(MP)(\boldsymbol{r}) \quad 4\text{-}20$$

where

$$K_f^{e,h}(MP)(\boldsymbol{r}) = \int_{E_{mp}}^{\infty} f(E,\boldsymbol{r})g(E)u_g(E)\sigma_{mp}^{e,h}(E,E_{mp})dE \quad 4\text{-}21$$

is the acceleration integral analogous to the SP expression. The Keldysh term is

$$\sigma_{mp}^{e}(E,E_{mp}) = \texttt{ELEC.MP.SIGMA}\left(\frac{E-\texttt{ELEC.MP.THRESH}}{K_bT}\right)^{\texttt{ELEC.MP.POWEI}} \quad 4\text{-}22$$

for electrons. And for holes, it is

$$\sigma_{mp}^{h}(E,E_{mp}) = \texttt{HOLE.MP.SIGMA}\left(\frac{E-\texttt{HOLE.MP.THRESH}}{K_bT}\right)^{\texttt{HOLE.MP.POWEI}} \quad 4\text{-}23$$

The threshold energies for MP processes are usually lower than threshold energies for the SP processes. This reflects the lower incident carrier energies required for the MP bond breaking processes. The ratio $\frac{P_u}{P_d}$ is raised to the power Nl, where Nl is the number of vibrational levels in the SiH bond. The bond energy itself is modeled as

$$E_b = \texttt{GF.EB} + \texttt{GF.DEBDF}F_{perp} \quad 4\text{-}24$$

where F_{perp} is the field perpendicular to the interface in the oxide. This is divided by GF.HBAROMEGA in order to obtain Nl. Thus, Nl will generally decrease with increasing gate bias.

It is clear from Equations 4-17 and 4-18 that the time evolution of the degradation charge is controlled by the rate P_{emi}, and that the saturated level is controlled by acceleration integral via the ratio $\left(\frac{P_u}{P_d}\right)^{Nl}$. In the absence of current, then this is $\exp(-E_b/K_bT)$. But in the current dominated case, it will be approximately unity.

The third component of the general framework model is a field enhanced thermal degradation. The rate of which is modeled as

$$P_{therm} = \texttt{GF.KTHRM}\exp(-E_b/K_bT) \quad 4\text{-}25$$

This term is only important at high temperature or high oxide field conditions. It has the same time dependence as the SP process and is simply added to $K_f^{e,h}(SP)(\boldsymbol{r})$ in the calculation of defects after stressing time *t*. The GF.KTHRM parameter is set to zero by default and should be obtained from fitting to experiment.

The way of using the General framework model is different to that of the other degradation models. With these models, a fully transient simulation is carried out. But with the General framework model, the time evolution is explicitly modeled by the analytical formulae above (see Equations 4-13, 4-14, 4-17, and 4-18), whereas a fully transient simulation is carried out in other models. You must first set the relevant Boltzmann Transport equation parameters on

the **MODELS** statement. Then, bias the device to the required stressing bias using the drift-diffusion approximation.

Next, specify DEVDEG.GF.E and/or DEVDEG.GF.H on the **SOLVE** statement for electron and/or hole based degradation, which will invoke the inhomogeneous Boltzmann solver for electrons and/or holes at that bias.

You must also specify an outfput file stem using the OUTFILE parameter and up to 10 different stressing times using the TD1, TD2 ... TD10 parameters. A structure file is output for each requested stressing time, where you can view the acceleration integral values and the degradation charges. Atlas will only solve the BTE on a **SOLVE** statement with DEVDEG.GF.E and/or DEVDEG.GF.H.

A Standard Structure File is written for each stressing time and will contain the calculated interface charge. For example

```
MODELS BTE.PP.E BTE.PP.H

.

.

.

.

SOLVE PREV DEVDEG.GF.H DEVDEG.GF.E TD1=1.0E-2 TD2=1.0E-1 TD3=1.0
TD4=10.0 OUTF=MOSFET.STR MASTER
```

will solve the BTE equation for both electrons and holes, calculate the interface degradation charge after 0.01,0.1,1.0, and 10.0 seconds of stressing, and output structure files MOSFET_1.00e-02s.STR, MOSFET_1.00e-01s.STR, MOSFET_1.00e+00s.STR, and MOSFET_1.00e+01s.STR. These files can then be used to study the shifts in threshold voltage and change in drain current caused by the degradation.

Parameters relevant to this model are shown in Table 4-5. They can be adjusted to fit to different experimental results. The complete parameter set for the Boltzmann transport equation is given in Section 3.17 “The Boltzmann Transport Equation Solver”.

Table 4-5 General Framework Degradation Model Statements

Statement	Parameter	Type	Default	Units
DEGRADATION	NTA.SP	Real	0.0	cm^{-2}
DEGRADATION	NTA.MP	Real	0.0	cm^{-2}
DEGRADATION	NTD.SP	Real	0.0	cm^{-2}
DEGRADATION	NTD.MP	Real	0.0	cm^{-2}
DEGRADATION	GF.NUEMI	Real	1×10^{12}	Hz
DEGRADATION	GF.NUPASS	Real	1×10^{12}	Hz
DEGRADATION	GF.BARREMI	Real	0.8	eV
DEGRADATION	GF.BARRPASS	Real	0.8	eV

Table 4-5 General Framework Degradation Model Statements				
Statement	**Parameter**	**Type**	**Default**	**Units**
DEGRADATION	GF.NUPHONON	Real	1×10^{11}	Hz
DEGRADATION	GF.HBAROMEGA	Real	0.075	eV
DEGRADATION	GF.EB	Real	1.5	eV
DEGRADATION	GF.DEBDF	Real	-5.6×10^{-8}	cm
DEGRADATION	GF.KTHRM	Real	0.0	Hz
MODELS	BTE.PP.E	Logical	False	
MODELS	BTE.PP.H	Logical	False	
MODELS	NEP.BTE	Real	100	
MODELS	NSUB.BTE	Real	4	
MATERIAL	ELEC.SP.SIGMA	Real	10^{-16}	cm^2
MATERIAL	ELEC.SP.THRESH	Real	1.75	eV
MATERIAL	ELEC.SP.POWER	Integer	2	
MATERIAL	ELEC.MP.SIGMA	Real	10^{-14}	cm^2
MATERIAL	ELEC.MP.THRESH	Real	1.0	eV
MATERIAL	ELEC.MP.POWER	Integer	3	
MATERIAL	HOLE.SP.SIGMA	Real	10^{-18}	cm^2
MATERIAL	HOLE.SP.THRESH	Real	1.75	eV
MATERIAL	HOLE.SP.POWER	Integer	3	
MATERIAL	HOLE.MP.SIGMA	Real	10^{-16}	cm^2
MATERIAL	HOLE.MP.THRESH	Real	1.0	eV
MATERIAL	HOLE.MP.POWER	Integer	3	
SOLVE	DEVDEG.GF.E	Logical	False	
SOLVE	DEVDEG.GF.H	Logical	False	
SOLVE	TD1 ... TD10	Real	0	s

4.1.4 Two Stage Negative Bias Temperature Instability Models

Negative Bias Temperature Instability (NBTI) is a type of degradation which occurs in pMOSFETs that are electrically stressed at large negative gate bias and elevated temperature. It results in positive charge being trapped and the threshold voltage shifted to a more negative gate bias. It can limit CMOS reliability and cause significant statistical variability, which is a concern in both analog and digital circuits.

Approaches adopted to try to model NBTI include tunneling of holes to states in the gate insulator and the creation of interface states by depassivation of dangling bonds. Atlas can model the tunneling of holes to states within a specified distance from the interface using the Heiman model [125]. By itself, this model cannot properly model the temperature dependence and other properties of NBTI degradation. For this reason, an enhanced hole trapping model, based on [136], has been included in Atlas. This takes into account structural relaxation of the hole traps and is described in Section 3.3.3 “Traps and Defects”, along with the Heiman model. The reaction-diffusion (RD) model considers that the NBTI degradation mechanism is the release of hydrogen from hydrogen passivated dangling bonds at the interface between the silicon channel and the insulating gate material. The interface states thus created can behave as amphoteric traps. This model has been given much consideration in the literature. It has been implemented in Atlas and is described in Section 4.1 “Operational Reliability”.

Recent studies suggest that neither hole tunneling models nor reaction-diffusion models can adequately model all the experimentally observed features of NBTI degradation [107]. This Two-stage model postulates the existence of near-interfacial oxygen vacancy states. These are located at energy levels approximately 1 eV below the silicon valence band and holes can tunnel into them via a multiphonon field assisted inelastic tunneling process (MPFAT), which is a high field extension of a thermally activated multiphonon emission model (MPE). When occupied by a hole these oxygen vacancies, or precursor states, undergo a structural transition and become positively charged with the trap energy lying within the silicon bandgap. These switching traps can exchange carriers with the channel in order to become neutral and also to switch back to being positively charged. When they are in a neutral state, they can undergo a reverse structural transition (a relaxation) to return to the uncharged precursor state. It is thought that this relaxation explains the recoverable aspect of NBTI. These transitions form the basic 3-state model. Further transitions of the switching trap are possible and these are described shortly.

Basic 3-State Model

The basic model requires a trap with 3 possible internal states. In Atlas, the multistate trap model described in Section 3.3.4 “Multistate Traps” is capable of this and so the Two-stage model is based upon the multistate trap model, but with the rates built into Atlas, rather than requiring a C-Interpreter function. You specify all necessary flags and parameters on the **INTTRAP** statement. The precursor state is labeled state 1, the positively charged switching trap is labeled state 2, and the neutral switching trap is labeled state 3. Using the notation for transition rates introduced in Figure 3-3, we can write them as

$$'_{12} = pv_{th}^{p}\texttt{GRA.SIGP}e^{(-\texttt{GRA.EB.HOLE}/K_BT)}e^{(F^2/\texttt{GRA.FC}^2)}S_p^{12} + nv_{th}^{n}\texttt{GRA.SIGN}e^{(-\texttt{GRA.EB.ELEC}/K_BT)}S_n^{1} \quad \text{4-26}$$

where n and p are the electron and hole concentrations in the channel, v_{th}^{n} and v_{th}^{p} are their respective thermal velocities, and GRA.SIGN and GRA.SIGP are the trap capture cross-

sections as specified on the **INTTRAP** statement. The MPFAT thermal activation energies are GRA.EB.ELEC for electron emission and GRA.EB.HOLE for hole capture respectively. The MPFAT field enhancement scaling factor is GRA.FC, which only applies to the hole capture at present. The field normal to the interface is denoted by *F*, and the rate a_{12} will increase rapidly with increasing field. The terms S_n^{12} and S_p^{12} are statistical factors and depend on the energy level of state 1 of the trap, GRA.ET1. GRA.ET1 is referenced relative to the valence band of the silicon channel. It will typically be below it and thus have a negative value. In this case, S_p^{12} is a Boltzmann factor for the energy difference between trap level and valence band. Otherwise, it is assumed that S_p^{12} is 1.

$$S_p^{12} = \begin{cases} e^{(\text{GRA.ET1}/K_BT)} & \text{if GRA.ET1} < 0 \\ 1 & \text{if GRA.ET1} \geq 0 \end{cases} \quad \text{4-27}$$

The statistical factor for electron emission is more complicated and depends on the sign of the trap level relative to the silicon conduction band, GRA.ET1-E_g, where E_g is the Silicon band gap. In the unlikely situation that the precursor energy is above the conduction band, S_n^{12} is the Boltzmann factor for the energy difference between the conduction band edge and the electron quasi-Fermi level. Otherwise, S_n^{12} is the Boltzmann factor for the energy difference between the trap energy level and the electron quasi-Fermi level.

$$S_n^{12} = \begin{cases} e^{((\text{GRA.ET1}+E_v-E_{qfn})/K_BT)} & \text{if GRA.ET1-}E_g < 0 \\ e^{((E_c-E_{qfn})/K_BT)} & \text{if GRA.ET1-}E_g \geq 0 \end{cases} \quad \text{4-28}$$

where E_c is the conduction band energy and E_v is the valence band energy. The positively charged switching trap, which we label as state 2, can transition to the neutral switching trap, state 3, via hole emission or electron capture. The rate is given by

$$r_{23} = pv_{th}^p \text{GRA.SIGP} e^{(-\text{GRA.EC.HOLE}/K_BT)} S_p^{23} + nv_{th}^n \text{GRA.SIGN} e^{(-\text{GRA.EC.ELEC}/K_BT)} S_n^{23} \quad \text{4-29}$$

and the energy level of the switching trap states is GRA.ET2, relative to silicon valence band edge, and the statistical factors are

$$S_p^{23} = \begin{cases} e^{((E_{qfp}-E_v)/K_BT)} & \text{if GRA.ET2} < 0 \\ e^{-((\text{GRA.ET2}+E_v-E_{qfp})/K_BT)} & \text{if GRA.ET2} \geq 0 \end{cases} \quad \text{4-30}$$

and

$$S_n^{23} = \begin{cases} 1 & \text{if GRA.ET2-}E_g < 0 \\ e^{((E_g-\text{GRA.ET2})/K_BT)} & \text{if GRA.ET2-}E_g \geq 0 \end{cases} \quad \text{4-31}$$

The neutral switching trap can become positively charged again by the processes of hole capture or electron emission. The rate is

$$r_{32} = pv_{th}^{p}\mathtt{GRA.SIGP}e^{(-\mathtt{GRA.EC.HOLE}/K_BT)}S_p^{32} + nv_{th}^{n}\mathtt{GRA.SIGN}e^{(-\mathtt{GRA.EC.ELEC}/K_BT)}S_n^{32} \quad 4\text{-}32$$

and the statistical factors are

$$S_p^{32} = \begin{cases} e^{(\mathtt{GRA.ET2}/K_BT)} & \text{if } \mathtt{GRA.ET2} < 0 \\ 1 & \text{if } \mathtt{GRA.ET2} \geq 0 \end{cases} \quad 4\text{-}33$$

for hole capture and

$$S_n^{32} = \begin{cases} e^{((\mathtt{GRA.ET2}+E_v-E_{qfn})/K_BT)} & \text{if } \mathtt{GRA.ET2}-E_g < 0 \\ e^{((E_c-E_{qfn})/K_BT)} & \text{if } \mathtt{GRA.ET2}-E_g \geq 0 \end{cases} \quad 4\text{-}34$$

for electron emission. The neutral state of the switching trap can revert back to the precursor state. The rate for this transition is

$$a_{31} = \mathtt{GRA.NU}e^{-(\mathtt{GRA.EA}/(K_BT))} \quad 4\text{-}35$$

where GRA.NU is an attempt frequency and GRA.EA a thermal activation energy. The trap emission and capture rates are assumed to be independent of their depth in the interface.

To enable the 3-state model, you specify either GRA.3.DET or GRA.3.STO on the **INTTRAP** statement. If you specify GRA.3.DET, then the parameters used for the capture rates are precisely as specified above. If you specify GRA.3.STO, then the parameters used for the capture rate are obtained from a uniform statistical distribution, in the range

$$\mathtt{GRA.X}-\mathtt{GRA.X.SD} < X < \mathtt{GRA.X}+\mathtt{GRA.X.SD} \quad 4\text{-}36$$

where X refers to any of the model parameters and the GRA.X.SD parameter gives the range about the average value of GRA.X. For example, the precursor energy would be GRA.ET1 in the deterministic model and chosen from the range from GRA.ET1–GRA.ET1.SD to GRA.ET1+GRA.ET1.SD in the stochastic model. The full range of deterministic parameters and their related variance parameters are given in Table 4-6. When using the GRA.3.STO model, the multistate trap associated with each interface point can be divided into GRA.SAMPLES subtraps. The trap density is divided equally among the subtraps, and each subtrap will generally have different rate parameters from all other subtraps. The default value of GRA.SAMPLES is 10 and a practical maximum is 1000. GRA.SAMPLES has no effect for the GRA.3.DET model.

The model is automatically initialized so that all multistate traps are in state 1. You set up a transient **SOLVE** statement and Atlas solves the equation system to give the time evolution of the occupancy of all the states. The changes in interface state charge and the transient recombination rates are included self-consistently in the simulation. The occupation fraction f_t of each trap can be saved in a log file by using the **PROBE** statement with FT.MSC and MSC.STATE parameters as described in Section 3.3.4 “Multistate Traps”. In the case of the GRA.3.STO model, a value averaged over the subtraps is logged. The occupation probability of each multistate trap can be written to a Standard Structure File by using the TRAPS flag on the **OUTPUT** statement followed by a **SAVE** statement. This is also described in Section 3.3.4 “Multistate Traps”. In the case of the GRA.3.STO model, the occupation fractions of each

subtrap are output (up to a maximum of 1000 per point). This then allows the simulation to be reinitialized with existing data, making it easier to separate the stress and recovery phases of a simulation.

4-State Model

The positively charged switching trap described in the previous section can be modeled as being part of a complex that also contains an unpassivated silicon dangling bond. Passivated silicon dangling bonds will typically already exist at the interface. It is claimed in [107] that it is energetically favorable for the hydrogen atoms associated with the passivated bonds to migrate into the insulator and passivate the dangling bonds created by the negative bias stress. The residual dangling bonds on the interface thus form a P_b center, which will be positively charged for most NBTI scenarios. In the 4-state model, this extra state couples to state 2 of the multistate trap model. The transition rates are as follows

$$a_{24} = \texttt{GRA.NU} e^{-(\texttt{GRA.ED} - \texttt{GRA.ET2} - \texttt{GRA.GAMMA} F)/K_B T} \quad \text{4-37}$$

$$a_{42} = \texttt{GRA.NU} e^{-(\texttt{GRA.ED} - \texttt{GRA.ET4} + \texttt{GRA.GAMMA} F)/K_B T} \quad \text{4-38}$$

where F is the field normal to the interface. If the rate a_{24} is much greater than a_{42} under recovery bias conditions, then state 4 will correspond to a non-recoverable part of the NBTI degradation.

To enable this model, specify either GRA.4.DET or GRA.4.STO on the **INTTRAP** statement. In addition to the parameters pertaining to the 3-state model, you optionally specify values for GRA.ED, GRA.GAMMA, and GRA.ET4. For the stochastic model, you can specify GRA.ED.SD, GRA.ET4.SD, and GRA.GAMMA.SD. The parameter values are obtained from a uniform distribution, as described above, except in the case of GRA.ED. For that energy, the stochastic values are chosen from a gaussian probability distribution with GRA.ED as the mean and GRA.ED.SD as the standard deviation. In addition the energy, GRA.ET4, of state 4 is restricted to be in the lower half of the silicon bandgap. This is because the P_b center is thought to be amphoteric and becomes negatively charged if it is in the upper half of the silcon bandgap. This possibility is not modeled here, although neutralization of positively charged P_b centers is considered in the 5-state model. The trap state occupation probabilities can be visualized in the same way as for the 3-state model.

5-State Model

In the 4-state model, it is assumed that state 4 is always positively charged, an assumption which will only be true if the Fermi-level in the channel is below the trap state. The trap quickly responds to changes in the position of the Fermi-level and can become neutral by electron capture and hole emission. If the Fermi-level decreases again it can capture a hole or emit an electron to regain its positive charge. This is modeled by adding a 5^{th} state to the multistate traps, which is a neutral P_b center. The transition rates are

$$a_{45} = n\texttt{GRA.SIGN} v_{th}^n + n_i e^{((E_i - E_t)K_B T)} \texttt{GRA.SIGP} v_{th}^p \quad \text{4-39}$$

$$a_{54} = p\texttt{GRA.SIGP} v_{th}^p + n_i e^{((E_i - E_t)K_B T)} \texttt{GRA.SIGN} v_{th}^n \quad \text{4-40}$$

where E_i is the intrinsic energy and E_t is the trap energy GRA.ET4 + E_v. The intrinsic concentration is given by n_i. To enable this model, specify either GRA.5.DET for the deterministic model or GRA.5.STO for the stochastic model. It requires no extra parameters

compared to the 4-state model. The trap state occupation probabilities can be visualized in the same way as for the 4-state model.

The parameters and flags for all these models are given in Table 4-6. The default values of these are given, but you should use your own suitable values for best results. A schematic diagram of the states and transitions utilized in this model are shown in Figure 4-1. The parameters for the **PROBE** statement that are useful for this model are given in Table 3-19 in Section 3.3.4 “Multistate Traps”.

Table 4-6 Parameters for 2-Stage NBTI Degradation Multistate Trap Model

Statement	Parameter	Type	Default	Units
INTTRAP	GRA.3.DET	Logical	False	
INTTRAP	GRA.3.STO	Logical	False	
INTTRAP	GRA.4.DET	Logical	False	
INTTRAP	GRA.4.STO	Logical	False	
INTTRAP	GRA.5.DET	Logical	False	
INTTRAP	GRA.5.STO	Logical	False	
INTTRAP	GRA.EA	Real	1.0	eV
INTTRAP	GRA.EA.SD	Real	0.025	eV
INTTRAP	GRA.EB.SD	Real	0.025	eV
INTTRAP	GRA.EB.ELEC	Real	0.054	eV
INTTRAP	GRA.EB.HOLE	Real	0.054	eV
INTTRAP	GRA.EC.ELEC	Real	0.02	eV
INTTRAP	GRA.EC.HOLE	Real	0.02	eV
INTTRAP	GRA.EC.SD	Real	0.025	eV
INTTRAP	GRA.ED	Real	0.1	eV
INTTRAP	GRA.ED.SD	Real	0.1, GRA.ED	eV
INTTRAP	GRA.ET1	Real	-0.6	eV
INTTRAP	GRA.ET2	Real	0.2	eV
INTTRAP	GRA.ET4	Real	0.25	eV
INTTRAP	GRA.ET1.SD	Real	0.025	eV
INTTRAP	GRA.ET2.SD	Real	0.025	eV
INTTRAP	GRA.ET4.SD	Real	0.025	eV
INTTRAP	GRA.FC	Real	1.0×10^{7}	V/cm

INTTRAP	GRA.FC.SD	Real	0.1, GRA.FC	V/cm
INTTRAP	GRA.GAMMA	Real	0.0	[Q] cm
INTTRAP	GRA.GAMMA.SD	Real	0.0	[Q] cm
INTTRAP	GRA.NU	Real	1.0×10^{13}	Hz
INTTRAP	GRA.NU.SD	Real	0.1, GRA.NU	Hz
INTTRAP	GRA.SAMPLES	Integer	10	
INTTRAP	GRA.SIGN	Real	3.0×10^{-14}	cm2
INTTRAP	GRA.SIGN.SD	Real	0.0	eV
INTTRAP	GRA.SIGP	Real	3.0×10^{-14}	cm2
INTTRAP	GRA.SIGP.SD	Real	0.0	eV

Note: GRA.ET1, GRA.ET2, and GRA.ET4 are automatically referenced to valence band energy, with a positive value meaning that the energy level is above the valence band.

Note: GRA.SAMPLES must be limited to 1000 if you wish to save and reinitialize a simulation from a standard structure file.

It is claimed in [107] that the 2-stage model is applicable to a range of technologies and not only Si/SiO^2 systems.

Examples

```
INTTRAP GRA.4.DET DENSITY=3.0e12 GRA.SIGN=1.23e-15
        GRA.SIGP=1.1e-13 X.MIN=0.65 X.MAX=0.85
        GRA.EA=0.77 GRA.NU=1.47e13  GRA.EB.ELEC=0.015
        GRA.EB.HOLE=0.016
        GRA.EC.ELEC=0.056 GRA.EC.HOLE=0.057 GRA.ET1=-1.0
        GRA.ET2=0.19 GRA.ET4=0.1 GRA.GAMMA=1.0e-8 GRA.ED=0.112
        GRA.FC=1.1e7
```

This sets up a 4-state deterministic model with an areal density of $3.0\times10^{12}cm^{-2}$, located on the interface between x=0.65 and x=0.85 microns.

```
INTTRAP GRA.3.STO GRA.SAMPLES=20 DENSITY=1.0e12 GRA.SIGN=1.0e-14
        GRA.SIGP=1.1e-13 X.MIN=0.0 X.MAX=0.6 GRA.EA=0.6
        GRA.NU=1.0e13  GRA.EB.ELEC=0.015 GRA.EB.HOLE=0.015
        GRA.EC.ELEC=0.056 GRA.EC.HOLE=0.056 GRA.ET1=-1.0
        GRA.ET2=0.19 GRA.FC=1.1e7 GRA.SIGN.SD=1.0e-15
        GRA.SIGP.SD=1.0e-14 GRA.ET1.SD=0.04 GRA.ET2.SD=0.02
```

```
GRA.EB.SD=0.0 GRA.NU.SD=2.0e12 GRA.EC.SD=0.0
GRA.EA.SD=0.001 GRA.FC.SD=1.0e6
```

This sets up a 3-state stochastic model with an areal density of $1.0\times10^{12}cm^{-2}$, located on the interface between x=0.0 and x=0.6 microns. There are 20 traps simulated per interface point, each with the relevant parameters chosen from a probability distribution.

```
PROBE   FT.MSC   MSC.STATE=1 [Position params]
PROBE   FT.MSC   MSC.STATE=2 [Position params]
PROBE   FT.MSC   MSC.STATE=3 [Position params]
PROBE   FT.MSC   MSC.STATE=4 [Position params]
PROBE   FT.MSC   MSC.STATE=5 [Position params]
```

This puts the value of occupation probabilities of each of 5 states at the specified interface location into a log file. If a stochastic version of the model has been chosen, the values returned will be sample averaged values.

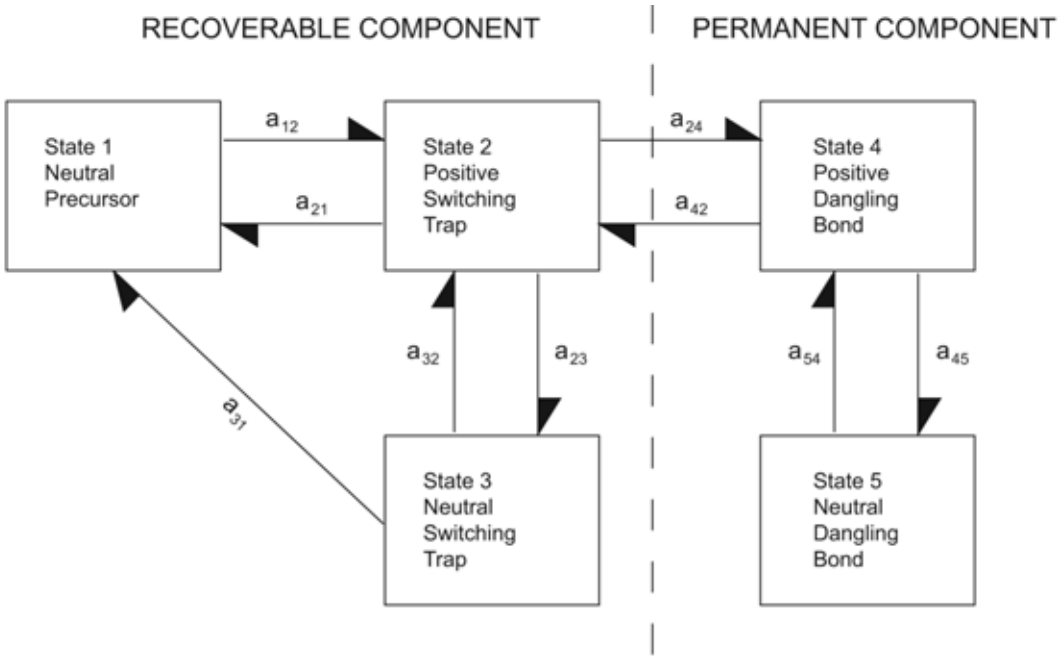


Figure 4-1: Schematic of transitions and states in 5-state, two-stage NBTI degradation model.

4.1.5 Charge Trapping Model for Bias Temperature Instability

The charge trapping model has been developed to explain the degradation in pMOSFETs upon application of large negative gate biases and at elevated temperatures. This degradation phenomenon is referred to as Negative Bias Temperature Instability (NBTI) and poses a serious reliability issue. It can strongly alter the device characteristics and can even result in device failure in the worst case. In the past, NBTI has been associated with various different physical mechanisms (see previous sections). Later studies have shown that the degradation involves a notable amount of hole trapping into oxide defects. The hole trapping mechanism

cannot be described by the HEIMAN model, which is inherently temperature independent in contrast to NBTI. However, hole trapping could be related to a temperature-activated process called MultiPhonon Field-Assisted Tunneling (MPFAT). This MPFAT process is the key component of the two-stage degradation model implemented in ATLAS (see 4.1.4 Two Stage Negative Bias Temperature Instability Models). Even though the model is suited for the simulation of large-area devices, in which numerous defects are present, the four-state nonradiative multiphonon (NMP) model can also explain hole trapping into single defects [105, 273]. Recently, the new model has been extended to electron trapping, observed for Positive Bias Temperature Instability (PBTI) stress [274]. Using the combination of electron and hole traps, the model has been shown to explain all four cases of degradation modes, i.e. NBTI and PBTI in n-channel and p-channel MOS transistors.

As shown in Figure 4-2, the four-state NMP model for hole traps is based on a set of four states, denoted as 1, 1', 2, and 2'. The trap is assumed to be neutral in the states 1 and 1' and positive in the states 2 and 2'. The states 1 and 2 are stable while the other states, marked by a prime (1', 2'), are metastable. The transitions $(1 \leftrightarrow 2')$ and $(1' \leftrightarrow 2)$ represent the actual hole capture or emission processes described by the NMP theory. By contrast, the transitions $(1 \leftrightarrow 1')$ and $(2 \leftrightarrow 2')$ correspond to pure defect deformations, which are independent of the local electric field. Even though several different transition pathways are possible in the state diagram of Figure 4-2, the whole charge capture and the whole emission process occur between the stable states 1 and 2 and are therefore always two-step transitions. For instance, the whole hole capture process is usually described by a transition that starts from state 1, proceeds over state 2', and ends up in state 2. During hole emission, the defect goes from state 2 to state 1 via the metastable states 1' or 2'.

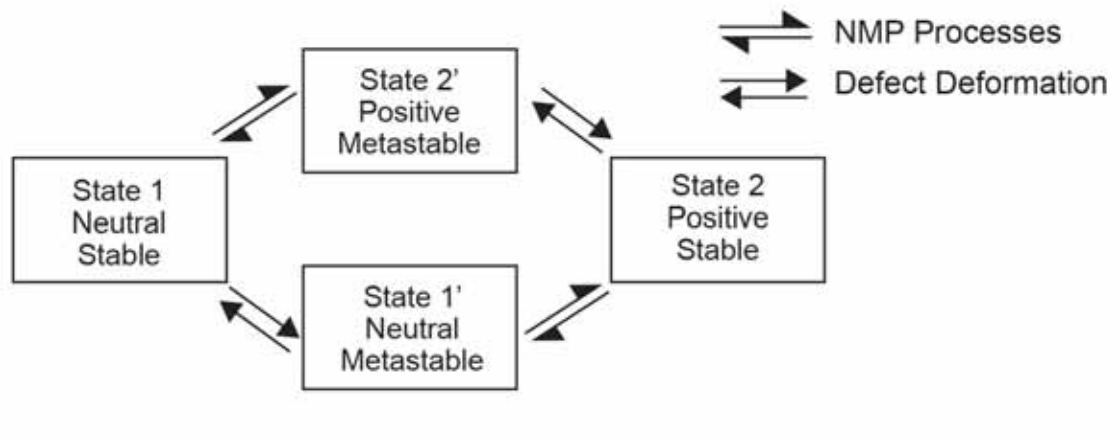


Figure 4-2: Schematic of Transitions and States in the Four-State NMP Model

NMP Processes

For hole capture, the NMP transition rate from state 1 to state 2' is given by

$$a_{12',p} = \text{SIGP}\, v_{th}^{p}\, p\, e^{-\varepsilon_{12'}/K_B T} f_{\text{WKB},p} \qquad 4\text{-}41$$

In the equation above, p is the channel concentration of the holes. Their thermal velocity and capture cross-section are denoted by v_{th}^{p} and SIGP, respectively. The factor $f_{\text{WKB},p}$ accounts

for the effect of hole tunneling between the substrate and the gate. The calculation of $f_{WKB,p}$ is based on the WKB approximation, where the energy barrier is formed by the valence band in the gate oxide. It is noted that the used tunneling mass for the holes must be set by the parameter MH.TUNNEL on the **MATERIAL** statement. $\varepsilon_{12'}$ corresponds to a thermal barrier that must be overcome during an NMP process. Its barrier height is calculated using the equation

$$\varepsilon_{12'} = \frac{\text{NMP4.S12S}}{(\text{NMP4.R12S}^2 - 1)^2}\left(1 \pm \text{NMP4.R12S}\sqrt{\frac{\text{NMP4.S12S} + \Delta E(\text{NMP4.R12S}^2 - 1)}{\text{NMP4.S12S}}}\right)^2 \quad \text{4-42}$$

where the relaxation energy NMP4.S12S and the energy NMP4.EPST2S are defined as shown in the configuration coordinate diagram of Figure 4-3.

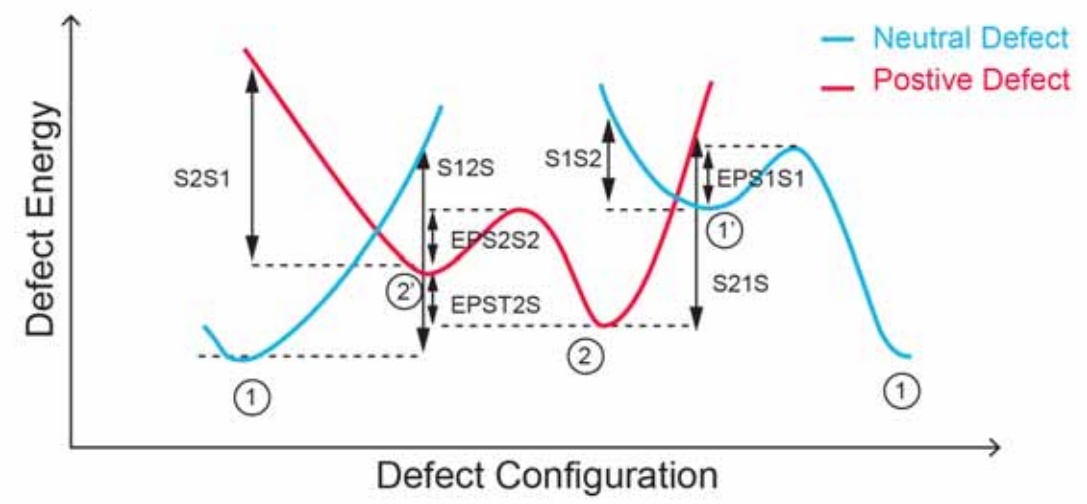


Figure 4-3: Configuration Coordinate Diagram of the Four-State NMP Model

In this diagram the defect energy is plotted as a function of its configuration for the cases when the defect is neutral (blue line) or positively charged (red line). The energy minima corresponds to one of the stable (1, 2) or metastable (1', 2') states in the four-state NMP model. The transitions between these states go either over the pure thermal barriers of the defect deformations ($1 \leftrightarrow 1'$, $2 \leftrightarrow 2'$) or over the NMP barriers ($1 \leftrightarrow 2'$, $1' \leftrightarrow 2$). The height of the latter is determined by the intersections between the corresponding energy curves in the configuration coordinate diagram. The parameter NMP4.R12S in Equation 4-42 can be obtained from the equation

$$\text{NMP4.R12S} = \sqrt{\frac{\text{NMP4.S12S}}{\text{NMP4.S2S1}}} \quad \text{4-43}$$

and ΔE corresponds to the energy gained or lost by the defect during the NMP process $1 \leftrightarrow 2'$. It is given by

$$\Delta E = E_v - \text{NMP4.ET1} + \text{NMP4.EPST2S} \quad \text{4-44}$$

where E_v denotes the substrate valence band and NMP4.ET1 is the trap level, referenced to the valence band of the gate dielectric. Furthermore, NMP4.EPST2S is the energy loss of the defect during the thermal transition from state 2' to state 2 (see Figure 4-3). As the barrier

$\varepsilon_{12'}$ in Equation 4-42 strongly varies with the position of the trap level NMP4.ET1, the hole capture rate $a_{2'1,p}$ shows a pronounced gate bias dependence characteristic for charge trapping.

The inverse process, hole emission, is an NMP transition rate from state 2' to state 1. Its rate is obtained from equation

$$a_{2'1,p} = a_{12',p}\, e^{-(\mathrm{NMP4.ET1} - \mathrm{NMP4.EPST2S} - E_{fp})/K_B T} \quad 4\text{-}45$$

where E_{fp} is the substrate Fermi level of the holes.

In the aforementioned hole capture process, the defect is assumed to trap a hole from the substrate valence band. Under certain bias condition, however, the defect can also emit an electron into the substrate conduction band. The respective electron emission rate is given by the equation

$$a_{2'1,n} = a_{12',n}\, e^{-(\mathrm{NMP4.ET1} - \mathrm{NMP4.EPST2S} - E_{fn})/K_B T} \quad 4\text{-}46$$

where E_{fn} denotes the substrate Fermi level of the electrons. $a_{2'1,n}$ corresponds to the electron capture rate, which is calculated using the equation

$$a_{12',n} = \mathrm{SIGN}\; v_{th}^{n}\; n\; e^{-\varepsilon_{2',1}/K_B T} f_{\mathrm{WKB},n} \quad 4\text{-}47$$

n is the channel concentration of the electrons and their thermal velocity and capture cross-section are denoted by v_{th}^{p} and SIGP, respectively. The term $f_{\mathrm{WKB},n}$ corresponds to the WKB factor for electrons, which tunnel over the energy barrier formed by the conduction band. Their tunneling mass must be specified by the parameter ME.TUNNEL on the **MATERIAL** statement again. The corresponding NMP barrier $\varepsilon_{2'1}$ is obtained from the equation

$$\varepsilon_{2'1} = \frac{\mathrm{NMP4.S2S1}}{(\mathrm{NMP4.R2S1}^2 - 1)^2}\left(1 \pm \mathrm{NMP4.R2S1}\sqrt{\frac{\mathrm{NMP4.S2S1} + \Delta E(\mathrm{NMP4.R2S1}^2 - 1)}{\mathrm{NMP4.S2S1}}}\right)^2 \quad 4\text{-}48$$

where NMP4.S2S1 is defined as in Figure 4-3 and NMP4.R2S1 is given by

$$\mathrm{NMP4.R2S1} = \sqrt{\frac{\mathrm{NMP4.S2S1}}{\mathrm{NMP4.S12S}}} \quad 4\text{-}49$$

The change in the defect energy ΔE is calculated using the equation

$$\Delta E = \mathrm{NMP4.ET1} - \mathrm{NMP4.EPST2S} - E_c \quad 4\text{-}50$$

The NMP transition rates between the states 1' and 2 are of the same form as the equations (4-41)-(4-50). The hole capture and emission rates $1' \leftrightarrow 2$ are given by

$$a_{1'2,p} = \mathrm{SIGN}\; v_{th}^{p}\; p\; e^{-\varepsilon_{1',2}/K_B T} f_{\mathrm{WKB},p} \quad 4\text{-}51$$

$$\varepsilon_{1'2} = \frac{\mathrm{NMP4.S1S2}}{(\mathrm{NMP4.R1S2}^2 - 1)^2}\left(1 \pm \mathrm{NMP4.R1S2}\sqrt{\frac{\mathrm{NMP4.S1S2} + \Delta E(\mathrm{NMP4.R1S2}^2 - 1)}{\mathrm{NMP4.S1S2}}}\right)^2 \quad 4\text{-}52$$

$$\Delta E = E_v - \text{NMP4.ET2} \tag{4-53}$$

$$\text{NMP4.R1S2} = \sqrt{\frac{\text{NMP4.S1S2}}{\text{NMP4.S21S}}} \tag{4-54}$$

$$a_{21',p} = a_{1'2,p}\, e^{-(\text{NMP4.ET2} - E_{fp})/K_B T} \tag{4-55}$$

The corresponding electron capture and emission rates read

$$a_{21',n} = \text{SIGN}\, v_{th}^{n}\, n\, e^{-\varepsilon_{21'}/K_B T} \tag{4-56}$$

$$a_{12',n} = a_{21',n}\, e^{(\text{NMP4.ET2} - E_{fn})/K_B T} f_{\text{WKB},n} \tag{4-57}$$

$$\varepsilon_{2'1} = \frac{\text{NMP4.S21S}}{(\text{NMP4.R21S}^2 - 1)^2}\left(1 \pm \text{NMP4.R21S}\sqrt{\frac{\text{NMP4.S21S} + \Delta E(\text{NMP4.R21S}^2 - 1)}{\text{NMP4.S21S}}}\right)^2 \tag{4-58}$$

$$\Delta E = \text{NMP4.ET2} - E_c \tag{4-59}$$

$$\text{NMP4.R21S} = \sqrt{\frac{\text{NMP4.S21S}}{\text{NMP4.S1S2}}} \tag{4-60}$$

In the equations NMP4.ET2 above denotes the trap level for the NMP processes $1' \leftrightarrow 2$. Again, the relaxation energies NMP4.S1S2 and NMP4.S21S are shown in Figure 4-3. The NMP transitions above are involved in the whole hole emission process that goes over the pathway $2 \leftrightarrow 1' \rightarrow 1$. This path becomes dominant if the trap level NMP4.ET2 falls below the substrate Fermi level.

Defect Deformation

In contrast to the above bias dependent NMP processes, the defect deformation $2' \rightarrow 2$ does not show any bias dependence and can only be thermally activated. The corresponding transition rate follows an Arrhenius-type law, given by

$$a_{2'2} = \text{NMP4.NU}\, e^{-\text{NMP4.EPS2S2}/K_B T} \tag{4-61}$$

In the above equation NMP4.NU denotes the attempt frequency and NMP4.EPS2S2 is the thermal barrier for the transition $2' \rightarrow 2$. The reverse process $2 \rightarrow 2'$ is obtained from the equation

$$a_{22'} = \text{NMP4.NU}\, e^{-(\text{NMP4.EPS2S2} + \text{NMP4.EPST2S})/K_B T} \tag{4-62}$$

The thermal barrier is now calculated as the sum of NMP4.EPS2S2 and NMP4.EPST2S, where the latter is the energy difference between metastable state 2’ and the stable state 2 (see Figure 4-3). The transition $2 \rightarrow 2'$ is part of the whole hole emission process $2 \leftrightarrow 2' \rightarrow 1$ and can give an explanation for the bias independent emission times, visible in time-dependent defect spectroscopy, for instance.

The defect deformation in the neutral charge state is described by the transition rates

$$a_{1'1} = \text{NMP4.NU}\, e^{-\text{NMP4.EPS1S1}/K_B T} \tag{4-63}$$

and

$$a_{11'} = \text{NMP4.NU}\, e^{-(\text{NMP4.EPS1S1} + \text{NMP4.ET2} - \text{NMP4.ET1})/K_B T} \quad 4\text{-}64$$

where NMP4.EPS1S1 denotes the thermal barrier from state 1’ to state 1.

Interface Reaction

The four-state NMP model has been extended by an interface reaction, which accounts for the permanent (or slowly recovering) component of NBTI. This reaction has been linked with the generation and the recovery of interface defects. However, the involved reactants have not been specified within this model. As such, the reaction can also be ascribed to the depassivation of P_b centers, leaving behind a positively charged Si dangling bond. In contrast to the RD model, the degradation is not dominated by the dynamics of the hydrogen diffusion but by the kinetics of the interface reaction.

The interface reaction is modeled by a double-well model with two stable states A and B (see Figure 4-4).

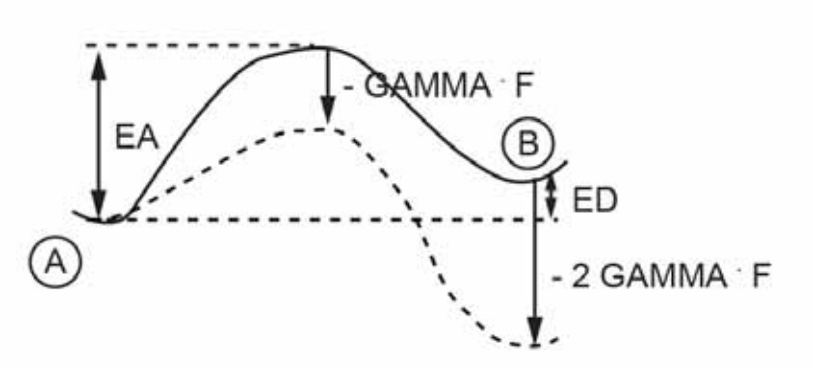


Figure 4-4: Configuration Coordinate Diagram of the Interface Reaction

The state A corresponds to a neutral interface defect and is energetically favorable compared to the positive interface defect in state B (in the absence of large oxide fields). However, an increase in the local electric field F lowers the energy of state B and thereby initiates a transition into the state B. The forward rate of this reaction reads

$$a_{AB} = \text{NMP4.NUP}\, e^{-(\text{NMP4.EA} - \text{NMP4.GAMMA}\ F)/K_B T} \quad 4\text{-}65$$

where NMP4.NUP and NMP4.EA are the attempt frequency and the barrier height, respectively. NMP4.GAMMA has a similar meaning as a dipole moment in an electric field. Therefore, the product of NMP4.GAMMA×F corresponds to the energy by which the barrier is reduced due to the electric field. The reverse rate of the interface reaction is given by

$$a_{BA} = \text{NMP4.NUP}\, e^{-(\text{NMP4.EA} - \text{NMP4.ED} + \text{NMP4.GAMMA}\ F)/K_B T} \quad 4\text{-}66$$

where NMP4.ED is the energy of the state B.

Electron Traps

The extended degradation model can account for electron as well as hole trapping in order to explain NBTI and PBTI. The electron traps are again described by a four-state NMP model but with the difference that the states 2’ and 2 correspond to negative charge states.

Analogous to hole traps, the whole charge capture and emission process occur as two-step transitions between the states 1 and 2. Electron capture usually proceed from state 1 to state 2 over the metastable 2'. Electron emission follows either of the transition pathways $2 \leftrightarrow 2' \rightarrow 1$ and $2 \leftrightarrow 1' \rightarrow 1$. The latter usually explain gate-bias dependent emission times (switching traps) while the former remain constant (fixed oxide traps) [338]. As in the case of hole traps, the defect deformations corresponds to the transitions $1 \leftrightarrow 1'$ and $2 \leftrightarrow 2'$. The actual charge capture and emission events are also caused by NMP processes, represented by the transitions $1 \leftrightarrow 2'$ and $1' \leftrightarrow 2$. Their corresponding NMP rates are given by the equations (4-41)-(4-60) again. However, the role of the forward and the reverse rates ($a_{12'} \leftrightarrow a_{2'1}$ and $a_{1'2} \leftrightarrow a_{21'}$) are exchanged since charge capture and emission are inverse processes if electron instead of hole trapping is considered. As a result of the reversed roles, the signs in front of NMP4.EPST2S are changed in the equations (4-41)-(4-60). The permanent component is again described by a double-well with a barrier that is reduced by an increased electric field at the interface and causes a negative interface charge.

Additional Comments

The four-state NMP model can be enabled by the flags NMP4.DET or NMP4.STO of the **TRAP** statement. The flag DONOR or ACCEPTOR can be set to chose between hole or electron traps, respectively. While the former are used to describe NBTI, the latter are suited for PBTI. In both cases, the simulated traps are assumed to reside in the region that is specified by the parameter MATERIAL. Their location can be further confined to a rectangle given by X.MIN, X.MAX, Y.MIN, and Y.MAX. In addition, the substrate region has to be selected by the parameter SUBCONTACT while the specification of a gate region is optional. If the gate should be considered, the gate region must be chosen by setting the GATECONTACT. The gate material can be either a polygate or a metal, with the latter requiring its workf unction to be specified on the **CONTACT** statement.

The four-state NMP model is implemented as a deterministic and a stochastic model. The former can be selected by the flag NMP4.DET and the latter by the flag NMP4.STO. In amorphous gate oxide materials, the parameter values are expected to differ from defect to defect. This suggests the use of the stochastic model with a multitude of traps for realistic device simulations. Similar to the two-stage degradation model, the randomly distributed trap parameters can be specified by their mean values and distribution widths. The latter has a denotation with the ending ".sd". The distribution widths must be set to a positive or negative value in order to chose a uniform or a normal distribution, respectively. It is noted that the random number generator also produces negative values for the distributed model parameters (except from NMP4.ED). Since negative values would be inconsistent with the basic assumptions of the four-state NMP model, only positive random numbers are considered. In addition, the user must also set the number of simulated traps per oxide node using the parameter NMP4.SAMPLES. As a rule of thumb, the parameter NMP4.SAMPLES should be set to a value larger than 5 for a device structure with more than 200 trap nodes.

The remaining parameters and flags of the four-state NMP model are listed in Table 4-7. Even though reasonable mean values for these parameters are given by default, they are subject to large device-to-device variations and change with different device technologies. Consequently, it is necessary to calibrate the model to experimental degradation data. In the following, you will find useful hints for the choice of reasonable parameter ranges:

- The device degradation is caused by hole traps which are shifted from below to above the substrate Fermi level. Therefore, their trap levels NMP4.ET must be within a range smaller than 1 eV around the substrate Fermi level.
- The trap levels NMP4.ET are usually assumed to be normally distributed. The mean value of this normal distribution should lie below the valence band edge for flat-band conditions.
- The trap level NMP4.ET2 must be equal or higher than NMP4.ET1 in energy. However, the distribution of trap levels NMP4.ET2 has to come close to NMP4.ET1 for the case that the recovery curves are strongly gate bias dependent.
- An increase relaxation energies NMP4.S12S and NMP4.S1S2 usually raises the heights of the corresponding NMP barriers, resulting in larger capture/emission times and a stronger temperature dependence.
- The quantities NMP4.R12S and NMP4.R1S2 are usually distributed around unity and vary by only a few tenths.
- In experiments the device degradation is found to last from µs up to at least 10^4s. This suggests wide distributions of the trap parameters that are shown in Figure 4-3. Their distribution widths usually extend to a few tenths of an electron Volt.
- The permanent component is often just a small contribution to the overall degradation. In those cases, the model should be calibrated without considering the permanent component at first. In a further step, the permanent component can be added for better calibration results.

The trapped interface and oxide charge can be saved in the structure file by using the parameter DEVDEG after the **OUTPUT** statement. The saved trap charges can be read in as fixed charges in subsequent simulations, in which the transfer characteristics can be calculated, for example. In addition, the occupation probabilities of the trap states are written out for visualization with TONYPLOT. In this tool, the states 1, 2', 2, and 1' correspond to the state numbers 1, 2, 3, and 4, respectively, and the state B of the interface reaction is assigned to the state 5.

Example

```
TRAP    Material=SiO2 SUBCONTACT=Silicon
        X.MIN=0.3 X.MAX=0.4 Y.MIN=-0.002 Y.MAX=-0.000
        NMP4.STO NMP4.SAMPLES=10
        DENSITY=1.0e+20 SIGN=1e-14 SIGP=1.0e-14
        NMP4.ET1=4.2 NMP4.ET1.SD=-0.4
        NMP4.S12S=2.0 NMP4.S12S.SD=1.0 NMP4.R12S=1.0 NMP4.R12S.SD=0.3
        NMP4.ET2=6.0
        NMP4.S1S2=3.0 NMP4.S1S2.SD=0.5 NMP4.R1S2=1.0
        NMP4.NU=1e13 NMP4.EPST2s=0.6 NMP4.EPST2S.SD=0.2
        NMP4.EPS2S2=0.2 NMP4.EPS2S.SD=0.2 NMP4.EPS1S1=3.0
        NMP4.NIT=1e11 NMP4.EA=1.65 NMP4.EA.SD=0.40
        NMP4.ED=0.15 NMP4.ED.SD=0.0 NMP4.GAMMA=8e-8 NMP4.NUP=1e12
```

Table 4-7 Default Values of the Four-State NMP Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
TRAP	NMP4.DET	Logical	False	
TRAP	NMP4.ST0	Logical	False	
TRAP	SUBCONTACT	Character		
TRAP	GATECONTACT	Character		
TRAP	SIGN	Real		cm^2
TRAP	SIGP	Real		cm^2
TRAP	NMP4.ET1	Real	0.0	eV
TRAP	NMP4.ET1.SD	Real	0.0	eV
TRAP	NMP4.ET2	Real	0.0	eV
TRAP	NMP4.ET2.SD	Real	0.0	eV
TRAP	NMP4.S12S	Real	2.5	eV
TRAP	NMP4.S12S.SD	Real	0.0	eV
TRAP	NMP4.R12S	Real	1.0	eV
TRAP	NMP4.R12S.SD	Real	0.0	eV
TRAP	NMP4.S1S2	Real	2.5	eV
TRAP	NMP4.S1S2.SD	Real	0.0	eV
TRAP	NMP4.R1S2	Real	1.0	eV
TRAP	NMP4.R1S2.SD	Real	0.0	eV
TRAP	NMP4.EPST2S	Real	0.3	eV
TRAP	NMP4.EPST2S.SD	Real	0.0	eV
TRAP	NMP4.EPS2S2	Real	0.3	eV
TRAP	NMP4.EPS2S2.SD	Real	0.0	eV
TRAP	NMP4.EPS1S1	Real	1.0	eV
TRAP	NMP4.EPS1S2.SD	Real	0.0	eV
TRAP	NMP4.NU	Real	10^{13}	eV
TRAP	NMP4.NU.SD	Real	0.0	eV
TRAP	NMP4.WKB	Logical	True	

Table 4-7 Default Values of the Four-State NMP Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
TRAP	NMP4.NIT	Real	0.0	cm^{-2}
TRAP	NMP4.NUP	Real	10^{13}	Hz
TRAP	NMP4.NUP.SD	Real	0.0	eV
TRAP	NMP4.EA	Real	1.3	eV
TRAP	NMP4.EA.SD	Real	0.0	eV
TRAP	NMP4.ED	Real	0.2	eV
TRAP	NMP4.ED.SD	Real	0.0	eV
TRAP	NMP4.GAMMA	Real	10^{-9}	cm/V
TRAP	NMP4.GAMMA.SD	Real	0.0	cm/V
TRAP	NMP4.SAMPLES	Integer	1	

Note: NMP4.ET1 and NMP4.ET2 are automatically referenced to valence band energy of the insulator.

4.1.6 Power-Law Degradation Mode

In this model, interface charge arising from degradation is given an explicit dependence on simulated time. The time factor used for evolution of electron traps is

$$\tau_e = \left(\mathrm{PL.E.NU0}\, exp\left(-\frac{\mathrm{PL.E.NA}}{K_B T}\right) t_{simulated}\right)^{\mathrm{PL.E.ALPHA}} \quad 4\text{-}67$$

where PL.E.NU0 is a rate prefactor, PL.E.EA is an activation energy for the rate constant, K_B is Boltzmann constant, and T is device temperature. The exponent PL.E.ALPHA gives the power law dependence of the time factor, and the elapsed simulated time is $t_{simulated}$. There is an entirely analagous expression for hole traps,

$$\tau_h = \left(\mathrm{PL.H.NU0}\, exp\left(-\frac{\mathrm{PL.H.NA}}{K_B T}\right) t_{simulated}\right)^{\mathrm{PL.H.ALPHA}} \quad 4\text{-}68$$

To enable electron trap modeling, you specify DEVDEG.E and DEVDEG.PL on the **MODELS** statement. The trap charge density along the interface is then given by

$$Q_{deg}(x) = -QNTA(x)\tau_e/(1+\tau_e) \quad 4\text{-}69$$

where $NTA(x)$ is the Si-H bond density and can be set by either the NTA or F.NTA parameter on the **DEGRADATION** statement. The units of charge are *Coulomb*/cm^2.

Similarly for hole traps with DEVDEG.H and DEVDEG.PL selected, the charge is

$$Q_{deg}(x) = +QNTD(x)\tau_h/(1+\tau_h) \quad 4\text{-}70$$

At large simulation times, the interface charge saturates at the value of the Si-H bond density. It is recommended to use the C-Interpreter functions F.NTA and F.NTD to define position dependent trap densities when using this model.

Table 4-8 Parameters for Power Law Degradation Model

Statement	Parameter	Type	Default	Units
DEGRADATION	PL.E.NU0	Real	1.0	/s
DEGRADATION	PL.H.NU0	Real	1.0	/s
DEGRADATION	PL.E.EA	Real	0.25	eV
DEGRADATION	PL.H.EA	Real	0.25	eV
DEGRADATION	PL.E.ALPHA	Real	0.5	
DEGRADATION	PL.H.ALPHA	Real	0.5	

The integrated charge density can be output to a log file by setting the DEVDEG parameter on the **LOG** statement.

4.1.7 Kinetic Degradation Model

An alternative model of MOSFET degradation based on [243] has been implemented in Atlas. Like the DEVDEG.RD model, it assumes that the depassivation of interface Si-H bonds is the main mechanism for device degradation. The interface equation is assumed to be

$$\frac{dN_{hb}}{dt} = -k_f N_{hb} + \gamma(N - N_{hb}) \qquad 4\text{-}71$$

where N_{hb} is the number of passivated Si-H bonds and N is the total number of passivated and unpassivated bonds and k_f and γ are rates. It is more convenient to write this in terms of the number of depassivated bonds, N_{it} where

$$N = N_{hb}^o + N_{it}^o = N_{hb} + N_{it} \qquad 4\text{-}72$$

where the superscript denotes the concentrations at the start of the simulation. The equation then becomes

$$\frac{dN_{it}}{dt} = k_f(N - N_{it}) - k_r\left(\frac{H}{H_o} + \Omega(N_{it} - N_{it}^o)\right)N_{it} \qquad 4\text{-}73$$

and the term γ has been replaced with a more detailed expression. The term $\frac{H}{H_o}$ is only included if hydrogen diffusion is also enabled, otherwise it is one. The model parameters are all set on the **DEGRADATION** statement, with the following symbolic identifications

$$\frac{N_{it}}{dt} = \texttt{A KC.KF0(KC.SIHTOT-}N_{it}\texttt{)} - \texttt{KC.KR0(}H/\texttt{KC.AMBIENTH + KC.VOLUME(}N_{it} - \texttt{KC.NIT0))}N_i \qquad 4\text{-}74$$

The initial value of the depassivated traps is KC.NIT0, a depassivation rate constant is KC.KF0, a repassivation rate constant is KC.KR0, and the maximum dangling bond density is

KC.SIHTOT. If the hydrogen diffusion model is enabled, then KC.AMBIENTH is a scaling factor for the repassivation of bonds due to excess hydrogen. You enable the hydrogen diffusion model by specifying the KC.HYDROGEN flag on the **MODELS** statement. If hydrogen diffusion is not enabled, then H/KC.AMBIENT = 1.

A non-zero value of the KC.VOLUME parameter enables a model for repassivation that doesn't require detailed modelling of the hydrogen diffusion. It assumes that the hydrogen density present at the interface is equal to the difference between the present and initial interface charge densities, N_{it} - KC.NIT0.

The parameter A in Equation 4-74 is quite complicated. It can be written as

$$A = \exp\left(\frac{KC.EA0}{K_B KC.T0} - \frac{KC.EA0+\nabla E_a}{\gamma T}\right) J_{eff} \quad 4\text{-}75$$

where

$$\gamma_T = K_B T + KC.DELTPARL\left|F_{||}\right|^{KC.RHOPARL} \quad 4\text{-}76$$

and

$$\Delta E_a = -KC.DELTAPERP\left|F_{\perp}\right|^{KC.RHOPERP} + 1(1+\beta)\gamma_T(\log(1+N_{it}) - \log(1+KC.NIT0)) \quad 4\text{-}77$$

and

$$\beta = -KC.BETA+KC.DELTAPERP\ \left|F_{\perp}\right| + KC.BETAPARL\ F_{|} \quad 4\text{-}78$$

The field at the interface is decomposed into parallel and perpendicular components, $F_{||}$ and $F_{\perp}$ respectively. The multiplicative factor J_{eff} is given by

$$J_{eff} = (1 + KC.E.HCCOEF\left|J_{hei}\right|^{KC.E.RHOHC} + KC.H.HCCOEF\left|J_{hhi}\right|^{KC.H.RHOHC}) \quad 4\text{-}79$$
$$\times (1 + KC.E.FNCOEF\left|J_{tun_e}\right|^{KC.E.RHOFN} + KC.H.FNCOEF\left|J_{tun_h}\right|^{KC.H.RHOFN})$$

The hot electron or hot hole flags, HEI or HHI, must be specified to obtain values for hot electron current, J_{hei}, and hot hole current, J_{hhi}, respectively. Similarly, a tunneling model such as FNORD or QTUNN must be enabled to obtain values for the gate tunnel currents for electrons, J_{tun_e}, and holes J_{tun_h}.

As with the DEVDEG.RD model, it is assumed that the dangling bonds become charged very soon after they are created.

The DEVDEG.E and DEVDEG.H parameters on the **MODELS** statement make the interface charge negative or positive respectively. The dangling bonds created at the Si/SiO_2 interface are commonly identified as P_b centers.

These are thought to act as amphoteric traps [41, 248] meaning that they can have either negative or positive charge. The DEVDEG.A parameter on the **MODELS** statement charges the traps according to the dominant carrier type at each point of the interface. If none of DEVDEG.A, DEVDEG.E, or DEVDEG.H are specified then the traps will be charged with electrons.

Specify the DEVDEG.KN flag on the **MODELS** statement to enable this model. The coupling between the interface state density and other parameters can be omitted from the Newton

iterations by clearing the KC.COUPLED flag on the **DEGRADATION** statement. This can improve runtime but may give lower quality results.

A summary of the parameters pertinent to the DEVDEG.KN model is given in Table 4-9.

Table 4-9 Parameters for the Kinetic Degradation Model

Statement	Parameter	Type	Default	Units
DEGRADATION	KC.AE0	Real	1.5	eV
DEGRADATION	KC.KF0	Real	10^{-5}	s
DEGRADATION	KC.KR0	Real	3.0×10^{-9}	s
DEGRADATION	KC.SIHTOT	Real	10^{12}	cm^{-2}
DEGRADATION	KC.NIT0	Real	0.0	cm^{-2}
DEGRADATION	KC.AMBIENTH	Real	1.0	cm^{-3}
DEGRADATION	KC.VOLUME	Real	0.0	cm^3
DEGRADATION	KC.T0	Real	300.0	Kelvin
DEGRADATION	KC.BETA	Real	0.0	
DEGRADATION	KC.BETAPERP	Real	0.0	cm/V
DEGRADATION	KC.BETAPARL	Real	0.0	cm/V
DEGRADATION	KC.DELTPARL	Real	0.0	[Q=1] cm
DEGRADATION	KC.DELTPERP	Real	0.0	[Q=1] cm
DEGRADATION	KC.RHOPARL	Real	1.0	
DEGRADATION	KC.RHOPERP	Real	1.0	
DEGRADATION	KC.E.HCCOEF	Real	0.0	cm^2/A
DEGRADATION	KC.H.HCCOEF	Real	0.0	cm^2/A
DEGRADATION	KC.E.FNCOEF	Real	0.0	cm^2/A
DEGRADATION	KC.H.FNCOEF	Real	0.0	cm^2/A
DEGRADATION	KC.E.RHOHC	Real	1.0	
DEGRADATION	KC.H.RHOHC	Real	1.0	
DEGRADATION	KC.E.RHOFN	Real	1.0	
DEGRADATION	KC.H.RHOFN	Real	1.0	
MODELS	DEVDEG.KN	Logical	False	
MODELS	KC.HYDROGEN	Logical	False	
MODELS	KC.COUPLED	Logical	False	

MODELS and DEGRADATION Statement Examples

```
MODELS MOS SRH DEVDEG.KN DEVDEG.A FNPP HEI

DEGRADATION     RD.SIHTOT=1.0E12     RD.KF0=1.0E-4     RD.KR0=1.0E-5
KC.VOLUME=1.0

RD.NIT0=1.0E6 KC.BETA=0.1 KC.BETAPERP=1.0E-11 KC.BETAPARL=-5.0E-8

KC.DELTPARL=1.0E-7         KC.DELTPERP=1.0E-8         KC.RHOPERP=1.0
KC.RHOPARL=1.0

KC.AMBIENTH=1.0     KC.T0=300.0     ^KC.COUPLED     KC.E.HCCOEF=1.0E4
KC.H.HCCOEF=0.0

KC.E.FNCOEF=1.0E10 KC.H.FNCOEF=0.0
```

4.1.8 Bulk Oxide Trap Degradation Model

It is thought that the hydrogenic species released by the depassivation of interface dangling bonds is also instrumental in the creation of defects in the bulk of the gate insulator layer itself [74]. Additionally, holes tunneling into the oxide are thought to catalyse bond breakage and defect creation in the oxide [207]. These defects give rise to stress induced leakage current (SILC) through the gate and ultimately gate dielectric breakdown (TDDB). This is not presently modeled with the DEVDEG models, although trapping of carriers injected into the oxide by bulk oxide states is, and is described in this section.

In order to simulate trapping of carriers in an insulator, the insulator must be made into a wide bandgap semiconductor by specifying the SEMICONDUCTOR flag on the **MATERIAL** statement. The properties of the traps are set by using the **DEVDEGBULKTRAP** statement. The parameters and their same for this statement are the same as the **NITRIDECHARGE** statement.

In order to get carrier injection into the insulator, you need to enable the THERMIONIC current injection model for the Semiconductor-insulator interface. To do this, you specify the parameters THERMIONIC and SITHERM on the **INTERFACE** statement. The hot carrier current calculated by the lucky electron model (HEI and HHI on the **MODELS** statement) is then injected into the insulator and undergoes drift-diffusion. It may possibly contribute to the carrier current of the gate contact or be transported back into the channel. The explicit hot carrier contributions to the terminal currents are included as part of the carrier current. The separate output of electrode hot carrier current is disabled in this case.

4.2 Environmental–Radiation and High/Low Temperature

4.2.1 Radiation

The Space Radiation Environment is complex, dynamic, and transient, consisting of Protons, Electrons, Alpha Particles, and Heavy Nuclei. Solar Events are the contributors to the dynamic and transient nature of the Space Radiation Environment.

Radiation introduces two fundamental elastic damage mechanisms, non-ionizing, and ionizing.

Non-ionizing introduces lattice displacement, caused by neutrons, protons, alpha particles, heavy ions, and very high-energy gamma photons. They change the arrangement of the atoms in the crystal lattice, creating lasting damage, and increasing the number of recombination centers, depleting the minority carriers and worsening the analog properties of the affected semiconductor junctions. It’s counter intuitive because higher doses over short time can cause partial annealing ("healing") of the damaged lattice, leading to a lower degree of damage than with the same doses delivered in low intensity over a long time. This type of problem is particularly significant in bipolar transistors, which are dependent on minority carriers in their base regions. Increased losses caused by recombination cause loss of the transistor gain.

Ionization effects are caused by charged particles, including the ones with energy too low to cause lattice effects. The ionization effects are usually transient, creating glitches and soft errors. But, this can lead to destruction of the device if they trigger a latch-up damage mechanism. Photocurrent caused by ultraviolet and x-ray radiation may belong to this category as well. Gradual accumulation of holes in the oxide layer in MOSFET transistors leads to worsening of their performance, up to device failure when the Total Ionizing Dose (TID) is high enough.

Spallation is a form of naturally occurring nuclear fission. It refers to the formation of elements from the inelastic impact of protons, neutrons, and heavy ions on atoms. This fission process emits multiple particles, which can cause ionizing and non-ionizing damage.

There are two units of measure of radiation: the Rad and the Gray. The Rad (radiation absorbed dose) is a unit used to measure a quantity called absorbed dose. This relates to the amount of energy actually absorbed in some material and is used for any type of radiation and any material. One Rad is defined as the absorption of 100 ergs per gram of material. The unit Rad can be used for any type of radiation, but it does not describe the biological effects of the different radiations.

The Gray (Gy) is a unit used to measure a quantity called absorbed dose. This relates to the amount of energy actually absorbed in some material and is used for any type of radiation and any material. One Gy is equal to one joule of energy deposited in one kg of a material. The unit gray can be used for any type of radiation, but it does not describe the biological effects of the different radiations. One Gy is equivalent to 100 rads.

4.2.2 Single Event Effects (SEE)

The capability of single event upset/photogeneration transient simulation is included in 2D and 3D using the **SINGLEEVENTUPSET** statement. It allows you to specify the radial, length, and time dependence of generated charge along tracks. There can be a single particle strike or multiple strikes. Each track is specified by an Entry Point Location (x0,y0,z0) and an Exit Point Location (x1,y1,z1). This is assumed to be a cylinder with the radius defined with the RADIUS parameter. In 2D, z0 and z1 should be neglected.

The entry and exit points are specified by the ENTRYPOINT and EXITPOINT parameters of the **SINGLEEVENTUPSET** statement. These are character parameters that represent the ordered triplet coordinates of the entry and exit points of the particle track.

The electron/hole pairs generated at any point is a function of the radial distance, r, from the center of the track to the point, the distance l along the track and the time, t. The implementation into Atlas allows you to define the generation rate as the number of electron-hole pairs per cm^3 along the track according to the equation:

$$G(r, l, t) = (\texttt{DENSITY}*L1(l) + S*\texttt{B.DENSITY}*L2(l))*R(r)*T(t) \quad \text{4-80}$$

where DENSITY and B.DENSITY are defined as the number of generated electron/hole pairs per cm^3.

In radiation studies, the ionizing particle is typically described by the linear charge deposition (LCD) value, which defines the actual charge deposited in units of pC/μm. You can use this definition within Atlas by specifying the PCUNITS parameter in the **SINGLEEVENTUPSET** statement. If the user- defined parameter, PCUNITS, is set in the **SINGLEEVENTUPSET** statement, then B.DENSITY is the generated charge, in pC/μm, and the scaling factor *S* is:

$$S = \frac{1}{q\pi\ \texttt{RADIUS}^2} \quad \text{4-81}$$

where RADIUS is defined on the **SINGLEEVENTUPSET** statement. If the PCUNITS parameter isn't set, then B.DENSITY is the number of generated electron/hole pairs in cm^{-3} and the scaling parameter, *S*, is unity.

Another common measure of the loss of energy of the SEU particle, as it suffers collisions in a material, is the linear energy transfer (LET) value, which is given in units of MeV/mg/cm^2 (or MeV-cm^2/mg). In silicon, you can convert energy to charge by considering you need approximately 3.6 eV of energy to generate an electron-hole pair. The conversion factor from the LET value to the LCD value is then approximately 0.01 for silicon. So for instance, a LET value of 25 MeV-cm^2/mg is equivalent to 0.25 pC/μm, which can then be defined with the B.DENSITY and PCUNITS parameters.

The factors, *L1* and *L2*, in Equation 4-80 define the variation of charge or carrier generation along the path of the SEU track. These variations are defined by the equations:

$$L1(l) = \texttt{A1} + \texttt{A2} \cdot l + \texttt{A3}\,exp(\texttt{A4} \cdot l) \quad \text{4-82}$$

$$L2(l) = \texttt{B1}(\texttt{B2} + l \cdot \texttt{B3})^{\texttt{B4}} \quad \text{4-83}$$

where the A1, A2, A3, A4, B1, B2, B3, and B4 parameters are user-definable as shown in Table 4-10.

Table 4-10 User-Specifiable Parameters for Equations 4-82 and 4-83			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	A1	1	
SINGLEEVENTUPSET	A2	0	cm^{-1}
SINGLEEVENTUPSET	A3	0	
SINGLEEVENTUPSET	A4	0	cm^{-1}
SINGLEEVENTUPSET	B1	1	
SINGLEEVENTUPSET	B2	1	
SINGLEEVENTUPSET	B3	0	cm^{-1}
SINGLEEVENTUPSET	B4	0	

Note: The default parameters in Table 4-10 were chosen to result in constant carrier or charge generation as a function of distance along the particle track.

The factor *R(r)* is the radial parameter, which is defined by one of two equations. The default is:

$$R(r) = exp\left(-\frac{r}{\mathtt{RADIUS}}\right) \quad 4\text{-}84$$

where *r* is the radial distance from the center of the track to the point and RADIUS is a user-definable parameter as shown in Table 4-11. You can choose an alternative expression if you specify the RADIALGAUSS parameter on the SINGLEEVENTUPSET statement. In this case, *R(r)* is given by:

$$R(r) = exp\left(-\frac{r}{\mathtt{RADIUS}}\right)^2 \quad 4\text{-}85$$

The time dependency of the charge generation *T*(*t*) is controlled with the TC parameter through two functions.

For TC=0:

$$T(t) = \mathtt{deltafunction}(t - \mathtt{T0}) \quad 4\text{-}86$$

For TC>0:

$$T(t) = \frac{2e^{-\left(\frac{t-\mathtt{TO}}{\mathtt{TC}}\right)^2}}{\mathtt{TC}\sqrt{\pi}erfc\left(\frac{-\mathtt{TO}}{\mathtt{TC}}\right)} \quad 4\text{-}87$$

where TO and TC are parameters of the SINGLEEVENTUPSET statement.

Table 4-11 User-Specifiable Parameters for Equations 4-80, 4-81, 4-86, and 4-87			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	DENSITY	0.0	cm^{-3}
SINGLEEVENTUPSET	B.DENSITY	0.0	cm^{-3} or pC/μm
SINGLEEVENTUPSET	TO	0.0	s
SINGLEEVENTUPSET	TC	0.0	s
SINGLEEVENTUPSET	RADIUS	0.05	μm

The following example shows a particle strike that is perpendicular to the surface along the Z plane. Therefore, only the Z coordinates change has a radius of 0.1 μm and a LET value of 20 MeV-cm^2/mg (B.DENSITY=0.2). The strike has a delay time of 60 ps and the Gaussian profile has a characteristic time of 10 ps. The generation track for this striking particle is from z=0 to z=10 μm and carrier generation occurs along its entire length.

```
single    entry="5.5,0.0,0.0" exit="5.5,0.0,10" radius=0.1 \
pcunits b.density=0.2 t0=60.e-12 tc=10.e-12
```

User-Defined SEU

In addition to the model described above, you can use the C-Interpreter to specify an arbitrary generation profile as a function of position and time. This is specified using the F.SEU parameter of the SINGLEEVENTUPSET statement.

```
SINGLEEVENTUPSET F.SEU=myseu.c
.
SOLVE DT=1e-14 TSTOP=1e-7
```

The F.SEU parameter indicates an external C-language sub-routine conforming to the template supplied. The file, myseu.c, returns a time and position dependent value of carrier generation. For more information about the C-Interpreter, see Appendix A "C-Interpreter Functions".

4.2.3 Total Ionizing Dose (TID)

Modeling Total Ionizing Dose (TID)

Several important radiation-induced failure modes in silicon devices, such as threshold voltage shift and leakage current increase, are associated with the buildup of charge (usually net positive) in the insulator regions. The following physical processes are required to model the radiation-induced charging in insulator regions:

- Electron-hole pair generation/recombination
- Electron and hole transport
- Electron and hole trapping

REM allows you to model these processes by treating the insulator region as if it is a wide bandgap semiconductor. A self-consistent, semiconductor-like system of equations are solved, specifically:

- Poisson's equation
- Electron and hole current continuity equations
- Electron and hole trapping equations

These equations are modified from the usual semicondutor equations to include radiation induced net electron-hole pair generation, dispersive transport and charge trapping.

To treat the insulator as a wide bandgap semicondutor, use the **MATERIAL** statement to change the insulator properties to that of a semiconductor. To do this, specify the parameter SEMICONDUCTOR. For example:

```
MATERIAL MATERIAL=oxide SEMICONDUCTOR
```

This statement must appear before any models or other material parameter definitions.

Insulator Equations

Poisson's Equation

To account for the electrostatic effects of mobile charge and the trapped charge, Poisson's equation in an insulator is written as follows:

$$\nabla \cdot (\varepsilon \nabla \psi) = -q(p - n + N_D - N_A + n_{t,\,rad} - p_{t,\,rad}) \quad \text{4-88}$$

where:

- n is the electron concentration.
- p is the hole concentration.
- $n_{t,rad}$ is the total trapped electron density due to radiation processes.
- $p_{t,rad}$ is the total trapped hole density due to radiation processes.
- Ψ is the electrostatic potential.
- ε is the permittivity.
- q is the electron charge.

Current Continuity Equations

Changes over time in densities of mobile electrons and holes are modelled using a coupled set of current continuity equations:

$$\frac{\partial n}{\partial t} = \frac{1}{q}\nabla \cdot \vec{J}_n + G_{net} - R_{n,\,disp} \quad \text{4-89}$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q}\nabla \cdot \vec{J}_p + G_{net} - R_{p,\,disp} \quad \text{4-90}$$

where:

- J_n is the electron current density.
- J_p is the hole current density.
- G_{net} is the net generation rate due to irradiation.
- $R_{n,disp}$ is the recombination/generation rate due to dispersive electron traps.

- $R_{p,disp}$ is the recombination/generation rate due to dispersive hole traps.

Electron and Hole Trapping Equations

The electron and hole total trap densities used in Poisson's equation are a product of the trap densities from the traps used in the dispersive transport model, $n_{t,disp}$ and $p_{t,disp}$, and the insulator charging traps, $n_{t,charge}$ and $p_{t,charge}$, that is:

$$n_{t,rad} = n_{t,disp} + n_{t,charge} \quad 4\text{-}91$$

$$p_{t,rad} = p_{t,disp} + p_{t,charge} \quad 4\text{-}92$$

See Section 4.2.5 “Dispersive Transport” and Section 4.2.6 “Insulator Charging” for more information.

4.2.4 Irradiation Generation Rate

A constant rate of electron-hole pair generation rate in an irradiated insulator, such as *a-SiO*$_2$ can be directly specified using:

$$G_{net} = \texttt{GMAX} \quad 4\text{-}93$$

Alternatively, a transient Gaussian pulse can be used if the GAUSSIAN parameter is specified, in which case G_{net} is given by:

$$G_{net} = \texttt{GMAX}\, exp\left(\frac{-(t-\texttt{TPEAK})^2}{2\ \texttt{SIGMAT}^2}\right) \quad 4\text{-}94$$

where:

- GMAX is the maximum net electron-hole pair generation rate.
- TPEAK is the peak time.
- SIGMAT is the standard deviation of the Gaussian pulse.

Dose Dependent Generation

The net generation rate can also be calculated from the radiation dose rate.

$$G_{net} = \texttt{G0 Y DOSERATE} \quad 4\text{-}95$$

where:

- G0 is the electron-hole pair generation factor which specifies the number of electron-hole pairs produced per unit *rad(SiO*$_2$*)*,
- DOSERATE is the dose rate (or rate of energy deposition) measured in units of *rad(SiO*$_2$*)*/s,
- Y is the geminate recombination yield function,

If the GAUSSIAN parameter is specified, G_{net} becomes:

$$G_{net} = \texttt{G0 Y DOSERATE}\ exp\left(\frac{-(t-\texttt{TPEAK})^2}{2\ \texttt{SIGMAT}^2}\right) \quad 4\text{-}96$$

The calculation of G0 takes into account the material density and the electron-hole pair creation energy (or average amount of energy absorbed for each electron-hole pair produced). Using 18 eV as the average electron-hole pair creation energy for $a\text{-}SiO_2$, you can calculate G0 to be $7.6\times10^{12}/cm^3\cdot rad(SiO_2)$. Simply stated, G0 is a constant that converts an exposure dose rate to a corresponding electron-hole pair generation rate.

Table 4-12 User Specifiable Parameters for Equations 4-93, 4-94, 4-95, and 4-96

Statement	Parameter	Type	Default	Units
RADIATION	DOSERATE	Real	0.0	$rad(SiO2)\cdot s^{-1}$
RADIATION	GO	Real	7.6×10^{12}	$cm^{-3}\cdot rad(SiO2)^{-1}$
RADIATION	GMAX	Real	1×10^{17}	$cm^{-3}s^{-1}$
RADIATION	SIGMAT	Real	0	s
RADIATION	TPEAK	Real	0	s

For Atlas/Device3D simulations, you must specify the logical parameter RADIATION on any SOLVE statement where the electron-hole pair generation due to irradiation is to be applied. For MixedMode/MixedMode3D simulations, use the .RAD statement instead.

Geminate Recombination

Geminate recombination occurs when an electron-hole pair recombines so soon after it is generated that neither carrier has a chance to transport [213]. For this reason, the geminate recombination term must be accounted for by reducing the generation rate itself rather than through a separate recombination term in the current continuity equations.

The geminate recombination yield function (Y) specifies the probability that an electron-hole pair escapes recombination. This is a function of the two opposing forces operating on the electron-hole pair. The first force arises from the Coulomb attraction between the electron and hole, which tends to cause the pair to recombine. The second is the electric field that arises from the applied biases, which tends to separate the carriers. This simple view best describes cases in which the electron-hole pairs are far enough apart from each other that they do not interact. Therefore, the term geminate (meaning sparsely distributed) is used. The yield function is given by:

$$Y = \left[\frac{|E| + (E0 \cdot Y0)}{|E| + E0}\right]^{A0} \qquad 4\text{-}97$$

where:

- $Y0$ is the yield at zero electric field.
- E is the electric field.
- $E0$ is the critical electric field value, which corresponds to the electric field at which the yield is $0.5*(1 + Y0)$.
- A0 moderates the growth rate of the function.

Table 4-13 User Specifiable Parameters for Equation 4-97			
Statement	**Parameter**	**Type**	**Units**
RADIATION	A0	Real	
RADIATION	E0	Real	$V \cdot cm^{-1}$
RADIATION	Y0	Real	

Table 4-14 Default Values for Geminate Yield Function				
Material	**Radiation Source**	**A0**	**E0(V/cm)**	**Y0**
Silicon	Unspecified	0.0	1.0	1.0
SiO_2	Unspecified	1.0	5.5×10^5	0.065
SiO_2	Co60	0.7	5.5×10^5	0.0
SiO_2	XRAY	0.9	1.36×10^6	0.0

Yield Function Dependence on Stopping Power

A method for calculating the yield function based on Brownian motion in a potential, representing both these forces was provided by Onsager to explain recombination in ionizing gases. The same basic explanation has been accepted for an experimentally observed yield function in irradiated insulators. Since this early work, several expressions for the yield function have been developed to explain varying experimental results. An important source of variation in the exact form is related to different radiation exposure conditions. In particular, the differences in the stopping power of the radiation used.

As noted above, the concept that geminate recombination can be understood on the basis of two opposing forces depends on the electron-hole pairs being spaced sufficient far apart from each other so that the interaction forces can be neglected. This criterion is satisfied in the case of radiation with low stopping power and gradually breaks down as the radiation stopping power increases. Eventually with very high-stopping power radiation, such as heavy ions, electron-hole pairs are created in a dense plasma and the concept of geminate recombination ceases to apply altogether. In this case, a different approach to modelling recombination, such as the columnar model for carrier recombination should be taken. For cases where the geminate model remains a good approximation, you may need to adjust the exact functional form of the yield function to suit exposure conditions. To facilitate this, REM allows the generation rate to be calculated from the field using a user-defined C-Interpreter function.

4.2.5 Dispersive Transport

The mobility of electrons in insulating materials is usually higher than holes under virtually all conditions (approximately 20 $cm^2/V \cdot s$ at room temperature and rising to 40 $cm^2/V \cdot s$ at

lower temperatures). At high fields, the electron velocity saturates at about 10^7 cm/s. Depending on experimental conditions, the hole mobility can range from 10^{-11} to 10^{-4} $cm^2/V \cdot s$ with a typical effective value of 10^{-5} $cm^2/V \cdot s$ at room temperature. Hole transport is also dispersive in nature, which gives rise to strong field and temperature dependence. More importantly, it gives rise to an apparent time and insulator thickness-dependence in the mobility. This makes it impossible to capture the details of transient behavior using simple drift-diffusion equations.

REM models dispersive transport using the multiple trapping-detrapping (MTD) approach [70]. This assumes that usual carrier (e.g., hole) transport is interrupted by trapping in shallow bandtail state. Transport resumes when they are detrapped according to a characteristic emission time, which is a function of the energy separation of the trap level from the valence band. These states are usually considered to have uniform spatial distributions and a (quasi) continuous energy distribution.

Multiple-Trapping Detrapping Model

The rate of trapping of hole at the donor sites used to simulate dispersive hole transport is given by:

$$\sum_{i}^{\mathrm{NUMD}} \frac{\partial p^{i}_{t,\,disp}}{\partial t} = \sum_{i}^{\mathrm{NUMD}} \left[\mathrm{SIGMA.DON}\ v_{th,\,p}\ p(N^{i}_{t,\,donor} - p^{i}_{t,\,disp}) - \frac{p^{i}_{t,\,disp}}{\tau^{i}_{p,\,donor}} \right] \quad \text{4-98}$$

where

- $N^{i}_{t,\,donor}$ is the density of donor states at the i^{th} donor level.
- $\tau^{i}_{p,\,donor}$ is the emission time for holes at the i^{th} donor level.
- SIGMA.DON is the capture cross-section for donor traps.
- $v_{th,p}$ is the thermal velocity of holes.
- $p_{t,disp}$ is the trapped hole density.
- NUMD is the number of donor trap levels.

The density of donor states ($N^{i}_{t,\,donor}$) in the i^{th} donor state is given by:

$$N^{i}_{t,\,donor} = \mathrm{N0.DON} \exp\left(-\frac{E^{i}_{t,\,donor} - E_v}{\mathrm{BETA.DON}} \right) \quad \text{4-99}$$

where:

- N0.DON is the density of states at the valence band edge,
- $E^{i}_{t,\,donor}$ is the energy level above the valence band of the i^{th} donor state,
- BETA.DON is the rate of change of the donor state density with energy level.

The characteristic hole emission time (hole lifetime) i^{th} level of the donor states is given by:

$$\tau^{i}_{p,\,donor} = \texttt{E0.DON} exp\left[-\frac{E^{i}_{t,\,donor} - E_v}{kT}\right] \quad \text{4-100}$$

where E0.DON is the pre-exponent emission time factor for holes in donor traps.

Table 4-15 User Specifiable Parameters for Equations 4-98, 4-99, and 4-100

Statement	Parameter	Type	Default	Units
DISPERSIVE	BETA.DON	Real	0.04	eV
DISPERSIVE	E0.DON	Real	1×10^{12}	s^{-1}
DISPERSIVE	N0.DON	Real	0.0	cm^{-3}
DISPERSIVE	NUM.DON	Real	10	
DISPERSIVE	SIGMA.DON	Real	1×10^{-15}	cm^2

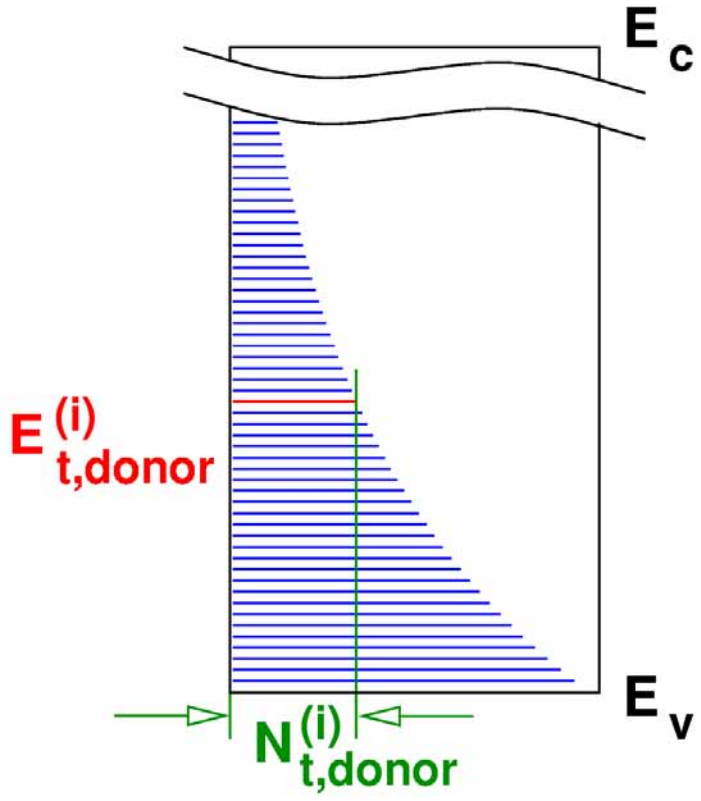


Figure 4-5: Donor States

A similar expression is used to describe the trapping of electrons at acceptor states:

$$\sum_{i}^{\text{NUMA}} \frac{\partial n^{i}_{t,\,disp}}{\partial t} = \sum_{i}^{\text{NUMA}} \left[\texttt{SIGMA.ACC}\ v_{th,\,n}\ n(N^{i}_{t,\,acceptor} - n^{i}_{t,\,disp}) - \frac{n^{i}_{t,\,disp}}{\tau^{i}_{n,\,acceptor}}\right] \quad \text{4-101}$$

where:

- $N^{i}_{t,\,acceptor}$ is the density of acceptor states at the i^{th} acceptor level.
- $\tau^{i}_{n,\,acceptor}$ is the emission time for electrons at the i^{th} acceptor level.
- SIGMA.ACC is the capture cross-section for acceptor traps.
- $v_{th,\,n}$ is the thermal velocity of electrons.
- $n^{i}_{t,\,disp}$ is the trapped electron density.
- NUMA is the number of acceptor trap levels.

$$N^{i}_{t,\,acceptor} = \mathtt{N0.ACC}\,exp\left(-\frac{E_C - E^{i}_{t,\,acceptor}}{\mathtt{BETA.ACC}}\right) \quad 4\text{-}102$$

where:

- N0.ACC is the density of states at the conductor band edge.
- BETA.ACC is the rate of change of the acceptor state density with energy level.

$$\tau^{i}_{n,\,acceptor} = \mathtt{E0.ACC}\,exp\left[-\frac{E_c - E^{i}_{t,\,acceptor}}{kT}\right] \quad 4\text{-}103$$

where E0.ACC is the pre-exponent emission time factor for electrons in acceptor traps.

Both models are available in REM for completeness.

Table 4-16 User Specifiable Parameters for Equations 4-98, 4-99, 4-100, 4-101, 4-102, and 4-103

Statement	Parameter	Type	Default	Units
DISPERSIVE	BETA.ACC	Real	0.04	eV
DISPERSIVE	E0.ACC	Real	1×10^{12}	s^{-1}
DISPERSIVE	N0.ACC	Real	0.0	cm^{-3}
DISPERSIVE	NUM.ACC	Real	10	
DISPERSIVE	SIGMA.ACC	Real	1×10^{-15}	cm^{2}

Recombination

The net recombination/generation rate for electrons and holes due to dispersive transport is given by:

$$R_{n,\,disp} = N^{i}_{t,\,acceptor} v_{th,\,n} \mathtt{SIGMA.ACC}\ n(1 - f^{i}_{t,\,acceptor}) - \frac{f^{i}_{t,\,acceptor}}{\tau^{i}_{n,\,acceptor}} \quad 4\text{-}104$$

$$R_{p,\,disp} = N^{i}_{t,\,donor} v_{th,\,p} \texttt{SIGMA.DON}\ p\ f^{i}_{t,\,donor} - \frac{(1 - f^{i}_{t,\,donor})}{\tau^{i}_{p,\,donor}} \quad \text{4-105}$$

where $f^{i}_{t,\,acceptor}$ is calculated from:

$$n^{i}_{t,\,disp} = N^{i}_{t,\,acceptor}\ \ (1 - f^{i}_{t,\,acceptor}) \quad \text{4-106}$$

and $f^{i}_{t,\,donor}$ is calculated from

$$p^{i}_{t,\,disp} = N^{i}_{t,\,acceptor}\ \ (1 - f^{i}_{t,\,donor}) \quad \text{4-107}$$

4.2.6 Insulator Charging

REM supports two separate models for insulator charging effects. The first model is known as the J model because it is based on the carrier current density.

The net rate of electron trapping is given by:

$$\frac{\partial n_{t,\,charge}}{\partial t} = \texttt{SIGMAT.N}\ \frac{\vec{J}_n}{q}(\texttt{NT.N} - n_{t,\,charge}) - \texttt{SIGMAN.P}\ \frac{\vec{J}_p}{q}\ n_{t,\,charge} - \frac{n_{t,\,charge}}{\texttt{TAU.N}} \quad \text{4-108}$$

The net rate of hole trapping is given by:

$$\frac{\partial p_{t,\,charge}}{\partial t} = \texttt{SIGMAT.P}\ \frac{\vec{J}_p}{q}(\texttt{NT.P} - p_{t,\,charge}) - \texttt{SIGMAP.N}\ \frac{\vec{J}_n}{q}\ p_{t,\,charge} - \frac{p_{t,\,charge}}{\texttt{TAU.P}} \quad \text{4-109}$$

where:

- SIGMAT.N is the electron capture cross-section for electron traps.
- SIGMAT.P is the hole capture cross-section for hole traps.
- NT.N is the initial density of electron traps in the insulator.
- NT.P is the initial density of hole traps in the insulator.
- SIGMAN.P is the electron capture cross-section for hole traps.
- SIGMAP.N is the hole capture cross-section for electron traps.
- $n_{t,charge}$ is the trapped electron density.
- $p_{t,charge}$ is the trapped hole density.

The second model is known as the V model because it is based on the thermal velocity:

$$\frac{\partial n_{t,\,charge}}{\partial t} = \texttt{SIGMAT.N}\ v_{th,\,n}\ n(\texttt{NT.N} - n_{t,\,charge}) - \texttt{SIGMAN.P}\ v_{th,\,p} - \frac{t,\,charge}{\texttt{TAU.N}} - \frac{p_{t,\,charge}}{\texttt{TAU.N}} \quad \text{4-110}$$

$$\frac{\partial p_{t,\,charge}}{\partial t} = \texttt{SIGMAT.P}\ v_{th,\,p}\ p(\texttt{NT.P} - p_{t,\,charge}) - \texttt{SIGMAP.N}\ v_{th,\,n}\ np_{t,\,charge} - \frac{p_{t,\,charge}}{\texttt{TAU.P}} \quad \text{4-111}$$

Table 4-17 User Specifiable Parameters for Equations 4-98, 4-99, 4-100, 4-108, 4-109, 4-110, and 4-111				
Statement	**Parameter**	**Type**	**Default**	**Units**
OXIDECHARGING	NT.N	Real	0.0	cm^{-3}
OXIDECHARGING	NT.P	Real	0.0	cm^{-3}
OXIDECHARGING	SIGMAN.P	Real	1×10^{-15}	cm^{2}
OXIDECHARGING	SIGMAP.N	Real	1×10^{-14}	cm^{-2}
OXIDECHARGING	SIGMAT.N	Real	1×10^{-16}	cm^{2}
OXIDECHARGING	SIGMAT.P	Real	1×10^{-14}	cm^{2}
OXIDECHARGING	TAU.N	Real	1×10^{30}	s
OXIDECHARGING	TAU.P	Real	1×10^{30}	s

The default algorithm for Insulator charging is to explicitly update the values of $n_{t,charge}$ and $p_{t,charge}$ at the end of each time step using the updated values of the carrier densities and electrostatic potential.

This is satisfactory for many devices. For some combinations of parameters, however, it is preferable to solve Equations 4-110 and 4-111 self-consistently with the current continuity and Poisson equations. This has been implemented for the V model. In this case, the recombination rates for current continuity equations are given as

$$R_n = \mathrm{SIGMAT.N}\ v_{th,n}\ n(\mathrm{NT\cdot N} - n_{t,charge}) - \frac{n_{t,charge}}{\mathrm{TAU.N}} + \mathrm{SIGMAP.N}\ v_{th,n}\ np_{t,charge} \qquad 4\text{-}112$$

$$R_p = \mathrm{SIGMAT.P}\ v_{th,p}\ p(\mathrm{NT\cdot P} - p_{t,charge}) - \frac{p_{t,charge}}{\mathrm{TAU.P}} + \mathrm{SIGMAN.P}\ v_{th,p}\ pn_{t,charge} \qquad 4\text{-}113$$

To enable this functionality, specify ICDE on the **MATERIAL** statement. This will cause both Equations 4-110 and 4-111 to be solved as part of a fully coupled NEWTON method. Specifying ICDE.ELEC will enable the solution of Equation 4-110. Specifying ICDE.HOLE will enable the solution of Equation 4-111.

Table 4-18 Miscellaneous REM Parameters			
Statement	**Parameter**	**Type**	**Default**
MATERIAL	SEMICONDUCTOR	Logical	False
METHOD	ICDE	Logical	False

Table 4-18 Miscellaneous REM Parameters			
Statement	**Parameter**	**Type**	**Default**
METHOD	ICDE.ELEC	Logical	False
METHOD	ICDE.HOLE	Logical	False
OXIDECHARGING	JMODEL.N	Logical	False
OXIDECHARGING	JMODEL.P	Logical	False
RADIATION	GAUSSIAN	Logical	False
SOLVE	RADIATION	Logical	False

4.2.7 REM Statements

DISPERSIVE

DISPERSIVE specifies the parameters for dispersive transport of carriers.

Syntax

```
DISPERSIVE  <parameter > [<parameter > *]
```

Parameter	Type	Default	Unit
BETA.ACC	Real	0.04	eV
BETA.DON	Real	0.04	eV
DEVICE	Character		
E0.ACC	Real	1×10^{12}	s^{-1}
E0.DON	Real	1×10^{12}	s^{-1}
ELIM1.ACC	Real	eV	
ELIM2.ACC	Real	eV	
ELIM1.DON	Real	eV	
ELIM2.DON	Real	eV	
F.DISPACC	Character		
F.DISPON	Character		
FILE.ACC	Character		
FILE.DON	Character		
MATERIAL	Character		

Parameter	Type	Default	Unit
N0.ACC	Real	0.0	cm^{-3}
N0.DON	Real	0.0	cm^{-3}
NUM.ACC	Integer	10	
NUM.DON	Integer	10	
REGION	Integer	All	
SIGMA.ACC	Real	1×10^{-15}	cm^2
SIGMA.DON	Real	1×10^{-15}	cm^2
X.MIN	Real		microns
X.MAX	Real		microns
Y.MIN	Real		microns
Y.MAX	Real		microns
Z.MIN	Real		microns
Z.MAX	Real		microns

Description

BETA.ACC	Specifies the rate of change of the acceptor state density with energy level in Equation 4-102.
BETA.DON	Specifies the rate of change of the donor state density with energy level given in Equation 4-99.
DEVICE	Specifies which device the statement should apply to in MixedMode simulation.
E0.ACC	Specifies the pre-exponent factor that is used to calculate the energy dependent emission time for electrons in acceptor states in Equation 4-103.
E0.DON	Specifies the pre-exponent factor that is used to calculate the energy dependent emission time for holes in donor states in Equation 4-104.
ELIM1.ACC	Specifies the lower limit of the effective energy range with respect to the conduction band edge for the acceptor states. The limit must be greater than zero and not exceed the bandgap energy.
ELIM2.ACC	Specifies the upper limit of the effective energy range with respect to the conduction band edge for the acceptor states. The limit must be greater than zero and not exceed the bandgap energy.

ELIM1.DON	Specifies the lower limit of the effective energy range with respect to the valence band edge for the donor states. The limit must be greater than zero and not exceed the bandgap energy.
ELIM2.DON	Specifies the upper limit of the effective energy range with respect to the valence band edge for the donor states. The limit must be greater than zero and not exceed the bandgap energy.
F.DISPACC	Specifies the name of a file containing a C-Interpreter function describing the distribution of acceptor state densities as a function of energy.
F.DISPON	Specifies the name of a file containing a C-Interpreter function describing the distribution of donor state densities as a function of energy.
FILE.ACC	Specifies the acceptor state density output file.
FILE.DON	Specifies the donor state density output file.
N0.ACC	Specifies the density of acceptor states at the conduction band edge.
N0.DON	Specifies the density of donor states at the valence band edge.
REGION	Specifies the region to which the **DISPERSIVE** statement applies.
NUM.ACC	Specifies the number of discrete acceptor levels that will be used starting from the conduction band edge.
NUM.DON	Specifies the number of discrete donor levels that will be used starting from the valence band edge.
SIGMA.ACC	Specifies the capture cross-section for acceptor traps.
SIGMA.DON	Specifies the capture cross-section for donor traps.
X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN**, and **Z.MAX**	Specifies the bounding box.
MATERIAL	Specifies the material to which the **DISPERSIVE** statement applies.

OXIDECHARGING

The various parameters of the charge trapping model can be specified using the **OXIDECHARGING** statement.

Syntax

```
OXIDECHARGING  <parameter > [<parameter > *]
```

Parameter	Type	Default	Unit
DEVICE	Character		
F.CHARGING	Character		

Parameter	Type	Default	Unit
JMODEL.N	Logical	False	
JMODEL.P	Logical	False	
MATERIAL	Character		
NAME	Character		
NT.N	Real	0.0	cm^{-3}
NT.P	Real	0.0	cm^{-3}
REGION	Real	All	
SIGMAN.P	Real	1×10^{-15}	cm^2
SIGMAP.N	Real	1×10^{-14}	cm^2
SIGMAT.N	Real	1×10^{-16}	cm^2
SIGMAT.P	Real	1×10^{-14}	cm^2
TAU.N	Real	1×10^{30}	s
TAU.P	Real	1×10^{30}	s
X.MIN	Real		microns
X.MAX	Real		microns
Y.MIN	Real		microns
Y.MAX	Real		microns
Z.MIN	Real		microns
Z.MAX	Real		microns

Description

DEVICE	Specifies a MixedMode device to which the OXIDECHARGING statement applies.
F.CHARGING	Specifies the name of a file containing a C-Interpreter function describing the position, electric field and current dependent net doping.

For Atlas, the arguments for this function are:

```
int charging( int ireg,
              double x,
              double y,
              double n,
              double p,
```

```
                doubleex,
                double ey,
                double jnx,
                double jny,
                double jpx,
                double jpy,
                double delt,
                int iter,
                double *nnetn,
                double *nnetp )
```

For Atlas 3D, the arguments for this function are:

```
int charging3( int ireg,
               double x,
               double y,
               double z,
               double n,
               doublep,
               double ex,
               double ey,
               double ez,
               double jnx,
               double jny,
               double jnz,
               double jpx,
               double jpy,
               double jpz,
               double delt,
               int iter,
               double *nnetn,
               double *nnetp )
```

where:

- `ireg` is the region number.
- x is the x location (microns).
- y is the y location (microns).
- `z` is the z location (microns), n is the electron concentration (cm^{-3}).
- `p` is the hole concentration (cm^{-3}).
- `ex` is the electric field in the x direction (V/cm).
- ey is the electric field in the y direction (V/cm).
- `ez` is the electric field in the z direction (V/cm),
- `jnx` is the electron current density in the x direction (A/cm^2).
- `jny` is the electron current density in the y direction (A/cm^2).
- `jnz` is the electron current density in the z direction (A/cm^2).
- `jpx` hole current density in the x direction (A/cm^2).
- `jpy` is the hole current density in the y direction (A/cm^2).
- `jpz` is the hole current density in the z direction (A/cm^2).

- `delt` is the time step (s).
- `iter` is the iteration number.
- `nnetn` is the calculated trapped electron density (cm^{-3}).
- `nnetp` is the calculated trapped hole density (cm^{-3}).

JMODEL.N	Specifies that the current dependent "J-model" for electron insulator charging (Equation 4-108) will be used.
JMODEL.P	Specifies that the current dependent "J-model" for hole insulator charging (Equation 4-109) will be used.
MATERIAL	Specifies the material to which the **OXIDECHARGING** statement applies.
NAME	Specifies the material name to which the **OXIDECHARGING** statement applies.
NT.N	Specifies the initial density of electron traps in the insulator. The electron traps are assumed to be uniformly distributed in the specified region, material or inside the bounding box.
NT.P	Specifies the initial density of hole traps in the insulator. The hole traps are assumed to be uniformly distributed in the specified region, material or inside the bounding box.
REGION	Specifies the region to which the **OXIDECHARGING** statement applies.
SIGMAP.N	Specifies the electron capture cross-section for trapped holes.
SIGMAN.P	Specifies the hole capture cross-section for trapped electrons.
SIGMAT.N	Specifies the electron capture cross-section for electron traps.
SIGMAT.P	Specifies the hole capture cross-section for hole traps.
TAU.N	Specifies the electron lifetime of the electron traps.
TAU.P	Specifies the electron lifetime of the hole traps.
X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN**, and **Z.MAX**	Specifies the bounding box.

RADIATION

RADIATION specifies the parameters of the ionizing radiation exposure.

Syntax

```
RADIATION <parameter > [<parameter > *]
```

Parameter	Type	Default	Unit
DEVICE	Character		
DOSERATE	Real	0.0	rad, s^{-1}

E0	Real	5.5×10^{5}	V, cm^{-1}
F.IRRADIATE	Character		
GAUSSIAN	Logical	false	
G0	Real	7.6×10^{12}	cm^{-3} rad^{-1}
GMAX	Real	1×10^{17}	cm^{-3} s^{-1}
MATERIAL	Character		
NAME	Character		
REGION	Real	All	
SIGMAT	Real	0	s
TPEAK	Real	0	s
X.MIN	Real		microns
X.MAX	Real		microns
Y0	Real	6.5×10^{-2}	
Y.MIN	Real		microns
Y.MAX	Real		microns
Z.MIN	Real		microns
Z.MAX	Real		microns

Description

DEVICE	Specifies a MixedMode device to which the **RADIATION** statement applies.
DOSERATE	Specifies the constant dose rate associated with an ionizing radiation exposure.
E0	Specifies the critical electric field.
F.IRRADIATE	Specifies the name of a file containing a C-Interpreter function describing the position, electric field and current dependent net radiation generation rate.

For Atlas, the arguments for this function are:

```
int irradiate( double  x,
               double  y,
               double  time,
               double  ex,
               double  ey,
               double* rat )
```

For Atlas 3D, the arguments for this function are:

```
int irradiate3( double  x,
                double  y,
                double  z,
                double  time,
                double  ex,
                double  ey,
                double  ez,
                double* rat )
```

where:

- x is the x location (microns).
- y is the y location (microns).
- z is the z location (microns).
- ex is the electric field in the x direction (V/cm).
- time is the current transient time(s).
- ey is the electric field in the y direction (V/cm).
- ez is the electric field in the z direction (V/cm).
- rat is the calculated radiation generation rate (cm^{-3} s^{-1}).

GAUSSIAN	Specifies that a transient Gaussian pulse will be used.
G0	Specifies the number of electron-hole pairs generated per unit volume by a total-dose exposure of 1 rad for the specified material.
GMAX	Specifies maximum net electron-hole pair generation rate.
MATERIAL	Specifies the material to which the **RADIATION** statement applies.
NAME	Specifies the material name to which the **RADIATION** statement applies.
REGION	Specifies the region to which the **RADIATION** statement applies.
SIGMAT	Specifies the standard deviation for the Gaussian pulse.
TPEAK	Specifies the peak time for Gaussian pulse.
Y0	Specifies the zero-field yield in Equation 4-94.
X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN, and Z.MAX	Specifies the bounding box.

.RAD

.RAD specifies that a radiation source defined on the **RADIATION** statement is to be enabled during a MixedMode/MixedMode 3D transient sweep.

Syntax

```
.RAD [default_value][transient_parameters]
```

Description

`default_value`	Specifies the default transient dose scale value used if no transient parameters are specified.
`transient_parameters`	Specifies the transient parameters as described in Section 13.4.3 "Transient Parameters".

Total Ionizing Dose (TID) Examples

remex01.in: Irradiation of a MOS Capacitor

Requires: S-Pisces/REM

MOS capacitor with irradiation. This example demonstrates:

- device formation using Atlas syntax,
- radiation specification using Gaussian generation rate,
- radiation specification using Gaussian dose rate,
- effects of dispersive transport and dispersive transport with Geminate recombination.

The mesh, region, electrodes and doping are defined using Atlas syntax. The oxide region is then changed from an insulator to a semiconductor using `material material=oxide semiconductor`.

The **`RADIATION`** statement is used to specify the dose rate applied to the device. A Gaussian pulse is used with a maximum generation rate (`gmax`) of 1.5e23, peak time (`tpeak`) of 1e-5s and a standard deviation (`sigmat`) of 2e-6s.

The **`SOLVE`** `init` statement is used to solve the thermal equilibrium case. After this, the gate voltage is ramped to 1V.

A transient simulation is performed starting from 1e-6s and finishing at 1e6s. The terminal characteristics saved to the file `remex01_0.log` as specified in the **`LOG`** statement. The `radiation` parameter is specified on each **`SOLVE`** statement where there is to be irradiation. The mininum and maximum time steps are carefully controlled by specifying `dt.min` and `dt.max` on the `method` statement.

Once this simulation is complete, a second Atlas simulation is performed. This simulation is identical to the first, except that the **`DISPERSIVE`** statement is used to specify the dispersive transport parameters. The transient terminal characteristics are stored in the log file `remex01_1.log`.

The third Atlas simulation uses `doserate` to specify the dose rate of the radiation. The also enables the Geminate recombination model. The results are stored in `remex01_2.log`.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DeckBuild, select the **Run** button to execute the example.

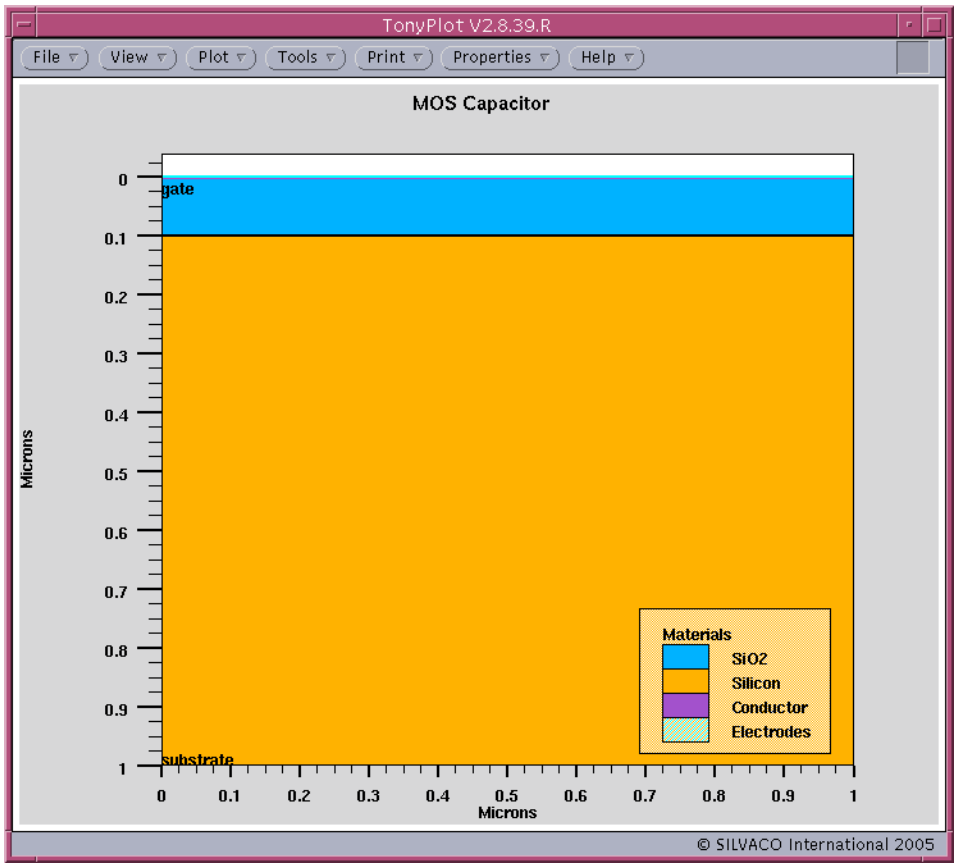


Figure 4-6: MOS Capacitor

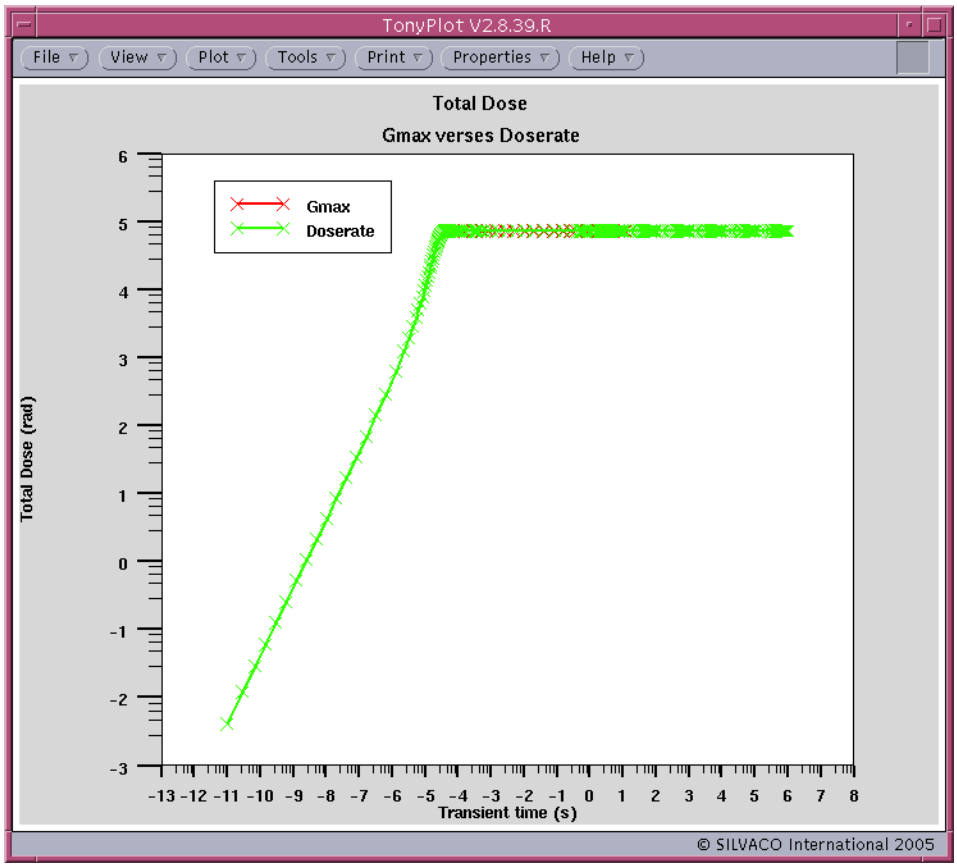


Figure 4-7: Total Dose as a function of time

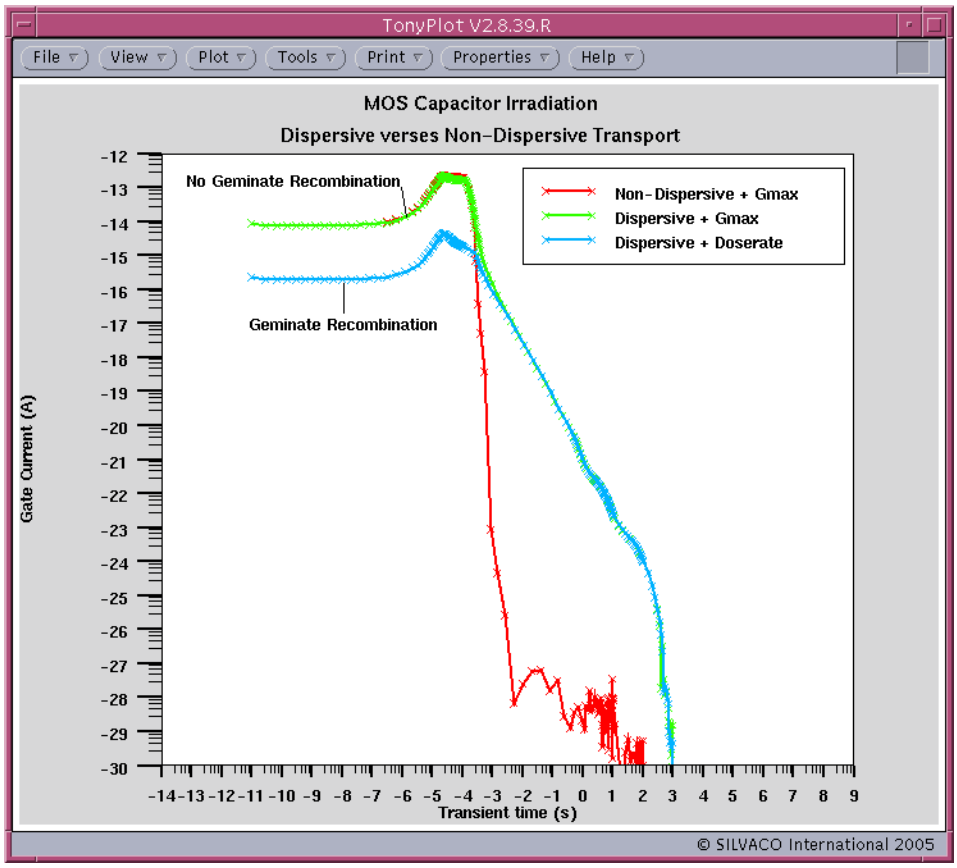


Figure 4-8: Dispersive verses Non-Dispersive Transport for MOS Capacitor

remex01.in Input File

```
go atlas
mesh
x.m l=0 sp=0.5
x.m l=1 sp=0.5

y.m l=0 sp=0.001
y.m l=0.1 sp=0.001
y.m l=0.11 sp=0.001
y.m l=1 sp=0.1

region num=1 oxide y.max=0.1
region num=2 silicon y.min=0.1

electrode name=gate top
```

```
electrode name=substrate bottom

doping material=silicon p.type conc=1e17 unif

material material=oxide semiconductor
material material=oxide nc300=1e19 nv300=1e19 eg300=9 mun=20 mup=1.0e-6

radiation gaussian gmax=1.5e23 tpeak=2.0e-5 sigmat=1e-5 y.max=0.005

method newton
solve init
solve prev
solve name=gate vstep=1.0 vstop=1.0

log outfile=remex01_0.log

method dt.max=1.0e-6
solve radiation tstep=1.0e-11 tstop=1.0e-4
save outfile=remex01_0.str

tonyplot remex01_0.str -set remex01_0.set

method dt.max=1.0e+6
solve radiation tstop=1e1

method dt.min=5.0e0
solve radiation tstop=1e2

method dt.min=5.0e1
solve radiation tstop=1e3

method dt.min=5.0e2
solve radiation tstop=1e4

method dt.min=5.0e3
solve radiation tstop=1e5
```

```
method dt.min=5.0e4
solve radiation tstop=1e6

save outfile=remex01_1.str


go atlas
mesh
x.m l=0 sp=0.5
x.m l=1 sp=0.5

y.m l=0 sp=0.001
y.m l=0.1 sp=0.001
y.m l=0.11 sp=0.001
y.m l=1 sp=0.1

region num=1 oxide y.max=0.1
region num=2 silicon y.min=0.1

electrode name=gate top
electrode name=substrate bottom

doping material=silicon p.type conc=1e17 unif

material material=oxide semiconductor
material material=oxide nc300=1e19 nv300=1e19 eg300=9 mun=20 mup=1.0e-6

dispersive material=oxide num.don=10 n0.don=1e16 beta.don=0.055 \
           sigma.don=1e-15 elim1.don=0.1 elim2.don=0.73

radiation gaussian gmax=1.5e23 tpeak=2.0e-5 sigmat=1e-5 y.max=0.005

method newton
solve init
solve prev
```

```
solve name=gate vstep=1.0 vstop=1.0

log outfile=remex01_1.log

method dt.max=1.0e-6
solve radiation tstep=1.0e-11 tstop=1.0e-4

method dt.max=1.0e+6
solve radiation tstop=1e1

method dt.min=5.0e0
solve radiation tstop=1e2

method dt.min=5.0e1
solve radiation tstop=1e3

method dt.min=5.0e2
solve radiation tstop=1e4

method dt.min=5.0e3
solve radiation tstop=1e5

method dt.min=5.0e4
solve radiation tstop=1e6

save outfile=remex01_2.str

go atlas
mesh
x.m l=0 sp=0.5
x.m l=1 sp=0.5

y.m l=0 sp=0.001
y.m l=0.1 sp=0.001
y.m l=0.11 sp=0.001
y.m l=1 sp=0.1
```

```
region num=1 oxide y.max=0.1
region num=2 silicon y.min=0.1

electrode name=gate top
electrode name=substrate bottom

doping material=silicon p.type conc=1e17 unif

material material=oxide semiconductor
material material=oxide nc300=1e19 nv300=1e19 eg300=9 mun=20 mup=1.0e-6

dispersive material=oxide num.don=10 n0.don=1e16 beta.don=0.055 \
           sigma.don=1e-15 elim1.don=0.1 elim2.don=0.73

radiation gaussian doserate=3e9 tpeak=2.0e-5 sigmat=1e-5 y.max=0.005

method newton
solve init
solve prev
solve name=gate vstep=1.0 vstop=1.0

log outfile=remex01_2.log

method dt.max=1.0e-6
solve radiation tstep=1.0e-11 tstop=1.0e-4

method dt.max=1.0e+6
solve radiation tstop=1e1

method dt.min=5.0e0
solve radiation tstop=1e2

method dt.min=5.0e1
solve radiation tstop=1e3
```

```
method dt.min=5.0e2
solve radiation tstop=1e4

method dt.min=5.0e3
solve radiation tstop=1e5

method dt.min=5.0e4
solve radiation tstop=1e6

save outfile=remex01_3.str

tonyplot -overlay remex01_0.log remex01_2.log -set remex01_1.set
tonyplot -overlay remex01_0.log remex01_1.log remex01_2.log -set remex-
01_2.set

quit
```

remex02.in: Irradiation of a Silicon NMOS

Requires: SSuprem4/S-Pisces/REM

Basic MOS Athena to Atlas example simulating an Ig/time curve during irradiation. This example demonstrates:

- process simulation of a MOS transistor in Athena,
- device simulation during irradiation with and without dispersive transport.

The process simulation in SSuprem4 follows a standard LDD MOS process. The process steps are simplified and default models are used to give a fast runtime. The polysilicon gate is formed by a simple geometrical etch. Before this point, the simulation is essentially one dimensional and hence is run in Athena's 1D mode. After the poly etch, the structure converts to 2D and the electrodes are defined.

In Atlas, the first task is to define the oxide as a semiconductor using `material material=oxide semiconductor`. The `models` and `material` parameters for the simulation are then defined. The `contact` statement defines the workfunction of the gate electrodes. The **`INTERFACE`** statement defines the fixed charge at the silicon/oxide interface. For simple MOS simulation, the `CVT` and `SRH` parameters define the recommended models. `CVT` sets a general purpose mobility model including concentration, temperature, parallel field and transverse field dependence.

The **`RADIATION`** statement is used to specify the dose rate applied to the device. A Gaussian pulse is used with a maximum dose rate (`doserate`) of 1e10 rad/s, peak time (`tpeak`) of 1e-5s and a standard deviation (`sigmat`) of 2e-6s.

The **`SOLVE`** `init` statement is used to solve the thermal equilibrium case. After this, the voltages can be ramped. A sequence of **`SOLVE`** statements is set to ramp the DC gate bias with the drain voltage at 0.1V. Solutions are obtained at 0.25V intervals up to 3.0V.

A transient simulation is performed starting from 1e-6s and finishing at 1e6s. The terminal characteristics saved to the file `remex02_0.log` as specified in the **LOG** statement. The `radiation` parameter is specified on each **SOLVE** statement where there is to be irradiation. The mininum and maximum time steps are carefully controlled by specifying `dt.min` and `dt.max` on the `method` statement.

Once this simulation is complete, a second Atlas simulation is performed. This simulation is identical to the first, except that the **DISPERSIVE** statement is used to specify the dispersive transport parameters. The transient terminal characteristics are stored in the log file `remex02_1.log`.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DeckBuild, select the **Run** button to execute the example.

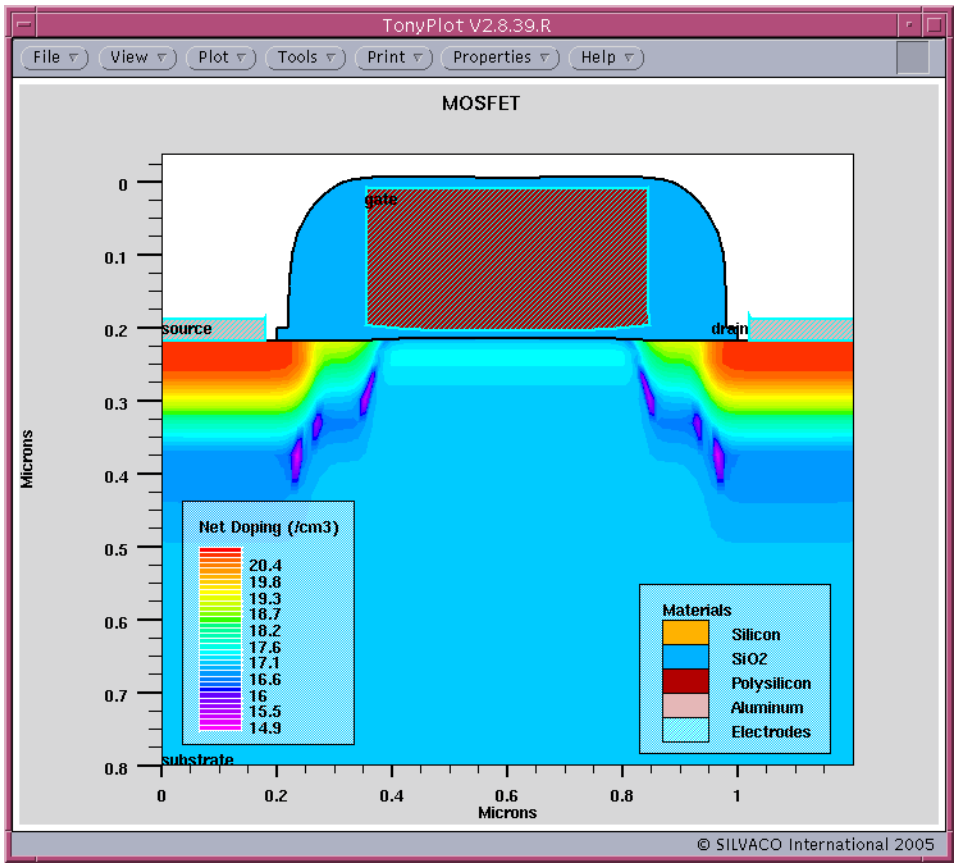


Figure 4-9: Figure NMOSFET

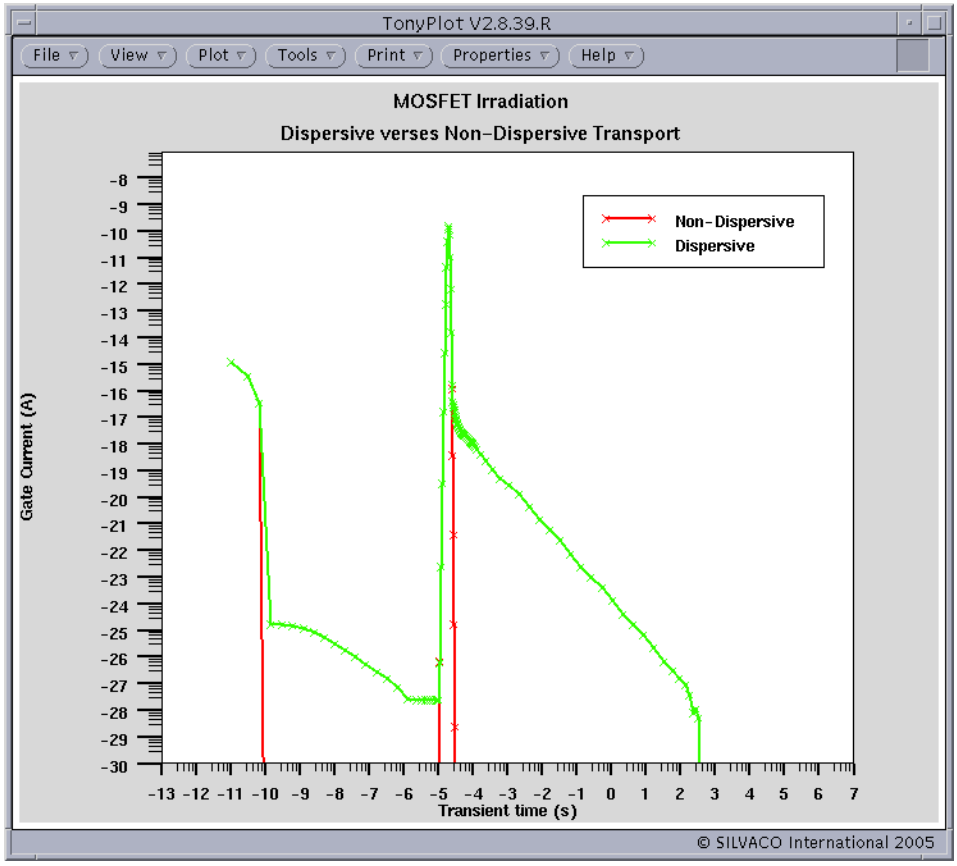


Figure 4-10: Effects of dispersive transport on Gate current

remex02.in Input File

```
go athena

#
line x loc=0.0 spac=0.1
line x loc=0.2 spac=0.006
line x loc=0.4 spac=0.006
line x loc=0.6 spac=0.01
#
line y loc=0.0 spac=0.002
line y loc=0.2 spac=0.005
line y loc=0.5 spac=0.05
line y loc=0.8 spac=0.15
#
init orientation=100 c.phos=1e14 space.mul=2
```

```
#pwell formation including masking off of the nwell
#
diffus time=30 temp=1000 dryo2 press=1.00 hcl=3
#
etch oxide thick=0.02
#
#P-well Implant
#
implant boron dose=8e12 energy=100 pears

#
diffus temp=950 time=100 weto2 hcl=3
#
#N-well implant not shown -
#
# welldrive starts here
diffus time=50 temp=1000 t.rate=4.000 dryo2 press=0.10 hcl=3
#
diffus time=220 temp=1200 nitro press=1
#
diffus time=90 temp=1200 t.rate=-4.444 nitro press=1
#
etch oxide all
#
#sacrificial "cleaning" oxide
diffus time=20 temp=1000 dryo2 press=1 hcl=3
#
etch oxide all
#
#gate oxide grown here:-
diffus time=11 temp=925 dryo2 press=1.00 hcl=3
#
# Extract a design parameter
extract name="gateox" thickness oxide mat.occno=1 x.val=0.05
```

```
#
#vt adjust implant
implant boron dose=9.5e11 energy=10 pearson

#
depo poly thick=0.2 divi=10
#
#from now on the situation is 2-D
#
etch poly left p1.x=0.35
#
method fermi compress
diffuse time=3 temp=900 weto2 press=1.0
#
implant phosphor dose=3.0e13 energy=20 pearson
#
depo oxide thick=0.120 divisions=8
#
etch oxide dry thick=0.120
#
implant arsenic dose=5.0e15 energy=50 pearson
#
method fermi compress
diffuse time=1 temp=900 nitro press=1.0
#

# pattern s/d contact metal
etch oxide left p1.x=0.2
deposit alumin thick=0.03 divi=2
etch alumin right p1.x=0.18

# Extract design parameters

# extract final S/D Xj
extract name="nxj" xj silicon mat.occno=1 x.val=0.1 junc.occno=1
```

```
# extract the N++ regions sheet resistance
extract name="n++ sheet rho" sheet.res material="Silicon" mat.occno=1
x.val=0.05 region.occno=1

# extract the sheet rho under the spacer, of the LDD region
extract name="ldd sheet rho" sheet.res material="Silicon" \
mat.occno=1 x.val=0.3 region.occno=1

# extract the surface conc under the channel.
extract name="chan surf conc" surf.conc impurity="Net Doping" \
material="Silicon" mat.occno=1 x.val=0.45

# extract a curve of conductance versus bias.
extract start material="Polysilicon" mat.occno=1 \
bias=0.0 bias.step=0.2 bias.stop=2 x.val=0.45
extract done name="sheet cond v bias" \
curve(bias,1dn.conduct material="Silicon" mat.occno=1  region.occno=1)\
outfile="extract.dat"

# extract the long chan Vt
extract name="n1dvt" 1dvt ntype vb=0.0 qss=1e10 x.val=0.49



structure mirror right

electrode name=gate x=0.5 y=0.1
electrode name=source x=0.1
electrode name=drain x=1.1
electrode name=substrate backside

structure outfile=remex02_0.str
tonyplot remex02_0.str -set remex02_0.set

go atlas
mesh infile=remex02_0.str
```

```
material material=oxide semiconductor
material material=oxide nc300=1e19 nv300=1e19 eg300=9 mun=20 mup=1.0e-6

contact name=gate n.poly

radiation gaussian doserate=1e10 tpeak=2.0e-5 sigmat=1e-6

interface qf=3e10

# set material models
models cvt srh print

method newton
solve init

# Bias the drain
solve vdrain=0.1

# Ramp the gate
solve vgate=0 vstep=0.25 vfinal=3.0 name=gate

method dt.max=1.0e-6
log outf=remex02_0.log
solve radiation tstep=1.0e-11 tstop=1.0e-4
save outfile=remex02_1.str

method dt.max=1.0e2
solve radiation tstop=1e4
save outfile=remex02_2.str

method dt.min=1.0e3
solve radiation tstop=1e5
save outfile=remex02_3.str

method dt.min=1.0e4
solve radiation tstop=1e6
```

```
save outfile=remex02_4.str

go atlas
mesh infile=remex02_0.str

material material=oxide semiconductor
material material=oxide nc300=1e19 nv300=1e19 eg300=9 mun=20 mup=1.0e-6

dispersive num.don=10 n0.don=1e16 beta.don=0.055 sigma.don=1e-15 \
           elim1.don=0.1 elim2.don=0.73 file.don=remex02.dat

contact name=gate n.poly

radiation gaussian doserate=1e10 tpeak=2.0e-5 sigmat=1e-6

interface qf=3e10

# set material models
models cvt srh print

method newton
solve init

# Bias the drain
solve vdrain=0.1

# Ramp the gate
solve vgate=0 vstep=0.25 vfinal=3.0 name=gate

method dt.max=1.0e-6

log outf=remex02_1.log master
solve radiation tstep=1.0e-11 tstop=1.0e-4
save outfile=remex02_5.str

method dt.max=1.0e2
```

```
solve radiation tstop=1e4
save outfile=remex02_6.str

method dt.min=1.0e3
solve radiation tstop=1e5
save outfile=remex02_7.str

method dt.min=1.0e4
solve radiation tstop=1e6
save outfile=remex02_8.str

tonyplot -overlay remex02_0.log remex02_1.log -set remex02_1.set

quit
```

4.2.8 Displacement Damage (DD)

Displacement Defect Production

Fluence (particles/area) of energetic particles incident on a solid lose their energy by ionization and non-ionization processes. The result of ionization processes is the creation of electron-hole pairs; the result of non-ionization processes is displacement damage. The collision between a particle with mass m with energy E and an atom, the primary Knock-on Atom (PKA) with mass M will have a maximum recoil energy R_{MAX} transferred to the PKA defined by

$$R_{MAX} = [(4 \cdot (mM)) / (m + M)^2] \cdot E \quad \text{4-114}$$

The energy of the PKA, E_{PKA}, lies between $0 < E_{PKA} < R_{MAX}$ and is described by a recoil energy spectrum.

Displacement Damage

When energetic particles collide with semiconductor lattice, the atoms may become dislodged from their lattice site and pushed into interstitial positions within the crystal. The former lattice site of this displaced atom is now called a vacancy. The displaced atom is called an interstitial (an atom not in a normal lattice position) and the interstitial-vacancy pair is called a Frenkel pair. If the energetic particle has a sufficiently large amount of energy, it may impart energy to the displaced atom to cause it to dislodge from other lattice site atoms, and then those atoms to dislodge from other lattice atoms. This cascade of collisions caused by energetic particle irradiation can result in a large disordered region called a defect cluster.

Displacement Damage is a complex function of particle type, particle energy, Flux/Dose Rate, pulsed or continuous irradiation, and Fluence.

Over time, all displaced atoms lose enough energy to acheive a thermal equilibrium with the lattice. Some atoms might slip back into the vacancy to reconstitute the local lattice structure, while other atoms might combine with dopant atoms to produce stable defects. Vacancies are

mobile at room temperature and can combine with other vacancies (a di-vacancy) with impunity atoms or donor atoms. This defect re-ordering is called annealing. Annealing can reverse the effect of the damage to the lattice by particle irradiation (forward annealing) but can also lead to more stable and effective defects (reverse annealing).

The general result from energetic particle irradiation is the production of defects within the band-gap.

Displacement Damage Terms

The observation that bulk damage is proportional to the total kinetic energy imparted to displaced lattice atoms is sometimes called Non-Ionizing Energy Loss (NIEL). NIEL units are *MeV-cm*^{2}*/g* or *keV-cm*^{2}*/g*. Another term often seen is Kinetic Energy Released in Matter (KERMA) with units in *MeV* or *keV*. The relationship between KERMA and NIEL is

$$KERMA(MeV) = NIEL(MeV-cm^2/g) \times \Phi(\#/cm2) \times mass(g) \quad \text{4-115}$$

where $\Phi(\#cm2)$ is the incident Fluence and *mass*(*g*) is the mass in gram of the irradiated sample.

Displacement Damage Cross Section (*D*) is sometimes used, with units of *MeV-mb*. Remembering a barn (*b*) is 1×10^{-24} cm, KERMA can be derived as

$$KERMA(MeV) = D(MeV-mb) \times \Phi(\#/cm2) \times (\#\ \text{Atoms irradiated}) \times (10^{-27} cm^2/mb) \quad \text{4-116}$$

NIEL can also be derived from *D* as

$$NIEL(MeV-cm^2/g) = D(MeV-mb) \times (10^{-27} cm^2/mb) \times (1/\text{Mole Atomic Mass}) \times (6.022 \times 10^{23}\text{Atoms/Mole}) \quad \text{4-117}$$

4.2.9 NIEL Values

Silicon

Proton NIEL Values

Proton Energy (MeV)	NIEL (MeV-cm^2/g)
$0.001 \le E < 0.01$	5.0
$0.01 \le E < 0.1$	1.0
$0.1 \le E < 1$	1.0×10^{-1}
$1 \le E < 10$	5.0×10^{-2}
$10 \le E < 100$	5.0×10^{-3}
$100 \le E < 1000$	1.5×10^{-3}

Neutron NIEL Values

Neutron Energy (MeV)	NIEL (MeV-cm^2/g)
$0.1 \le E < 0.2$	1.0×10^{-4}

Neutron NIEL Values

$0.3 \leq E < 3$	2.0×10^{-3}
$3 \leq E < 100$	5.0×10^{-3}
$100 \leq E < 1000$	1.5×10^{-3}

GaAs

Proton NIEL Values

Proton Energy (MeV)	NIEL (MeV-cm^2/g)
$0.001 \leq E < 0.01$	5.0
$0.01 \leq E < 0.1$	1.0
$0.1 \leq E < 1$	1.0×10^{-1}
$1 \leq E < 10$	1.0×10^{-2}
$10 \leq E < 1000$	5.0×10^{-3}

4.2.10 Radiation Fluence Model

The radiation fluence model allows the simulation of the defect dislocation generation rate due to energetic particle bombardment in the semiconductor. The total defect density of states (N_T) due to a radiation fluence with a specific irradiation energy and species is given by:

$$N_T = \alpha_D \ \rho \ E_L \ \mathtt{FLUENCE} \qquad 4\text{-}118$$

where:

- α_D is the damage factor (DAM.ALPHA, DAM.ELECTRON, DAM.ION, DAM.NEUTRON, DAM.PHOTON, DAM.PROTON or DAM.USER on the **MATERIAL** statement).
- ρ is the density of the material (DAM.DENSITY on the **MATERIAL** statement).
- E_L is the non-ionizing energy loss (*NIEL*) and is calculated using tables in the Atlas common directory or by specifying DAM.NIEL on the **MATERIAL** statement.
- FLUENCE is the total radiation flux (FLUENCE on the **RADIATION** or **SOLVE** statements).

Traps, as defined on the **DEFECTS** statement, are then created with a density of states of $\frac{N_T}{\mathtt{NUMA}}$ for acceptor traps and $\frac{N_T}{\mathtt{NUMD}}$ for donor traps.

To enable the model, specify a value of the fluence using the FLUENCE parameter on the **RADIATION** statement. Also, specify the irridiation energy (ENERGY parameter on the **RADIATION** statement) and species (by specifying either PROTON, ELECTRON, NEUTRON, ALPHA, ION, PHOTON or USER on the **RADIATION** statement). You should specify the logical parameter FLUENCE.MODEL on any **DEFECTS** statement where the density of states is to be calculated from the fluence.

The acceptor and donor lifetimes should be specified using SIGTAE, SIGTAH, SIGTDE, and SIGTDH. See Chapter 15 "TFT: Thin-Film Transistor Simulator" for more information of **DEFECTS**.

Table 4-19 Radiation Fluence Parameters				
Statement	**Parameter**	**Type**	**Default**	**Units**
DEFECTS	NUMA	Real	10	
DEFECTS	NUMD	Real	10	
DEFECTS	SIGTAE	Real	0	cm^2
DEFECTS	SIGTAH	Real	0	cm^2
DEFECTS	SIGTDE	Real	0	cm^2
DEFECTS	SIGTDH	Real	0	cm^2
DEFECTS	FLUENCE.MODEL	Logical	False	
MATERIAL	DAM.ALPHA	Logical	False	
MATERIAL	DAM.DENSITY	Real	0	cm^{-3}
MATERIAL	DAM.ELECTRON	Logical	False	
MATERIAL	DAM.ION	Logical	False	
MATERIAL	DAM.NEIL	Real	0	$MeVcm^2gm^{-1}$
MATERIAL	DAM.NEUTRON	Logical	False	
MATERIAL	DAM.PHOTON	Logical	False	
MATERIAL	DAM.PROTON	Logical	False	
MATERIAL	DAM.USER	Logical	False	
RADIATION	ALPHA	Logical	False	
RADIATION	FLUENCE	Real	0	cm^{-2}
RADIATION	ENERGY	Real	0	MeV
RADIATION	PROTON	Logical	False	
RADIATION	ELECTRON	Logical	False	
RADIATION	NEUTRON	Logical	False	
RADIATION	ION	Logical	False	
RADIATION	PHOTON	Logical	False	
RADIATION	USER	Logical	False	
SOLVE	FLUENCE	Real	0	cm^{-2}

Description

ALPHA	Specifies that the irradiation species are alpha particles.
DAM.ALPHA	Specifies the damage factor for an alpha particle.
DAM.DENSITY	Specifies the damage density of the material.
DAM.ELECTRON	Specifies the damage factor for an electron.
DAM.ION	Specifies the damage factor for an ion.

DAM.NEIL	Specifies the non-ionizing energy loss (NIEL) for the material. If this is not specified, then the NIEL table values will be used. Additional NIEL tables should be located in the ATLAS_COMMON_DIR directory and have the name <Material>.niel, where <Material> is the material name (e.g., Silicon or GaN). The table should be in the form below.
	`Energy (MeV)  NIEL`
	For example:
	`4`
	`1.8  3.1`
	`15  0.27`
	`40  0.1`
	`105  0.05`

DAM.NEUTRON	Specifies the damage factor for a neutron.
DAM.PHOTON	Specifies the damage factor for a photon.
DAM.PROTON	Specifies the damage factor for a proton.
DAM.USER	Specifies the damage factor for a user defined particle.
ELECTRON	Specifies that the irradiation species are electrons.
ENERGY	Specifies the irradiation energy.
FLUENCE	Specifies the initial irradiation fluence.
FLUENCE.MODEL	Specifies the the fluence model is used to calculate the acceptor and donor density of states.
ION	Specifies that the irradiation species are ions.
NEUTRON	Specifies that the irradiation species are neutrons.
NUMA	Specifies the number of acceptor levels to be used.
NUMD	Specifies the number of donor levels to be used.
PHOTON	Specifies that the irradiation species are photons.
PROTON	Specifies that the irradiation species are protons.

SIGTAE	Specifies the electron capture cross-section for the acceptor traps.
SIGTAH	Specifies the hole capture cross-section for the acceptor traps.
SIGTDE	Specifies the electron capture cross-section for the donor traps.
SIGTDH	Specifies the hole capture cross-section for the donor traps.
USER	Specifies that the irradiation species are user defined.

remex03.in: Effects of Defect Dislocation Damage on a GaAs MESFET

Requires: FLASH/Blaze/REM

This example demonstrates fabrication and electrical analysis of a MESFET structure using the FLASH module of Athena and the Blaze capability of Atlas. The example uses DevEdit at various points in the process to optimize the grid. The example shows:

- MESFET fabrication using FLASH,
- re-meshing of structure in DevEdit,
- fluence dependent defects definition,
- electrical simulation of Id/Vgs curves at different fluence levels.

This example starts by interfacing Athena and DevEdit to provide a GaAs MESFET structure using silicon and beryllium implants. Details of using FLASH are found in the ATHENA_FLASH examples section.

The **CONTACT** statement sets the gate workfunction for the Schottky contact. The low recombination lifetimes typical of GaAs are set in the **MATERIAL** statement. The **MODELS** statement is used to specify appropriate models within the simulation. The `fldmob` parameter turns on the electric field dependent mobility. `conmob` specifies the concentration dependent mobility. These values are taken from a look-up table and exist only for room temperature.

The solution sequence for Id/Vgs curves is first to obtain the initial solution at zero bias on all contacts. The drain is set to 0.1V and the gate is ramped to 0.5V. The log file `remex03_0.log` is opened and the gate voltage is swept from 0.5V to -2V.

The second Atlas simulation uses the `radiation` statement to define the initial fluence level, species and energy while the **MATERIAL** statement specifies the damage factor and `NIEL` value. The `fluence.model` parameter is specified on the **DEFECTS** statement so that the trap densities of state are calculated using the damage model. The same sequence of **SOLVE** statements is used to generate the Id/Vgs curve and the results are stored in the log file `remex03_11.log`.

Once the gate has reached -2V, the log file is closed using the **LOG** `off` command. The fluence level is changed and the thermal equilibrium solution is found using **SOLVE** `init fluence=1e12`. The drain is again set to 0.1V and the gate varied from 0.5V to -2V. The results are stored in `remex03_12.log`. The sequence is repeated with a new fluence level and the results stored in `remex03_13.log`.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DeckBuild, select the **Run** button to execute the example.

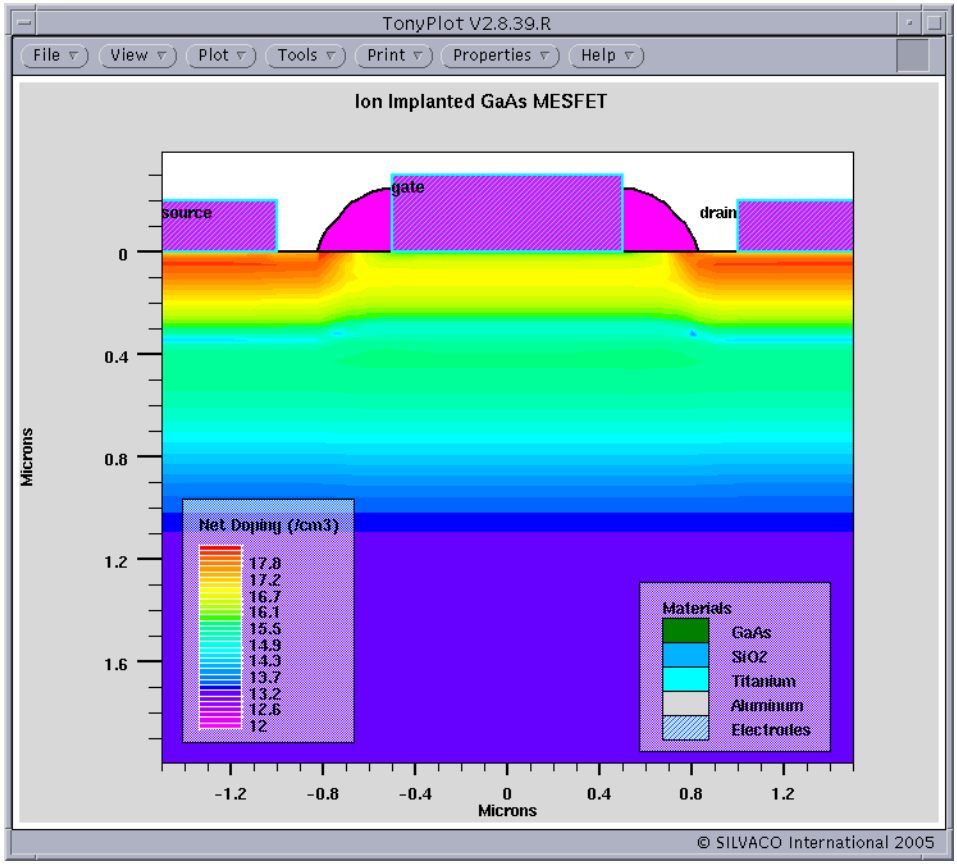


Figure 4-11: MESFET

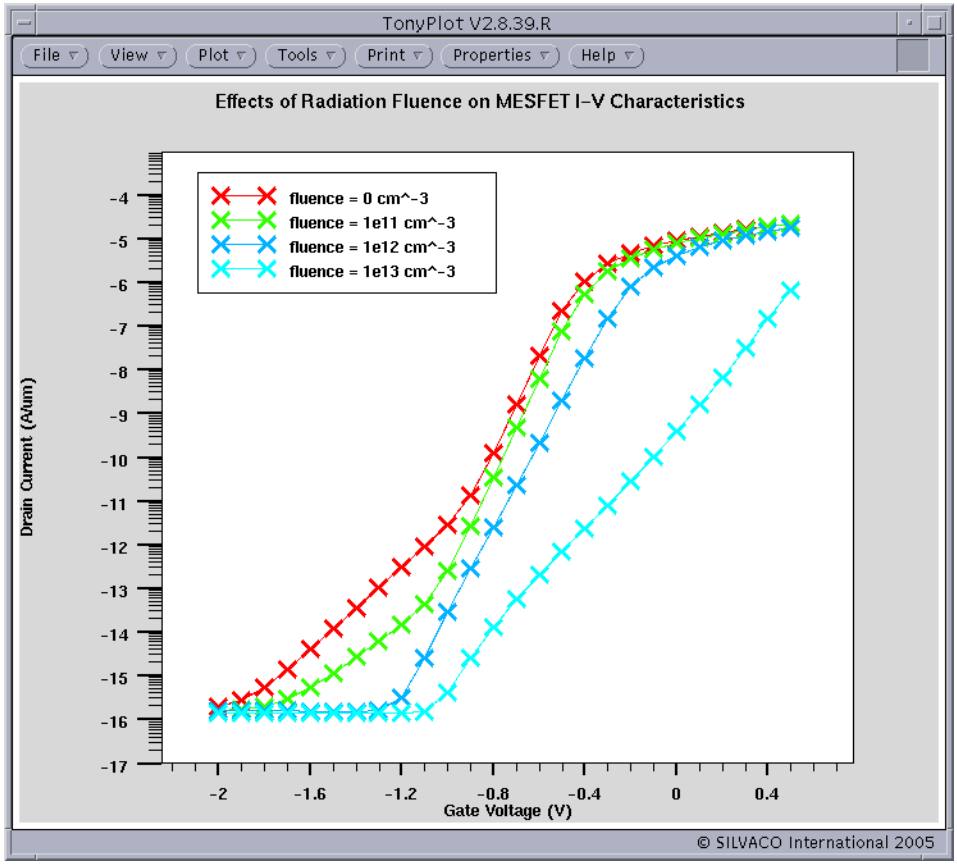


Figure 4-12: Effects of dislocation damage on MESFET I-V characteristics

remex03.in Input File

```
go athena

# GaAs MESFET fabrication and analysis using DevEdit

line x loc=-1.5 spac=0.2
line x loc=-.7  spac=0.1
line x loc=-.5  spac=0.05
line x loc=0.0  spac=0.1
line x loc=0.5  spac=0.05
line x loc=0.7  spac=0.1
line x loc=1.5  spac=0.2
#
line y loc=0.00 spac=0.02
line y loc=2.00 spac=0.5
```

```
#
init gaas c.beryllium=1.0e13 orient=100 space.mult=1

implant beryllium energy=100 dose=2e11
implant silicon energy=100 dose=1e12

diffus time=10 temp=850

# deposit and pattern gate metal
deposit titanium thick=.3 divisions=10
etch titanium right p1.x=0.5
etch titanium left  p1.x=-0.5

deposit oxide thick=0.35 divisions=8

etch oxide thick=.4

# regrid before implant
go devedit

base.mesh height=0.08 width=0.08
bound.cond apply=false max.ratio=300
bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straightening=2 \
        align.points
constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
        min.height=0.0001 min.width=0.0001
constr.mesh type=Semiconductor default
constr.mesh type=Insulator default
constr.mesh type=Metal default


imp.refine min.spacing=0.02
imp.refine imp="net doping" sensitivity=1
mesh

go athena
```

```
# perform source/drain implant
implant silicon energy=50 dose=1e13

# regrid to reduce grid in un-implanted areas
go devedit

base.mesh height=0.4 width=0.4
bound.cond apply=false max.ratio=300
bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straightening=2 \
        align.points
constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
        min.height=0.0001 min.width=0.0001
constr.mesh type=Semiconductor default
constr.mesh type=Insulator default max.angle=180
constr.mesh type=Metal default

imp.refine min.spacing=0.03
imp.refine imp="net doping" sensitivity=.5
mesh


go athena

diff time=10 temp=850

# deposit ohmic metal
deposit aluminum thick=.2 divisions=4

etch aluminum        start x=-1 y=10
etch cont                  x=-1 y=-10
etch cont                  x=1 y=-10
etch done                  x=1 y=10
#
electrode name=source x=-1.4
electrode name=drain x=1.4
```

```
electrode name=gate x=0.0

# regrid to resolve new junction position
go devedit

base.mesh height=0.1 width=0.1
bound.cond apply=false max.ratio=300
bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straightening=2 \
        align.points
constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
        min.height=0.0001 min.width=0.0001
constr.mesh type=Semiconductor default
constr.mesh type=Insulator default max.angle=180
constr.mesh type=Metal default

imp.refine min.spacing=0.03
imp.refine imp="net doping" sensitivity=1
mesh

structure outfile=remex03_0.str

tonyplot remex03_0.str -set remex03_0.set

go atlas
mesh infile=remex03_0.str

# set work function for gate
contact name=gate work=4.87

# specify lifetimes in GaAs and models
material material=GaAS taun0=1.e-8 taup0=1.e-8
models conmob fldmob srh optr print

# Begin solution with no fluence
solve init
solve vdrain=0.1
```

```
solve vstep=0.1 vfinal=0.5 name=gate

# Ramp gate and log results
log outfile=remex03_0.log master
solve vgate=0.5 vstep=-0.1 vfinal=-2 name=gate

extract init infile="remex03_0.log"
extract name="vt" (xintercept(maxslope(curve((v."gate"),(i."drain"))))) \
        datafile="remex03_vt.dat"

save outfile=remex03_1.str

go atlas
mesh infile=remex03_0.str

radiation proton fluence=1e11 energy=1.8
material dam.proton=1e3 dam.niel=3.1
defects fluence.model \
  sigtae=1.e-17 sigtah=1.e-15 sigtde=1.e-15 sigtdh=1.e-17  \
  siggae=2.e-16 siggah=2.e-15 siggde=2.e-15 siggdh=2.e-16

# set work function for gate
contact name=gate work=4.87

# specify lifetimes in GaAs and models
material material=GaAS taun0=1.e-8 taup0=1.e-8
models conmob fldmob srh optr print

# Begin solution with fluence=1e11
solve init
solve vdrain=0.1
solve vstep=0.1 vfinal=0.5 name=gate

# Ramp gate and log results
log outfile=remex03_11.log master
solve vgate=0.5 vstep=-0.1 vfinal=-2 name=gate
```

```
save outfile=remex03_11.str
log off

extract init infile="remex03_11.log"
extract name="vt" (xintercept(maxslope(curve((v."gate"),(i."drain"))))) \
        datafile="remex03_vt.dat"



# Begin solution with fluence=1e12
solve init fluence=1e12
solve vdrain=0.1
solve vstep=0.1 vfinal=0.5 name=gate

# Ramp gate and log results
log outfile=remex03_12.log master
solve vgate=0.5 vstep=-0.1 vfinal=-2 name=gate
save outfile=remex03_12.str
log off

extract init infile="remex03_12.log"
extract name="vt" (xintercept(maxslope(curve((v."gate"),(i."drain"))))) \
        datafile="remex03_vt.dat"



# Begin solution with fluence=1e13
solve init fluence=1e13
solve vdrain=0.1
solve vstep=0.1 vfinal=0.5 name=gate

# Ramp gate and log results
log outfile=remex03_13.log master
solve vgate=0.5 vstep=-0.1 vfinal=-2 name=gate
save outfile=remex03_13.str

extract init infile="remex03_13.log"
```

```
extract name="vt" (xintercept(maxslope(curve((v."gate"),(i."drain"))))) \
        datafile="remex03_vt.dat"


tonyplot -overlay remex03_0.log remex03_11.log remex03_12.log remex03_13.log
-set remex03_1.set
quit
```

Chapter 5
S-Pisces: Silicon Based 2D Simulator

5.1 Overview

S-Pisces is a powerful and accurate two-dimensional device modeling program that simulates the electrical characteristics of silicon based semiconductor devices including MOS, bipolar, SOI, EEPROM, and power device technologies.

S-Pisces calculates the internal distributions of physical parameters and predicts the electrical behavior of devices under either steady-state, transient, or small signal AC conditions. This is performed by solving Poisson's Equation (see Section 3.1.1 "Poisson's Equation") and the electron and hole carrier continuity equations in two dimensions (see Section 3.1.2 "Carrier Continuity Equations").

S-Pisces solves basic semiconductor equations on non-uniform triangular grids. The structure of the simulated device can be completely arbitrary. Doping profiles and the structure of the device may be obtained from analytical functions, experimentally measured data, or from process modeling programs SSuprem3 and Athena.

For more information on semiconductor equations, see Section 3.1 "Basic Semiconductor Equations".

You should be familiar with Atlas before reading this chapter any further. If not, read Chapter 2 "Getting Started with Atlas".

5.2 Simulating Silicon Devices Using S-PISCES

5.2.1 Simulating MOS Technologies

Physical Models for MOSFETs

S-Pisces provides special physical models tuned for simulating MOS devices. Most of these models are accessed from the **MODELS** statement. The MOS parameter of the **MODELS** statement can be specified to turn on a default set of physical models that are most useful for MOS simulation. The MOS parameter enables Shockley-Read-Hall (SRH), Fermi Statistics (FERMI), and the Lombardi Mobility model (CVT) for transverse field dependence. To set the default MOS simulation models , use:

```
MODEL MOS PRINT
```

The transverse field dependent mobility models are of particular importance for simulating MOS devices. S-Pisces currently supports several different transverse field dependent mobility models. The CVT parameter selects the Lombardi CVT model. The YAMA parameter selects the Yamaguchi model. The TASCH parameter selects the Tasch model. The WATT parameter selects the Watt Surface Model, which can be operated in a more accurate mode with the extra parameter, MOD.WATT, on the **MOBILITY** statement.

You will find that the **MOBILITY** statement can be used to modify some of the parameters of the various models, to apply different models to different regions, or to apply different models to electrons and holes.

Meshing for MOS Devices

In device simulation of MOS devices, the key areas for a tight mesh are:

- very small vertical mesh spacing in the channel under the gate. The exact size of mesh required depends on the transverse field or surface mobility model chosen. See Figure 5-1 for an example of the effect of mesh size on drain current.
- lateral mesh spacing along the length of the channel for deep sub-micron devices. This is required to get the correct source-drain resistance and to resolve the channel pinch-off point.
- lateral mesh at the drain/channel junction for breakdown simulations. This is required to resolve the peak of electric field, carrier temperature and impact ionization.
- several vertical grid spacings inside the gate oxide when simulating gate field effects such as gate induced drain leakage (GIDL) or using any hot electron or tunneling gate current models

Figures 5-1 and 5-2 show the effect of mesh size in the MOS channel on IV curves. In Figure 5-1, the mesh density in the vertical direction is increased. As the mesh density increases, the resolution of the electric field and carrier concentration is improved. This example uses the CVT mobility model. Improvements in transverse electric field resolution lead to a reduced mobility in the channel and a stronger IV roll-off.

But Figure 5-2 shows the effect of surface channel mesh in MOSFETs is model dependent. This result shows the current at Vds=3.0V and Vgs=0.1V versus the size of the first grid division into the silicon. Results vary for each model but note that for all models, a very fine grid is required to reduce the grid dependence to acceptable levels.

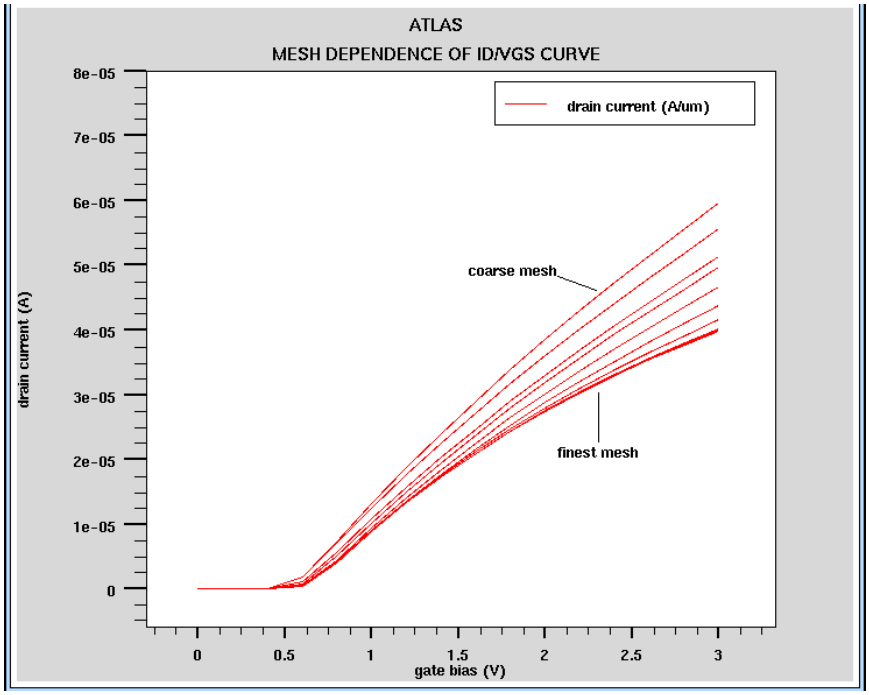


Figure 5-1: Effect on MOS IV curve of progressive refinement of the vertical mesh spacing at the surface of the MOS channel

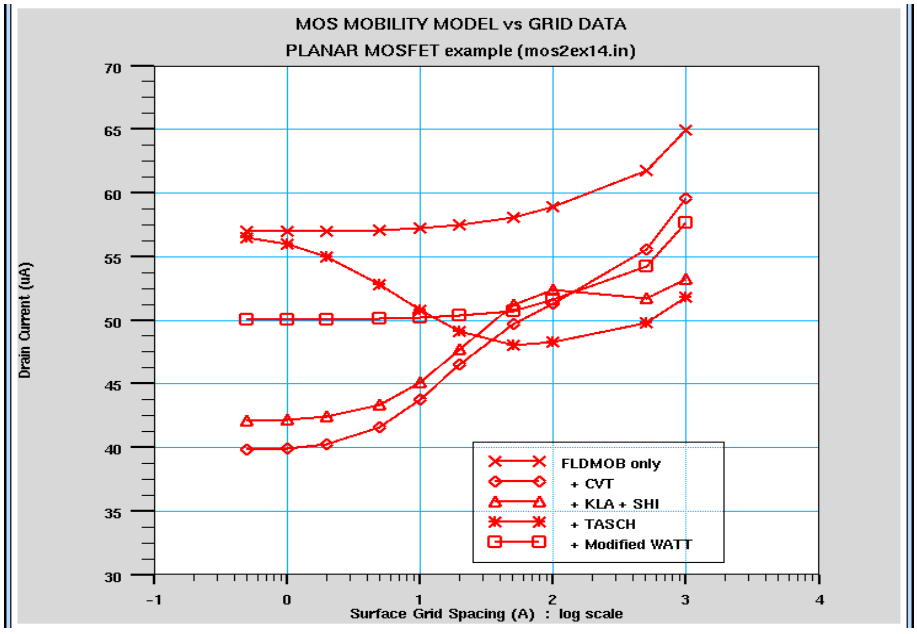


Figure 5-2: Effect of surface mesh spacing on simulated current for several MOS Mobility Models

MOS Electrode Naming

For MOS simulation, S-Pisces allows you to use standard electrode names to reduce confusion with the use of electrode indices. These names include: `source`, `drain`, `gate`, and `substrate`. Electrode names can be defined in Athena or DevEdit or in the **ELECTRODE** statement in Atlas. These names can be used in the **SOLVE** statements for setting bias voltages such as:

```
SOLVE VGATE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=GATE
```

Gate Workfunction

In MOS simulations, the workfunction of the gate is an important parameter. This must be set in each input deck using the WORK parameter of the **CONTACT** statement. For example:

```
CONTACT NAME=GATE WORK=4.17
```

would set the workfunction on the gate at 4.17 eV.

Certain material names can also be used to set the workfunction of common gate materials. For example:

```
CONTACT NAME=GATE N.POLY
```

would set the workfunction on the gate to that of n type polysilicon.

Note: The gate workfunction should be set on a **CONTACT** statement even though the material or workfunction might be set from Athena or DevEdit.

Interface Charge

For accurate simulation of MOS devices, the interface charge at the oxide, specify semiconductor interface. To do this, set the QF parameters for the **INTERFACE** statement. Typically, a value of 3×10^{10} cm^{-2} is representative for the interface charge found in silicon MOS devices. The proper syntax for setting the value for this interface fixed charge is:

```
INTERFACE QF=3e10
```

You can also try to model the fixed charge more directly by using interface traps to simulate the surface states. To do this, use the **INTTRAP** statement. But this is rarely done in practice.

Single Carrier Solutions

Frequently for MOS simulation, you can choose to simulate only the majority carrier. This will significantly speed up simulations where minority carrier effects can be neglected. This can be done by turning off the minority carrier. To do this, use the Atlas negation character, ^ , and one of the carrier parameters (ELECTRONS or HOLES) in the **METHOD** statement. For example, to simulate electrons only, you can specify one of the following:

```
METHOD CARRIERS=1 ELECTRONS
```

```
METHOD ^HOLES
```

Single carrier solutions should not be performed where impact ionization, any recombination mechanism, lumped element boundaries, or small signal AC analysis are involved.

Energy Balance Solutions

As MOS devices become smaller and smaller, non-local carrier heating effects becomes important. You can perform accurate simulation of non-local carrier heating effects using the energy balance model (EBM). As a general rule, you can use the gate length as a gauge to predict when non-local effects are important. Generally, for drain currents, energy balance should be applied for gate lengths less than 0.5 microns. For substrate currents, energy balance should be applied for gate lengths less than 1.0 micron.

To enable energy balance for electrons/holes, set either the HCTE.EL or HCTE.HO parameters on the **MODELS** statement. For example:

```
MODEL HCTE.EL
```

enables the energy balance model for electrons.

ESD Simulation

In some cases, lattice heating may be important to MOS simulation. This typically occurs in cases with very high currents, just like the case with ESD simulation. In these cases, Giga should be used to simulate the heat-flow in the device. To enable heat flow simulation, set the LAT.TEMP parameter of the **MODELS** statement (a license for Giga is required). For example, the statement:

```
MODEL LAT.TEMP
```

enables heat-flow simulation.

5.2.2 Simulating Silicon Bipolar Devices

Physical Models for BJTs

S-Pisces provides special physical models for bipolar device simulation. You can select these models using the **MODELS** statement. The BIPOLAR parameter of the **MODELS** statement enables a reasonable default set of bipolar models. These include: concentration dependent mobility (CONMOB), field dependent mobility (FLDMOB), bandgap narrowing (BGN), concentration-dependent lifetime (CONSRH) and Auger recombination (AUGER). If LAT.TEMP is also specified on the **MODELS** statement, or the TEMPERATURE parameter differs from 300 Kelvin by more than 10 Kelvin, then the ANALYTIC model is used instead of CONMOB. This is because of the narrow valid temperature range for CONMOB.

For the most accurate bipolar simulations, the recommended mobility model is the Klaassen model (KLA). This includes doping, temperature and carrier dependence. It applies separate mobility expressions for majority and minority carriers. You should also use this model with Klaassen's Auger model (KLAAUG) and Klaassen's concentration dependent SRH model (KLASRH). Combine the mobility model with FLDMOB to model velocity saturation. For surface (or lateral) bipolar devices, you can use the Shirahata model (SHI) to extend the Klaassen model with a transverse electric field dependence. The most accurate and appropriate model statement for bipolar devices is therefore:

```
MODELS KLA FLDMOB KLASRH KLAAUG BGN FERMI PRINT
```

You can choose this set of models by using the BIPOLAR2 parameter from the MODEL statement.

Note: For a complete syntax including description of models and method for simulating polysilicon emitter bipolar devices, see the BJT directory in the on-line examples.

Meshing Issues for BJTs

The most important areas to resolve in bipolar transistors are the emitter/base and base/collector junctions. Typically, a very fine mesh throughout the base region is required. The gain of the bipolar device is determined primarily by the recombination at the emitter/base junction or inside the emitter. Therefore, these regions need to be resolved with a fine mesh.

BJT Electrode Naming

S-Pisces also provides special electrode names for bipolar simulation that can be used to ease confusion over electrode indices. These electrode names include: “emitter”, “base”, “collector”, “anode”, and “cathode”. Electrode names can be defined in Athena or DevEdit or in the **ELECTRODE** statement in Atlas. The electrode names are used on **SOLVE** statement. For example:

```
SOLVE VBASE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=BASE
```

Dual Base BJTs

It is possible in S-Pisces to tie two or more contacts together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initially defined.

If the structure is defined using Atlas syntax, it is possible to have multiple **ELECTRODE** statements with the same NAME parameter defining separate locations within the device structure. In this case, the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both **ELECTRODE** areas.

Similarly, if two separate metal regions in Athena are defined using the Athena **ELECTRODE** statement to have the same name, then in Atlas these two electrodes will be considered as shorted together.

If the electrodes are defined with different names, the following syntax can be used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base

.

SOLVE VBASE=0.1
```

Here, the electrode, base1, will be linked to the electrode, base. Then, the applied 0.1V on base will also appear on base1. But Atlas will calculate and store separate currents for both base and base1. This can be a useful feature. In some cases, however, such as where functions of the currents are required in **EXTRACT** or TonyPlot, it is undesirable. You can add the SHORT parameter to the **CONTACT** statement above to specify that only a single base current will appear combining the currents from base and base1.

When loading a structure from Athena or DevEdit where two defined electrode regions are touching, Atlas will automatically short these and use the first defined electrode name.

Creating an Open Circuit Electrode

It is often required to perform a simulation with an open circuit, such as for an open-base breakdown voltage simulation, on one of the defined electrodes. There are three different methods that will accomplish this. The first method is to delete an electrode from the structure file. The second method is to add an extremely large lumped resistance, for example $10^{20}\Omega$, onto the contact to be made open circuited. The third method is to first switch the boundary conditions on the contact to be made open circuited from voltage controlled to current controlled. Then, specify a very small current through that electrode.

Each of these methods are feasible. But if a floating region is created within the structure while using these methods, then numerical convergence may be affected. As a result, it is normally recommended to use the second method to ensure that no floating region is created.

Solution Techniques for BJTs

To obtain bipolar solutions, you almost always need to simulate using two carriers. This is due to the importance of minority carriers to device operation.

In certain cases, non-local carrier heating may be important to accurately simulate bipolar devices. In these cases, use the energy balance model. To model non-local carrier heating for electrons/holes, set the HCTE.EL and HCTE.HO parameters in the **MODELS** statement. For example, the statement:

```
MODEL HCTE.EL
```

invokes the carrier heating equation for electrons.

5.2.3 Simulating Non-Volatile Memory Technologies (EEPROMs, FLASH Memories)

To simulate non-volatile memory devices, you must become familiar with the basics of MOSFET simulation (see Section 5.2.1 “Simulating MOS Technologies”).

Defining Floating Gates

To simulate non-volatile memory technologies, such as EEPROMs or FLASH EEPROMs, specify one or more electrodes as floating gates. To do this, set the FLOATING parameter of the **CONTACT** statement. For example:

```
CONTACT NAME=fgate FLOATING
```

This specifies that the electrode named fgate is simulated as a floating gate. This means that the charge on the floating gate is calculated as the integral of the gate current at that gate during a transient simulation.

Modeling the correct coupling capacitance ratio between the floating gate and control gate often requires adding an extra lumped capacitor from the floating gate to the control gate or one of the other device terminals. This is often required since S-Pisces is performing a 2D simulation whereas the coupling of the gates is often determined by their 3D geometry. Parameters on the **CONTACT** statement are used to apply these extra lumped capacitances. For example, to add a capacitor of 1fF/mm between the control and floating gates the syntax is:

```
CONTACT NAME=fgate FLOATING FG1.CAP=1.0e-15 EL1.CAP=cgate
```

Gate Current Models

You can supply the gate currents for the floating gate structure by one of three sources: hot electron injection (HEI or N.CONCAN), hot hole injection (HHI or P.CONCAN) and Fowler-Nordheim tunneling current (FNORD).

These currents are of importance, depending on whether electrons are being moved onto the gate or off the floating gate. In the case of placing electrons on the floating gate, use hot electron injection and Fowler-Nordheim tunneling. In the case of removing electrons from the floating gate, set hot hole injection and Fowler-Nordheim tunneling.

In drift diffusion simulations, perform hot electron injection by setting the HEI parameter of the **MODELS** statement. To simulate hot hole injection, use the HHI parameter of the **MODELS** statement. To enable Fowler-Nordheim tunneling, set the FNORD parameter of the **MODELS** statement. The following example demonstrated the proper setting for Flash EPROM programming.

```
MODELS MOS HEI PRINT
```

This next example is appropriate for EEPROM erasure.

```
MODELS MOS HHI FNORD PRINT
```

With energy balance simulations, use the Concannon Models for EPROM programming and erasing. For more information about these models, see "Concannon's Injection Model" on page 273.

Note: Writing and erasure of floating gate devices should be done using transient simulation.

Gate Current Assignment (NEARFLG)

The actual calculation of floating gate current magnitude is done at the silicon-oxide interface. The question of distribution of oxide currents to the various electrodes near the interface is resolved using one of two models. The actual flow of carriers in oxides is not well known. Accepted physical models of carrier flow in oxides are still under research. As such, S-Pisces provides two heuristic models to choose from. The default is to distribute currents calculated at points along the interface to the electrode in the direction of highest contributing field. This model is somewhat analogous to a purely drift model of oxide carrier transport. The alternative is to set the NEARFLG parameter of the **MODELS** statement. In this case, the currents calculated at points along the interface are distributed to the geometrically closest electrode. This model is analogous to a purely diffusion model of carrier transport in oxide.

5.2.4 Simulating SOI Technologies

Silicon substrates are now being produced that contain an oxide layer buried below the surface of the silicon at some predefined depth. The existence of this buried oxide layer has resulted in a change not only in the fabrication process used to manufacture a device in the surface silicon but also in the challenges facing device simulation.

All of the issues raised previously about MOS device simulation should be considered with some extra SOI specific problems.

The most common device technology that uses these SOI substrates is the SOI MOSFET. This section summarizes the simulation requirements for SOI using this particular technology as a reference.

Meshing in SOI devices

The mesh requirements for SOI MOSFETs is very similar to that described in the previous section for bulk MOS transistors. In addition to these requirements, there are some additional points to meshing these devices. These are:

- Two channel regions may exist: one underneath the top (front) gate oxide and the other above the buried (back gate) oxide.
- Inside the buried oxide layer, the mesh constraints can be relaxed considerably compared with the top gate oxide.
- The active silicon underneath the top gate can act as the base region of a bipolar transistor and as such may require a finer mesh when compared to bulk MOS transistors.

Physical Models for SOI

SOI MOSFET simulations are based upon the physical operation of the device, which exhibits both MOS and bipolar phenomena. As a result a more complex set of physical models will be required than for either MOS or bipolar technologies. Table 5-1 shows these models.

Table 5-1 SOI Physical Models

Model	Description
Mobility	Klaassen's Model (KLA) is recommended to account for lattice scattering, impurity scattering, carrier-carrier scattering and impurity clustering effects at high concentration. The Shirahata mobility model (SHI) is needed to take into account surface scattering effects at the silicon/oxide interface, which is a function of the transverse electric field. High electric field velocity saturation is modelled through the field dependent mobility model (FLDMOB). You can tune model parameters using the **MOBILITY** statement syntax.
Interface Charge	In SOI transistors, there exist two active silicon to oxide interfaces on the wafer. The top interface, under the top gate, is similar to MOS technology. The bottom interface is quite different and typically contains significantly more charge. You can set different interface charges in SPISCES using the **INTERFACE** statement with region specific parameters.
Recombination	To take account of recombination effects, we recommend the use of the Shockley-Read-Hall (SRH) model. This simulates the leakage currents that exist due to thermal generation. You may also need to simulate the presence of interface traps at the silicon/oxide interface. Then, turn on the direct recombination model (AUGER). The parameters for both models can be tuned in the **MOBILITY** statement.
Bandgap Narrowing	This model (BGN) is necessary to correctly model the bipolar current gain when the SOI MOSFET behaves like a bipolar transistor. Specify the parameters for this model in the **MODELS** statement.

Table 5-1 SOI Physical Models	
Model	**Description**
Carrier Generation	Impact ionization significantly modifies the operation of SOI MOSFETs. To account for this phenomena, switch on the impact ionization model (**IMPACT**) and calibrate for SOI technology. The calibration parameters are set in the **IMPACT** statement.
Lattice Heating	When you switch a device on, there can be significant current density within the silicon. This could generate a significant amount of heat. In bulk MOS devices, the silicon substrates behaves like a good heat conductor. This generated heat is then quickly removed. But this isn't the case with SOI substrates as the buried oxide layer allows this generated heat to be retained. For SOI MOSFETs, this can be a significant amount and can drastically affect the operation of the device. In such cases, take account of this by using the Giga module. Note that when you switch on lattice heating by using the LAT.TEMP parameter in the **MODELS** statement, you also need to specify a thermal boundary condition with the **THERMCONTACT** statement. See Chapter 8 "Giga: Self-Heating Simulator" for more details.

Carrier Heating	In deep submicron designs, you may need to switch on the additional energy balance equations. These take into account the exchange of energy between carriers and between the carriers and the lattice. See Section 5.2.1 "Simulating MOS Technologies" for more information.
	
	An example of a set of typical models for a partially depleted SOI MOSFET could be:
	
	`MODEL KLA SHI FLDMOB SRH AUGER BGN LAT.TEMP`
	`INTERFACE QF=1e10 Y.MAX=0.05`
	`INTERFACE QF=1e11 Y.MIN=0.05`
	`THERMCONTACT NUM=1 ELEC.NUM=4 EXT.TEMP=300`
	`IMPACT SELB`

Numerical Methods for SOI

One important issue with SOI device simulation is the choice of numerical methods. In SOI technology, the potential in the channel (or body) region is commonly referred to as "floating". This occurs because there exists no direct contact to it by any electrode. As a result, when a bias is applied, or increased, on a contact there may be some convergence problem. This occurs because the guess used in the numerical solution scheme for (ψ, n, p) may be poor, particularly in the "floating" region. This is particularly true if impact ionization is used. To overcome the problem of poor initial guess, use the following numerical methods syntax in isothermal drift-diffusion simulations.

```
METHOD GUMMEL NEWTON
```

This method initially performs a GUMMEL iteration to obtain an improved initial guess for the NEWTON solution scheme. Although this method is more robust, this is slower than using the NEWTON scheme alone. For more information on the numerical schemes available, see Chapter 21: "Numerical Techniques".

SOI Physical Phenomena

The physical models and the numerical schemes described above should allow S-Pisces to study all the important SOI phenomena. These include:

- full or partial depletion effects
- threshold voltage
- subthreshold slopes
- front to back gate coupling effects
- leakage current analysis
- high frequency analysis
- device breakdown
- snapback effects
- the kink effect in the output Ids-Vds characteristics
- negative differential resistance

Chapter 6
Blaze: Compound Material 2D Simulator

6.1 Overview

If you are unfamiliar with the general operation of the Atlas framework, read Chapter 2 "Getting Started with Atlas" before reading this chapter.

Blaze is a general purpose 2D device simulator for III-V, II-VI materials, and devices with position dependent band structure (e.g., heterojunctions). Blaze accounts for the effects of positionally dependent band structure by modifications to the charge transport equations and Poisson's equation. Blaze is applicable to a broad range of devices such as: HBTs, HEMTs, LEDs, lasers, heterojunction photodetectors (e.g., APDs, solar cells) and heterojunction diodes.

This chapter is composed of several sections. Section 6.1.1 "Basic Heterojunction Definitions" gives an overview of the basic heterojunction band parameters. Section 6.1.2 "Alignment" explains heterojunction alignment. Heterojunction charge transport is covered in Section 6.1.3 "Temperature Dependence of Electron Affinity". This section includes the details of how Blaze modifies the basic transport models to simulate heterodevices. Section 6.3 "The Physical Models" covers the Blaze models specific to compostionally dependent materials. Detailed information about the material systems encountered in heterojunction simulation is covered in Section 6.4 "Material Dependent Physical Models". This includes the relationships between the compound elemental concentrations and bandgap, dielectric constant, low field mobility, and other important material and transport parameters. Finally, Section 6.5 "Simulating Heterojunction Devices with Blaze" covers how to define materials and models for heterojunction devices with Blaze.

See Appendix B "Material Systems" for the defaults parameters and for some of the materials discussed in this chapter.

6.1.1 Basic Heterojunction Definitions

Figure 6-1 shows the band diagrams and band parameters for a basic p-n heterojunction device under equilibrium conditions. This diagram illustrates two materials with different bandgaps and electron affinities and a Schottky barrier in contact to the n-type material.

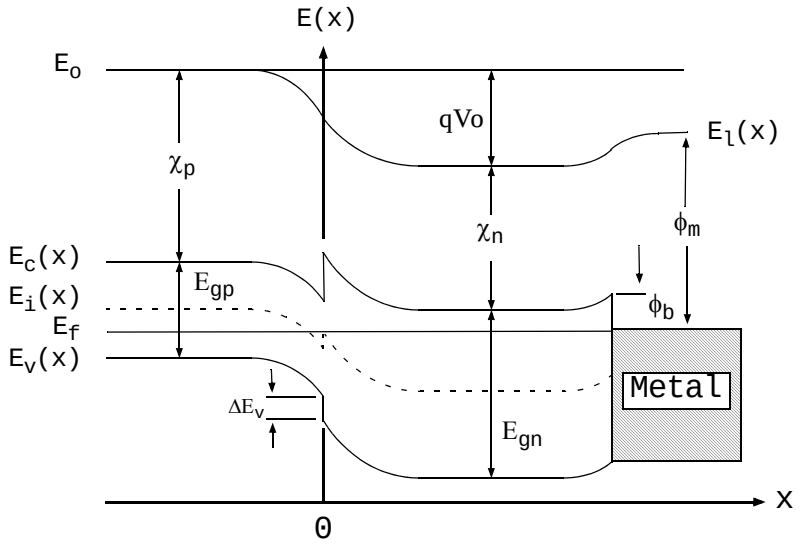


Figure 6-1: Band Diagram of p-n Heterojunction

Referring to Figure 6-1:

- $E_c(x)$, $E_v(x)$ and $E_i(x)$ are the spatially dependent conduction band, valence band and intrinsic energy levels respectively.
- E_f and E_o are the Fermi and Vacuum level energies.
- E_{gp} and E_{gn} are the p-type and n-type material bandgaps.
- ΔE_v is the fraction of the difference between E_{gp} and E_{gn} that appears in the valence band at the heterojunction interface.
- ΔE_c (not labelled) is the fraction of the difference between E_{gp} and E_{gn} that appears in the conduction band at the heterojunction interface.
- *qVo* is the built-in potential of the heterojunction.
- χ_p and χ_n are the electron affinities of the p-type and n-type materials.
- ϕ_m is the work function of the metal.
- ϕ_b is the barrier height between the metal and the semiconductor.

The basic band parameters for defining heterojunctions in Blaze are bandgap parameter, EG300, AFFINITY, and the conduction and valence band density of states, NC300 and NV300. To define these parameters for each material, use the **MATERIAL** statement. Other transport parameters relating these basic definitions to compound elemental concentrations (X.COMPOSITION and Y.COMPOSITION) can also be defined.

See Section 6.4 "Material Dependent Physical Models" for a description of these relationships for each material system and the Section 6.5 "Simulating Heterojunction Devices with Blaze" for their usage.

The work function of metals is defined using the **CONTACT** statement.

Note: In the following, the affinites are defined by the vacuum energy minus the lowest conduction band edge energy.

6.1.2 Alignment

As shown from Figure 6-1, the difference between the two materials bandgaps creates conduction and valence band discontinuities. The distribution of bandgap between the conduction and valence bands has a large impact on the charge transport in these heterodevices. There are two methods for defining the conduction band alignment for a heterointerface The first method is the Affinity Rule, which uses the ALIGN parameter on the **MATERIAL** statement. The second method is to adjust the material affinities using the AFFINITY parameter on the **MATERIAL** statement.

The Affinity Rule

This is the default method in Blaze for assigning how much of the bandgap difference appears as the conduction band discontinuity. The affinity rule assigns the conduction band discontinuity equal to the difference between the two materials electron affinities (AFFINITY on the **MATERIAL** statement). The affinity rule method is used by default for all materials where the ALIGN parameter has not been defined on the **MATERIAL** statement.

Using the ALIGN parameter in the MATERIAL statement

Experimental measurements of band discontinuities can differ from what is assigned using the affinity rule with the standard material electron affinities. Therefore, Blaze allows ΔE_c to be calculated by specifying the ALIGN parameter on the **MATERIAL** statement. ALIGN specifies the fraction of the bandgap difference, which will appear as the conduction band discontinuity. This bandgap difference is between the material, which is specified by the ALIGN parameter, and the smallest bandgap material in the overall structure (the reference material). Use the ALIGN parameter to modify the electron affinity of the material and Blaze will create the desired conduction band offset.

In many applications, a Schottky barrier or more than two different semiconductor materials are present. Keep the reference material bandgap and these assigned affinities in mind when defining offsets for multiple materials or Schottky barrier heights. Examples for multiple materials and Schottky barriers are given in the following section. In the following examples, ΔE_g, ΔX, ΔE_c, and E_v are positive.

Manually Adjusting Material Affinity

You can use the AFFINITY parameter on the **MATERIAL** statement to adjust the conduction band offset. The following examples describe the procedure for aligning heterojunctions using this method in Blaze.

EXAMPLE 1

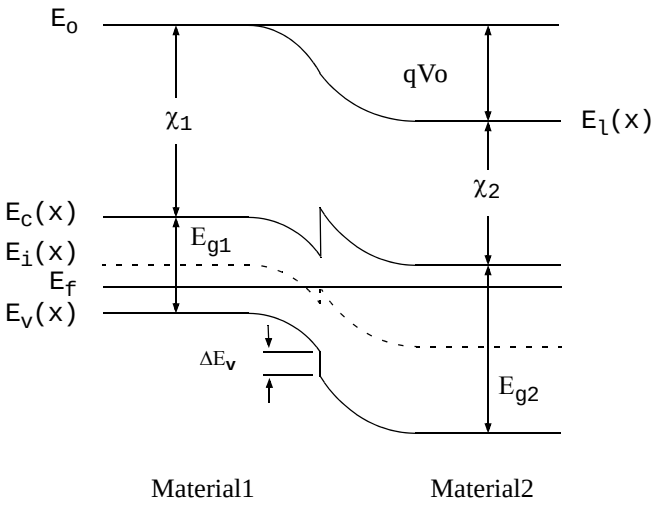


Figure 6-2: Band diagram of heterojunction with band offset.

Figure 6-2 shows a heterojunction consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities χ_1 and χ_2. This example is similar to the bandstructure of a HFET or HEMT. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_c = \chi_1 - \chi_2 \quad \text{6-1}$$

$$\Delta E_g = E_{g2} - E_{g1} \quad \text{6-2}$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \quad \text{6-3}$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface. ΔE_v is the amount of the valence band discontinuity.

Note: The Affinity Rule is used to calculate the conduction band offset for a material if the ALIGN parameter is not specified on the **MATERIAL** statement for that material.

Using the ALIGN parameter on the MATERIAL statement

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset. Then, define the ALIGN parameter on the **MATERIAL** statement for Material2 using

```
MATERIAL MATERIAL=Material2 ALIGN=0.80
```

Then:

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 \quad \text{6-4}$$

Internally, the affinity of Material2 is adjusted so that ΔE_c equals this value. This value of electron affinity will override any electron affinity specification for Material2. This has an impact on any calculation where this material's electron affinity is used and considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

```
MATERIAL MATERIAL=Material2 AFFINITY=VALUE
```

where VALUE is adjusted to provide for the desired conduction band offset as calculated by the affinity rule, that is, relative to all other assigned affinities. This value of electron affinity will override any prior specification of electron affinity for Material2. This has an impact on any calculation where this material's electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations

---

Note: If you do not specify the ALIGN parameter on the **MATERIAL** statement, Blaze will use the Affinity Rule and either the default electron affinity or the affinity assigned using the AFFINITY parameter on the **MATERIAL** statement to calculate the conduction band offsets.

EXAMPLE 2

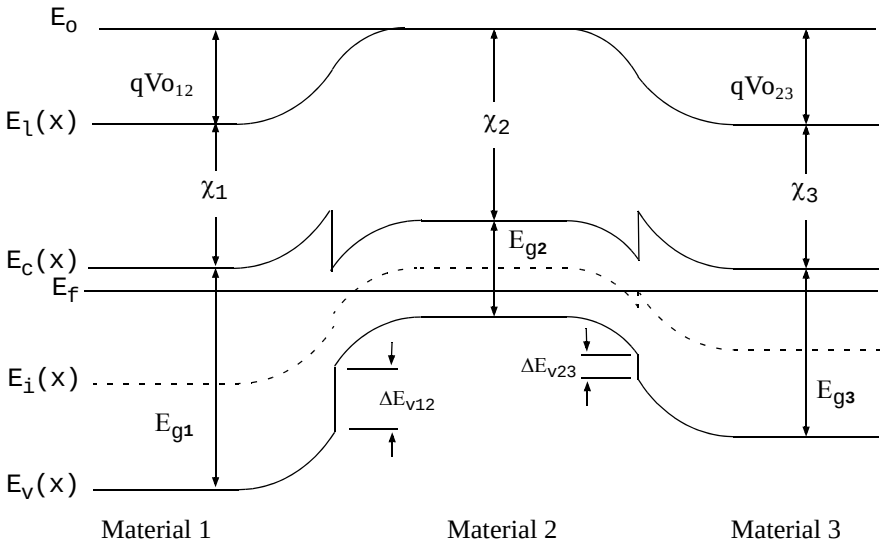


Figure 6-3: Band diagram of three material system (lowest E_g in center)

Figure 6-3 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1}, E_{g2}, and E_{g3}, and electron affinities χ_1, χ_2, and χ_3. This is similar to the band diagram of a Double Heterojunction Bipolar Transistor. For this example, $E_{g1} > E_{g2} < E_{g3}$ and $\chi_1 < \chi_2 > \chi_3$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_{c12} = \chi_2 - \chi_1 \quad \text{6-5}$$

$$\Delta E_{g12} = E_{g1} - E_{g2} \quad \text{6-6}$$

and

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \quad \text{6-7}$$

for the heterojunction between `Material1` and `Material2` and:

$$\Delta E_{c23} = \chi_2 - \chi_3 \quad \text{6-8}$$

$$\Delta E_{g23} = E_{g2} - E_{g3} \quad \text{6-9}$$

and

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \quad \text{6-10}$$

for the heterojunction between Material2 and Material3.

Notice the reference material (i.e., material) with the smallest bandgap. In this case, Material2, is located between the two larger bandgap materials, Material1, and Material3.

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset for this heterojunction. Then, define the ALIGN parameter on the **MATERIAL** statement for Material 1 using:

```
MATERIAL MATERIAL=Material1 ALIGN=0.8
```

then:

$$\Delta E_{c12} = (E_{g1} - E_{g2}) \cdot 0.80 \quad \text{6-11}$$

Internally, the affinity of Material1 is adjusted so that ΔE_{c12} equals this value.

Let's assign 70% of the bandgap difference between Material3 and Material2 to the conduction band offset for this heterojunction. Defining the ALIGN parameter on the **MATERIAL** statement for Material3 using:

```
MATERIAL MATERIAL=Material3 ALIGN=0.70
```

then:

$$\Delta E_{c23} = (E_{g3} - E_{g2}) \cdot 0.70 \quad \text{6-12}$$

Internally, the affinity of Material3 is adjusted so that ΔE_{c23} equals this value.

These new values of electron affinity for Material1 and Material3 will override any electron affinity specification for these materials. This has an impact on any calculation where these materials's electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

You can assign the conduction band offsets for each heterojunction by setting the electron affinities for Material1 and Material3 using the AFFINITY parameter on the **MATERIAL** statement.

Let's assume an electron affinity for Material2 of 4eV (~ that of GaAs). Let's decide that between Material1 and Material2, the conduction band offset is 0.3eV and that between Material3 and Material2, the conduction band offset is 0.2eV. Then, for Material1:

```
MATERIAL MATERIAL=Material1 AFFINITY=3.7
```

and for Material3:

```
MATERIAL MATERIAL=Material3 AFFINITY=3.8
```

This is the easiest method to define the conduction band offsets for multiple materials. This value of electron affinity will override any electron affinity specification. This has an impact on any calculation where this material's electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 3

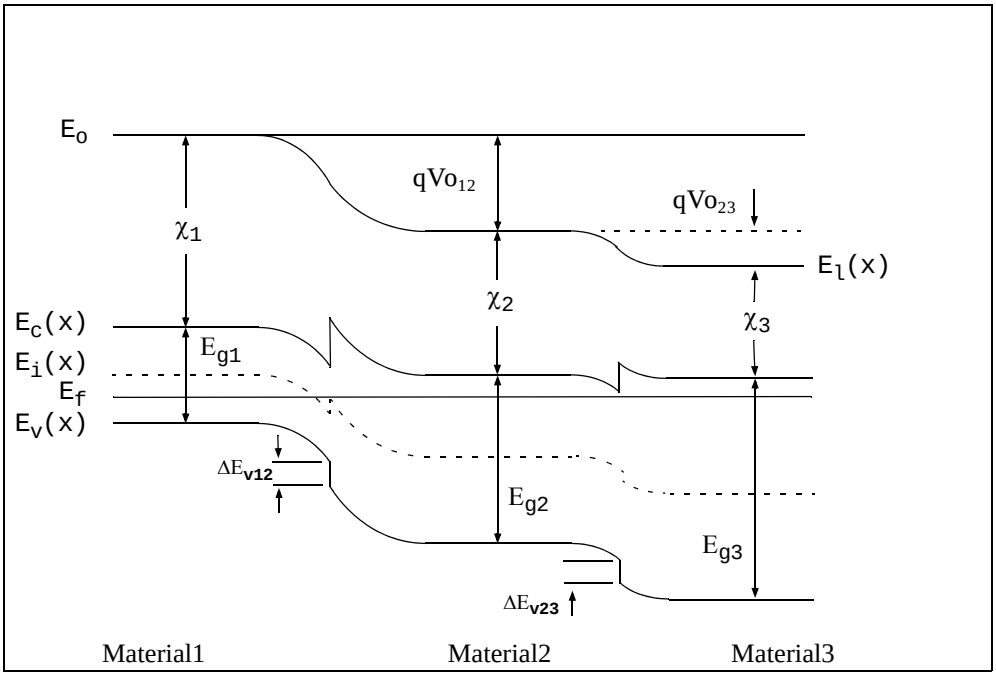


Figure 6-4: Band diagram of three material system (lowest E_g not in center)

Figure 6-4 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1}, E_{g2}, and E_{g3} and electron affinities χ_1, χ_2, and χ_3. This is similar to the "EXAMPLE 2" on page 466, except that the narrow bandgap material is not located in between the other larger bandgap materials. This will add extra complexity to the conduction and valence band offset calculations. For this example, $E_{g1} < E_{g2} < E_{g3}$ and $\chi_3 < \chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_{c12} = \chi_1 - \chi_2 \quad \text{6-13}$$

$$\Delta E_{g12} = E_{g2} - E_{g1} \quad \text{6-14}$$

and

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \quad \text{6-15}$$

for the heterojunction between `Material1` and `Material2` and:

$$\Delta E_{c23} = \chi_2 - \chi_3 \quad \text{6-16}$$

$$\Delta E_{g23} = E_{g3} - E_{g2} \quad (6\text{-}17)$$

and

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \quad (6\text{-}18)$$

for the heterojunction between Material2 and Material3.

Using the ALIGN parameter on the MATERIAL statement

Notice that the reference material (i.e., the material with the smallest bandgap), Material1, is not shared between the two larger bandgap materials, Material2, and Material3. This will be important in calculating the conduction band offsets for the heterojunction formed by Material2 and Material3.

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset for this heterojunction. Since the reference material is one of the materials of this heterojunction, we can proceed as before. Define the ALIGN parameter in the **MATERIAL** statement for Material2 using:

```
MATERIAL MATERIAL=Material2 ALIGN=0.8
```

then

$$\Delta E_{c12} = (E_{g2} - E_{g1}) \cdot 0.80 \quad (6\text{-}19)$$

Let's assign 70% of the bandgap difference between Material3 and Material2 to the conduction band offset for this heterojunction. Since the reference material is not one of the materials in this heterojunction, another procedure will be used. Since Blaze always uses the bandgap of the reference material (the smallest bandgap material in overall structure) when calculating the conduction band offset using the ALIGN parameter on the **MATERIAL** statement, the actual value for the ALIGN parameter needs to be calculated as follows:

$$\texttt{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot (\Delta E_{c32} / \Delta E_{g32}) \quad (6\text{-}20)$$

Once calculated, you can use this value for the ALIGN parameter on the **MATERIAL** statement for Material3. For this example, let's assume that

$$\Delta E_{g32} = 0.2 \quad (6\text{-}21)$$

$$\Delta E_{g} = E_{g2} - E_{g1} \quad (6\text{-}22)$$

and

$$\Delta E_{g31} = 0.4 \quad (6\text{-}23)$$

then for a desired conduction band offset fraction of 0.70:

$$\texttt{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot (\Delta E_{c32} / \Delta E_{g32}) = (0.2 / 0.4) \times 0.70 = 0.35 \quad (6\text{-}24)$$

You can assign the proper value of the ALIGN parameter reflecting the desired conduction band offset as:

```
MATERIAL MATERIAL=Material3 ALIGN=0.35
```

This assigns 70% of the bandgap difference between Material3 and Material2 as the conduction band offset.

Note: Calculating ALIGN in this manner is only necessary when the reference material is not in contact with the material where the ALIGN parameter will be specified.

These new values of electron affinity for Material2 (from the first heterojunction band offset calculation) and Material3 (from the second heterojunction band offset calculation) will override any electron affinity specification for these materials. This has an impact on any calculation where these materials's electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

You can assign the conduction band offsets for each heterojunction by setting the electron affinities for Material2 and Material3 using the AFFINITY parameter on the **MATERIAL** statement. The electron affinity for Material2 is adjusted relative to Material1 and Material3 is adjusted relative to Material2 by the amount of the desired conduction band offset for each heterojunction. Since Material1 affinity is larger than that for Material2 and Material2 affinity is larger than that for Material3, the affinities for Material2 and Material3 are reduced to provide the desired conduction band offsets.

Let's assume an electron affinity for Material1 of 4eV (~ that of GaAs). Let's decide that between Material1 and Material2, the conduction band offset is 0.3eV and that between Material2 and Material 3, the conduction band offset is 0.2eV. Then, for Material2:

```
MATERIAL MATERIAL=Material2 AFFINITY=3.7
```

and for Material3:

```
MATERIAL MATERIAL=Material3 AFFINITY=3.5
```

This is the easiest method to define the conduction band offsets for multiple materials. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 471 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 4

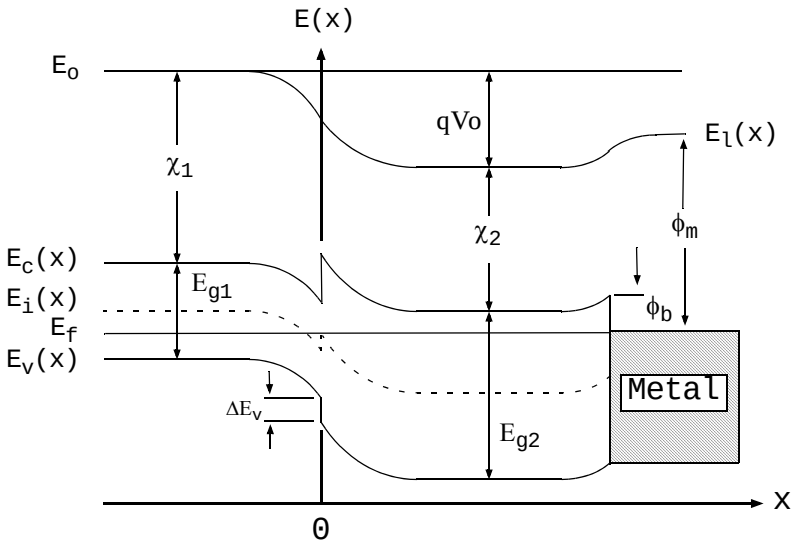


Figure 6-5: Schematic band diagram for an abrupt heterojunction

Figure 6-5 details a heterostructure device consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities χ_1 and χ_2 and a Schottky barrier. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

This example will first define the heterojunction band offsets and then the Schottky barrier height. Schottky contact barrier heights are calculated by Blaze using the metal work function and the semiconductor electron affinity as:

$$\phi_b = \phi_m - \chi_s \quad \text{6-25}$$

where ϕ_b is the Schottky barrier height, ϕ_m is the work function of the metal, and χ_s is the semiconductor electron affinity. ϕ_m is set using the WORKFUN parameter in the CONTACT statement. Therefore, the semiconductor electron affinity as modified or defined during the heterojunction alignment process plays an important role in determining the value of the metal workfunction needed to provide the desired barrier height. Let's assume for this example that a Schottky barrier height of 0.2eV is desired and calculate the appropriate metal workfunction for each case.

Using the Affinity rule for the heterojunction

$$\Delta E_c = \chi_1 - \chi_2 \quad \text{6-26}$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \quad \text{6-27}$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface. ΔE_v is the amount of the valence band discontinuity.

Note: The Affinity Rule is used for a material if the ALIGN parameter is not specified on the **MATERIAL** statement for that material.

Let's use an electron affinity for Material1 of 4eV and for Material2 of 3.5eV. Since the affinity of the material on which the Schottky barrier is formed was not modified with this method of alignment, the metal work function needed to provide for a Schottky barrier height of 0.2eV is:

$$\phi_m = 3.5 + 0.2 = 3.7 \quad \text{6-28}$$

You can now assign this value to the WORKFUN parameter on the **CONTACT** statement as:

```
CONTACT NUMBER=1 WORKFUN=3.7
```

This produces a Schottky barrier height of 0.2eV between the metal and Material 2 using the ALIGN parameter on the **MATERIAL** statement:

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset, an electron affinity for Material1 of 4eV, and ΔE_g of 0.2eV. Then, define the ALIGN parameter on the **MATERIAL** statement for Material2 using:

```
MATERIAL MATERIAL=Material2 ALIGN=0.80
```

Then:

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 = 0.2 \cdot 0.80 = 0.16 \quad \text{6-29}$$

Internally, the affinity of Material2 is reduced by 0.16eV so:

$$\chi_2 = \chi_1 - \Delta E_c = (4 - 0.16) = 3.84 \quad \text{6-30}$$

You can use this value of electron affinity to assign the proper value of WORKFUN on the **CONTACT** statement to provide for a Schottky barrier height of 0.2eV.

$$\phi_m = 3.84 + 0.2 = 4.04 \quad \text{6-31}$$

You can now assign this value to the WORKFUN parameter on the **CONTACT** statement as:

```
CONTACT NUMBER=1 WORKFUN=4.04
```

producing a Schottky barrier height of 0.2eV between the metal and Material2.

Manually Adjusting Material Affinity

Let's assign a conduction band offset between Material1 and Material2 of 0.15eV, an electron affinity for Material1 of 4eV, and the desired Schottky barrier height of 0.2eV. The affinity for Material2 is calculated from the affinity of Material1 and the desired conduction band offset as:

$$\chi_2 = \chi_1 - 0.15 = 4 - 0.15 = 3.85 \quad \text{6-32}$$

This is then used to assign the value of AFFINITY using:

```
MATERIAL MATERIAL=Material2 AFFINITY=3.85
```

You can now use this value of electron affinity to calculate the metal workfunction necessary to produce a Schottky barrier height of 0.2eV as:

$$\phi_m = 3.85 + 0.2 = 4.05 \quad \text{6-33}$$

You can now assign this value to the WORKFUN parameter on the **CONTACT** statement as:

```
CONTACT NUMBER=1 WORKFUN=4.05
```

This produces a Schottky barrier height of 0.2eV between the metal and Material2.

6.1.3 Temperature Dependence of Electron Affinity

The electron affinity of a semiconductor is generally a function of lattice temperature. This temperature dependence is usually modelled implicitly in Atlas by making it a fraction of the temperature dependence of the bandgap. The default value of the fraction is 0.5 but you can assign it a value in the range [0,1] using the CHI.EG.TDEP.

If you specify the ALIGN parameter, then CHI.EG.TDEP will be ignored. In that case, the affinity is adjusted to keep the conduction band offset at the specified ratio of the bandgap differences at the heterojunction at the particular temperature required.

Example 1

For a material where the affinity is given only at 300 K (e.g., GaAs) the electron affinity at temperature T would be

$$\chi(T) = \chi(300) - \texttt{CHI.EG.TDEP} \times (Eg(T_L) - Eg(300)) \quad \text{6-34}$$

Example 2

For HgCdTe, the affinity depends on bandgap and consequently temperature (see Equation 6-171). In this case, you must include the following statement.

```
MATERIAL MATERIAL=HGCDTE CHI.EG.TDEP=0
```

Example 3

If you are using F.BANDCOMP to specify the electron affinity and your function includes its full temperature dependence, then specify CHI.EG.TDEP = 0 on the **MATERIAL** statement.

If your function excludes temperature dependence of affinity, then you can use the value of CHI.EG.TDEP to adjust this.

Note: The ASYMMETRY parameter on the **MATERIAL** statement will be used to assign a fraction of bandgap narrowing to the affinity. See Section 3.2.9 "Bandgap Narrowing" for more information.

6.2 Transport Models

6.2.1 The Drift Diffusion Transport Model

Drift-Diffusion with Position Dependent Band Structure

The current continuity equations for electrons and holes, and the Poisson Equation (see Equation 3-1) are the same as for the homogeneous case. Although the changing dielectric constant is taken into account. The current density expressions, however, must be modified to take into account the nonuniform band structure [192]. This procedure starts with the current density expressions:

$$\vec{J}_n = -q\mu_n n\nabla\phi_n \quad (6\text{-}35)$$

$$\vec{J}_p = -q\mu_p p\nabla\phi_p \quad (6\text{-}36)$$

where ϕ_n and ϕ_p are quasi-Fermi potentials.

$$\phi_n = -\frac{1}{q}E_{FN} \quad (6\text{-}37)$$

$$\phi_p = -\frac{1}{q}E_{FP} \quad (6\text{-}38)$$

The conduction and valence band edge energies can be written as:

$$E_c = q(\psi_0 - \psi) - \chi \quad (6\text{-}39)$$

$$E_v = q(\psi_0 - \psi) - \chi - E_g \quad (6\text{-}40)$$

where:

- ψ_0 is some reference potential.
- χ is the position-dependent electron affinity.
- E_g is the position-dependent bandgap.
- ψ_0 can be selected in the form:

$$\psi_0 = \frac{\chi_r}{q} + \frac{kT_L}{q}\ln\frac{N_{cr}}{n_{ir}} = \frac{\chi_r + E_g}{q} - \frac{kT_L}{q}\ln\frac{N_{vr}}{n_{ir}} \quad (6\text{-}41)$$

where n_{ir} is the intrinsic carrier concentration of the arbitrarily selected reference material, and r is the index that indicates all of the parameters are taken from reference material.

Fermi energies are expressed in the form:

$$E_{FN} = E_c + kT_L\ln\frac{n}{N_c} - kT_L\ln\gamma_n \quad (6\text{-}42)$$

$$E_{FP} = E_v - kT_L \ln\frac{p}{N_v} + kT_L \ln\Upsilon_p \quad 6\text{-}43$$

The final terms in Equations 6-42 and 6-43 are due to the influence of Fermi-Dirac statistics. These final terms are defined as follows:

$$\gamma_n = \frac{F_{1/2}(\eta_n)}{e^{\eta_n}}, \eta_n = \frac{E_{FN} - E_c}{kT_L} = F_{1/2}^{-1}\left(\frac{n}{N_c}\right) \quad 6\text{-}44$$

$$\gamma_p = \frac{F_{1/2}(\eta_p)}{e^{\eta_p}}, \eta_p = \frac{E_v - E_{FP}}{kT_L} = F_{1/2}^{-1}\left(\frac{p}{N_v}\right) \quad 6\text{-}45$$

where N_c and N_v are position-dependent and γ_n=γ_p= 1 for Boltzmann statistics.

By combining Equations 6-35 to 6-45, you can obtain the following expressions for current densities.

$$\vec{J}_n = kT_L\mu_n\nabla n - q\mu_n n\nabla\left(\psi + \frac{kT_L}{q}\ln\gamma_n + \frac{\chi}{q} + \frac{kT_L}{q}\ln N_c\right) \quad 6\text{-}46$$

$$\vec{J}_p = -kT_L\mu_p\nabla p - q\mu_p p\nabla\left(\psi - \frac{kT_L}{q}\ln\gamma_p + \frac{\chi + E_g}{q} - \frac{kT_L}{q}\ln N_v\right) \quad 6\text{-}47$$

6.2.2 The Thermionic Emission and Field Emission Transport Model

You can activate alternative current density expressions for electron and hole current [358, 364], which take into account thermionic emission dominated current in abrupt heterojunctions. These equation applies only at the node points along the interface of the heterojunction and take the form:

$$\vec{J}_n = q(1+\delta)\left(v_{n+}n^+ - v_{n-}n^-\exp\left(\frac{-Q.\texttt{THERMIONIC}\Delta E_C}{kT_L}\right)\right) \quad 6\text{-}48$$

$$\vec{J}_p = (-q)(1+\delta)\left(v_{p+}p^+ - v_{p-}p^-\exp\left(\frac{-Q.\texttt{THERMIONIC}\Delta E_V}{kT_L}\right)\right) \quad 6\text{-}49$$

`Q.THERMIONIC` is the thermionic emission quality factor and can be specified on the CONTACT statement (default is 1). QN.BARRIER and QP.BARRIER can be used to set different quality factors for the different carriers. QN.BARRIER sets the quality for electrons (equation 6-48) and QP.BARRIER sets the quality for holes (equation 6-49).

where J_n and J_p are the electron and hole current densities from the "-" region to the "+" region. v_{n-}, v_{n+}, v_{p-}, and v_{p+} are the electron and hole thermal velocities in the "-" and "+" regions. ΔE_c is conduction band energy change going from the "-" region to the "+" region. ΔE_v is the valence band energy change going from the "-" region to the "+" region. The δ parameter represents the contribution due to thermionic field emission (tunneling).

The thermal velocities υ_n and υ_p are given by:

$$\upsilon_n = \frac{A_n^* T_L^{\ 2}}{qN_C} \quad \text{6-50}$$

$$\upsilon_p = \frac{A_p^* T_L^{\ 2}}{qN_V} \quad \text{6-51}$$

where T_L is the lattice temperature, N_C is the conduction band density of states, N_V is the valence band density of states, and A_n^* and A_p^* are the electron and hole Richardson constants.

You can specify the Richardson constants with the ARICHN and ARICHP parameters of the **MATERIAL** statement. If the Richardson constants aren't specified, the following expressions will be used:

$$A_n^* = \frac{4\pi q k^2 \mathtt{M.VTHN}\, m_0}{h^3} \quad \text{6-52}$$

$$A_p^* = \frac{4\pi q k^2 \mathtt{M.VTHP}\, m_0}{h^3} \quad \text{6-53}$$

The electron and hole effective masses can be specified with the M.VTHN and M.VTHP parameters of the **MATERIAL** statement. If the effective masses aren't specified, then they will be calculated from the conduction and valence band densities of states using Equations 3-31 and 3-32. To enable the thermionic transport model, you must specify THERMIONIC on the **INTERFACE** statement. You must also set the S.S parameter on the **INTERFACE** statement to indicate that the model applies to semiconductor-semiconductor interfaces.

The tunneling factor δ in Equation 6-48 is zero when the tunneling mechanism is neglected. When you account for tunneling by specifying the TUNNEL parameter in the **INTERFACE** statement, the tunneling factor, δ, is calculated by using the expression:

$$\delta = \frac{1}{kT}\int_{E_{min}}^{E_C^+} \exp\left(\frac{E_C^+ - E_x}{kT}\right) \exp\left(\frac{-4\pi}{h}\int_0^{X_E} [2m_n^*(E_C - E_x)]^{0.5}\, dx\right) dE_x \quad \text{6-54}$$

where E_x is the energy component in the X direction and $E_{min} = max[E_c(0\text{-}), E_c(W)]$ as described in Figure 6-6.

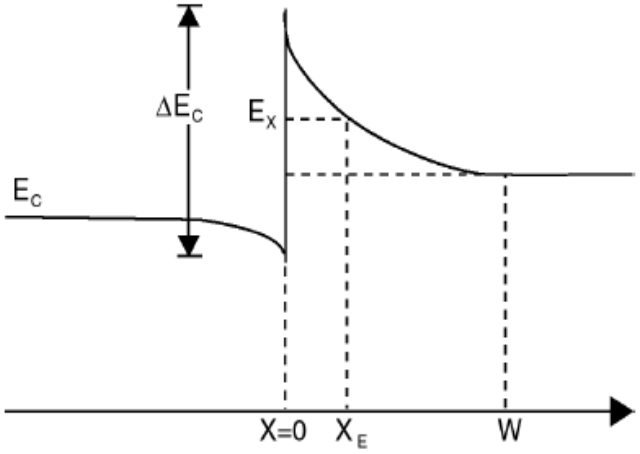


Figure 6-6: Band Diagram of p-n heterojunction

The band edge is approximated as a linear function of position in the wide-band gap material. The slope corresponds to the electric field at the interface.

If S.S THERMIONIC is activated for an interface between two organic semiconductors modeled with Gaussian band structures (see Section 16.2.1 "Gaussian Band Structure"), the probability of a carrier jumping the barrier is given by Equation 28 in [300].

6.2.3 Non-local Heterojunction Tunneling Model

A weakness of the TUNNEL model described above is that it is a local model. Specifically, it calculates the tunneling current based on the value of the field at the interface and the current is injected entirely at the interface. Tunneling is possible over a range of energies. And assuming that the process is elastic, it will give rise to carrier generation or recombination throughout much of the depletion region. To calculate this tunneling current properly, you need to take into account the non-linear shape of the potential barrier as shown in Figure 6-6. Atlas has a non-local tunneling model that calculates the tunneling path and the contribution to the tunneling current for each energy at which elastic tunnelling is possible. In order to use this model, specify one or more **QTREGION**s covering the junction. The tunneling paths of the **QTREGION** should begin on the narrow bandgap side of the heterojunction and end at the boundary of the depletion region in the wide bandgap material. The syntax for the **QTREGION** statement is described in Section 3.6.6 "Band-to-Band Tunneling". Tunneling current is evaluated for all energies at which tunneling is possible, up to the maximum of the band energy at the interface. The current in a range E_x to E_x+dE is given by Equation 3-534, where $T(E)$ is evaluated using the WKB approximation. It is then converted into a recombination or generation rate and inserted into the current continuity equation at each end of its tunnelling path as shown in Figure 6-6.

To enable this model you specify SHJ.EL or SHJ.HO or both on the **MODELS** statement. SHJ.EL is for electron tunneling and SHJ.HO is for hole tunnelling. Convergence is generally improved by including non-local derivatives in the jacobian. Specify SHJ.NLDERIVS on the **MODELS** statement to do this. The recombination/generation rates are automatically output with any Standard Structure File when one or both of SHJ.EL and SHJ.HO are set.

You can specify the QTREGION parameter on the **MODELS** statement in order to restrict the application of the model to the specified **QTREGION**. The default behavior is to apply the model to all defined **QTREGION**s.

Table 6-1 MODEL Statement Parameters

Parameter	Type	Default	Units
SHJ.EL	Logical	False	
SHJ.HO	Logical	False	
SHJ.NLDERIVS	Logical	False	

This model is only suitable for calculating tunneling through single heterojunctions. To calculate tunneling through single or multiple quantum barriers, you need the SIS.EL or SIS.HO models. This is described below.

6.2.4 Non-local Quantum Barrier Tunneling Model

You can calculate the tunneling current between two semiconducting regions separated by a quantum barrier using the SIS.EL and SIS.HO models. The barrier layer may be either an insulating layer or a wide bandgap semiconductor. The model assumes that the charge tunnels across the whole barrier with the source/sink at the interface with the semiconductor regions. Tunneling current is evaluated for all energies at which tunneling is possible, up to the maximum of the band energy at the interface. The current in a range E_x to E_x+dE is given by Equation 3-534. The transmission probability, $T(E)$, is evaluated at each energy from a solution of the one-dimensional Schrodinger equation. The default is to use a transmission matrix method, identical to that used in the QTUNN model of gate oxide tunnelling. To use the alternative Wentzel-Kramers-Brillouin (WKB) approximation to solve the Schrodinger equation, specify TUN.WKB on the METHOD statement. The TUN.WKB option will omit some quantum interference effects, but will result in a much quicker calculation. The tunneling current is then converted into a recombination or generation rate and inserted into the current continuity equation at each end of its tunneling path.

Alternatively, a custom transmission probability profile can be provided using the QT.FILE option, with the following format:

```
NEnergy <number of energy points>
NBias <number of bias points>
<energy 1> <bias 1> <transmission value>
...
<energy NEnergy> <bias 1> <transmission value>
.
.
.
<energy 1> <bias NBias> <transmission value>
<energy 2> <bias NBias> <transmission value>
...
```

```
<energy NEnergy> <bias NBias> <transmission value>
```

where energy values are in electron volts (eV), bias values are in volts (V), and transmission probability values are unitless numbers between 0 and 1, inclusive. In calculating the tunneling rates, the transmission probability values will be linearly interpolated on the two-dimensional energy-bias grid.

In order to use this model, you need to define one or more quantum tunneling regions by using the **QTREGION** statement. The **QTREGION**s should define tunneling paths which start and end in the semiconductor regions on opposite sides of the barrier. There is no requirement for the **QTREGION**s to start and end in flat-band regions as is the case for the BBT.NONLOCAL model. It is possible to simulate coherent tunneling through multiple quantum barriers by defining a **QTREGION** which covers all the barriers. You invoke the non-local tunneling by specifying SIS.EL for electron tunneling, and SIS.HO for hole tunneling, on the **MODELS** statement. In case of poor convergence, you set the flag SIS.NLDERIVS on the **MODELS** statement. This puts in non-local couplings into the system matrix.

You can also model sequential tunneling by defining a single **QTREGION** for each barrier. For structures with a large number of barriers, this can be an onerous task. A more convenient format for setting up the **QTREGION** is to specify the QTREGION parameter on each **REGION** statement defining a quantum barrier. The **QTREGION** formed is optimized in terms of system matrix size because it avoids the use of interpolation. It also injects the tunneling current at the exact position of the interface and into the lower carrier energy side of the interface.

You set QTREGION=x to set up that **MODELS** as a **QTREGION** number x. There are some restrictions with this format. First, it will only work for structures with rectangular meshing. Second, the interfaces at the boundaries of the **QTREGION** must be of a special type. You enable this by specifying INTERFACE THERMIONIC S.S for semiconducting barriers or INTERFACE S.I SITHERM for insulating barriers. If the barrier material has been reassigned as a semiconductor by using the SEMICONDUCTOR on the **MATERIAL** statement, you specify INTERFACE S.I SITHERM. You may additionally need to specify localization parameters. And finally, each **REGION** statement must use a unique value of the **QTREGION** number (x) and the maximum value for x is the total number of **QTREGION**s required to be set up using this method. It is advised to sequentially increment the **QTREGION** numbers by one.

This format is currently restricted to SIS.EL and SIS.HO models.

You may use a combination of **REGION** statements and **QTREGION** statements to set up several **QTREGION**s. The **REGION** statements must occur before the **QTREGION** statements, and the NUMBER parameter of the first **QTREGION** statement must be given a value of one more than the number of **QTREGION**s set up using the **REGION** statements.

To restrict the SIS.EL or SIS.HO model to a specific **QTREGION** specify the region through the QTREGION parameter on the **MODELS** statement. The default behavior is to apply the model to all defined **QTREGION**s.

Table 6-2 MODEL Statement Parameters

Parameter	Type	Default	Units
SIS.EL	Logical	False	
SIS.HO	Logical	False	
SIS.NLDERIVS	Logical	False	

Table 6-3 REGION Statement Parameters			
Parameter	**Type**	**Default**	**Units**
QTREGION	Integer		

Example

```
MODELS SIS.EL QTREGION=3 SIS.NLDERIVS
MODELS SHJ.HO QTREGION=2 SHJ.NLDERIVS
```

This applies the SIS model for electrons to **QTREGION** number 3 and the Single Heterojunction Tunneling model for holes to **QTREGION** 2. In both cases, non-local couplings and derivatives are included to improve convergence.

6.2.5 Quantum Barrier Trap Assisted Tunneling Model

Basic Model

The SIS.EL and SIS.HO models do not consider the possibility of indirect tunneling through the trap levels. The SIS.TAT model does include this, and has two levels of sophistication. The simplest model is the extension of the TAT.NONLOCAL model described in Section 3.3.3 "Traps and Defects". This can be viewed as being a non-local SRH recombination process through the barrier. The trap depth is specified by the TAT.NLDEPTH parameter on the **MODELS** statement. It is recommended to use the TAT.RELEI flag on the **MODELS** statement in order to make the trap depth relative to the Intrinsic Fermi level. The other two parameters are the basic recombination lifetimes, SISTAT.TN and SISTAT.TP. You can enable this model for some or all **QTREGION**s by specifying SIS.TAT on the **MODELS** statement. You can specify the parameter SIS.NLDERIVS on the **MODELS** statement to improve the convergence properties by including non-local Jacobian couplings.

Table 6-4 Parameters for the basic SIS.TAT model.				
Statement	**Parameter**	**Type**	**Default**	**Units**
MODELS	SIS.TAT	Logical	False	
MODELS	SISTAT.TN	Real	1.0e-6	s
MODELS	SISTAT.TP	Real	1.0e-6	s
MODELS	TAT.NLDEPTH	Real	0.0	eV
MODELS	TAT.RELEI	Logical	False	

Advanced Model

The Ielmini trap assisted tunneling model has been described in Section 3.6.7 "Gate Current Models", where it is applied to Metal-Insulator-Semiconductor structures, and it has also been

applied to Metal-Insulator-Metal structures in the subsequent section. You can use it for Quantum barrier structures by specifying ITAT.SC.EL, ITAT.SC.HO, or RTAT.SC on the **MODELS** statement along with SIS.TAT. The tunneling current through the barrier is coupled self-consistently with the current continuity equations on either side. The post-processing option does not exist, so the parameters ITAT.PP.EL, ITAT.PP.HO, RTAT.PP enable the equivalent self-consistent modes instead. The barrier material can be either an insulator, a semiconductor, or an insulator changed to a wide bandgap semiconductor using the SEMICONDUCTOR parameter on the **MODELS** statement.

Which model is most appropriate depends on the energy of the trap with respect to the bandgap of the semiconductors on either side of the barrier. If the energy is above the conduction band then the ITAT.SC.EL model will model the electron transfer. If the energy is below the valence band, then the ITAT.SC.HO model will model the hole transfer. Otherwise, the RTAT.SC option will model transfer of both carriers, including recombination at the trap centers. The model parameters are exactly the same as in the Ielmini models for MIS and MIM structures, and the traps can be set up using the **TRAP**, **DOPING**, or **DEFECTS** statements. The model can be applied to selected or all **QTREGION**s using the QTREGION parameter on the **MODELS** statement.

Table 6-5 Parameters for the Basic SIS.TAT Model

Statement	Parameter	Type	Default
MODELS	SIS.TAT	Logical	False
MODELS	ITAT.SC.EL	Logical	False
MODELS	ITAT.SC.HO	Logical	False
MODELS	RTAT.SC	Logical	False

6.2.6 Current flow mechanisms for Type II heterojunctions

A Type II heterojunction is a relatively abrupt junction between two semiconductors, where the conduction band of semiconductor 2 is at a lower electron energy than the conduction band of semiconductor 1, and the valence band energy of semiconductor 2 is at a lower energy than the valence band energy of material 1. We extend this definition to also include the case where the conduction band of material 2 is also at a lower energy than the valence band of material 1. See figure 6-7 for a schematic of the two cases. The thermionic transport model of section 6.2.2 and the SHJ.EL and SHJ.HO models of sections 6.2.3 will still apply for same carrier type transport in these junctions. The barrier tunneling models of section 6.2.4 and 6.2.5 will still also work for same carrier type transport in these materials.

At certain conditions of bias and doping there will be a large concentration of electrons in semiconductor 2 in close spatial proximity to a large concentration of holes in semiconductor 1. This will lead to a recombination current component, and ATLAS has two models for this vertical transport. In the Danielsson model, traps located at the interface mediate the current, and the local carrier densities at the interface are used to apply an SRH-like majority carrier recombination current across the interface. The other model is based on the non-local trap assisted tunneling model of section 6.2.5. It requires bulk traps to be present in one of the semiconductors, and relies on non-local tunneling into these traps by carriers from the other

semiconductor. Both models require the THERMIONIC model to be enabled on the INTERFACE statement for the type II interface.

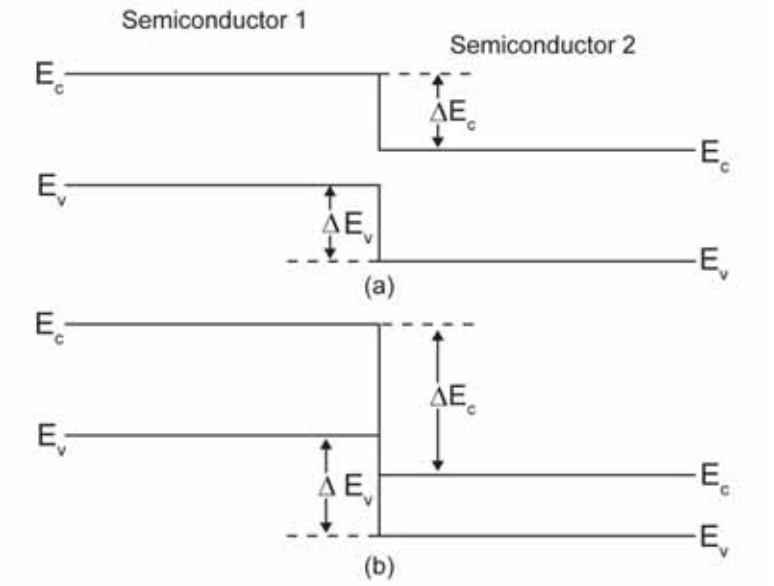


Figure 6-7: Schematic Type-II Heterojunctions

Danielsson model

Danielsson et al [1] proposed a model for interface tunneling between Gallium Nitride and 4H-SiC. This model is applicable to other type-II heterojunction interfaces. It assumes that a recombination center is located at the heterojunction and this gives a recombination between electrons on the lower Ec side of the heterojunction and holes on the higher E_v side of the heterojunction. Denoting these concentrations by n_2 and p_1 respectively, and their equilibrium values by n_2^{eq} and p_1^{eq}, the recombination rate is

$$U_{Dan} = \frac{(n_2 p_1 - n_2^{eq} p_1^{eq})}{(v_n^{-1}(n_2 + n_2^{eq} a_0) + v_p^{-1}(p_1 + p_1^{eq}/a_0))} \quad 6\text{-}55$$

where v_n is an electron recombination velocity calculated from

$$v_n = \texttt{DNLS.CN}\ \texttt{DNLS.NT} \quad 6\text{-}56$$

and DNLS.CN is a user specified capture coefficient in units of $cm^3 s^{-1}$ and DNLS.NT is the trap sheet density in cm^{-2}.

Similarly, V_P is a hole recombination velocity calculated from

$$v_p = \texttt{DNLS.CP}\ \texttt{DNLS.NT} \quad 6\text{-}57$$

The quantity a_0 is given by

$$a_0 = \exp(\texttt{DNLS.ET}/(K_b T_l)) \quad 6\text{-}58$$

where DNLS.ET is the trap level relative to the Fermi-Level in equilibrium, K_b is the Boltzmann constant and T_l is the Lattice temperature. All of these parameters are set on the INTERFACE statement.

The interface traps can be modelled as neutral or charged. To model as acceptor traps (to give a negative interface charge), you specify DNLS.ACC and to model as donor traps (to give a positive interface charge), you specify DNLS.DON. Omitting either of these parameters models neutral interface traps.

The model is enabled by setting the flag DANIELSSON on the INTERFACE statement. The flags S.S and THERMIONIC on the INTERFACE statement must also be set for this model to work properly. The localisation parameters of the INTERFACE statement can be used to restrict the domain of application of the MODEL.

Table 6-6 User Specified Parameters for the DANIELSSON Type-II Interface Tunneling Model

Statement	Parameter	Type	Default	Units
INTERFACE	DANIELSSON	Logical	False	
INTERFACE	DNLS.ACC	Logical	False	
INTERFACE	DNLS.DON	Logical	False	
INTERFACE	DNLS.CN	Real	0.0	cm^3s^{-1}
INTERFACE	DNLS.CP	Real	0.0	cm^3s^{-1}
INTERFACE	DNLS.NT	Real	0.0	cm^{-3}
INTERFACE	DNLS.ET	Real	0.0	eV

Trap Assisted Recommendation

The Danielsson model includes all details of tunnelling via the capture co-efficients and restricts the recombination to the interface. In the case where there are trap levels near to the valence band in Semiconductor 1, (see figure 6-8) and extending into Semiconductor 1, then electron tunneling under forward bias from the conduction band in Semiconductor 2 into these trap levels is possible. This will then augment hole capture from the valence band in Semiconductor 1 and result in a forward bias current. This requires detailed modelling of the tunnelling as a function of depth in Semiconductor 1.

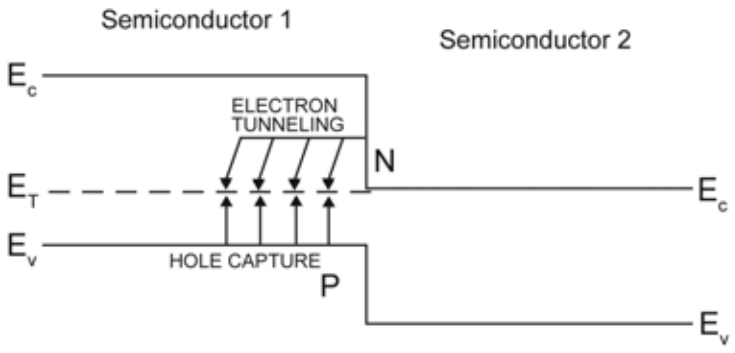


Figure 6-8: Trap Assisted Recombination Current For Type-II Heterojunctions

To include the tunneling from the interface, the thermal trap capture and emission rates as given in the right hand sides of equations 3-113 and 3-114 must have extra terms added. For donor traps, where $F_{tD}(x)$ is the probability that the trap is occupied by a hole,the terms

$$DENSITY(B_h(x)(f_h(E_{trap}, x) - F_{tD}(x)) + B_e(x)(1 - f_e(E_{trap}, x) - F_{tD}(x))) \quad 6\text{-}59$$

are added. Be and Bh are the rates for inelastic tunneling, for electrons and holes respectively, into the traps at distance x from the interface in Semiconductor 1. These rates are calculated from equation 3-543, using either the electron energy and electron tunneling mass or hole energy and hole tunneling mass. The quantities $f_h(E_{trap}, x)$ and $f_e(E_{trap}, x)$ are the occupation probabilities of the trap levels at distance x from the interface, but calculated using the relevant quasi-Fermi levels at the interface. The quantity f_h refers to a probability of finding a hole and f_e refers to probability of finding an electron.

To obtain the steady state occupation probabilities the time derivative on the left hand side of equation 3-113 is set to zero and the equation is solved for $f_{tD}(x)$. Dividing by DENSITY and writing $c_pD = v_pSIGP_p$ and adopting the shorthand of section 3.3.3, the resulting equation is

$$c_pD(1 - F_{tD}) - F_{tD}e_{pD} - c_{nD}F_{tD} + (1 - F_{tD})e_{nD} + B_h(x)(f_h(E_{trap}, x) - F_{tD}(x)) + B_e(x)(1 - f_e(E_{trap}, x) - F_{tD}(x)) = 0 \quad 6\text{-}60$$

this gives a value of trap occupation probability of

$$F_{tD} = \frac{c_{pD} + e_{nD} + B_h(x)f_h(E_{trap}, x) + B_e(x)(1 - f_e(E_{trap}, x))}{c_{pD} + e_{pD} + c_{nD} + e_{nD} + B_e(x) + B_h(x)} \quad 6\text{-}61$$

It can be seen from this equation that, for example, if the dominant term at a particular position is $B_h(x)$, then the trap occupation probability is equal to $f_h(E_{trap}, x)$, and thus controlled by the hole quasi-Fermi level at the interface. Similarly if the dominant term is $B_e(x)$, then the trap occupation probability is controlled by the electron quasi-Fermi level at the interface. Further from the interface $B_h(x)$ and $B_e(x)$ become negligible and the trap occupation probabilities are given by equations 3-85.

For acceptor traps, a similar analysis applies. The probability of the trap being occupied by an electron is F_{tA}. The steady state occupation probability with the inclusion of tunneling becomes

$$F_{tA} = \frac{c_{nA} + e_{pA} + B_h(x)(1 - f_h(E_{trap}, x)) + B_e(x)f_e(E_{trap}, x)}{c_{pA} + e_{pA} + c_{nA} + e_{nA} + B_e(x) + B_h(x)} \quad 6\text{-}62$$

Once the steady state occupation probability has been obtained, the nett capture rate of conduction band electrons and valence band holes in Semiconductor 1 can be calculated, as can the current due to the tunneling to or from the interface. These components are inserted into the current contunuity equations in the relevant Semiconductor to provide the current flow through the junction. In order to model this component of current flow in ATLAS you must follow these steps. First, define the traps in the material required using the TRAP statement, and using the flag HJ.TNL.TRAP to specify that tunneling into the traps is to be considered. Second, it is necessary to then create a QTREGION in the Semiconductor in which the traps occur, which starts at the interface between the two semiconductors and extends a user specified distance into the semiconductor with traps. This can only be done by using the QTREGION parameter on the REGION statement. and for this it is also necessary to make the interface thermionic using the INTERFACE S.S THERMIONIC syntax. The depth of the QTREGION is specified in microns by using the QT.EXTENT parameter. The

direction of tunneling is deduced automatically, but this can be explicitly overwritten by using the QT.XDIR or QT.YDIR flags. Finally the model can be enabled on the MODELS statement by using the flag ITAT.HJ.EL for electron tunneling or ITAT.HJ.HO for hole tunneling.

Table 6-7 User Specified Parameters for the Type-II Interface non-local Tunneling Model

Statement	Parameter	Type	Default	Units
MODELS	ITAT.HJ.EL	Logical	False	
MODELS	ITAT.HJ.HO	Logical	False	
REGION	QT.EXTENT	Real	0.0	microns
REGION	QT.XDIR	Logical	False	
REGION	QT.YDIR	Logical	False	
TRAP	HJ.TNL.TRAP	Logical	False	
MATERIAL	ME.TUNNEL	Real	Material Dependent	
MATERIAL	MH.TUNNEL	Real	Material Dependent	

Example

```
REGION NUMBER=2 QTREGION=1 QT.EXTENT=1.0E-2 QT.YDIR

INTERFACE S.S THERMIONIC

TRAP ACCEPTOR E.LEVEL=5.0 DEGEN.FAC=2 DENSITY=1E18 SIGN=1E-15
SIGP=1E-18 MATERIAL=GAASP

HJ.TNL.TRAP

MODELS ITAT.HJ.EL
```

The analagous situation for holes tunneling from the valence band in Semiconductor 1 to traps in the conduction band in Semiconductor 2 can also be modelled.

6.3 The Physical Models

Section 3.6 “Physical Models” already describes a comprehensive set of physical models for silicon. It is known, however, that for III-V, II-VI, and ternary compounds that special consideration are required such as for mole fraction dependence and material properties. The following sections describe the material dependent physical models that have been implemented into Blaze to account for these effects.

First, a description of the common mobility model equations is given. These equations are applicable to all materials unless stated in the following material sections. Following this are sections describing the physical models on a material-per-material basis. For each material, there are descriptions of the models for bandgap narrowing, electron affinity, density of states, dielectric permittivity, and low field mobility.

6.3.1 Common Physical Models

Low Field Mobility Models

The default low field mobility models used for most materials in Blaze are given by the following:

$$\mu_{n0}(T_L) = \mathtt{MUN}\left(\frac{T_L}{300}\right)^{-\mathtt{TMUN}} \quad \text{6-63}$$

$$\mu_{p0}(T_L) = \mathtt{MUP}\left(\frac{T_L}{300}\right)^{-\mathtt{TMUP}} \quad \text{6-64}$$

where T_L is the temperature in degrees Kelvin and the MUN, MUP, TMUN, and TMUP parameters are user-definable as shown in Table 6-8.

Note: All the mobility models described in Chapter 3 “Physics”, except for the TASCH model, can be used in Blaze. But all default coefficients exist only for Silicon and therefore are not suitable for compound materials.

Table 6-8 User-Specifiable Parameters for Equations 6-63 and 6-64

Statement	Parameter	Units
MOBILITY	MUN	cm2/V·s
MOBILITY	TMUN	
MOBILITY	MUP	cm2/V·s
MOBILITY	TMUP	

Parallel Electric Field-Dependent Mobility Models

There are two types of electric field-dependent mobility models used in Atlas/Blaze. These models are called Standard Mobility Model and Negative Differential Mobility Model. Both of these models contain appropriate default values of parameters for different materials. You need to specify which type of mobility will be used for each material and which material parameters you want to alter.

The Standard Mobility Model that takes account of velocity saturation is defined according to:

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}E}{\mathrm{VSATN}}\right)^{\mathrm{BETAN}}}\right]^{1/\mathrm{BETAN}} \quad 6\text{-}65$$

$$\mu_p(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}E}{\mathrm{VSATP}}\right)^{\mathrm{BETAP}}}\right]^{1/\mathrm{BETAP}} \quad 6\text{-}66$$

where VSATN and VSATP are the saturation velocities for electrons and holes, BETAN and BETAP are constants given in Table 6-9, and $\mu_{n0,p0}$ are the electron and hole low field mobilities. This model is activated by specifying the FLDMOB and the EVSATMOD=0 parameter in the **MODELS** statement.

Table 6-9 User-Specifiable Parameters for Equations 6-65 and 6-66

Statement	Parameter	Units
MOBILITY	BETAN	
MOBILITY	BETAP	
MOBILITY	VSATN	cm/s
MOBILITY	VSATP	cm/s

The Negative Differential Mobility Model of Barnes et. al. [23] has been implemented to account for certain devices where the carrier drift velocity peaks at some electric field before reducing as the electric field increases. This model takes account of this through the carrier mobility with equations of the form:

$$\mu_n(E) = \frac{\mu_{n0} + \frac{\mathrm{VSATN}}{E}\left(\frac{E}{\mathrm{ECRITN}}\right)^{\mathrm{GAMMAN}}}{1+\left(\frac{E}{\mathrm{ECRITN}}\right)^{\mathrm{GAMMAN}}} \quad 6\text{-}67$$

$$\mu_p(E) = \frac{\mu_{p0} + \frac{\text{VSATP}}{E}\left(\frac{E}{\text{ECRITP}}\right)^{\text{GAMMAP}}}{1 + \left(\frac{E}{\text{ECRITP}}\right)^{\text{GAMMAP}}} \quad \text{6-68}$$

where VSATN and VSATP are the electron and hole saturation velocities, E_0 is a constant, and $\mu_{n0,p0}$ are the low-field electron and hole mobilities. To activate this mobility model, specify EVSATMOD=1 in the **MODELS** statement.

Table 6-10 User-Specifiable Parameters for Equations 6-67 and 6-68

Statement	Parameter	Default	Units
MOBILITY	ECRITN	4.0×10^3	V/cm
MOBILITY	ECRITP	4.0×10^3	V/cm
MOBILITY	GAMMAN	4.0	
MOBILITY	GAMMAP	1.0	

Note: The Negative Differential Mobility Model introduces an instability in the solution process and is not recommended for general use. Only activate this model in cases where the device operation directly depends on negative differential mobility (e.g., a Gunn diode).

For both the standard and negative differential models, an empirical temperature-dependent model for saturation velocity in GaAs [186] is implemented according to:

$$\text{VSATN} = \text{VSATN.T0} - \text{VSATN.T1}\ T_L \quad \text{6-69}$$

$$\text{VSATP} = \text{VSATP.T0} - \text{VSATP.T1}\ T_L \quad \text{6-70}$$

where VSATN and VSATP are expressed in cm/sec and T_L is the temperature in degrees Kelvin.

Alternatively, you can set the saturation velocities to constant values using the VSATN and VSATP parameters of the **MATERIAL** statement.

Table 6-11 User-Specifiable Parameters for Equations 6-69 and 6-70

Statement	Parameter	Default	Units
MOBILITY	VSATN.T0	1.13×10^7	cm/s
MOBILITY	VSATN.T1	1.2×10^4	cm/s·K
MOBILITY	VSATP.T0	1.13×10^7	cm/s
MOBILITY	VSATP.T1	1.2×10^4	cm/s·K

Velocity Saturation with Energy Balance Transport Model

When the Energy Balance Transport Model is activated, the mobility can be made a function of carrier energy. In Section 3.6.1 “Mobility Modeling”, physical models for the dependence of carrier mobility on carrier energy were introduced. The same models are applicable for use within Blaze with one additional model, which applies when the negative differential mobility model is used.

The carrier temperature dependence is activated when the EVSATMOD=1 parameter is on the **MODELS** statement. This model can be derived in a similar fashion as in the case of EVSATMOD=0 described in “Parallel Electric Field-Dependent Mobility” on page 205. These expressions, however, require several piecewise approximations, which are too in-depth to show in this manual. To say that these piecewise expressions provide a continuous velocity saturation model for mobility versus carrier temperature are completely like the expressions for drift diffusion given in Equations 3-341 and 3-342.

6.3.2 Recombination and Generation Models

The recombination and generation models for compound semiconductors are the same as the models previously described in Section 3.6.3 “Carrier Generation-Recombination Models”. The default parameters for different materials are automatically used, unless new values are specified. The default parameter values are listed in Appendix B “Material Systems”.

6.4 Material Dependent Physical Models

6.4.1 Cubic III-V Semiconductors

The basic band parameters (*Eg*, χ, ε, *Nc*, and *Nv*) of most III-V cubic can be modeled using composition dependent interpolation schemes from the parameters of the constituent binary semiconductors. Table 6-12 lists the compound semiconductors you can model using this approach. The first column of this table gives the material name used on the **REGION**, **DOPING**, **ELECTRODE**, **MODELS**, **MATERIAL**, **INTERFACE**, **IMPACT**, **MOBILITY**, **TRAP**, **DEFECTS**, **PROBE**, **LASER**, and **ODEFECTS** statements. The second column of Table 6-12 shows the association of composition fractions with the constituent elements. The third column indicates whether this model is used by default (see Section 3.2.5 "Rules for Evaluation of Energy Bandgap"). If this model is not used, then there is another default model for that material described in the section listed in the last column of the table.

You can apply the Cubic III-V model to materials, which use an alternative model described in the **Section** column of Table 6-12, by setting the CUBIC35 parameter of the **MODELS** statement.

In specifying a material name, Atlas will recognize alternate spellings switching the order of cations or anions as long as you remember to maintain the associations of elements with composition fractions as given in Table 6-12 (e.g., Al(x)Ga(1-x)As = Ga(1-x)Al(x)As).

Table 6-12 Cubic III-V Interpolation Model Applicability Chart

Atlas Name	Composition	Default	Section
GaAs		No	6.4.2
AlP		Yes	
AlAs		No	6.4.3
AlSb		Yes	
GaSb		Yes	
GaP		No	6.4.4
InP		No	6.4.4
InSb		Yes	
InAs		No	6.4.4
AlGaAs	Al(x)Ga(1-x)As	No	6.4.3
InAlP	In(1-x)Al(x)P	Yes	
InGaAs	In(1-x)Ga(x)As	No	6.4.4
InGaP	In(1-x)Ga(x)P	No	6.4.4
GaSbP	GaSb(1-y)P(y)	Yes	
InAlAs	In(1-x)Al(x)As	Yes	

Table 6-12 Cubic III-V Interpolation Model Applicability Chart			
InAsP	InAs(1-y)P(y)	No	6.4.4
GaAsP	GaAs(1-y)P(y)	No	6.4.4
InGaSb	In(1-x)Ga(x)Sb	Yes	
InAlSb	In(1-x)Al(x)Sb	Yes	
AlGaSb	Al(x)Ga(1-x)Sb	Yes	
InAsSb	InAs(y)Sb(1-y)	Yes	
GaAsSb	GaAs(y)Sb(1-y)	Yes	
AlAsSb	AlAs(y)Sb(1-y)	Yes	
InPSb	InSb(1-y)P(y)	Yes	
AlPSb	AlP(y)Sb(1-y)	Yes	
AlPAs	AlP(y)As(1-y)	Yes	
AlGaP	Al(x)Ga(1-x)P	Yes	
InPAsSb	InP(x)As(y)Sb(1-x-y)	Yes	
InGaAsP	In(1-x)Ga(x)As(1-y)P(y)	No	6.4.4 (See the Note below)
AlGaAsP	Al(x)Ga(1-x)As(1-y)P(y)	Yes	
AlGaAsSb	Al(x)Ga(1-x)As(y)Sb(1-y)	Yes	
InAlGaAs	In(1-x-y)Al(x)Ga(y)As	Yes	
InAlGaP	In(1-x-y)Al(x)Ga(y)P	Yes	
InAlAsP	In(1-x)Al(x)As(1-y)P(y)	Yes	
InGaAsSb	In(1-x)Ga(x)As(y)Sb(1-y)	Yes	
InAlAsSb	In(1-x)Al(x)As(y)Sb(1-y)	Yes	

Note: Materials in the InGaAsP system by default use the models in Section 6.4.3 “Al(x)Ga(1-x)As System”. Make sure that the usage of the y composition fraction is different in the two models.

Energy Bandgap

The energy bandgaps of the binary compounds are calculated for transitions in each of the Γ, X and L directions using the universal formula for temperature dependent bandgap (see Equation 3-38). Table 6-7 shows the default parameters for Equation 3-38 for the binary compounds.

Table 6-13 Default Bandgap Parameters for Cubic III-V Binary Compounds [246]

Binary	$Eg^{\Gamma}(0)$ (eV)	α^{Γ} (meV/K)	β^{Γ} (K)	$Eg^{X}(0)$ (eV)	α^{X} (meV/K)	β^{X} (K)	$Eg^{L}(0)$ (eV)	α^{L} (meV/K)	β^{L} (K)
GaAs	1.519	0.5405	204.0	1.981	0.4600	204.0	1.815	0.6050	204
AlP	3.63	0.5771	372.0	2.52	0.3180	588.0	3.57	0.3180	588
AlAs	3.099	0.885	530.0	2.240	0.700	530.0	2.460	0.605	240
AlSb	2.386	0.42	140.0	1.696	0.39	140.0	2.329	0.58	140
GaSb	0.812	0.417	140.0	1.141	0.475	94.00	0.875	0.597	140
GaP	2.886	0.0	0.0	2.350	0.5771	372.0	2.720	0.5771	372
InP	1.4236	0.363	162.0	2.3840	0.0	0.0	2.0140	0.363	162
InSb	0.235	0.32	170.0	0.630	0.00	0.0	0.930	0.00	0.0
InAs	0.417	0.276	93.0	1.433	0.276	93.0	1.133	0.276	93

For the ternary materials the bandgap in each principal direction is approximated by the function [2]:

$$Tabc(x) = xTac + (1-x)Tbc - x(1-x)Cabc \quad \text{6-71}$$

where *Tabc* is the ternary bandgap, *Tac* and *Tbc* are the binary bandgaps, *x* is the composition fraction and *Cabc* is the bowing factor as given for each of the ternary compounds in each of the principal directions in Table 6-14.

Table 6-14 Bowing Factors and Alignment for Cubic III-V Ternary Compounds [246]

Ternary	$C(Eg^{\Gamma})$ (eV)	$C(Eg^{X})$ (eV)	$C(Eg^{L})$ (eV)	dEc/dEg
AlGaAs	1.31*x-0.127	0.055	0.0	0.65
InAlP	-0.48	0.38	0.0	0.5
InGaAs	0.477	1.4	0.33	0.4
InGaP	0.65	0.2	1.3	0.4
GaSbP	2.558	2.7	2.7	0.5
GaSbAs	1.43	1.2	1.2	0.5
InAlAs	0.7	0.0	0.0	0.7
InAsP	0.1	0.27	0.27	0.4

Table 6-14 Bowing Factors and Alignment for Cubic III-V Ternary Compounds [246]				
GaAsP	0.19	0.24	0.16	0.4
InGaSb	0.415	0.330	0.4	0.5
InAlSb	0.43	0.0	0.0	0.5
AlGaSb	1.22*x-0.044	0.0	0.0	0.5
InAsSb	0.67	0.6	0.6	0.5
GaAsSb	1.43	1.2	1.2	0.5
AlAsSb	0.84	0.28	0.28	0.5
InPSb	1.9	1.9	1.9	0.5
AlPSb	3.56	2.7	2.7	0.5
AlPAs	0.22	0.22	0.22	0.5
AlGaP	0.0	0.13	0.0	0.5

For the quaternary compounds, the bandgap in each principal direction is approximated by the function [2]:

$$Qabcd(x,y) = \frac{x(1-x[yTabc(x)+(1-y)Tabd(x)]+y(1-y)[xTacd(y)+(1-x)Tbcd(y)])}{x(1-x)+y(1-y)} \quad 6\text{-}72$$

or for compounds with three anions [2]:

$$Qabcd(x,y) = \frac{xyTabc(u)+y(1-x-y)Tacd(v)+(1-x-y)xTabd(w)}{xy+y(1-x-y)+(1-x-y)x} \quad 6\text{-}73$$

where *Tabc*, *Tabd*, *Tacd*, and *Tbcd* are the ternary bandgaps, x and y are the composition fractions and u, v and w are given as follows:

$$u = (1-x+y)/2 \quad 6\text{-}74$$

$$v = (2-x-2y)/2 \quad 6\text{-}75$$

and

$$w = (2-2x-y)/2 \quad 6\text{-}76$$

For any case (binary, ternary or quaternary), the bandbap used in the drift-diffusion calculations is taken as the minimum of Eg^{Γ}, Eg^{χ} or Eg^{L}.

Electron Affinity

Table 6-15 shows the electron affinities for the binary compounds in the given direction in *k* space.

Table 6-15 Default Electron Affinities and Alignments for Cubic III-V Binary Compounds [246]

Binary	Affinity (eV)	Direction	dEc/dEg
GaAs	4.07	Γ	0.65
AlP	3.98	X	0.7
AlAs	3.85	X	0.65
AlSb	3.60	X	0.5
GaSb	4.06	Γ	0.5
GaP	4.0	X	0.4
InP	4.4	Γ	0.4
InSb	4.59	Γ	0.5
InAs	4.90	Γ	0.65

The affinities for the other directions in *k* space not given in Table 6-15 are calculated by the following:

$$\chi^{\Gamma} = \chi^{X} - (Eg^{X} - Eg^{\Gamma})(dEc/dEg) \quad 6\text{-}77$$

$$\chi^{\chi} = \chi^{\Gamma} - (Eg^{\Gamma} - Eg^{X})(dEc/dEg) \quad 6\text{-}78$$

$$\chi^{L} = \chi^{X/\Gamma} - (Eg^{X/\Gamma} - Eg^{L})(dEc/dEg) \quad 6\text{-}79$$

where X^{Γ}, X^{X} and X^{L} are the affinities in each of the principal directions, Eg^{Γ}, Eg^{X} and Eg^{L} are the energy bandgaps in each of the principal directions as calculated above and dEc/dEg is an alignment parameter describing the proportion of change in bandgap that can be associated with the conduction band edge. Table 6-15 lists the default values of dEc/dEg. You can define the value of dEc/dEg by using the `DEC.DEG` parameter of the **`MATERIAL`** statement.

For ternary compounds, the affinities are calculated from the binary values using Equation 6-71 except *Tabc* is the ternary affinity, *Tac* and *Tbc* are the binary affinities and *Cabc* is the bowing factor for affinity which is given by:

$$Cabc(\chi) = -Cabc(Eg)dEc/dEg \quad 6\text{-}80$$

where *Cabc(χ)* is the bowing factor for affinity, *Cabc(Eg)* is the bowing factor for bandgap as given in Table 6-14 and dEc/dEg is the alignment parameter as given in Table 6-14.

For quaternaries, the affinities are approximated by the Equations 6-72 and 6-73 with affinities replacing energy bandgaps.

Finally for all cases (binary, ternary and quaternary), the affinity used in the drift-diffusion calculation is chosen as the one in the same direction (Γ, X, or L) as chosen for energy bandgap.

Density of States

For the III-V cubic semiconductor model, the density of states (*Nc* and *Nv*) are each calculated from the effective masses using Equations 3-31 and 3-32. For binary materials, the density of states effective masses are given in Table 6-16.

Table 6-16 Default Cubic III-V Effective Masses and Permittivities

Binary	m_c^Γ (×mo)	m_{ct}^X (×mo)	m_{cl}^X (×mo)	m_{ct}^Γ (×mo)	m_{cl}^Γ (×mo)	m_{hh} (×mo)	m_{lh} (×mo)	ε_r
GaAs	0.067	0.23	1.3	0.0754	1.9	0.49	0.16	12.91
AlP	0.22	0.155	2.68	0.0	0.0	0.63	0.20	9.8
AlAs	0.15	0.22	0.97	0.15	1.32	0.76	0.15	10.06
AlSb	0.14	0.123	1.357	0.23	1.64	0.94	0.14	12.04
GaSb	0.039	0.22	1.51	0.10	1.3	0.28	0.05	15.69
GaP	0.13	0.253	2.0	0.15	1.2	0.79	0.14	11.10
InP	0.0795	0.88	0.0	0.47	0.0	0.56	0.12	12.61
InSb	0.0135	0.0	0.0	0.25	0.0	0.43	0.015	17.70
InAs	0.026	0.16	1.13	0.05	0.64	0.57	0.025	15.15

For ternary materials, the effective masses are interpolated using the following expression [2]:

$$Mabc = xMac + (1-x)Mbc \quad (6\text{-}81)$$

where *Mabc* is the ternary effective mass, *Mac* and *Mbc* are the binary effective masses and x is the composition fraction.

For quaternary compounds the effective masses are interpolated using the following expression[2]:

$$Mabcd = xyMac + x(1-y)Mad + (1-x)yMbc + (1-x)(1-y)Mbd \quad (6\text{-}82)$$

or for compounds with 3 anions [2]:

$$Mabcd = xMab + yMab + (1-x-y)Mad \quad (6\text{-}83)$$

where *Mabcd* is the quaternary effective mass, *Mac*, *Mad*, *Mbc* and *Mbd* are the binary effective masses and *x* and *y* are the composition fractions.

For the calculation of conduction band density of states, the conduction band density of states mass is calculated for the same direction (Γ, X, or L) as is used for energy bandgap as follows:

$$m_c = m_c^\Gamma \quad (6\text{-}84)$$

$$m_c = [(m_{ct}^X)^2 \cdot (m_{cl}^X)]^{1/3} \quad (6\text{-}85)$$

or

$$m_c = [(m_{ct}^L)^2 \cdot (m_{cl}^L)]^{1/3} \quad 6\text{-}86$$

For the calculation of valence band density of states, the valence band density of states mass is calculated as follows:

$$m_v = (m_{hh}^{3/2} + m_{lh}^{3/2})^{2/3} \quad 6\text{-}87$$

Static Permittivity

Table 6-16 shows the permittivities of the binary compounds. The ternary and quaternary permittivities are approximated by the interpolation formulas analogous to those used for effective mass (Equations 6-81, 6-82, and 6-83) replacing masses by permittivities.

6.4.2 Gallium Arsenide (GaAs) Physical Models

Bandgap Narrowing

Following Klausmeier-Brown [161], the bandgap narrowing effects are important only for p-type regions. By default, Blaze uses the bandgap narrowing values shown in Table 6-17.

Table 6-17 Default Bandgap Narrowing Values

Concentration (cm^{-3})	Bandgap Narrowing (meV)
1.0×10^{18}	31.0
2.0×10^{18}	36.0
4.0×10^{18}	44.2
6.0×10^{18}	48.5
8.0×10^{18}	51.7
1.0×10^{19}	54.3
2.0×10^{19}	61.1
4.0×10^{19}	64.4
6.0×10^{19}	61.9
8.0×10^{19}	56.9
1.0×10^{20}	53.2
2.0×10^{20}	18.0

Note: This table is only used for GaAs. No data is available at present for other materials. The C-Interpreter function for bandgap narrowing, however, allows a user-defined model for bandgap narrowing to be applied for these materials. See Appendix A "C-Interpreter Functions" for more information about on these functions.

Low Field Mobility

You can make the mobility in GaAs concentration dependent by setting the CONMOB parameter of the **MODELS** statement. In this model, mobility is interpolated from the values in Table 6-18.

Table 6-18 Default Concentration Dependent Mobility for GaAs

Concentration (cm^{-3})	Mobility in GaAs (cm^2/v-s)	
	Electrons	Holes
1.0×10^{14}	8000.0	390.0
2.0×10^{14}	7718.0	380.0
4.0×10^{14}	7445.0	375.0
6.0×10^{14}	7290.0	360.0
8.0×10^{14}	7182.0	350.0
1.0×10^{15}	7300.0	340.0
2.0×10^{15}	6847.0	335.0
4.0×10^{15}	6422.0	320.0
6.0×10^{15}	6185.0	315.0
8.0×10^{15}	6023.0	305.0
1.0×10^{16}	5900.0	302.0
2.0×10^{16}	5474.0	300.0
4.0×10^{16}	5079.0	285.0
6.0×10^{16}	4861.0	270.0
8.0×10^{16}	4712.0	245.0
1.0×10^{17}	4600.0	240.0
2.0×10^{17}	3874.0	210.0
4.0×10^{17}	3263.0	205.0
6.0×10^{17}	2950.0	200.0

Table 6-18 Default Concentration Dependent Mobility for GaAs		
Concentration (cm^{-3})	Mobility in GaAs (cm^2/v-s)	
	Electrons	Holes
8.0×10^{17}	2747.0	186.9
1.0×10^{18}	2600.0	170.0
2.0×10^{18}	2060.0	130.0
4.0×10^{18}	1632.0	90.0
6.0×10^{18}	1424.0	74.5
8.0×10^{18}	1293.0	66.6
1.0×10^{20}	1200.0	61.0

If you specify the ANALYTIC flag on the **MODELS** statement, Equations 3-207 and 3-208 are used. The default parameter values for these equations for GaAs are shown in Table 6-19.

Table 6-19 ANALYTIC Model Parameters for GaAs				
Statement	Parameter	Type	Default	Units
MOBILITY	MU1N.CAUG	500.0	Real	$cm^2/(V\ s)$
MOBILITY	MU1P.CAUG	20.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2N.CAUG	9400.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2P.CAUG	491.5	Real	$cm^2/\ (V\ s)$
MOBILITY	ALPHAN.CAUG	0.0	Real	–
MOBILITY	ALPHAP.CAUG	0.0	Real	–
MOBILITY	BETAN.CAUG	-2.1	Real	–
MOBILITY	BETAP.CAUG	-2.2	Real	–
MOBILITY	GAMMAN.CAUG	-1.182	Real	–
MOBILITY	GAMMAP.CAUG	-1.14	Real	–
MOBILITY	DELTAN.CAUG	0.394	Real	–

Table 6-19 ANALYTIC Model Parameters for GaAs				
MOBILITY	DELTAP.CAUG	0.38	Real	–
MOBILITY	NCRITN.CAUG	6×10^{16}	Real	cm^{-3}
MOBILITY	NCRITP.CAUG	1.48×10^{17}	Real	cm^{-3}

The Sotoodeh low field mobility model [298] is also available for GaAs. Specify N.SOTOODEH and P.SOTOODEH on the **MOBILITY** statement to enable it for electrons and holes respectively.

6.4.3 Al(x)Ga(1-x)As System

The Al(x)Ga(1-x)As material system is commonly used for the fabrication of heterojunction devices. These materials are available in Blaze by specifying the material name GaAs, AlGaAs, or AlAs. As a ternary material system, different material properties are obtained by adjusting the molar fraction of Aluminum and Gallium. This mole fraction is represented by the x as written in Al(x)Ga(1-x)As. GaAs material parameters are identical to the those of AlGaAs with the mole fraction x set equal to zero. AlAs material parameters are identical to the those of AlGaAs with mole the fraction x set equal to one. Fundamental in the proper simulation with the AlGaAs material system is the relationship between this mole fraction x, and the material parameters for that composition. In the following sections, the relationship between mole fraction and material parameters for the AlGaAs material system will be described. You can specify the x-composition fraction of a material in the **REGION** statement with the X.COMP parameter.

Bandgap

There are three primary conduction bands in the AlGaAs system, depending on mole fraction, that determine the bandgap. These are named Gamma, L, and X. The default bandgaps for each of these conduction band valleys are as follows:

$$E_{g\Gamma} = \mathrm{EG300} + \mathrm{x.comp} \cdot (1.155 + 0.37 \cdot \mathrm{X.COMP}) \quad (6\text{-}88)$$

$$E_{gL} = 1.734 + \mathrm{x.comp} \cdot (0.574 + 0.055 \cdot \mathrm{X.COMP}) \quad (6\text{-}89)$$

$$E_{gX} = 1.911 + \mathrm{x.comp} \cdot (0.005 + 0.245 \cdot \mathrm{X.COMP}) \quad (6\text{-}90)$$

The bandgap used for any given Al concentration is the minimum as calculated from these equations. EG300 is the bandgap at 300K and specified on the material statement. x.composition is the Aluminum mole fraction and can be user-defined in the REGION statement.

The temperature dependence of the bandgap is calculated according to:

$$E_g(T_L) = E_g(300) + \mathrm{EGALPHA} \cdot \left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}} \right] \quad (6\text{-}91)$$

The value of $E_g(300)$ is taken as the minimum of $E_{g\Gamma}$, E_{gx}, and E_{gL}. The default temperature dependent bandgap parameters for AlGaAs are listed in Table 6-20.

Table 6-20 Default Bandgap Parameters for Al(x)Ga(1-x)As

Statement	Parameter	Default	Units
MATERIAL	EG300	1.59	eV
MATERIAL	EGALPHA	5.405×10^{-4}	eV/K
MATERIAL	EGBETA	204	K

Electron Affinity

As indicated in the introduction, the semiconductor electron affinity χ is a key parameter for determining the alignment of heterojunctions. For AlGaAs, χ is a function of $E_{g\Gamma}$ and is given by:

$$\chi_{AlGaAs} = 4.07 - 0.85 \cdot (E_{g\Gamma}(\mathrm{X.COMP}) - E_{gGaAs}) \quad (6\text{-}92)$$

Density of States and Effective Mass

The valence and conduction band densities of states, N_C and N_v, are calculated from the effective masses according to the following equations:

$$N_c = 2\left(\frac{2\pi m_e^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad (6\text{-}93)$$

$$N_v = 2\left(\frac{2\pi m_h^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad (6\text{-}94)$$

For the AlGaAs system the conduction band and valence band effective masses, for electrons and holes, are given by [192]:

$$m_e = \begin{matrix} 0.067 + 0.083x (0 < x < 0.45) \\ 0.85 - 0.14x (x > 0.45) \end{matrix} \quad (6\text{-}95)$$

$$m_{lh} = 0.087 + 0.063x \quad (6\text{-}96)$$

$$m_{hh} = 0.62 + 0.14x \quad (6\text{-}97)$$

$$m_h = (m_{lh}^{3/2} + m_{hh}^{3/2})^{2/3} \quad (6\text{-}98)$$

Dielectric Permittivity

The default static dielectric constant for AlGaAs is given by:

$$\varepsilon_{AlGaAs} = 13.18 - 2.9 \cdot \mathrm{X.COMP} \quad \text{6-99}$$

Low Field Mobility

The default low field electron mobility for AlGaAs is a function of the composition fraction, x, within the system. The following equations outline this relationship.

Table 6-21 Low Field Mobility Equations

AlGaAs Low Field Mobility	Mole Fraction Range
$\mu_n = 8000 - (1.818\times10^4 \cdot \mathrm{X.COMP})$	$(0 < \mathrm{X.COMP} < 0.429)$
$\mu_n = 90 + 1.1435\times10^4 \cdot (\mathrm{X.COMP} - 0.46)^2$	$(0.429 < \mathrm{X.COMP} < 0.46)$
$\mu_n = 90 + 3.75\times10^4 \cdot (\mathrm{X.COMP} - 0.46)^2$	$(0.46 < \mathrm{X.COMP} < 0.5)$
$\mu_n = 200 - (2/(\mathrm{X.COMP}-0.46))$	$(0.5 < \mathrm{X.COMP} < 1.0)$

In the case of AlAs (but not AlGaAs alloy), Equations 3-207 and 3-208 can be used. You specify ANALYTIC on the **MODELS** statement to enable these. The default parameter values for these equations for AlAs are shown in Table 6-22.

Table 6-22 ANALYTIC Model Parameters for AlAs

Statment	Parameter	Type	Default	Units
MOBILITY	MU1N.CAUG	10.0	Real	$cm^2/(V\ s)$
MOBILITY	MU1P.CAUG	10.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2N.CAUG	400.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2P.CAUG	200.0	Real	$cm^2/(V\ s)$
MOBILITY	ALPHAN.CAUG	0.0	Real	–
MOBILITY	ALPHAP.CAUG	0.0	Real	–
MOBILITY	BETAN.CAUG	-2.1	Real	–
MOBILITY	BETAP.CAUG	-2.24	Real	–
MOBILITY	GAMMAN.CAUG	-3.0	Real	–
MOBILITY	GAMMAP.CAUG	-1.464	Real	–
MOBILITY	DELTAN.CAUG	1.0	Real	–

Table 6-22 ANALYTIC Model Parameters for AlAs				
MOBILITY	DELTAP.CAUG	0.488	Real	–
MOBILITY	NCRITN.CAUG	5.46×10^{17}	Real	cm^{-3}
MOBILITY	NCRITP.CAUG	3.48×10^{17}	Real	cm^{-3}

The Sotoodeh low field mobility model [298] is also available for AlAs and AlGaAs. Specify N.SOTOODEH and P.SOTOODEH on the **MOBILITY** statement to enable it for electrons and holes respectively.

6.4.4 In(1-x)Ga(x)P System

The In(1-x)Ga(x)P system is commonly used in optoelectronics, especially solar cells, that are lattice matched to GaAs. Particularly, important is are composition fractions around 0.5.

Bandgap

The default energy bandgap for InGaP latticed matched to GaAs used in Blaze is given by Equation 6-100 [199].

$$Eg = 1.35 + X.COMP*0.73 + X.COMP*X.COMP*0.7 \quad \text{6-100}$$

Electron Affinity

The default electron affinity for InGaP lattice matched to GaAs used in Blaze is given by Equation 6-101 [140].

$$\chi = 4.38 - 0.58*X.COMP \quad \text{6-101}$$

Density of States Effective Mass

The electron density of states effective mass is given by Equations 6-102 and 6-103 [140].

$$m_c = 0.088 \qquad X.COMP < 0.74 \quad \text{6-102}$$

$$m_c = 0.63 + 0.13*X.COMP \qquad X.COMP \geq 0.74 \quad \text{6-103}$$

The heavy hole and light hole density of states effective masses are given by Equations 6-104 and 6-105 [140].

$$m_{hh} = 0.6 + 0.19*X.COMP \quad \text{6-104}$$

$$m_{lh} = 0.09 + 0.05*X.COMP \quad (6\text{-}105)$$

Dielectric Permittivity

The electron density of states effective mass is given by Equations 6-102 and 6-103 [140].

$$\varepsilon = 12.5 - 1.4*X.COMP \quad (6\text{-}106)$$

Low Field Mobility

In the case of InP and GaP (but not InGaP alloy), Equations 3-207 and 3-208 can be used. You specify ANALYTIC on the **MODELS** statement to enable these. The default parameter values for these equations for GaP are shown in Table 6-23. The default parameter values for these equations for InP are shown in Table 6-24.

Table 6-23 ANALYTIC Model Parameters for GaP

Statment	Parameter	Type	Default	Units
MOBILITY	MU1N.CAUG	10.0	Real	$cm^2/(V\ s)$
MOBILITY	MU1P.CAUG	10.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2N.CAUG	152.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2P.CAUG	147.0	Real	$cm^2/\ (V\ s)$
MOBILITY	ALPHAN.CAUG	0.0	Real	–
MOBILITY	ALPHAP.CAUG	0.0	Real	–
MOBILITY	BETAN.CAUG	-1.6	Real	–
MOBILITY	BETAP.CAUG	-1.98	Real	–
MOBILITY	GAMMAN.CAUG	-0.568	Real	–
MOBILITY	GAMMAP.CAUG	-0.0	Real	–
MOBILITY	DELTAN.CAUG	0.8	Real	–
MOBILITY	DELTAP.CAUG	0.85	Real	–
MOBILITY	NCRITN.CAUG	4.4×10^{18}	Real	cm^{-3}
MOBILITY	NCRITP.CAUG	1.0×10^{18}	Real	cm^{-3}

Table 6-24 ANALYTIC Model Parameters for InP				
Statment	**Parameter**	**Type**	**Default**	**Units**
MOBILITY	MU1N.CAUG	400.0	Real	$cm^2/(V\ s)$
MOBILITY	MU1P.CAUG	10.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2N.CAUG	5200.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2P.CAUG	170.0	Real	$cm^2/\ (V\ s)$
MOBILITY	ALPHAN.CAUG	0.0	Real	–
MOBILITY	ALPHAP.CAUG	0.0	Real	–
MOBILITY	BETAN.CAUG	-2.0	Real	–
MOBILITY	BETAP.CAUG	-2.0	Real	–
MOBILITY	GAMMAN.CAUG	-1.5275	Real	–
MOBILITY	GAMMAP.CAUG	-1.86	Real	–
MOBILITY	DELTAN.CAUG	0.47	Real	–
MOBILITY	DELTAP.CAUG	0.62	Real	–
MOBILITY	NCRITN.CAUG	3.0×10^{17}	Real	cm^{-3}
MOBILITY	NCRITP.CAUG	4.87×10^{17}	Real	cm^{-3}

The Sotoodeh low field mobility model [298] is also available for InP, GaP and InGaP. Specify N.SOTOODEH and P.SOTOODEH on the **MOBILITY** statement to enable it for electrons and holes respectively.

6.4.5 In(1-x)Ga(x)As(y)P(1-y) System

The In(1-x)Ga(x)As(y)P(1-y) material system is commonly used for the fabrication of heterojunction devices. These include laser diodes, photodiodes, Gunn diodes, and high speed heterostructure transistors. As a quaternary material, two different mole fraction parameters, x and y, are necessary to specify any particular combination. This produces a wide array of InGaAsP materials and characteristics. Of particular interest in this system are materials that are lattice matched to InP.

The default material characteristics in Blaze for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP. The xcomposition and ycomposition are specified in the **REGION** statement with the X.COMP and Y.COMP parameters. The relationship between x and y that satisfy this condition is given by:

$$x = \frac{0.1896 \cdot \mathrm{Y.COMP}}{0.4176 - (0.0125 \cdot \mathrm{Y.COMP})} \qquad 0 < \mathrm{Y.COMP} < 1 \qquad \text{6-107}$$

Many of the parameter models for the In(1-x)Ga(x)As(y)P(1-y) system are functions of the composition fraction Y.COMP only. The composition fraction, X.COMP, can be deduced from the preceding relationship. Again, the default material characteristics in Blaze for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP.

Note: Don't use this material system to form GaAs by setting x=1 and y=1, specify GaAs as the material instead.

Bandgap

The default energy bandgap for the InP lattice matched In(1-x)Ga(x)As(y)P(1-y) system used in Blaze is given by:

$$E_g(InGaAsP) = 1.35 + \mathrm{X.COMP}\cdot(0.642 + (0.758\cdot\mathrm{X.COMP})) + (0.101\cdot\mathrm{Y.COMP} - 1.101)\cdot\mathrm{Y.COMP} - (0.28\cdot\mathrm{X.COMP} - 0.109\cdot\mathrm{Y.COMP} + 0.159)\cdot\mathrm{X.COMP}\cdot\mathrm{Y.COMP} \quad 6\text{-}108$$

Electron Affinity

The electron affinities for materials in the InP lattice matched InGaAsP system are derived from conduction band offsets and from the assumption that the affinity of InP is 4.4eV. The default conduction band edge offset between lattice matched InGaAsP and InP is then:

$$\Delta E_c = 0.268\cdot\mathrm{Y.COMP} + 0.003\cdot(\mathrm{Y.COMP})^2 \quad 6\text{-}109$$

Density of States and Effective Mass

The density of states is defined, as before, as a function of the effective masses of electrons and holes according to Equation 3-31. For the InGaAsP system, the default conduction and valence band effective masses, for electrons and holes, are given by the following.

For the conduction band:

$$m_e^* = 0.08 - (0.116\cdot\mathrm{Y.COMP}) + (0.026\cdot\mathrm{X.COMP}) - 0.059\cdot(\mathrm{X.COMP}\cdot\mathrm{Y.COMP}) + (0.064 - 0.02\cdot\mathrm{Y.COMP})\cdot(\mathrm{X.COMP})^2 + (0.06 + 0.032\cdot\mathrm{X.COMP})\cdot(\mathrm{Y.COMP})^2 \quad 6\text{-}110$$

For the valence band the hole effective mass is defined by:

$$m_h^* = (m_{lh}^{1.5} + m_{hh}^{1.5})^{\frac{2}{3}} \quad 6\text{-}111$$

where the default light hole effective mass is given by:

$$m_{lh} = 0.120 - (0.116\cdot\mathrm{Y.COMP}) + 0.03\cdot(\mathrm{X.COMP})^2 \quad 6\text{-}112$$

and the default heavy hole effective mass is a constant and is given by:

$$m_{hh} = 0.46 \quad 6\text{-}113$$

Dielectric Permittivity

The default static dielectric constant for lattice matched InGaAsP to InP is given by

$$\varepsilon_{InGaAsP} = (14.6 \cdot (1 - \texttt{X.COMP}) \cdot \texttt{Y.COMP}) + 12.5 \cdot (1 - \texttt{X.COMP}) \cdot (1 - \texttt{Y.COMP}) + 13.18 \cdot \texttt{X.COMP} \cdot \texttt{Y.COMP} + 11.11 \cdot \texttt{X.COMP} \cdot (1 - \texttt{Y.COMP}) \quad (6\text{-}114)$$

Low Field Mobility

The default low field mobility parameters for electrons and holes for lattice matched InGaAs are given by linear interpolations from the binary compounds GaAs and InP. The following formulas are used:

$$\mu_{n1} = 33000 + (8500 - 33000) \cdot \texttt{X.COMP} \quad (6\text{-}115)$$

$$\mu_{p1} = 460 + (400 - 460) \cdot \texttt{X.COMP} \quad (6\text{-}116)$$

$$\mu_{n2} = 4600 + (300 - 4600) \cdot \texttt{X.COMP} \quad (6\text{-}117)$$

$$\mu_{p2} = 150 + (100 - 150) \cdot \texttt{X.COMP} \quad (6\text{-}118)$$

$$\mu_{n0} = \mu_{n1} + (1 - \texttt{Y.COMP})(\mu_{n2} - \mu_{n1}) \quad (6\text{-}119)$$

$$\mu_{p0} = \mu_{p1} + (1 - \texttt{Y.COMP})(\mu_{p2} - \mu_{p1}) \quad (6\text{-}120)$$

In the case of InAs (but not InGaAsP alloy), Equations 3-207 and 3-208 can be used. You specify ANALYTIC on the **MODELS** statement to enable these. The default parameter values for these equations for InAs are shown in Table 6-25.

Table 6-25 ANALYTIC Model Parameters for InAs

Statment	Parameter	Type	Default	Units
MOBILITY	MU1N.CAUG	1000.0	Real	$cm^2/(V\ s)$
MOBILITY	MU1P.CAUG	20.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2N.CAUG	3400.0	Real	$cm^2/(V\ s)$
MOBILITY	MU2P.CAUG	530.0	Real	$cm^2/\ (V\ s)$
MOBILITY	ALPHAN.CAUG	0.0	Real	–
MOBILITY	ALPHAP.CAUG	0.0	Real	–
MOBILITY	BETAN.CAUG	-1.57	Real	–
MOBILITY	BETAP.CAUG	-2.3	Real	–

Table 6-25 ANALYTIC Model Parameters for InAs				
MOBILITY	GAMMAN.CAUG	-0.96	Real	–
MOBILITY	GAMMAP.CAUG	-1.38	Real	–
MOBILITY	DELTAN.CAUG	0.32	Real	–
MOBILITY	DELTAP.CAUG	0.46	Real	–
MOBILITY	NCRITN.CAUG	1.1×10^{18}	Real	cm^{-3}
MOBILITY	NCRITP.CAUG	1.1×10^{17}	Real	cm^{-3}

Sotoodeh's Low Field Mobility Model

Sotoodeh et al [298] describe a methodology of interpolating values of the parameters of the Caughey-Thomas mobility model to fit various composition, doping, and temperature for III-V compounds related to InGaAsP. The model is enabled by specifying `N.SOTOODEH` for electrons and `P.SOTOODEH` for holes on the **`MOBILITY`** statement. This model uses the Caughey-Thomas expression, fitted to closely match experimental data. The parameters are fixed, and you cannot change them.

The model is available for arbitrary temperature, doping, and composition of the following materials: GaAs, GaP, InP, InAs, AlGaAs, InGaP, InGaAs, InAlAs, and InGaAsP.

6.4.6 The Si(1-x)Ge(x) System

Advances in the growth of Silicon and Si(1-x)Ge(x) alloys have allowed the potential for using bandgap engineering to construct heterojunction devices such as HBTs and HEMTs using these materials. Blaze supports the SiGe material system by providing composition dependent material parameters. These parameters are accessed by specifying the material name SiGe.

The following sections describe the functional relationship between Ge mole fraction x, and the SiGe material characteristics necessary for device simulation.

Bandgap [173]

Bandgap is one of the most fundamental parameters for any material. For SiGe, the dependence of the bandgap on the Ge mole fraction, *x.composition*, is divided into ranges as follows:

for $x \leq 0.245$

$$E_g = 1.08 + \texttt{X.COMP}(0.945 - 1.08) / 0.245 \quad \text{6-121}$$

for $0.245 < x \leq 0.35$

$$E_g = 0.945 + (\texttt{X.COMP} - 0.245) \cdot (0.87 - 0.945) / (0.35 - 0.245) \quad \text{6-122}$$

for $0.35 < x \leq 0.5$

$$E_g = 0.87 + (\mathrm{X.COMP} - 0.35) \cdot (0.78 - 0.87) / (0.5 - 0.35) \quad 6\text{-}123$$

for $0.5 < x \leq 0.6$

$$E_g = 0.78 + (\mathrm{X.COMP} - 0.5) \cdot (0.72 - 0.78) / (0.6 - 0.5) \quad 6\text{-}124$$

for $0.6 < x \leq 0.675$

$$E_g = 0.72 + (\mathrm{X.COMP} - 0.6) \cdot (0.69 - 0.72) / (0.675 - 0.6) \quad 6\text{-}125$$

for $0.675 < x \leq 0.735$

$$E_g = 0.69 + (\mathrm{X.COMP} - 0.675) \cdot (0.67 - 0.69) / (0.735 - 0.675) \quad 6\text{-}126$$

for $0.735 < x \leq 1$

$$E_g = 0.67 \quad 6\text{-}127$$

The temperature dependence of the bandgap of SiGe is calculated the same as for Silicon except that EGALPHA and EGBETA are a function of Ge mole fraction x as follows:

$$E_g(T_L) = E_g + \mathrm{EGALPHA}\left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}}\right] \quad 6\text{-}128$$

$$\mathrm{EGALPHA} = (4.73 + \mathrm{X.COMP} \cdot (4.77 - 4.73)) \times 10^{-4} \quad 6\text{-}129$$

$$\mathrm{EGBETA} = 636 + \mathrm{X.COMP} \cdot (235 - 636) \quad 6\text{-}130$$

where E_g is dependent upon the mole fraction as above.

Electron Affinity

The electron affinity χ of SiGe is taken to be constant (4.17) with respect to composition.

Density of States

The density of states for SiGe is defined differently compared to the previous materials by not being a function of the effective masses. Instead the density of states have been made to depend upon the Ge mole fraction, *x.composition*,according to:

$$N_c = 2.8 \times 10^{19} + \mathrm{X.COMP} \cdot (1.04 \times 10^{19} - 2.8 \times 10^{19}) \quad 6\text{-}131$$

$$N_v = 1.04 \times 10^{19} + \mathrm{X.COMP} \cdot (6.0 \times 10^{18} - 1.04 \times 10^{19}) \quad 6\text{-}132$$

Dielectric Function

The compositional dependence of the static dielectric constant of SiGe is given by

$$\varepsilon = 11.8 + 4.2 \cdot \mathrm{X.COMP} \quad 6\text{-}133$$

Low Field Mobility

No specific SiGe low field mobility models have been implemented into Blaze.

Velocity Saturation

In SiGe, the temperature dependent velocity saturation, used in the field dependent mobility model is defined by the following equations.

$$\mathrm{VSATN} = 1.38 \times 10^{7} \cdot \sqrt{\left(tanh\left(\frac{175}{T_L}\right)\right)} \quad \text{6-134}$$

$$\mathrm{VSATP} = 9.05 \times 10^{6} \cdot \sqrt{\left(tanh\left(\frac{312}{T_L}\right)\right)} \quad \text{6-135}$$

Note: All other defaults used for SiGe are taken from Silicon.

6.4.7 Silicon Carbide (SiC)

Silicon carbide materials are of interest for high power, high temperature applications. The main characteristics of silicon carbide are that they have a very wide bandgap, high thermal conductivity, high saturation velocity, and high breakdown strength. Silicon carbide is commercially available in three polytypes called 6H-SiC, 3H-SiC, and 4H-SiC. Atlas supports all three these polytypes. The following paragraphs describe the material defaults for these materials.

Band Parameters for SiC

SiC band parameter equations are identical to those used for Silicon but with the values adjusted for 3C-SiC, 4H-SiC, and 6H-SiC. The physical band parameter values are shown in Tables B-15 and B-16.

SiC Mobility Parameters

Isotropic Mobility

By default, mobility is assumed to be entirely isotropic in nature. That is, there is no directional component. The default low field mobilities of electrons and holes for 3C-SiC, 4H-SiC, and 6H-SiC are shown in Table 6-26.

Table 6-26 Silicon Carbide Low Field Mobility Defaults

Statement	Parameter	6H-SiC	3C-SiC	4H-SiC	Units
MOBILITY	MUN	330	1000	460	$cm^2/V \cdot s$
MOBILITY	MUP	60	50	124	$cm^2/V \cdot s$

Anisotropic Mobility

The mobility behavior within SiC is now known to be anisotropic in nature, which dramatically alters the electrical performance of a device. A anisotropic model has been implemented into Atlas to correctly model this behavior. Following the ideas of Lindefelt [184] and Lades [171], the mobility within the drift diffusion equations has been made a tensor property. As a result the mobility has become:

$$\mu = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_2 \end{bmatrix} \quad \text{6-136}$$

where μ_1 represents the mobility defined in one plane and μ_2 the mobility defined in a second plane. In the case of SiC, μ_1 represents the mobility of the direction <1100> while μ_2 represents the mobility of the direction <0001>. These mobilities are defined for both holes and electrons.

Anisotropy is handled by allowing you to specify mobility parameters in two directions normal to each other. To enable the model, specify the N.ANGLE or P.ANGLE or both parameters of the **MOBILITY** statement. When you specify the parameter(s), it signifies that the mobility parameter values and model flags in the enclosing **MOBILITY** statement apply along the specified angle, with respect to the X axis, for electrons (N.ANGLE) and holes

(P.ANGLE). The parameter values and model flags specified in a second **MOBILITY** statement (without the N.ANGLE or P.ANGLE specification) apply to all normal directions (in the x-y plane) to the angle specified in the other statement.

When calculating the current along an element edge, the direction vector of the edge is used to calculate the weighting of the two mobilities specified by the two **MOBILITY** statements. The weights (w_1 and w_2) are taken to be the dot product of the current direction vector with the angle specified on the N.ANGLE or P.ANGLE parameter for the associated parameters and models, the dot product of the current direction vector, and the normal to the specified angle for the models and parameters not associated with the N.ANGLE or P.ANGLE parameter specification.

The net weighted mobility is calculated by a weighted application of Mattheissen's rule as given by Equation 6-137.

$$\mu = \frac{1}{\frac{w_1}{\mu_1} + \frac{w_2}{\mu_2}} \qquad 6\text{-}137$$

Defining Anisotropic Mobility in Atlas

To define a material with anisotropic mobility, specify two **MOBILITY** statements. In each statement, the N.ANGLE and P.ANGLE parameters are used to specify the direction where that particular mobility is to apply. The following example shows how this is done.

```
# FIRST DEFINE MOBILITY IN PLANE <1100>
#
MOBILITY MATERIAL=4H-SiC VSATN=2E7 VSATP=2E7 BETAN=2 BETAP=2 \
   MU1N.CAUG=10 MU2N.CAUG=410 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=20 MU2P.CAUG=95 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
#
# NOW DEFINE MOBILITY IN PLANE <0001>
#
MOBILITY MATERIAL=4H-SiC N.ANGLE=90.0 P.ANGLE=90.0 VSATN=2E7 VSATP=2E7 \
   BETAN=2 BETAP=2 MU1N.CAUG=5 MU2N.CAUG=80 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=2.5 MU2P.CAUG=20 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
```

In this example, the second **MOBILITY** statement applies to the direction 90° with respect to the X axis (or along the Y axis). The first **MOBILITY** statement without the N.ANGLE and P.ANGLE parameter assignments applies along the X axis.

Carrier-Carrier Scattering Contribution to Mobility

The expression for the carrier-carrier scattering contribution to mobility given by Lades [170] is the same as the Conwell-Weisskopf expression given in Equation 3-220. The default values of the coefficients D.CONWELL and F.CONWELL for 4H-SiC and 6H-SiC are set to the values given in Table 6-27.

Table 6-27 Default Values of Conwell-Weisskopf Mobility Model Parameters for SiC

Statement	Parameter	Default (4H-SiC)	Default (6H-SiC)	Units
MOBILITY	D.CONWELL	6.9×10^{20}	3.045×10^{20}	$(cm\ V\ s)^{-1}$
MOBILITY	F.CONWELL	7.452×10^{13}	7.452×10^{13}	cm^{-2}

Doping Dependent Low Field Mobility for 4H-SiC and 6H-SiC

The analytic low field model (Caughey-Thomas) is given in Equations 3-207 and 3-208. Specific parameters for 4H-SiC and 6H-SiC are given by Lades [170]. These are listed in Table 6-28.

Table 6-28 Default Values of Analytic Mobility Model Parameters for SiC.

Statement	Parameter	Default (4H-SiC)	Default (6H-SiC)	Units
MOBILITY	MU1N.CAUG	0.0	0.0	$cm^2/(V\cdot s)$
MOBILITY	MU1P.CAUG	15.9	6.8	$cm^2/(V\cdot s)$
MOBILITY	MU2N.CAUG	947.0	415.0	$cm^2/(V\cdot s)$
MOBILITY	MU2P.CAUG	124.0	99.0	$cm^2/(V\cdot s)$
MOBILITY	ALPHAN.CAUG	0.0	0.0	
MOBILITY	ALPHAP.CAUG	0.0	0.0	
MOBILITY	BETAN.CAUG	-1.8	-1.8	
MOBILITY	BETAP.CAUG	-1.8	-1.8	
MOBILITY	GAMMAN.CAUG	0.0	0.0	
MOBILITY	GAMMAP.CAUG	0.0	0.0	
MOBILITY	DELTAN.CAUG	0.61	0.59	
MOBILITY	DELTAP.CAUG	0.34	0.31	
MOBILITY	NCRITN.CAUG	1.94×10^{17}	1.11×10^{18}	cm^{-3}
MOBILITY	NCRITP.CAUG	1.76×10^{19}	2.1×10^{19}	cm^{-3}

The parameters for the low field mobility for the CVT model have also been changed so they give the same mobility as the ANALYTIC model.

Temperature Dependent Saturated Velocity for 4H-SiC and 6H-SiC

The parallel field dependent mobility model for electrons enabled by specifying FLDMOB on the **MODELS** statement has a saturated velocity given by Equation 3-325. For 4H-SiC and 6H-SiC, a different model for saturated velocity is preferable. This is given by

$$v_n = \text{ALPHAN.FLD}\left(\frac{T_L}{\text{TNOMN.FLD}}\right)^{\text{DELTAN.FLD}} \quad \text{6-138}$$

and applies to electrons in 4H-SiC and 6H-SiC. If the BETA exponents of this model are temperature dependent, as in the Canali modification of Equation 3-327, then the specific default values of N.BETA0 and N.BETAEXP for SiC are used. All defaults are given in Table 6-29.

Table 6-29 Default Values of Saturated Electron Velocity Model Parameters for SiC.

Statement	Parameter	Default (4H-SiC)	Default (6H-SiC)	Units
MOBILITY	ALPHAN.FLD	2.2×10^7	1.9×10^7	cm/s
MOBILITY	TNOMN.FLD	300	300	Kelvin
MOBILITY	DELTAN.FLD	-0.44	-1.0	
MOBILITY	N.BETA0	1.2	1.7	
MOBILITY	N.BETAEXP	1.0	1.25	

Anisotropic Impact Ionization of 4H-SiC

To enable a model [120] for the anisotropic impact ionization rate of 4H-SiC, specify the ANISO parameter of the IMPACT statement. In this model, the following equations describe the ionization rate:

$$\alpha(E_xE_y) = a\text{EXP}\left(-c\sqrt{1-A^2c^2\left(\frac{E_xE_y}{b_xb_y}\right)^2}\right) \quad \text{6-139}$$

$$A = ln(a_y/a_x) \quad \text{6-140}$$

$$c = \frac{b}{E} = \left(\frac{E_x^2}{b_x^2}+\frac{E_y^2}{b_y^2}\right)^{-1/2} \quad \text{6-141}$$

$$a = a_x^{\frac{c^2E_x^2}{b_x^2}}\,a_y^{\frac{c^2E_y^2}{b_y^2}} \quad \text{6-142}$$

Here, E_x and E_y are the electric field magnitude in the x and y directions. The values parameters a_x, a_y, b_x, and b_y depend upon the crystal orientation. The crystal orientation can be either 0001 or $11\bar{2}0$ as specified by either the SIC4H0001 or the SIC4H1120 flag of the **IMPACT** statement. By default, SIC4H0001 is selected.

The values associated with the y coordinates (a_y and b_y) correspond to the selected orinetation. The values of the unselected orientation are then associated with the x coordinates (a_x and b_x). Table 6-30 shows the default values of a_x, b_x, a_y, and b_y, where AE* and BE* are for electrons and AH* and BH* are for holes.

Table 6-30 Anisotropic Impact Ionization Model Defaults

Parameter	Statement	Default	Units
AE0001	**IMPACT**	1.76×10^{8}	cm^{-1}
BE0001	**IMPACT**	3.30×10^{7}	V/cm
AH0001	**IMPACT**	3.41×10^{8}	cm^{-1}
BH0001	**IMPACT**	2.50×10^{7}	V/cm
AE1120	**IMPACT**	2.10×10^{7}	cm^{-1}
BE1120	**IMPACT**	1.70×10^{7}	V/cm
AH1120	**IMPACT**	2.96×10^{7}	cm^{-1}
BH1120	**IMPACT**	1.60×10^{7}	V/cm

Thermal Parameters

The equations governing these effects are identical to those for Silicon but with adjusted coefficients. See Appendix B "Material Systems" for a list of all these parameters.

Two level Incomplete Ionization Model

The model for incomplete ionization described in Section 3.1.2 "Carrier Continuity Equations" assumes a single ionisation energy for donor impurities and a single ionization level for acceptor impurities. In 4H-SiC and 6H-SiC, the dopant locations may have either a locally cubic arrangement of surrounding atoms, or a locally hexagonal arrangement [170]. These different arrangements result in different ionization energies for the same dopant species. In 6H-SiC, there are 2 cubic environment sites for every hexagonal environment site and in 4H-SiC there is one cubic site for every hexagonal site. The ionized n-type doping is given by

$$N_d^+ = \frac{\mathrm{ALPHA.INCOMPLETE}\,N_d}{1+GCB\exp\left(\dfrac{E_{Fn}-E_c+\mathrm{ED.CUBIC}}{KT_L}\right)} + \frac{(1-\mathrm{ALPHA.INCOMPLETE})N_d}{1+GCB\exp\left(\dfrac{E_{Fn}-E_c+\mathrm{ED.HEXAGONAL}}{KT_L}\right)} \quad \text{6-143}$$

and the ionized p-type doping given by

$$N_a^- = \frac{\mathrm{ALPHA.INCOMPLETE}\,N_a}{1+GVB\exp\left(\dfrac{E_v-E_{Fp}+\mathrm{EA.CUBIC}}{KT_L}\right)} + \frac{(1-\mathrm{ALPHA.INCOMPLETE})N_a}{1+GVB\exp\left(\dfrac{E_v-E_{Fp}+\mathrm{EA.HEXAGONAL}}{KT_L}\right)} \quad \text{6-144}$$

The ionization energies for donor impurities, ED.CUBIC and ED.HEXAGONAL, are given default values corresponding to Nitrogen doping. The ionization levels for acceptor impurities are given default values corresponding to Boron doping. The default values are from [170] and are given in Table 6-31.

Table 6-31 Default Values of Parameters for Site Specific Ionization Levels

Statement	Parameter	Default (4H-SiC)	Default (6H-SiC)	Units
MATERIAL	ALPHA.INCOMPLETE	1/2	2/3	
MATERIAL	ED.CUBIC	0.09	0.14	eV
MATERIAL	ED.HEXAGONAL	0.05	0.08	eV
MATERIAL	EA.CUBIC	0.32	0.32	eV
MATERIAL	EA.HEXAGONAL	0.32	0.32	eV

The ionised densities obtained using this model can be quite different from those obtained by using a single, averaged, ionization energy. To use this model, specify INCOMPLETE on the MODELS statement and it will be used in any 4H-SiC or 6H-SiC regions.

6.4.8 GaN, InN, AlN, Al(x)Ga(1-x)N, In(x)Ga(1-x)N, Al(x)In(1-x)N, and Al(x)In(y)Ga(1-x-y)N

The following sections describe the relationship between mole fraction, X.COMP, and the material parameters and various physical models specific to the Al/In/GaN system.

Bandgap

By default, the bandgap for the nitrides is calculated in a two step process. First, the bandgap(s) of the relevant binary compounds are computed as a function of temperature, *T*, using Equations 6-145 through 6-147 [335].

$$Eg(GaN) = 3.507 - \frac{0.909 \times 10^{-3} T^2}{T + 830.0} \quad \text{6-145}$$

$$Eg(InN) = 1.994 - \frac{0.245 \times 10^{-3} T^2}{T + 624.0} \quad \text{6-146}$$

$$Eg(AlN) = 6.23 - \frac{1.799 \times 10^{-3} T^2}{T + 1462.0} \quad \text{6-147}$$

Note: The bandgap for InN contradicts recent observations [341,72,359] of a bandgap of less than 1.0 eV. Despite these observations, we keep the above specified value to maintain consistency with ternary and quaternary blends. You should carefully consider whether to use the default values, or supply your own values using one of the standard methods of user specification. The simplest method is to specify EG300 on the MATERIAL statement.

Then, the dependence on composition fraction, *x*, is described by Equations 6-148 and 6-150 [246].

$$Eg(In_xGa_{1-x}N) = Eg(InN)x + Eg(GaN)(1-x) - 3.8x(1-x) \quad 6\text{-}148$$

$$Eg(Al_xGa_{1-x}N) = Eg(AlN)x + Eg(GaN)(1-x) - 1.3x(1-x) \quad 6\text{-}149$$

$$Eg(Al_xIn_{1-x}N) = Eg(AlN)x + Eg(InN)(1-x) \quad 6\text{-}150$$

Electron Affinity

The electron affinity is calculated such that the band edge offset ratio is given by [246].

$$\frac{\Delta Ec}{\Delta Ev} = \frac{0.7}{0.3} \quad 6\text{-}151$$

You can overide this ratio by specifying the ALIGN parameter of the **MATERIAL** statement.

Permittivity

The permittivity of the nitrides as a function of composition fraction, *x*, is given by linear interpolations of the values for the binary compounds as in Equations 6-152 and 6-153 [8].

$$\varepsilon(In_xGa_{1-x}N) = 15.3x + 8.9(1-x) \quad 6\text{-}152$$

$$\varepsilon(Al_xGa_{1-x}N) = 8.5x + 8.9(1-x) \quad 6\text{-}153$$

Density of States Masses

The nitride density of states masses as a function of composition fraction, x, is given by linear interpolations of the values for the binary compounds as in Equations 6-154 through 6-157 [335].

$$m_e(In_xGa_{1-x}N) = 0.12x + 0.2(1-x) \quad 6\text{-}154$$

$$m_h(In_xGa_{1-x}N) = 0.17x + 1.0(1-x) \quad 6\text{-}155$$

$$m_e(Al_xGa_{1-x}N) = 0.314x + 0.2(1-x) \quad 6\text{-}156$$

$$m_h(Al_xGa_{1-x}N) = 0.417x + 1.0(1-x) \quad 6\text{-}157$$

Low Field Mobility

The Albrecht Model

You can choose to model low field mobility following the work of Albrecht et.al [6] by specifying ALBRCT on the **MODELS** statement or ALBRCT.N or ALBRCT.P or both on the **MOBILITY** statement for separate control over electrons and holes. This model is described as follows:

$$\frac{1}{\mu(N,T_L)} = \frac{\mathtt{AN.ALBRCT}\cdot N}{\mathtt{N0N}\cdot\mathtt{ALBRCT}}\left(\frac{T_L}{\mathtt{T0N}\cdot\mathtt{ALBRCT}}\right)^{-3/2} ln\left[1+3\left(\frac{T_L}{\mathtt{T0N}\cdot\mathtt{ALBRCT}}\right)^{2}\left(\frac{N}{\mathtt{N0N}\cdot\mathtt{ALBRCT}}\right)^{-2/3}\right] + \mathtt{BN.ALBRCT}\times\left(\frac{T_L}{\mathtt{T0N}\cdot\mathtt{ALBRCT}}\right)^{3/2} + \frac{\mathtt{CN.ALBRCT}}{exp(\mathtt{T1N}\cdot\mathtt{ALBRCT}/T_L)-1} \quad \text{6-158}$$

where $\mu(N,T)$ is the mobility as a function of doping and lattice temperature, N is the total doping concentration, and T_L is the lattice temperature. AN.ALBRCT, BN.ALBRCT, CN.ALBRCT, N0N.ALBRCT, T0N.ALBRCT and T1N.ALBRCT are user-specifiable parameters on the **MOBILITY** statement. You can use a similar expression for holes with the user-defined parameters AP.ALBRCT, BP.ALBRCT, CP.ALBRCT, N0P.ALBRCT, T0P.ALBRCT and T1P.ALBRCT.

Table 6-32 shows the default values for the parameters of the Albrecht Model.

Table 6-32 Default Parameter Values for the Albrecht Model

Parameter	Default	Parameter	Default	Units
AN.ALBRCT	2.61×10^{-4}	AP.ALBRCT	2.61×10^{-4}	V*s/(cm^2)
BN.ALBRCT	2.9×10^{-4}	BP.ALBRCT	2.9×10^{-4}	V*s/(cm^2)
CN.ALBRCT	170.0×10^{-4}	CP.ALBRCT	170.0×10^{-4}	V*s/(cm^2)
N0N.ALBRCT	1.0×10^{17}	N0P.ALBRCT	1.0×10^{17}	cm^{-3}
T0N.ALBRCT	300.0	T0P.ALBRCT	300.0	K
T1N.ALBRCT	1065.0	T1P.ALBRCT	1065.0	K

Farahmand Modified Caughey Thomas

You can use a composition and temperature dependent low field model by specifying the FMCT.N and FMCT.P in the **MOBILITY** statement. FMCT stands for Farahmand Modified Caughey Thomas. This model [86] was the result of fitting a Caughey Thomas like model to Monte Carlo data. The model is similar to the analytic model described by Equations 3-207 and 3-208. This modified model is described by Equations 6-159 and 6-160 for electrons and holes.

$$\mu_n(T,N) = \mathtt{MU1N.FMCT}\left(\frac{T}{300}\right)^{\mathtt{BETAN.FMCT}} + \frac{(\mathtt{MU2N.FMCT}-\mathtt{MU1N.FMCT})\left(\frac{T}{300}\right)^{\mathtt{DELTAN.FMCT}}}{1+\left[\frac{N}{\mathtt{NCRITN.FMCT}\left(\frac{T}{300}\right)^{\mathtt{GAMMAN.FMCT}}}\right]^{\mathtt{ALPHAN.FMCT}(T/300)^{\mathtt{EPSP\cdot FMCT}}}} \quad \text{6-159}$$

$$\mu_p(T,N) = \mathtt{MU1P.FMCT}\left(\frac{T}{300}\right)^{\mathtt{BETAP.FMCT}} + \frac{(\mathtt{MU2P.FMCT}-\mathtt{MU1P.FMCT})\left(\frac{T}{300}\right)^{\mathtt{DELTAP.FMCT}}}{1+\left[\frac{N}{\mathtt{NCRITP.FMCT}\left(\frac{T}{300}\right)^{\mathtt{GAMMAP.FMCT}}}\right]^{\mathtt{ALPHAP.FMCT}(T/300)^{\mathtt{EPSP\cdot FMCT}}}} \quad \text{6-160}$$

In these equations, T is the lattice temperature and N is the total doping. Table 6-33 shows the user-definable parameters.

Table 6-33 User-specifiable parameters for the Faramand modified Caughey Thomas model for Nitrides.			
Parameter	**Statement**	**Type**	**Units**
MU1N.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU1P.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU2N.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU2P.FMCT	MOBILITY	Real	$cm^2/(V*s)$
ALPHAN.FMCT	MOBILITY	Real	
ALPHAP.FMCT	MOBILITY	Real	
BETAN.FMCT	MOBILITY	Real	
BETAP.FMCT	MOBILITY	Real	
GAMMAN.FMCT	MOBILITY	Real	
GAMMAP.FMCT	MOBILITY	Real	
DELTAN.FMCT	MOBILITY	Real	
DELTAP.FMCT	MOBILITY	Real	
EPSN.FMCT	MOBILITY	Real	
EPSP.FMCT	MOBILITY	Real	
NCRITN.FMCT	MOBILITY	Real	cm^{-3}
NCRITP.FMCT	MOBILITY	Real	cm^{-3}

Tables 6-34 and 6-35 show the default parameters as taken from the Monte Carlo fits for various nitride compositions.

Table 6-34 Default Nitride Low Field Mobility Model Parameter Values [86]				
MATERIAL	**MU1N.FMCT (cm^2/V.s)**	**MU2N.FMCT (cm^2/V.s)**	**ALPHAN.FMCT**	**BETAN.FMCT**
InN	774	3138.4	0.68	-6.39
$In_{0.8}Ga_{0.2}N$	644.3	1252.7	0.82	-1.81
$In_{0.5}Ga_{0.5}N$	456.4	758.1	1.04	-1.16
$In_{0.2}Ga_{0.8}N$	386.4	684.2	1.37	-1.36
GaN	295.0	1460.7	0.66	-1.02

Table 6-34 Default Nitride Low Field Mobility Model Parameter Values [86]				
MATERIAL	MU1N.FMCT (cm^2/V.s)	MU2N.FMCT (cm^2/V.s)	ALPHAN.FMCT	BETAN.FMCT
$Al_{0.2}Ga_{0.8}N$	132.0	306.1	0.29	-1.33
$Al_{0.5}Ga_{0.5}N$	41.7	208.3	0.12	-0.6
$Al_{0.8}Ga_{0.2}N$	47.8	199.6	0.17	-0.74
AlN	297.8	683.8	1.16	-1.82

Table 6-35 Default Nitride Low Field Mobility Model Parameter Values [86]				
MATERIAL	DELTAN.FMCT	GAMMAN.FMCT	EPSN.FMCT	NCRITN.FMCT
InN	-1.81	8.05	-0.94	10^{17}
$In_{0.8}Ga_{0.2}N$	-1.30	4.84	-0.41	10^{17}
$In_{0.5}Ga_{0.5}N$	-1.74	2.21	-0.22	10^{17}
$In_{0.2}Ga_{0.8}N$	-1.95	2.12	-0.99	10^{17}
GaN	-3.84	3.02	0.81	10^{17}
$Al_{0.2}Ga_{0.8}N$	-1.75	6.02	1.44	10^{17}
$Al_{0.5}Ga_{0.5}N$	-2.08	10.45	2.00	10^{17}
$Al_{0.8}Ga_{0.2}N$	-2.04	20.65	0.01	10^{17}
AlN	-3.43	3.78	0.86	10^{17}

For composition fractions not listed in Tables 6-34 and 6-35, the default parameters are linearly interpolated from the nearest composition fractions on the table. You can override these defaults by specifying any of the parameters listed in Table 6-33 on the **MOBILITY** statement. Currently, this model has only been calibrated for electrons. We only recommend that you use this model for holes when you define a set of real default parameters.

High Field Mobility

You can select nitride specific field dependent mobility model by specifying GANSAT.N and GANSAT.P on the **MOBILITY** statement. This model [86] is based on a fit to Monte Carlo data for bulk nitride, which is described in Equations 6-161 and 6-162.

$$\mu_n = \frac{\mu_{n0}(T,N) + \mathrm{VSATN}\dfrac{E^{\mathrm{N1N}\cdot\mathrm{GANSAT}-1}}{\mathrm{ECN}\cdot\mathrm{GANSAT}^{\mathrm{N1N}\cdot\mathrm{GANSAT}}}}{1 + \mathrm{ANN}\cdot\mathrm{GANSAT}\left(\dfrac{E}{\mathrm{ECN}\cdot\mathrm{GANSAT}}\right)^{N2N.\mathrm{GANSAT}} + \left(\dfrac{E}{\mathrm{ECN}\cdot\mathrm{GANSAT}}\right)^{\mathrm{N1N.GANSAT}}} \quad \text{6-161}$$

$$\mu_p = \frac{\mu_{p0}(T,N) + \mathrm{VSATP}\dfrac{E^{\mathrm{N1P}\cdot\mathrm{GANSAT}-1}}{\mathrm{ECP}\cdot\mathrm{GANSAT}^{\mathrm{N1P}\cdot\mathrm{GANSAT}}}}{1 + \mathrm{ANP}\cdot\mathrm{GANSAT}\left(\dfrac{E}{\mathrm{ECP}\cdot\mathrm{GANSAT}}\right)^{N2P.\mathrm{GANSAT}} + \left(\dfrac{E}{\mathrm{ECP}\cdot\mathrm{GANSAT}}\right)^{N1P.\mathrm{GANSAT}}} \quad \text{6-162}$$

In these equations, $\mu_0(T,N)$ is the low field mobility and E is the electric field. Table 6-36 lists the user-definable parameters.

Table 6-36 User Definable Field Nitride Mobility Model Parameters

Parameter	Statement	Type	Units
N1N.GANSAT	**MOBILITY**	Real	
N1P.GANSAT	**MOBILITY**	Real	
N2N.GANSAT	**MOBILITY**	Real	
N2P.GANSAT	**MOBILITY**	Real	
ANN.GANSAT	**MOBILITY**	Real	
ANP.GANSAT	**MOBILITY**	Real	
ECN.GANSAT	**MOBILITY**	Real	V/cm
ECP.GANSAT	**MOBILITY**	Real	V/cm

Table 6-37 shows the default parameters as taken from the Monte Carlo fits for various nitride compositions.

Table 6-37 Default Nitride Field Dependent Mobility Model Parameter Values [86]

MATERIAL	VSATN (cm/S)	ECN.GANSAT (kV/cm)	N1N.GANSAT	N2N.GANSAT	ANN.GANSAT
InN	1.3595×10^7	52.4242	3.8501	0.6078	2.2623
$In_{0.8}Ga_{0.2}N$	0.8714×10^7	103.4550	4.2379	1.1227	3.0295

Table 6-37 Default Nitride Field Dependent Mobility Model Parameter Values [86]					
$In_{0.5}Ga_{0.5}N$	0.7973×10^{7}	148.9098	4.0635	1.0849	3.0052
$In_{0.2}Ga_{0.8}N$	1.0428×10^{7}	207.5922	4.7193	1.0239	3.6204
GaN	1.9064×10^{7}	220.8936	7.2044	0.7857	6.1973
$Al_{0.2}Ga_{0.8}N$	1.1219×10^{7}	365.5529	5.3193	1.0396	3.2332
$Al_{0.5}Ga_{0.5}N$	1.1459×10^{7}	455.4437	5.0264	1.0016	2.6055
$Al_{0.8}Ga_{0.2}N$	1.5804×10^{7}	428.1290	7.8166	1.0196	2.4359
AlN	2.1670×10^{7}	447.0339	17.3681	0.8554	8.7253

For composition fractions not listed in Table 6-37, the default parameters are linearly interpolated from the nearest composition fractions on the table.

You can override these defaults by specifying any of the parameters listed in Table 6-36 in the **MOBILITY** statement. Currently, this model has only been calibrated for electrons. We only recommend that you use this model for holes when you define a set of real default parameters. Also note that these models exhibit negative differential mobility and may exhibit poor convergence. In such cases, you may do well to use the simpler model given in Equations 3-323 and 3-324 with a reasonable value of saturation velocity.

Figure 6-9 shows the modeled velocity-field curves for pure GaN for various low-field mobilities.

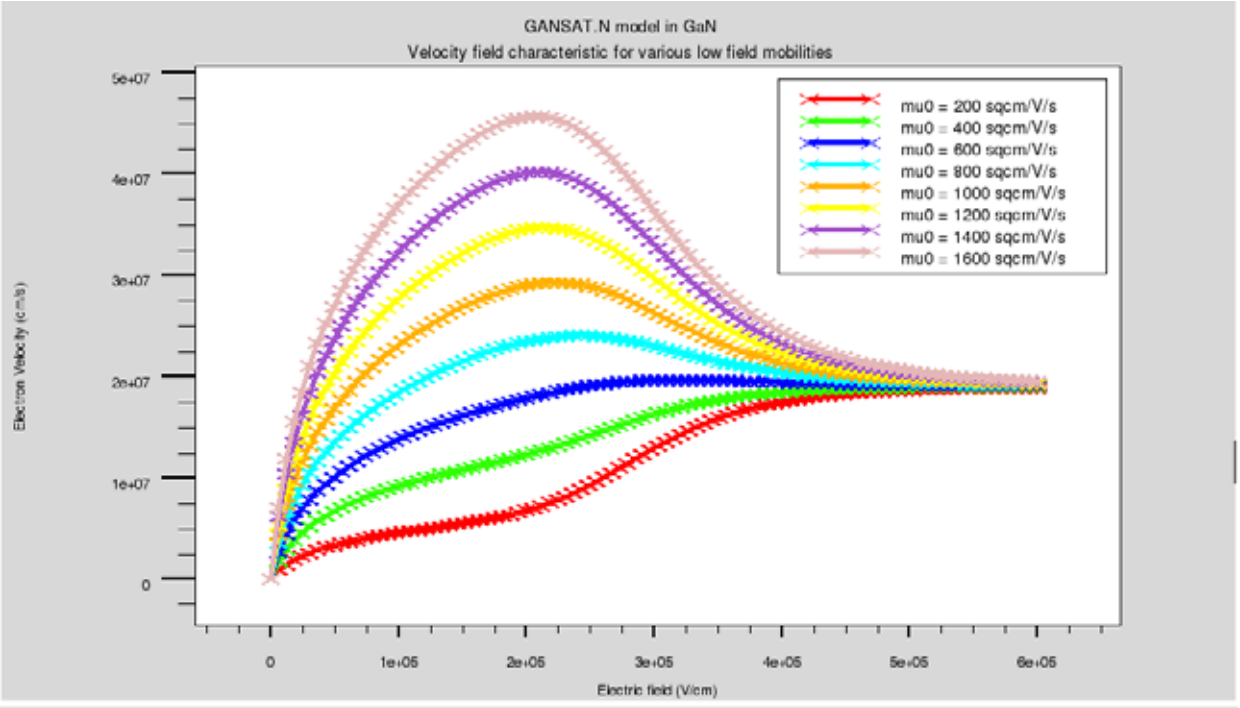


Figure 6-9: Velocity-Field Curves for Various Low-Field Mobilities

Curve Fit Velocity Saturation Mobility Models

We have used the Deckbuild Optimizer to optimize parameters for the default GANSAT mobility model, Equation 6-161 for electrons and the silicon saturation model, Equation 3-323. We have built these models in accessable by setting logical flags on the **MOBILITY** statement as listed in Table 6-38.

Table 6-38 Comparison of fit velocity saturation models

MOBILITY statement	Reference	Description
GANSAT.N	[86]	Monte Carlo InAlGaN electrons
CHEN.N	[52]	Monte Carlo GaN electrons
CHEN.P	[52]	Monte Carlo GaN holes
OZGUR.N	[227]	Monte Carlo ZnO electrons

Figure 6-10 shows a comparison of the various velocity saturation mobility models.

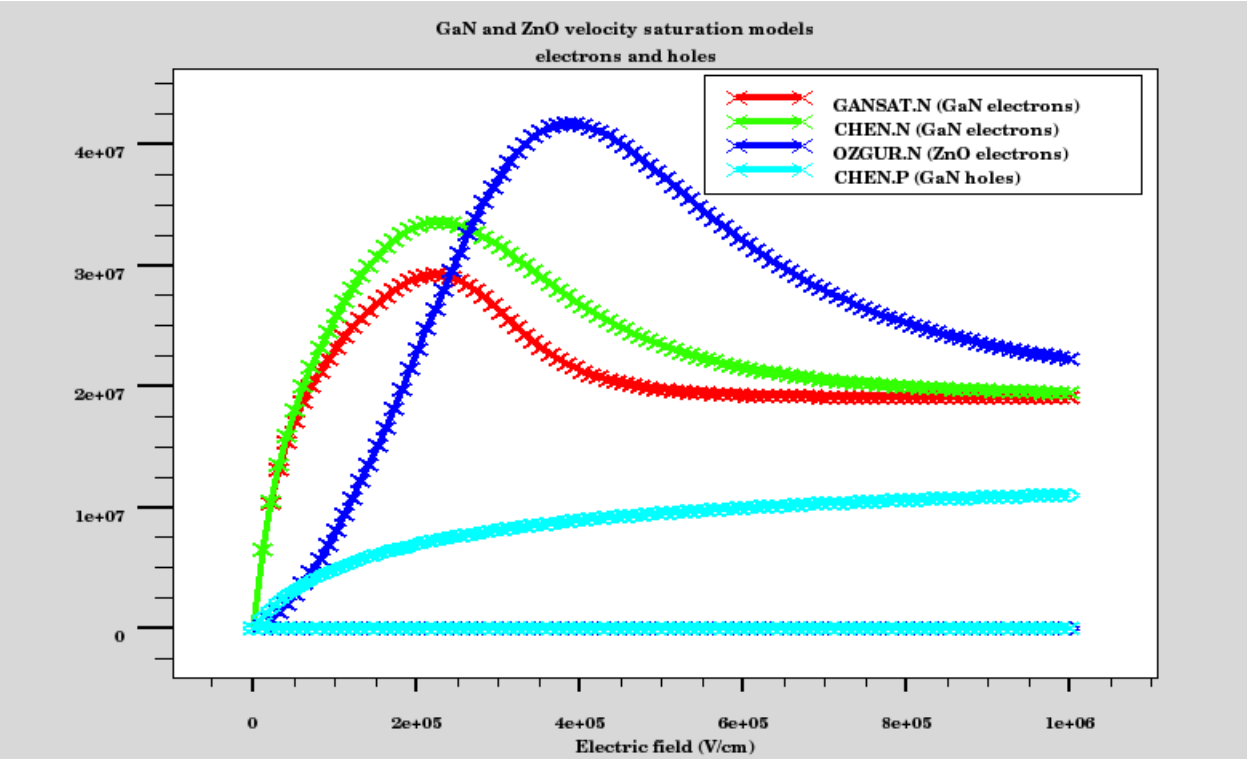


Figure 6-10: Comparison of velocity saturation characteristics of various fit models.

The default parameters for the various electron saturation models using Equation 6-161 are given in Table 6-39.

Table 6-39 Default parameters for fit models using Equation 6-161

MODEL $cm^2/(V*s)$	MUN0 cm/s	VSATN V/cm	ECN.GANSAT	N1N.GANSAT	N2N.GANSAT	ANN.GANSAT
GANSAT.MU	1000	1.9e7	220e3	7.2044	0.7857	6.1973
CHEN.N	1100	1.9e7	198e3	4.56	0.664	5.131
OZGUR.N	1000	1.9e7	256e3	4.12	-0.578	6.7

The default parameters for the hole saturation model using Equation 3-323 is listed in Table 6-40.

Table 6-40 Default parameters for fit models using Equation 3-323

MODEL $cm^2/(V*s)$	MUP0 cm/s	BETAP	VSATP
CHEN.P	100	0.8	1.4e7

Impact Ionization Parameters

Table 6-41 shows the extracted default values for the Selberherr impact ionization model from [228] for GaN.

Table 6-41 Default Impact Ionization Parameters (IMPACT statement)

Parameter	Units	GaN	InN	AlN	InGaN	AlGaN
AN1	cm^{-1}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}
AN2	cm^{-1}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}	2.52×10^{8}
BN1	V/cm	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}
BN2	V/cm	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}	3.41×10^{7}
AP1	cm^{-1}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}
AP2	cm^{-1}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}	5.37×10^{6}
BP1	V/cm	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}
BP2	V/cm	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}	1.96×10^{7}
BETAN		1.0	1.0	1.0	1.0	1.0
BETAP		1.0	1.0	1.0	1.0	1.0
EGRAN	V/cm	0.0	0.0	0.0	0.0	0.0

Supplied Impact Ionization Rate Tables

Using the tabular Selberherr model, we have provided build-in tabular representations of results from several recent publications. We have provided convenient access using logicals on the **IMPACT** statement. The association between logical parameter name and publication is contained in Table 6-42.

Table 6-42 Association Between IMPACT Statement Parameter and Reference

IMPACT statement	Reference	Description
N.CHEN	[52]	Monte Carlo
P.CHEN	[52]	Monte Carlo
N.MCCLINTOCK	[205]	measured
P.MCCLINTOCK	[205]	measured
N.OGUZMAN	[228]	Monte Carlo
P.OGUZMAN	[228]	Monte Carlo
N.TUT	[315]	measured Al(0.4)Ga(0.6)N
P.TUT	[315]	measured Al(0.4)Ga(0.6)N

Figures 6-11 and 6-12 show a comparison of the various tabular ionization rate models against the default GANSAT.N and GANSAT.P models for the case of extrapolation as indicated by the TBL.EXTRAP parameter of the **IMPACT** statement.

Figures 6-13 and 6-14 show a comparison of the various tabular ionization rate models against the default GANSAT.N and GANSAT.P models for the case of disabled extrapolation as indicated by ^TBL.TXTRACT on the **IMPACT** statement.

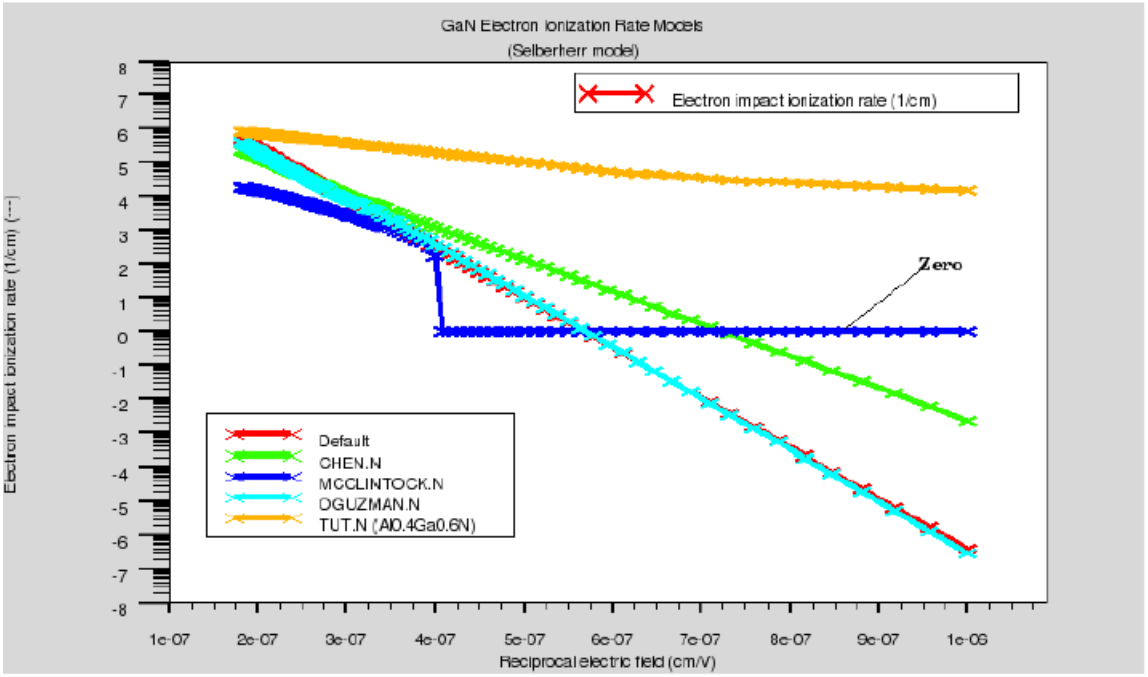


Figure 6-11: Comparison of electron ionization rates as a function of reciprocal field with extrapolation enabled.

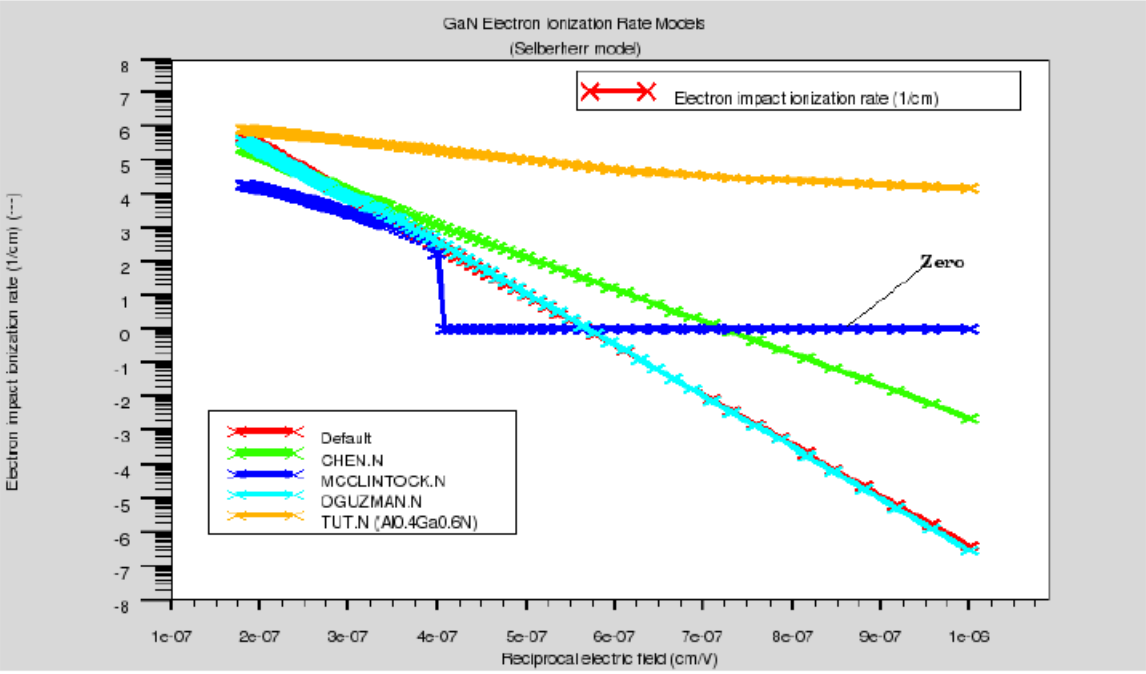


Figure 6-12: Comparison of hole ionization rates as a function of reciprocal field with extrapolation enabled.

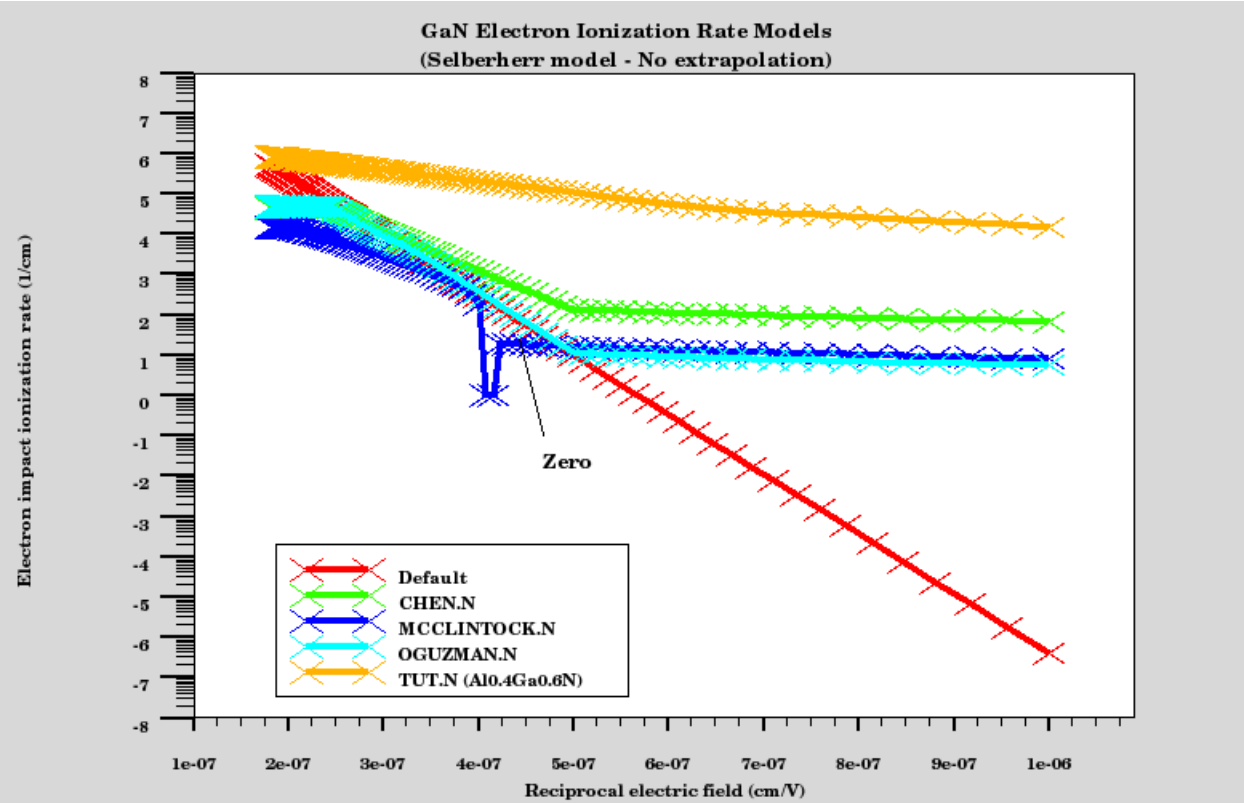


Figure 6-13: Comparison of electron ionization rates as a function of reciprocal field with extrapolation disabled.

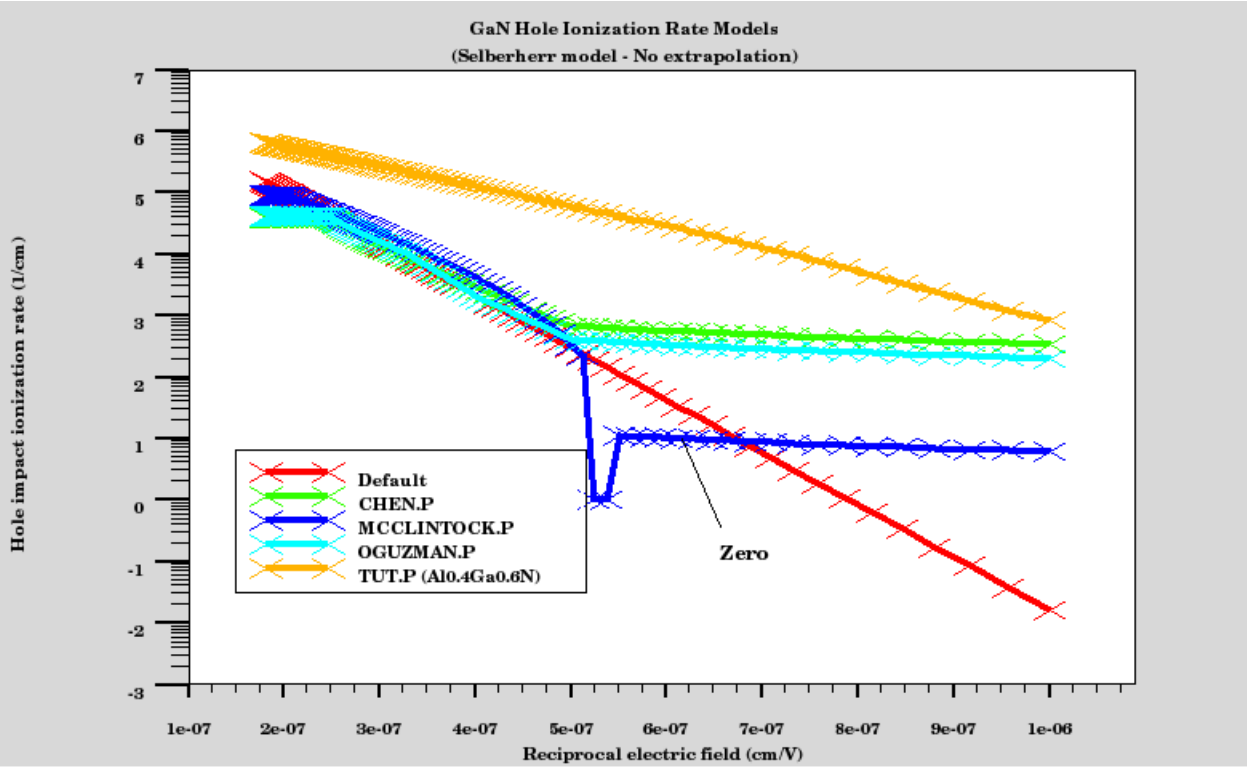


Figure 6-14: Comparison of hole ionization rates as a function of reciprocal field with extrapolation disabled.

Recombination Parameters

Table 6-43 shows the default values for radiative rates for the the binary wurtzite nitride compounds.

Table 6-43 Default Radiative Rates for Nitrides (MATERIAL COPT)

Material	COPT	Units	Reference
GaN	1.1×10^{-8}	cm^3/s	[217]
InN	2.0×10^{-10}	cm^3/s	[378]
AlN	0.4×10^{-10}	cm^3/s	[340]

Note: Currently, no default formula have been implemented for ternary and quaternary compounds for COPT.

Default Models for Heat Capacity and Themal Conductivity

By default, the heat capacity and thermal conductivities used in self-consistent heat flow simulations (Giga and Giga 3D) for the GaN/AlGaN/InGaN system are given by the following equations [246].

$$C_L(T) = C_L(300K)\frac{20-(\Theta_D/T)^2}{20-(\Theta_D/300K)^2}\rho_L \quad 6\text{-}163$$

$$\kappa_L(T) = \kappa_L(300K)\left(\frac{T}{300K}\right)^{\delta_\kappa} \quad 6\text{-}164$$

Here, T is the local temperature and the other parameters are given in Table 6-44 [246].

Table 6-44 Default Heat Capacity and Thermal Conductivity Parameters for GaN

Parameter Unit	κ_L (W/Kcm)	C_L (Ws/gK)	ρ_L (g/cm^3)	Θ_D (K)	δ_κ
GaN	1.30	0.49	6.15	600	-0.28
AlN	2.85	0.6	3.23	1150	-1.64
InN	0.45	0.32	6.81	660	0.0

Values for ternary and quaternary compositions are obtained by linear interpolation as a function of composition.

Adachi's Refractive Index Model [246]

For the III-V nitride compounds, Adachi's refractive index model is expressed by Equation 6-165.

$$n_r(\omega) = \sqrt{A\left(\frac{h\omega}{E_g}\right)^{-2}\left\{2-\sqrt{1+\frac{h\omega}{E_g}}-\sqrt{1-\frac{h\omega}{E_g}}\right\}+B} \quad 6\text{-}165$$

where E_g is the bandgap, ω is the optical frequency and A and B are material composition dependent parameters. For $Al_xGa_{1-x}N$, the compositional dependence of the A and B parameters are given by the expressions in Equations 6-166 and 6-167.

$$A(x) = 9.827 - 8.216X - 31.59X^2 \quad 6\text{-}166$$

$$B(x) = 2.736 + 0.842X - 6.293X^2 \quad 6\text{-}167$$

For $In_xGa_{1-x}N$, the compositional dependence of the A and B parameters are given by the expressions in Equations 6-168 and 6-169.

$$A(x) = 9.827(1+X) - 53.57X \quad 6\text{-}168$$

$$B(x) = 2.736(1-X) - 9.19X \quad 6\text{-}169$$

GANFET Convenience Model

We have introduced a convenience flag for enabling a specific set of defaults useful for simulation of most GaN FET type devices. The GANFET parameter of the **MODELS** statement enables POLAR, CALC.STRAIN, the Albrecht mobility model for electrons, and SRH and defaults POLAR.SCALE to 1.0.

6.4.9 The Hg(1-x)Cd(x)Te System

The following sections describe the relationship between mole fraction, X.COMP, and the material parameters of the of the Hg(1-x)Cd(x)Te system. This data has been taken from [346].

Bandgap

Equation 6-170 is used in the calculation of bandgap as a function of composition fraction.

$$Eg = -0.302 + 1.93*\mathrm{X.COMP} - 0.810*\mathrm{X.COMP}^2 + 0.832*\mathrm{X.COMP}^3 + 5.354\times 10^{-4}*\left(1 - 2*\mathrm{X.COMP}\right) T_L \quad \text{6-170}$$

Here, T_L is lattice temperature.

Electron Affinity

Equation 6-171 is used in the calculation of electron affinity as a function of composition fraction.

$$\chi = 4.23 - 0.813*(Eg(T_L) - 0.083) \quad \text{6-171}$$

Permittivity

Equation 6-172 is used in the calculation of relative permittivity as a function of composition fraction.

$$\varepsilon = 20.5 - 15.5*\mathrm{X.COMP} + 5.7*\mathrm{X.COMP}^2 \quad \text{6-172}$$

Density of States Masses

Equations 6-173 and 6-174 are used to calculate the density of states effective masses of electrons and holes.

$$\frac{me}{m_0} = \left[-0.6 + 6.333*\left(\frac{2}{Eg(TL)} + \frac{1}{Eg(TL)+1}\right)\right]^{-1} \quad \text{6-173}$$

$$\frac{me}{m_0} = 0.55 \quad \text{6-174}$$

Mobility

Equations 6-175 and 6-176 are used to calculate the electron and hole mobilities as functions of composition fraction.

$$\mu_e = 9 \times 10^8 \left(\frac{0.2}{\mathrm{X.COMP}} \right)^{7.5} T_L^{-2} \left(\frac{0.2}{\mathrm{X.COMP}} \right)^{0.6} \quad \text{6-175}$$

$$\mu_h = 0.01 * \mu e \quad \text{6-176}$$

6.4.10 CIGS (CuIn(1-x)Ga(x)Se2), CdS, and Zn0

The following table gives the default material parameter values for CIGS, CdS, and Zn0 [100, 174, 290].

Table 6-45 Default Parameters ofr CIGS, CdS, and Zn0

Parameter	Zn0	CdS	CIGS
εr	9	10	13.6
μn	100	100	100
μp	25	25	25
Eg	3.3	2.48	Equation 6-177
χ	4.5	4.18	4.58
Nc	2.2×10^{18}	2.41×10^{18}	2.2×10^{18}
Nv	1.8×10^{19}	2.57×10^{19}	1.8×10^{19}

$$Eg = (1.036 + 4.238\times10^{-5} * T_L + -8.875\times10^{-5} * 170.0/(\tanh(170.0*0.5/T_L) - 1.0)(1 - \mathrm{X.COMP}) \quad \text{6-177}$$
$$+ (1.691 + 8.820 \times 10^{-5} * T_L - 16.0 \times 10^{-5} * 189.0/\tanh(189.8*0.5/T_L) - 1.0)\mathrm{X.COMP} - 0.02$$
$$*\mathrm{X.COMP}*(1-\mathrm{X.COMP})$$

Zn0 Velocity Saturation

The electron velocity saturation model described in Figure 6-10 and Table 6-38 [227] can be used by setting the OZGUR.N parameter of the **MOBILITY** statement.

6.5 Simulating Heterojunction Devices with Blaze

6.5.1 Defining Material Regions with Positionally-Dependent Band Structure

Step Junctions

The easiest way to define a device with positionally dependent band structure is to specify two adjacent semiconductor regions with dissimilar bandgap. In this case, Blaze would simulate an abrupt heterojunction between the two materials.

For example, you want to simulate an abrupt heterojunction parallel to the X axis at a location of y=0.1 microns. For values of y greater than 0.1 specify, GaAs. For values of y less than 0.1, specify AlGaAs with a composition fraction of 0.3. The following statements would specify this situation.

```
REGION Y.MIN=0.1 MATERIAL=GaAs

REGION Y.MAx=0.1 MATERIAL=AlGaAs x.COMP=0.3
```

This fragment specifies that the two regions form an abrupt heterojunction at Y=0.1. The first region is composed of GaAs while the second is composed of AlGaAs.

These two material names are used by Blaze to choose default material models for the two regions. For a complete list of the materials available in Atlas/Blaze, see Appendix B "Material Systems". For the AlGaAs region a composition fraction of 0.3 is specified.

Graded Junctions

A grading can be applied to this heterojunction with a simple modification. For example:

```
REGION Y.MIN=0.1 MATERIAL=GaAs

REGION Y.MAx=0.1 MATERIAL=AlGaAs x.COMP=0.3 GRAD.34=0.01
```

specifies that the composition fraction of the AlGaAs region decreases from 0.3 at y=0.1 microns to 0.0 at y=0.11 microns. The GRAD parameter specifies the distance over which the mole fraction reduces to zero. The GRAD parameter is indexed such that GRAD.12 corresponds to the Y.MIN side of the region, GRAD.23 corresponds to the X.MAX side of the region, GRAD.34 corresponds to the Y.MAX side of the region, and GRAD.41 corresponds to the X.MIN side of the region. In most cases, the GRAD.n parameter acts to increase the size of the region. By default, the GRAD.n parameters are set to zero and all heterojunctions are abrupt. Note that the GRAD parameter acts just like the other region geometry parameters in that later defined regions overlapping the graded part of the region will overlap the grading. If in the previous example the grading had been applied to the GaAs region, it would be over lapped by the AlGaAs region. This would have produced an abrupt interface. A solution would be to limit Y.MAX in the AlGaAs region to 0.09. Make sure you specify regions in the proper order to avoid such problems.

Along similar lines, you can use the overlapping of regions to an advantage in forming graded heterojunctions between two materials in the same system with different non-zero composition fractions. For example:

```
REGION Y.MIN=0.1 MATERIAL=AlGaAs x.COMP=0.3 GRAD.12=0.02

REGION Y.MAx=0.11 MATERIAL=AlGaAs x.COMP=0.1
```

specifies a graded heterojunction with a composition of 0.3 at y = 0.1 falling to 0.1 at y = 0.11.

6.5.2 Defining Materials and Models

Materials

For example to set the bandgap for the material, InP, use the following syntax.

```
MATERIAL MATERIAL=InP EG300=1.35
```

Models

Blaze has two ways of simulating the physical effects of variations in semiconductor composition. For relatively gradual variations in composition, the standard modifications to the drift-diffusion equations can be considered adequate for simulation purposes. For abrupt heterojunctions, it has been suggested that thermionic emission may be the dominant factor in the behavior of heterojunction behavior.

Individual material parameters and models can be defined for each material or region. These models are set in the **MATERIAL**, **MODELS**, and **IMPACT** statements.

This statement uses the MATERIAL parameter to select all regions composed of the material "InP". The bandgap in these regions will be set to 1.35. There are two ways to set the parameters of a particular region. The first is to use the region index. For example:

```
MODEL REGION=1 BGN
```

In this case, the bandgap narrowing model is enabled in the region indexed number 1.

The second is to use the region name. For example:

```
IMPACT NAME=substrate SELB
```

This example turns on the Selberherr Impact Ionization Model in the region named substrate. You can then set the parameters for all regions and materials by omitting the MATERIAL, REGION, or NAME parameters, as in the following:

```
MODEL BGN
```

This statement sets the bandgap narrowing model for all regions and materials

Parser Functions

To use the C-Interpreter functions, you need to know the C programming language. See Appendix A “C-Interpreter Functions” for a description of the parser functions.

To specify a completely arbitrary spatial variation of varying composition fraction, use a parser function. To define the parser function for composition fraction, write a C function describing the composition fraction as a function of position. A template for the function called COMPOSITION is provided with this release of Atlas. Once you define the COMPOSITION function, store it in a file. To use the function for composition, set the F.COMPOSIT parameter to the file name of the function.

Chapter 7
3D Device Simulator

7.1 3D Device Simulation Programs

This chapter aims to highlight the extra information required for 3D simulation as compared to 2D. You should be familiar with the equivalent 2D models before reading this chapter.

This chapter describes the set of Atlas products that extends 2D simulation models and techniques and applies them to general non-planar 3D structures. The structural definition, models and material parameters settings and solution techniques are similar to 2D. You should be familiar with the simulation techniques described in Chapter 2 "Getting Started with Atlas" and the equivalent 2D product chapters before reading the sections that follow. The products that form 3D device simulation in Atlas are:

- Device 3D – silicon compound material and heterojunction simulation
- Giga 3D – non-isothermal simulation
- MixedMode 3D – mixed device-circuit simulation
- TFT 3D – thin film transistor simulation
- Quantum 3D – quantum effects simulation
- Luminous 3D – photodetection simulation

The 3D modules, Thermal3D, are described in Chapter 18 "Thermal 3D: Thermal Packaging Simulator".

In a similar manner to the 2D products, Giga 3D, MixedMode 3D, Luminous 3D and Quantum 3D should be combined with both Device 3D or Blaze3D depending on the semiconductor materials used.

7.1.1 Device 3D

Device 3D provides semiconductor device in 3D. Its use is analogous to the 2D simulations in S-Pisces and Blaze. See Chapter 5 "S-Pisces: Silicon Based 2D Simulator" for more information on S-Pisces. See Chapter 6 "Blaze: Compound Material 2D Simulator" for more information on Blaze.

7.1.2 Giga 3D

Giga 3D is an extension of Device 3D that accounts for lattice heat flow in 3D devices. Giga 3D has all the functionality of Giga (see Chapter 8 "Giga: Self-Heating Simulator") with a few exceptions. One exception is that additional syntax has been added to account for the three dimensional nature of thermal contacts. You can specify the `Z.MIN` and `Z.MAX` parameters on the **`THERMCONTACT`** statement to describe the extent of the contact in the Z direction. Another exception is that there is no `BLOCK` method available in the 3D version.

7.1.3 TFT 3D

TFT 3D is an extension of Device 3D that allows you to simulate amorphous and polycrystalline semiconductor materials in three dimensions. TFT 3D is completely analogous to the TFT simulator described in Chapter 15 "TFT: Thin-Film Transistor Simulator". The complete functionality of the TFT simulator is available in TFT 3D for three dimensional devices.

7.1.4 MixedMode 3D

MixedMode 3D is an extension of Device 3D or Blaze3D that allows you to simulate physical devices embedded in lumped element circuits (SPICE circuits). MixedMode 3D is completely analogous to the MixedMode simulator, which is described in Chapter 13 "MixedMode: Mixed Circuit and Device Simulator". The complete functionality of the MixedMode simulator is available in MixedMode 3D for three dimensional devices.

7.1.5 Quantum 3D

Quantum 3D is an extension of Device 3D or Blaze3D, which allows you to simulate of the effects of quantum confinement using the Quantum Transport Model (Quantum Moments Model). Quantum 3D is completely analogous to the Quantum model but applies to three dimensional devices. See Chapter 14 "Quantum: Quantum Effect Simulator" for more information on Quantum.

7.1.6 Luminous 3D

Luminous 3D is an extension of Device 3D that allows you to simulate photodetection in three dimensions. Luminous 3D is analogous to the Luminous simulator, which is described in Chapter 11 "Luminous: Optoelectronic Simulator" with a few significant differences described in Section 7.3.11 "Luminous 3D Models".

7.2 3D Structure Generation

All 3D programs in Atlas supports structures defined on 3D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension.

There are two methods for creating a 3D structure that can be used with Atlas. One way is through the command syntax of Atlas. Another way is through an interface to DevEdit3D.

A direct interface from Athena to 3D Atlas impossible. But DevEdit3D provides the ability to read in 2D structures from Athena and extend them non-uniformly to create 3D structures for Atlas.

Atlas Syntax For 3D Structure Generation

Mesh generation

Section 2.6.3 “Using The Command Language To Define A Structure” covers the generation of 2D and 3D mesh structures using the Atlas command language. The `Z.MESH` statement and the `NZ` and `THREE.D` parameters of the `MESH` statement are required to extend a 2D mesh into 3D.

Conventionally, slices are made perpendicular to the Z axis. The mesh is triangular in XY but rectangular in XZ or YZ planes.

Region, Electrode, and Doping definition

Section 2.6.3 “Using The Command Language To Define A Structure” also covers the definition of 2D regions, electrodes and doping profiles. To extend the regions into 3D, use the `Z.MIN` and `Z.MAX` parameters. For example:

```
REGION NUM=2 MATERIAL=Silicon X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1
Z.MIN=0 Z.MAX=1

ELECTRODE NAME=gate X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1 Z.MIN=0
Z.MAX=1

DOPING GAUSS N.TYPE CONC=1E20 JUNC=0.2 Z.MIN=0.0 Z.MAX=1.0
```

For 2D regions or electrodes defined with the command language, geometry is limited to rectangular shapes. Similarly, in 3D regions and electrodes are composed of rectangular parallelopipeds.

DevEdit3D Interface

DevEdit3D is a graphical tool that allows you to draw 3D device structures and create 3D meshes. It can also read 2D structures from Athena and extend them into 3D. These structures can be saved from DevEdit3D as structure files for Atlas. Also, save a command file when using DevEdit3D. This file is used to recreate the 3D structure inside DevEdit3D, which is important, since DevEdit3D doesn’t read in 3D structure files.

Atlas can read structures generated by DevEdit3D using the command:

```
MESH INF=<filename>
```

The program is able to distinguish automatically between 2D and 3D meshes read in using this command.

Defining Devices with Circular Masks

DevEdit3D makes a triangular mesh in the XY plane and uses Z plane slices. This means, that normally the Y direction is vertically down into the substrate. But in the case of using circular masks, you need to rotate the device.

With defining devices using circular masks in DevEdit3D, the XY plane should be the surface of the device and the Z direction should be into the substrate.

7.3 Model And Material Parameter Selection in 3D

Models and material parameters are chosen in 3D in common with other 2D modules using the **MODELS**, **IMPACT**, **MATERIAL**, **MOBILITY**, **INTERFACE**, and **CONTACT** statements. The following models are available in 3D device simulation programs. All of these models are documented in Chapter 3 "Physics" or in the 2D product chapters.

7.3.1 Mobility

- Table for 300K (CONMOB)
- Thomas's model (ANALYTIC)
- Arora's model (ARORA)
- Klaassen's model (KLAASSEN)
- Lombardi's model (CVT)
- Yamaguchi's model (YAMA)
- Parallel field dependence (FLDMOB)
- Parallel field dependence with negative differential mobility (FLDMOB EVSATMOD=1)

7.3.2 Recombination

- Shockley Read Hall (SRH)
- Concentration dependent lifetime SRH (CONSRH)
- Klaassen's concentration dependent lifetime SRH (KLASRH)
- Auger (AUGER)
- Klaassen's concentration dependent Auger recombination model (KLAAUG)
- Optical recombination (OPTR)
- Bulk and interface traps (TRAP, INTTRAP)
- Continuous defect states (DEFECT)

7.3.3 Generation

- Selberherr's impact ionization (IMPACT SELB)
- Crowell's impact ionization (IMPACT CROWELL)
- Hot electron injection (HEI)
- Fowler Nordheim tunneling (FNORD)

7.3.4 Carrier Statistics

- Boltzmann (default)
- Fermi (FERMI)
- Band gap narrowing (BGN)
- Incomplete ionization (INCOMPLETE)
- Quantum mechanical effects (QUANTUM)
- Heterojunction thermionic field emission

7.3.5 Boundary Conditions

- Ohmic and Schottky
- Current boundary conditions
- Lumped element boundary conditions
- Distributed contact resistance

7.3.6 Optical

- Photogeneration with ray tracing (Luminous 3D).

7.3.7 Single Event Upset Simulation

- Single event upset simulation.

All these models, with the exception of **SINGLEEVENTUPSET**, are documented in the Chapter 3 “Physics” or in the 2D product chapters of this manual.

7.3.8 Boundary Conditions in 3D

External Passive Elements

You can attach external lumped resistors, capacitors and inductors to any contact. The syntax is the same as for the 2D products, which is:

```
CONTACT NAME=drain RES=1e3 CAP=1e-12 L=1e-6
```

You can also apply distributed resistances to contacts. The algorithm used for estimating contact area for 3D distributed contact resistance multiplies the contact perimeter in a given Z plane by the displacement in the Z direction. This algorithm will only work properly for planar contacts that do not vary in the Z direction. They may however abruptly terminate or start in the Z direction.

The units of lumped external passive elements are ohms for resistors, Farads for capacitors and Henrys for inductors. Distributed contact resistance is defined in ohms.cm^3.

Thermal Contacts for Giga 3D

Thermal contacts for non-isothermal simulation in Giga 3D are defined in an analogous manner to the 2D thermal contacts in Giga. The Z.MIN and Z.MAX parameters are used to define the extent of the thermal contact in the Z plane. The units of the thermal resistance parameter ALPHA are scaled in 3D to W/(cm.K). For more information about Giga 3D, see Chapter 8 “Giga: Self-Heating Simulator”.

7.3.9 TFT 3D Models

Models for simulating thin-film transistors made from amorphous or polycrystalline semiconductors are supported in TFT 3D. The definition of the continuous defect states in the bandgap is performed using the same parameters as in 2D simulations with TFT. The models for continuous defect (or trap) densities are documented in Chapter 15 “TFT: Thin-Film Transistor Simulator”.

7.3.10 Quantum 3D Models

Models for simulating quantum effects semiconductors are supported in Quantum 3D. The definition of the quantum moments solver is the same as in 2D simulations with Quantum. The models for simulating quantum effects and the parameters to control the model are shown in Chapter 14 “Quantum: Quantum Effect Simulator”.

7.3.11 Luminous 3D Models

Many of the models for simulating photodetection in Luminous 3D are similar to those for simulating photodetection in Luminous. For more information about Luminous, see Chapter 11 “Luminous: Optoelectronic Simulator”. This section, however, shows several important differences between Luminous and Luminous 3D.

7.4 Numerical Methods for 3D

7.4.1 DC Solutions

There are several differences between the default numerical methods applied in 2D Atlas and those applied in 3D Atlas. For example, with respect to non-linear iteration strategies, the current version of the 3D simulator does not support the BLOCK method. The NEWTON and GUMMEL iteration strategies are supported for 3D simulations, whereas NEWTON, GUMMEL and BLOCK are all supported for 2D simulations.

In solving the linear subproblem, the default approach in 3D is to use an iterative solver. This is believed to be the most efficient method for general 3D problems. In 2D, the direct solver is used by default. You may find it desirable to use direct methods in 3D problems due to improved convergence of computational efficiency. You can select the direct method by specifying DIRECT in the **METHOD** statement. Also, in 3D there are two linear iterative solution methods available. The defaults are ILUCGS (incomplete lower-upper decomposition conjugate gradient system) and BICGST (bi-conjugate gradient stabilized). Historically, tests have shown that the current implementation of ILUCGS is slightly more stable than BICGST and is the default iterative solver in 3D. You can define the BICGST solver by specifying BICGST in the **METHOD** statement.

Note: Iterative solvers are recommended for large problems, typically greater than 5000 node points, due to lower solution times and memory usage.

7.4.2 Transient Solutions

In transient mode, a semi-implicit scheme is used in addition to the default TR-BDF algorithm [3]. This algorithm is recommended for complex simulations such as Single Event Upset. To select this method, use:

```
METHOD HALFIMPLICIT
```

It is available for solving the drift-diffusion equations and if LAT.TEMP is specified on the **MODELS** statement. If a carrier temperature solution is requested (HCTE.EL or HCTE.HO or both), then HALFIMPLICIT cannot be used.

7.4.3 Obtaining Solutions In 3D

Atlas3D programs can perform DC and transient analysis in an equivalent manner to 2D. The SOLVE statement is used to define the solution procedure. The syntax used is described in Section 2.9 “Obtaining Solutions”.

7.4.4 Interpreting the Results From 3D

The log files produced by 3D Atlas can be plotted in TonyPlot exactly as those that result from S-Pisces or Blaze. The only difference is the units of the currents produced. Log files from 3D simulations save current in Amperes, whereas the 2D simulations use Amperes/micron.

The solution files produced by 3D Atlas should be plotted using TonyPlot3D. These files cannot be read directly into the 2D TonyPlot program. TonyPlot3D contains features that allows you to make slices of the 3D structure, which can be plotted in 2D TonyPlot. For more information about TonyPlot 3D, see the TonyPlot3D User's Manual.

7.4.5 More Information

Many examples using 3D Atlas have been provided on the distribution package. You can find more information about 3D Atlas by reading the text associated with each example. You can also find find some more information at www.silvaco.com.

Chapter 8
Giga: Self-Heating Simulator

8.1 Overview

Giga extends Atlas to account for lattice heat flow and general thermal environments. Giga implements Wachutka's thermodynamically rigorous model of lattice heating [336], which accounts for Joule heating, heating, and cooling due to carrier generation and recombination, and the Peltier and Thomson effects. Giga accounts for the dependence of material and transport parameters on the lattice temperature. Giga also supports the specification of general thermal environments using a combination of realistic heat-sink structures, thermal impedances, and specified ambient temperatures. Giga works with both S-Pisces and Blaze and with both the drift-diffusion and energy balance transport models. See the "Drift-Diffusion Transport Model" on page 102 and the "Energy Balance Transport Model" on page 104 for more information on these models.

Before continuing with this chapter, you should be familiar with Atlas. If not, read Chapter 2 "Getting Started with Atlas", along with Chapter 5 "S-Pisces: Silicon Based 2D Simulator" or Chapter 6 "Blaze: Compound Material 2D Simulator".

8.1.1 Applications

A major application of Giga is the simulation of high-power structures including bipolar, MOS, IGBT, and thyristor devices. Another important application is the simulation of electrostatic discharge (ESD) protection devices. Thermal effects are also important in SOI device operation due to the low thermal conductivity of the buried oxide, and in devices fabricated in III-V material systems due to the relatively low thermal conductivity of these materials.

Recent studies have demonstrated that accounting self-consistently for lattice heating is necessary for accurate simulation of bipolar VLSI devices. This is due to the sensitive temperature dependence of the carrier injection process. Since bipolar devices are key components of CMOS technologies and many devices can be impacted by parasitic bipolar effects, the applications of Giga are very general.

For other more information on Giga's application, see Section 8.3 "Applications of GIGA".

8.1.2 Numerics

Giga supplies numerical techniques that provide efficient and robust solution of the complicated systems of equations that result when lattice heating is accounted for. These numerical techniques include fully-coupled and block iteration methods. When Giga is used with the energy balance equations, the result is a "six equation solver", which defines the state-of-the-art for general purpose device simulation.

For more information on numerical techniques, see Chapter 21: "Numerical Techniques".

8.2 Physical Models

8.2.1 The Lattice Heat Flow Equation

Giga adds the heat flow equation to the primary equations that are solved by Atlas. The heat flow equation has the form:

$$C\frac{\partial T_L}{\partial t} = \nabla(\kappa\nabla T_L) + H \qquad 8\text{-}1$$

where:

- C is the heat capacitance per unit volume.
- κ is the thermal conductivity.
- H is the heat generation
- T_L is the local lattice temperature.

The heat capacitance can be expressed as $C = \rho C_p$, where C_p is the specific heat and ρ is the density of the material.

Specifying the LAT.TEMP parameter in the **MODELS** statement includes the lattice heat flow equation in Atlas simulations.

Giga supports different combinations of models. For example, if the HCTE and LAT.TEMP parameters are specified in the **MODELS** statement and both particle continuity equations are solved, all six equations are solved. If HCTE.EL is specified instead of HCTE, only five equations are solved and the hole temperature T_p is set equal to lattice temperature T_L.

8.2.2 Specifying Thermal Conductivity

Standard Model

The value of thermal conductivity, k, for each region should be specified in the **MATERIAL** statement. Because thermal conductivity is generally temperature dependent, the following four models are available.

Table 8-1 Standard Thermal Conductivity Models

Equation	Units	MATERIAL Flag
$k(T) = \text{TC.CONST}$	$(W/cm\cdot K)$	TCON.CONST
$k(T) = (\text{TC.CONST})/(T_L/300)^{\text{TC.NPOW}}$	$(W/cm\cdot K)$	TCON.POWER
$k(T) = 1/(\text{TC.A} + (\text{TC.B})*T_L + (\text{TC.C})*T_L^2)$	$(W/cm\cdot K)$	TCON.POLYN
$k(T) = (\text{TC.E})/(\text{T}_\text{L} - \text{TC.D})$	$(W/cm\cdot K)$	TCON.RECIP

The TC.CONST, TC.NPOW, TC.A, TC.B, TC.C, TC.D and TC.E parameters are all user-specifiable in the **MATERIAL** statement. As described in Table 8-1, the choice of models is also user-specified in the **MATERIAL** statement.

To clarify model selection and parameter value definition, the following syntax examples are given.

Enabling TCON.POWER:

```
material mat=Diamond TCON.POWER TC.NPOW=2 TC.CONST=1.4
```

Enabling TCON.POLYN:

```
material mat=GaN TCON.POLYN TC.A=1 TC.B=2 TC.C=4
```

Enabling TCON.RECIP:

```
material mat=silicon TCON.RECIP TC.B=2 TC.D=4
```

The thermal model and parameter selection can also be completed through REGION statements as follows:

```
MATERIAL REGION=5 TC.A=<n> TC.B=<n> TC.C=<n>
MATERIAL REGION=6 TC.A=<n> TC.B=<n> TC.C=<n>
```

It is also advised that you include PRINT on the **MODELS** statement to verify model selection and parameter values used.

Compositionally Dependent Thermal Conductivity (TCON.COMP) [236]

The key material dependent parameters in Equation 8-1 are the thermal conductivity and the heat capacity. In general, both thermal conductivity and heat capacity are both composition and temperature dependent. In later sections, we will show how you have a great deal of flexibility in specifying these model dependencies.

For most materials, we provide reasonable default values for these material parameters, which you can examine by specifying PRINT on the **MODELS** statement. These models allow user specification but are not material composition dependent. As such, we have provided default models for some of the more popular materials that are both composition and temperature dependent. These models described here apply to the following materials: Si, Ge, GaAs, AlAs, InAs, InP, GaP, SiGe, AlGaAs, InGaAs, InAlAs, InAsP, GaAsP and InGaP.

The default built-in model derives temperature dependency in a manner very similar to the user specifiable models discussed later in this section.

The composition depencencies of the ternary compounds and the binary, SiGe, are derived from interpolation from the binary compounds (or the values for Si and Ge in the case of SiGe).

The basic model for thermal conductivity is given by:

$$\kappa(T_L) = K_{300} \cdot \left(\frac{T_L}{300}\right)^{\alpha} \qquad 8\text{-}2$$

where $\kappa(T_L)$ is the temperature dependent thermal conductivity, T_L is the lattice temperature and K_{300} and α are material dependent parameters.

Table 8-2 shows the default values for K_{300} and α for Si, Ge and the binary III-V compounds.

Table 8-2 Default Parameter Values for Thermal Conductivity

Material	K_{300}(W/Kcm)	α
Si	1.48	-1.65
Ge	0.60	-1.25
GaAs	0.46	-1.25
AlAs	0.80	-1.37
InAs	0.273	-1.1
InP	0.68	-1.4
GaP	0.77	-1.4

The parameters K_{300} and α are interpolated as a function of composition fraction x using Equations 8-3 and 8-4.

$$K_{300}^{AB} = \frac{1}{\left(\frac{1-x}{K_{300}^{A}} + \frac{x}{K_{300}^{B}} + \frac{(1-x)x}{C}\right)} \quad \text{8-3}$$

$$\alpha^{AB} = (1-x)\alpha^{A} + x\alpha^{B} \quad \text{8-4}$$

The C parameter in Equation 8-3 is a bowing factor used to account for the non-linear aspects of the variation of thermal conductivity with composition. Table 8-3 shows the default values of the bowing factor, C, for the various ternary compounds and SiGe.

Table 8-3 Default Bowing Parameter Values for Thermal Conductivity

Material	C(W/K cm)
SiGe	0.028
AlGaAs	0.033
InGaAs	0.014
InAlAs	0.033
InAsP	0.033
GaAsP	0.014
InGaP	0.014

C-Interpreter Defined Thermal Conductivity

You can use the C-Interpreter to define the thermal conductivity, κ, as a function of the lattice temperature, position, doping and fraction composition. This is defined using the syntax:

```
MATERIAL REGION=<n> F.TCOND=<filename>
```

where the <filename> parameter is an ASCII file containing the C-Interpreter function. For more information about the C-Interpreter, see Appendix A "C-Interpreter Functions".

Anisotropic Thermal Conductivity

The flux term in Equation 8-1 models the thermal conductivity κ as being isotropic by default. You can specify an anisotropic thermal conductivity κ_{aniso}. In Giga2D, the anisotropic value is applied in the Y direction. In Giga 3D, it is applied in the Z direction by default. In this case, the thermal conductivity tensor is

$$\kappa = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_1 & 0 \\ 0 & 0 & \kappa_2 \end{bmatrix} \qquad 8\text{-}5$$

You can also change the anisotropic direction to be the Y direction instead of the Z direction by specifying the YDIR.ANISO parameter or by specifying ^ZDIR.ANISO.

The thermal conductivity is temperature dependent, and the anisotropic thermal conductivity must have a model selected for its dependence on temperature. To do this, select TANI.CONST, TANI.POWER, TANI.POLYNOM or TANI.RECIP. These are analagous to TCON.CONST, TCON.POWER, TCON.POLYNOM and TCON.RECIP respectively. You can specify different temperature dependency models for the κ(T) and κ_{aniso}(T). The parameters for the chosen model for κ_{aniso}(T) are specified by using the relevant parameters from A.TC.CONST, A.TC.A, A.TC.B, A.TC.C, A.TC.D, A.TC.E and A.TC.NPOW. These are equivalent to the isotropic parameters but with a prefix of A. to distinguish them. No built-in models are available for the anisotropic component of thermal conductivity. To give the anisotropic component the same temperature dependence as for the built-in model (Equation 8-2), specify the TANI.POWER.

As in the case of anisotropic permittivity, the discretization of the flux term is modified. For the simple cases of a rectangular mesh and a diagonal thermal conductivity tensor, the discretization chooses the value of thermal conductivity appropriate to the direction.

If the coordinate system where the thermal conductivity tensor is diagonal and the Atlas coordinate system are non-coincident, then the coordinate transformation can be specified as a set of vectors

```
X = (X.X, X.Y, X.Z)
Y = (Y.X, Y.Y, Y.Z)
Z = (Z.X, Z.Y, Z.Z)
```

where the vector components can be specified on the **MATERIAL** statement. The default is that

```
X = (1.0, 0.0, 0.0)
Y = (0.0, 1.0, 0.0)
Z = (0.0, 0.0, 1.0)
```

and you should specify all necessary components. You do not need to normalize the vectors to unit length. Atlas will normalize them and transform the thermal conductivity tensor according to the usual rules. If you specify any component of X, Y or Z, then a more complete form of discretization will be used.

This is similar to the complete discretization carried out for the case of a generally anisotropic dielectric permittivity described in Section 3.12 "Anisotropic Relative Dielectric Permittivity".

To enable the more complete form of discretization, use the TC.FULL.ANISO parameter on the **MATERIAL** statement.

Table 8-4 shows the parameters used in Anisotropic Thermal Conductivity.

Table 8-4 MATERIAL Statement Parameters

Parameter	Type	Default	Units
A.TC.CONST	Real	0.0	(W/cm/K)
A.TC.A	Real	0.0	cm K /W
A.TC.B	Real	0.0	cm /W
A.TC.C	Real	0.0	cm / W /K
A.TC.D	Real	0.0	K
A.TC.E	Real	0.0	W/cm
A.TC.NPOW	Real	0.0	-
TANI.CONST	Logical	False	
TANI.POWER	Logical	False	
TANI.POLYNOM	Logical	False	
TANI.RECIP	Logical	False	
TC.FULL.ANISO	Logical	False	
YDIR.ANISO	Logical	False	
ZDIR.ANISO	Logical	True	
X.X	Real	1.0	
X.Y	Real	0.0	
X.Z	Real	0.0	
Y.X	Real	0.0	
Y.Y	Real	1.0	
Y.Z	Real	0.0	

Table 8-4 MATERIAL Statement Parameters			
Parameter	**Type**	**Default**	**Units**
Z.X	Real	0.0	
Z.Y	Real	0.0	
Z.Z	Real	1.0	

8.2.3 Specifying Heat Capacity

Standard Model

For transient calculations, specify heat capacities for every region in the structure. These are also functions of the lattice temperature and are modeled as:

$$C = \mathrm{HC.A} + \mathrm{HC.B}\,T_L + \mathrm{HC.C}\,T_L^{\,2} + \frac{\mathrm{HC.D}}{T_L^{\,2}} \quad (J/cm^3/K) \qquad 8\text{-}6$$

Default values of HC.A, HC.B, HC.C, and HC.D are provided for common materials. You can specify these values in the **MATERIAL** statement.

The following statements will be used to specify the temperature dependent heat capacities of the regions previously defined.

```
MATERIAL REGION=5 HC.A=<n> HC.B=<n> HC.C=<n> HC.D=<n>
MATERIAL REGION=6 HC.A=<n> HC.B=<n> HC.C=<n> HC.D=<n>
```

Table 8-5 User Specifiable Parameters for Equation 8-6		
Statement	**Parameter**	**Units**
MATERIAL	HC.A	$J/cm^3/K$
MATERIAL	HC.B	$J/cm^3 \cdot K^2$
MATERIAL	HC.C	$J/cm^3 \cdot K^3$
MATERIAL	HC.D	JK/cm^3

To select the standard model for heat capacity, specify HC.STD on the **MATERIAL** statement.

Compositionally Dependent Heat Capacity [236]

The temperature dependence of heat capacity can be expressed by:

$$C(T_L) = \rho\left[C_{300} + C_1\frac{\left(\frac{T_L}{300}\right)^{\beta} - 1}{\left(\frac{T_L}{300}\right)^{\beta} + \frac{C_1}{C_{300}}}\right] \quad 8\text{-}7$$

where $C(T_L)$ is the temperature dependent heat capacity, ρ is the mass density, C_{300}, C_1 and β are material dependent parameters. Table 8-6 shows the default values of ρ, C_{300}, C_1 and β for the binary compounds and Si and Ge.

Table 8-6 Default Parameter Values for Heat Capacity

Material	ρ (g/cm^3)	C_{300} (J/K/kg)	C_1 (J/K/kg)	β
Si	2.33	711	255	1.85
Ge	5.327	360	130	1.3
GaAs	5.32	322	50	1.6
AlAs	3.76	441	50	1.2
InAs	5.667	394	50	1.95
InP	4.81	410	50	2.05
GaP	4.138	519	50	2.6

For the ternary compounds and SiGe, the parameters ρ, C_{300}, C_1 and β are interpolated versus composition using a simple linear form as in Equation 8-8.

$$P_{ab} = (1-x)P_a + xP_b \quad 8\text{-}8$$

When the lattice heat flow equation is solved, the global device temperature parameter is the maximum temperature at any mesh point in the device.

To select the compositionally dependent heat capacity, specify `HC.COMP` on the **`MATERIAL`** statement.

C-Interpreter Defined Thermal Capacity

You can use the C-Interpreter to define the thermal capacity, `TCAP`, as a function of the lattice temperature, position, doping and fraction composition. This is defined using the syntax:

```
MATERIAL REGION=<n.region> F.TCAP=<filename>
```

where the `<filename>` parameter is an ASCII file containing the C-Interpreter function. For more information about the C-Interpreter, see Appendix A "C-Interpreter Functions".

You can override the default values for ρ, C_{300}, C_1, and β by specifying `HC.RHO`, `HC.C300`, `HC.C1`, `HC.BETA` respectively on the **`MATERIAL`** statement.

8.2.4 Specifying Heat Sink Layers

You can define regions to only include thermal calculations. These regions will typically consist of layers associated with heat sinks. They are defined using the **REGION** statement. Even though in reality the heat sink materials are typically metal conductors, it is easier to specify these layers with the material type, INSULATOR. This is because as insulators the program will only solve heat flow and not attempt to solve current continuity in these layers. The region number is subsequently used as an identifier when thermal conductivities and heat capacities are assigned to these regions.

The following statements specify two layers of a heat sink for inclusion in the thermal calculation.

```
REGION NUM=5 Y.MIN=0.5 Y.MAX=2.0 INSULATOR
REGION NUM=6 Y.MIN=2.0 Y.MAX=3.0 INSULATOR
```

8.2.5 Non-Isothermal Models

Effective Density Of States

When lattice heating is specified with the drift-diffusion transport model, the effective density of states for electrons and holes are modeled as functions of the local lattice temperature as defined by Equations 3-31 and 3-32.

When lattice heating is specified with the energy balance model, the effective densities of states are modeled as functions of the local carrier temperatures, T_n and T_p, as defined by Equations 3-148 and 3-149.

Non-isothermal Current Densities

When Giga is used, the electron and hole current densities are modified to account for spatially varying lattice temperatures:

$$\vec{J}_n = -q\mu_n n(\nabla\phi_n + P_n\nabla T_L) \quad 8\text{-}9$$

$$\vec{J}_p = -q\mu_p p(\nabla\phi_p + P_p\nabla T_L) \quad 8\text{-}10$$

Here, P_n and P_p are the absolute thermoelectric powers for electrons and holes. P_n and P_p are modeled as follows:

$$P_n = -\frac{k_B}{Q}\left(\frac{5}{2} + ln\frac{N_c}{n} + \mathrm{KSN} + z_n\right) \quad 8\text{-}11$$

$$P_p = \frac{k_B}{Q}\left(\frac{5}{2} + ln\frac{N_v}{p} + \mathrm{KSP} + z_p\right) \quad 8\text{-}12$$

Here, k_B is Boltzmann's constant. KSN and KSP are the exponents in the power law relationship between relaxation time (or mobility) and carrier temperature. They are also used in the energy balance model. Their values typically range between -1 and 2 in Silicon and they are set on the **MODELS** statement.

You can also model them using the C-Interpreter functions F.KSN and F.KSP if a more complicated dependence is required.

The quantities ζ_n and ζ_p are the phonon drag contribution to the thermopower. These are only significant in low doped material and at low temperatures. Atlas has a built in model for this phonon drag contribution, which you enable by specifying the PHONONDRAG parameter on the **MODELS** statement. The built-in model is

$$z_n = \left(\frac{k_B}{Q}\right)\mathrm{PDA.N}\left(\frac{T_L}{300}\right)^{\mathrm{PDEXP.N}} \quad 8\text{-}13$$

for electrons and

$$z_p = \left(\frac{k_B}{Q}\right)\mathrm{PDA.P}\left(\frac{T_L}{300}\right)^{\mathrm{PDEXP.P}} \quad 8\text{-}14$$

for holes. A theoretically derived value for PDEXP.N and PDEXP.P is -7/2 [126] but experimentally obtained results generally give a value of around -5/2 [98]. The values of PDA.N and PDA.P depend on doping level and can also depend on sample size. Therefore, it is recommended that you choose values to fit to your sample. Alternatively, empirical data or more complicated formulae can be incorporated by using the C-Interpreter functions F.PDN and F.PDP on the **MATERIAL** or **MODELS** statements [25], [118], [324].

The thermopower can be thought of as having 3 components. The first is the derivative of the Fermi Potential with respect to temperature. For Maxwell-Boltzmann statistics, this is

$$-\frac{k_B}{Q}\left(\frac{3}{2} + ln\frac{Nc}{n}\right) \quad 8\text{-}15$$

for electrons and

$$\frac{k_B}{Q}\left(\frac{3}{2} + ln\frac{Nv}{p}\right) \quad 8\text{-}16$$

for holes.

Atlas incorporates this effect indirectly through the boundary conditions. It does not contribute directly to the temperature gradient term in the expressions for current, see Equations 8-9 and 8-10.

The second term is due to carrier scattering and is

$$-\frac{k_B}{Q}(1 + \mathrm{KSN}) \quad 8\text{-}17$$

for electrons and

$$\frac{k_B}{Q}(1 + \mathrm{KSP}) \quad 8\text{-}18$$

for holes. The third term is the phonon drag contribution -ζ_n and ζ_p, and the second and third terms are included directly into the temperature gradient term in the expressions for current.

Seebeck Effect

This is the phenomenon in which an open circuited voltage is generated across a semiconductor having a temperature gradient. To observe this in practice, you need two materials each with a different thermopower. In simulation, you can directly observe the Seebeck effect because you can measure the open circuit voltage between two contacts at different temperatures. Current boundary conditions are applied to one contact and solved for zero current.

This will result in a voltage difference between the two contacts.

The Seebeck coefficient is given as the ratio of the Voltage difference between the contacts to the temperature difference, and the most convenient units for expressing it are mV/K. To theoretically calculate the Voltage difference generated by the temperature gradient, integrate the thermopower as a function of temperature between the temperatures on the contact

$$V_1 - V_0 = -\int_{T_0}^{T_1} \frac{\mu_n N_D P_n(T) + \mu_p N_A P_p(T) dT}{\mu_n N_D + \mu_p N_A} \qquad 8\text{-}19$$

for a compensated material, where T_0 and T_1 are the temperatures on the contacts and V_0 and V_1 are the voltages. The effective Seebeck coefficient for the device is obtained by dividing the voltage difference so obtained by the temperature difference (T_1 - T_0).

Table 8-7 shows the default values for Equations 8-11 and 8-12.

Table 8-7 User-Specifiable Parameters for Equations 8-11 and 8-12

Statement	Parameter	Default	Units
MODELS	F.KSN		
MODELS	F.KSP		
MODELS	KSN	-1	None
MODELS	KSP	-1	None
MODELS	PHONONDRAG	False	
MODELS	F.PDN		
MODELS	F.PDP		
MATERIAL	F.PDN		
MATERIAL	F.PDP		
MATERIAL	PDA.N	0.2	
MATERIAL	PDA.P	0.2	
MATERIAL	PDEXP.N	-2.5	
MATERIAL	PDEXP.P	-2.5	

8.2.6 Heat Generation

When carrier transport is handled in the drift-diffusion approximation the heat generation term, H, used in Equation 8-1 has the form:

$$H = \frac{|\vec{J_n}|^2}{q\mu_n n} + \frac{|\vec{J_p}|^2}{q\mu_p p} - T_L(\vec{J_n}\nabla P_n) - T_L(\vec{J_p}\nabla P_p) \quad 8\text{-}20$$

$$+ q(R-G)\left[T_L\left(\frac{\partial \phi_n}{\partial T}_{n,p}\right) - \phi_n - T_L\left(\frac{\partial \phi_p}{\partial T}_{n,p}\right) + \phi_p\right]$$

$$-T_L\left[\left(\frac{\partial \phi_n}{\partial T}\right)_{n,p} + P_n\right] divJ_n - T_L\left[\left(\frac{\partial \phi_p}{\partial T}\right)_{n,p} + P_p\right] divJ_p$$

In the steady-state case, the current divergence can be replaced with the net recombination. Equation 8-20 then simplifies to:

$$H = \left[\frac{|\overrightarrow{J_n}|^2}{q\mu_n n} + \frac{|\overrightarrow{J_p}|^2}{q\mu_p p}\right] + q(R-G)[\phi_p - \phi_n + T_L(P_p - P_n)] - T_L(\vec{J_n}\nabla P_n + \vec{J_p}\nabla P_p) \quad 8\text{-}21$$

where:

$\left[\frac{|\vec{J_n}|^2}{q\mu_n n} + \frac{|\vec{J_p}|^2}{q\mu_p p}\right]$ is the Joule heating term,

$q(R-G)[\phi_p - \phi_n + T_L(P_p - P_n)]$ is the recombination and generation heating and cooling term,

$-T_L(\vec{J_n}\nabla P_n + \vec{J_p}\nabla P_p)$ accounts for the Peltier and Joule-Thomson effects.

A simple and intuitive form of H that has been widely used in the past is:

$$H = (\vec{J_n} + \vec{J_p}) \cdot \vec{E} \quad 8\text{-}22$$

Giga can use either Equations 8-21 or 8-22 for steady-state calculations. By default, Equation 8-22 is used. Equation 8-21 is used if the HEAT.FULL parameter is specified in the **MODELS** statement. To enable/disable the individual terms of Equation 8-21, use the JOULE.HEAT, GR.HEAT, and PT.HEAT parameters of the **MODELS** statement.

If the general expression shown in Equation 8-20 is used for the non-stationary case, the derivatives $\left(\frac{\partial \phi_n}{\partial T_L}\right)_{n,p}$ and $\left(\frac{\partial \phi_p}{\partial T_L}\right)_{n,p}$ are evaluated for the case of an idealized non-degenerate semiconductor and complete ionization.

The heat generation term, H, is always set equal to 0 in insulators. For semiconductors, two numerical approaches to discretizing the Joule heating term have been implemented. The first method approximates the term as a special type of flux term. The second method approximates the term as a source term. These may give different results in some cases. The first method is the default, while the second method may be enabled by specifying

```
METHOD ^FLUX.JOULE
```

to explicitly clear the inclusion of Joule heating as a flux term. This option is included primarily for compatibility with Victory Device.

For conductors $H = \frac{(\nabla V)^2}{\rho}$.

When electron and hole transport are modeled in the energy balance approximation (by specifying HCTE on the **MODELS** statement), the following expression is used for H:

$$H = W_n + W_p + E_g \ U, \tag{8-23}$$

where U, W_n, and W_p are defined by Equations 3-150 through 3-152.

If the energy balance model is enabled for only electrons or only holes, then a hybrid of Equations 8-23 and 8-21 or 8-22 is used. For example if the energy balance equation is solved for electrons, but not for holes, then H is evaluated as follows if HEAT.FULL is specified.

$$H = W_n + E_g U + \frac{\left|\vec{J_p}\right|^2}{q\mu_p p} - T_L \vec{J_p} \nabla P_p \tag{8-24}$$

A simpler form for the heating will be used if HEAT.FULL is not specified.

$$H = W_n + E_g U + \vec{J_p} \cdot \vec{E} \tag{8-25}$$

If the energy balance equation is solved for holes, but drift-diffusion is solved for holes, then the heating term is analagous to Equations 8-24 and 8-25 with hole terms swapped for electron terms and vice versa.

The first terms of W_n and W_p are output to structure files as Joule heating. The last term in Equation 8-24 is output as Peltier Thomson Heat power. The remaining terms are output as recombination heat power.

8.2.7 Thermal Boundary Conditions

At least one thermal boundary condition must be specified when the lattice heat flow equation is solved. The thermal boundary conditions used have the following general form:

$$\sigma(\vec{J}_{tot}^{u} \cdot \vec{s}) = \alpha(T_L - T_{ext}) \quad 8\text{-}26$$

where σ is either 0 or 1, $\vec{J}_{tot}^{u}$ is the total energy flux and $\vec{s}$ is the unit external normal of the boundary. The projection of the energy flux onto s is:

$$\left(\overrightarrow{J_{tot}^{u}} \cdot \vec{s}\right) = -\kappa\frac{\partial T_L}{\partial n} + (T_L P_n + \phi_n)\vec{J}_n \cdot \vec{s} + (T_L P_p + \phi_p)(\vec{J}_p \cdot \vec{s}) \quad 8\text{-}27$$

When $\sigma = 0$, Equation 8-26 specifies a Dirichlet (fixed temperature) boundary condition:

$$T_L = \texttt{TEMPER} \quad 8\text{-}28$$

where you define TEMPER in the **THERMCONTACT** statement as shown in the next section. You can specify dirichlet boundary conditions for an external boundary (which may coincide with an electrode) or for an electrode inside the device. When σ = 1, Equation 8-26 takes the form:

$$\left(\overrightarrow{J_{tot}^{u}} \cdot \vec{s}\right) = \frac{1}{R_{th}}(T_L - \texttt{TEMPER}) \quad 8\text{-}29$$

where the thermal resistance, R_{th}, is given by:

$$R_{th} = \frac{1}{\texttt{ALPHA}} \quad 8\text{-}30$$

and ALPHA is user-definable on the **THERMCONTACT** statement.

Giga can also model thermal boundary conditions due to energy loss by blackbody radiation. If you specify the BLACKBODY flag on the **THERMCONTACT** statement, then the rate

$$\texttt{STEFAN}(T_L^4 - \texttt{TEMPER}^4) \quad 8\text{-}31$$

will be added to the heat flow given by Equation 8-29, where STEFAN is user-definable on the **THERMCONTACT** statement.

Specifying Thermal Boundary Conditions

Setting thermal boundary conditions is similar to setting electrical boundary conditions. The **THERMCONTACT** statement is used to specify the position of the thermal contact and any optional properties of the contact. You can place thermal contacts anywhere in the device (including sidewalls). Equation 8-29 is used if a value is specified for α. Otherwise, Equation 8-28 is used.

The following command specifies that thermal contact number 1 is located between x=0 μm and x=2 μm at y=0 μm, and that the temperature at the contact is 300K.

```
THERMCONTACT NUM=1 X.MIN=0 X.MAX=2 Y.MIN=0 Y.MAX=0 TEMP=300
```

You can use simpler statement if the coordinates of a thermal contact coincide with the coordinates of an electrical contact. In this case, it is permissible to specify the location of the

thermal contact by referring to the electrode number of the electrical contact. For example, the statement:

```
THERMCONTACT NUM=1 ELEC.NUM=3 TEMP=400
```

specifies that thermal contact number 1 is located in the same position as electrode number 3 and that the contact temperature is 400K.

Specifying the BOUNDARY parameter gives you flexibility in applying thermal boundary conditions. This parameter is set by default and means that the thermal boundary condition will only be set on the outside surface of the thermal contact, where it forms the exterior of the device. If the parameter is cleared by specifying ^BOUNDARY in the **THERMCONTACT** statement, the boundary conditions will be applied to the interior of the thermal contact and to the part of the surface of the thermal contact that forms an interface with the interior of the device. In a 2D model, you may have a thermal contact that is internal to the device and in this case you would need to specify ^BOUNDARY. Otherwise, the contact will be ignored.

For example:

```
THERMCONTACT ELEC.NUM=1 ^BOUNDARY TEMPER=450 ALPHA=2.5
```

where the ELECTRODE extends into the device and will set the interior points of the thermalcontact to 450K and will apply the flux boundary condition on all faces of the thermal contact that interface with the interior of the device.

Table 8-8 User Specifiable Parameters for Equations 8-28 and 8-29

Statement	Parameter	Units	Default
THERMCONTACT	ALPHA	$W/(cm^2K)$	∞
THERMCONTACT	BLACKBODY	Logical	False
THERMCONTACT	BOUNDARY	-	True
THERMCONTACT	STEFAN	$W/(cm^{-2}K^{-4})$	5.67051×10^{-12}
THERMCONTACT	TEMPER	K	300

Representing a thermal environment, in terms of thermal impedances, leads to efficient solutions. But, thermal impedance representations are typically only approximations. Detailed thermal modeling (e.g., the effect of heat sink design changes) typically requires using detailed modeling of thermal regions with specified external ambient temperatures.

Note: You can't alter the value of a thermal resistor within a sequence of **SOLVE** statements. Rerun the input file whenever a thermal resistor is changed.

At temperatures of around 1000 K, heat loss by radiation may become significant. To model blackbody radiation, specify the flag BLACKBODY on the relevant **THERMCONTACT** statement. This adds blackbody emission proportional to the fourth power of lattice temperature at the contact and blackbody absorption proportional to the fourth power of the specified environmental temperature, set with the TEMPER parameter.

You can set the constant of proportionality using the STEFAN parameter, which has the default value for an ideal blackbody of 5.67051×10^{-12} $W/(cm^{-2}K^{-4})$.

For a non-ideal emitter, you set this parameter to the appropriate value. The blackbody emission is added to the thermal resistance boundary conditions. To model only radiation energy losses, you must set ALPHA to be zero to eliminate conductive heat losses. In practice, both types of heat loss may be present.

8.2.8 Temperature Dependent Material Parameters

Giga automatically uses the built-in temperature dependence of the physical models that are specified. When lattice heating is specified, temperature dependent values are evaluated locally at each point in the device. When lattice heating is not solved, the models only provide global temperature dependence. In other words, all points in the device are assumed to be at the specified temperature.

The non-isothermal energy balance model uses the same carrier temperature dependencies of the mobility and impact ionization models as in the pure energy balance case, but with coefficients that depend on the local lattice temperature. Impact ionization coefficients depend on lattice temperature. Almost all other models and coefficients depend on lattice temperature.

When lattice heating is used, there is no point in specifying models that do not include temperature dependence. For mobilities, don't specify CONMOB. Instead, specify ANALYTIC or ARORA. For impact ionization coefficients, specify the SELBERHERR model.

Giga can account for the temperature dependence of the minority carrier lifetimes for electrons or holes or both. The LT.TAUN (electrons) and LT.TAUP (holes) parameters of the **MATERIAL** statement are used to select this model. This model is turned on whenever the value of LT.TAUN or LT.TAUP is greater than 0, which is the default.

The temperature dependence of electron and hole lifetimes in the SRH recombination model have the forms:

$$\tau_n = \mathrm{TAUN0}\left(\frac{T_L}{300}\right)^{\mathrm{LT.TAUN}} \quad \text{8-32}$$

$$\tau_p = \mathrm{TAUP0}\left(\frac{T_L}{300}\right)^{\mathrm{LT.TAUP}} \quad \text{8-33}$$

Table 8-9 User-Specifiable Parameters for Equations 8-32 and 8-33

Statement	Parameter	Default	Units
MATERIAL	TAUN0	1×10^{-7}	s
MATERIAL	TAUP0	1×10^{-7}	s
MATERIAL	LT.TAUN	0	
MATERIAL	LT.TAUP	0	

See Equations 3-352 and 3-353 for information regarding concentration dependent lifetimes.

There is an alternative model for the temperature dependence of the recombination lifetimes. This is

$$\tau_n(T_L) = \texttt{TAUN0}\exp(\texttt{TCOEFF.N}(T_L/300 - 1.0)) \quad 8\text{-}34$$

$$\tau_p(T_L) = \texttt{TAUP0}\exp(\texttt{TCOEFF.P}(T_L/300 - 1.0)) \quad 8\text{-}35$$

To enable this model, use SRH.EXPTEMP on the **MODELS** statement along with SRH or CONSRH. You can set the parameters TCOEFF.N and TCOEFF.P on the **MATERIAL** statement (see Table 8-10).

Table 8-10 Material Parameters for Exponential Temperature Dependence Model

Statement	Parameter	Type	Default
MATERIAL	TCOEFF.N	Real	2.55
MATERIAL	TCOEFF.P	Real	2.55

8.2.9 Phase Change Materials (PCMs)

Phase Change Materials, PCMs, are materials that display a change in resistivity with temperature that can be associated with a phase change in the material between crystalline and amorphous states. Typically, the material exhibits a low resistivity in the crystalline state and a relatively higher resistivity in the amorphous state. Once a state (crystalline/amorphous or low/high resistivity) is obtained, it can be "frozen" into the material by rapidly lowering the temperature. This characteristic enables PCM materials to exhibit hysterisis under such conditions and makes them suitable to store information. Although this model was designed for PCM materials, it is also applicable to other materials exhibiting similar characteristics.

To enable the PCM model, specify PCM in the **MODELS** statement. When you enable this model, the resistivity as a function of lattice temperature will be described (see Figure 8-1).

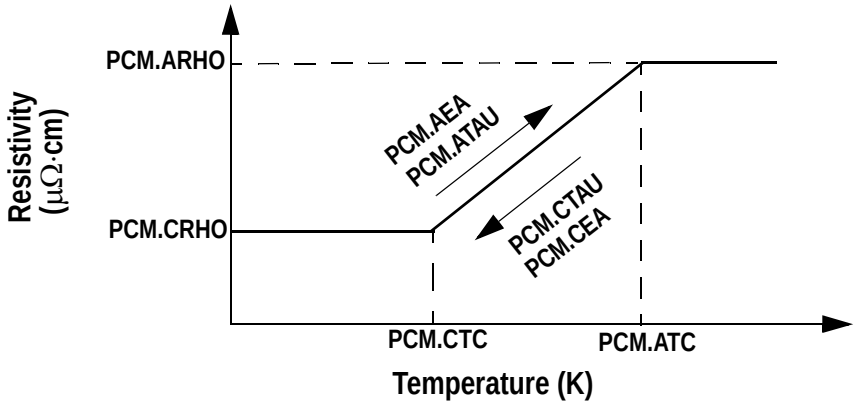


Figure 8-1: PCM Temperature Dependent Resistivity Model

Here at low temperatures, less than or equal to PCM.CTC. The resistivity is constant and equal to the value specified by PCM.CRHO. At high temperatures above PCM.ATC, the resistivity is

constant and equal to the value of PCM.ARHO. Between the temperatures PCM.CTC and PCM.ATC, the resistivity varies either linearly or logarithmically with temperature depending on the value of the PCM.LOG parameter of the **MODELS** statement.

The resistivity as a function of temperature is as given in Equations 8-36 through 8-38 for the case where PCM.LOG is not true.

$$\rho = \text{PCM.CRHO} \qquad (T < \text{PCM.CTC}) \tag{8-36}$$

$$\rho = \text{PCM.ARHO} \qquad (T > \text{PCM.ATC}) \tag{8-37}$$

$$\rho = \frac{(T - \text{PCM.CTC})(\text{PCM.ARHO} - \text{PCM.CRHO})}{(\text{PCM.ATC} - \text{PCM.CTC})} + \text{PCM.CRHO} \qquad (\text{PCM.CTC} < T < \text{PCM.ATC}) \tag{8-38}$$

For the case where PCM.LOG is true, the resistivity between the critical temperatures is given by Equations 8-39 and 8-40.

$$\rho = 10^{x} \tag{8-39}$$

$$x = \left(\frac{T - \text{PCM.CTC}}{\text{PCM.ATC} - \text{PCM.CTC}}\right)(\log_{10}(\text{PCM.ARHO}) - \log_{10}(\text{PCM.CRHO})) + \log_{10}(\text{PCM.CRHO}) \qquad (\text{PCM.CTC} < T < \text{PCM.ATC}) \tag{8-40}$$

Hysterisis is built into the model in two ways. In static simulations, the resistivity is updated according to Figure 8-1 only for increasing temperature. This is suitable since phase changes in memory devices are generally made during increasing temperature ramps.

For time domain simulations, the hysterisis is described using the Johnson-Mehl-Avrami model as given in Equations 8-41 and 8-42. Here, t is the time at the last time step, dt is the size of the time step, T is the lattice temperature, $\hat{\rho}(T)$ is the temperature dependent static resistivity given by Equations 8-36 through 8-40, $\rho(t,T)$ is the resistivity at the last time step, and $\rho(t+d+T)$ is the resistivity at the new time step.

Equation 8-41 applies to increasing temperature. Equation 8-42 applies to decreasing temperature.

$$\rho(t + \Delta t, T) = \rho(t, T) + [\hat{\rho}(T) - \rho(t, T)] \cdot \left\{1 - \exp\left[(-dt/\text{PCM.ATAU})^{\text{PCM.AP}} \exp\left(\frac{-\text{PCM.AEA}}{kT}\right)\right]\right\} \tag{8-41}$$

$$\rho(t+\Delta t, T) = \rho(t, T) + [\hat{\rho}(T) - \rho(t, T)] \cdot \left\{1 - \exp\left[(-dt/\texttt{PCM.CTAU})^{\texttt{PCM.CP}} \exp\left(\frac{-\texttt{PCM.CEA}}{kT}\right)\right]\right\} \quad 8\text{-}42$$

Table 8-11 shows the parameters of Equations 8-36 through 8-42.

Table 8-11 Material Parameters for PCM Materials

Statement	Parameter	Type	Default	Units
MATERIAL	PCM.AEA	Real	3.2	eV
MATERIAL	PCM.AP	Real	1.0	
MATERIAL	PCM.ARHO	Real	10^8	μΩcm
MATERIAL	PCM.ATAU	Real	2×10^{-36}	s
MATERIAL	PCM.ATC	Real	620	K
MATERIAL	PCM.CEA	Real	3.7	eV
MATERIAL	PCM.CP	Real	1.0	
MATERIAL	PCM.CRHO	Real	10^8	μΩcm
MATERIAL	PCM.CTAU	Real	2×10^{-36}	s
MATERIAL	PCM.CTC	Real	600	K
MATERIAL	PCM.LATHEAT	Real	622	J

Latent heat can be also be introduced into the source term of the heatflow equation (Equation 8-1) by specifying PCM.LATHEAT on the **MATERIAL** statement. The latent heat is introduced as described in Equation 8-43 for increasing temperatures and Equation 8-44 for decreasing temperatures where $H(t+dt,T)$ is the the latent heat at the current time step and $H(t, T)$ is the latent heat at the previous time step. $\hat{H}(T)$ is the temperature dependent static latent heat given by Equation 8-45 for increasing temperature and Equation 8-47 for decreasing temperature.

Note: The time integration performed in Equations 8-41 and 8-42 above and Equations 8-43 and 8-44 below critically depends on the choice of the parameters PCM.AEA, PCM.CEA, PCM.ATAU, and PCM.CTAU relative to the operating temperatures and time steps. In some cases, you may find that the simulator produces no discernable changes in resistivity. In this case, you should carefully reconsider the specifications of these parameters.

$$H(t+\Delta t, T) = H(t, T) + [\hat{H}(T) - H(t, T)] \cdot \left\{1 - \exp\left[(-dt/\mathtt{PCM.ATAU})^{\mathtt{PCM.AP}} \exp\left(\frac{-\mathtt{PCM.AEA}}{kT}\right)\right]\right\} \quad 8\text{-}43$$

$$H(t+\Delta t, T) = H(t, T) + [\hat{H}(T) - H(t, T)] \cdot \left\{1 - \exp\left[(-dt/\mathtt{PCM.CTAU})^{\mathtt{PCM.CP}} \exp\left(\frac{-\mathtt{PCM.CEA}}{kT}\right)\right]\right\} \quad 8\text{-}44$$

$$\hat{H}(T) = \left(\frac{T - \mathtt{PCM.CTC}}{\mathtt{PCM.ATC} - \mathtt{PCM.CTC}}\right)\mathtt{PCM.LATHEAT} \qquad (\mathtt{PCM.CTC} < T < \mathtt{PCM.ATC}) \quad 8\text{-}45$$

$$\hat{H}(T) = -\left(\frac{\mathtt{PCM.ATC} - T}{\mathtt{PCM.ATC} - \mathtt{PCM.CTC}}\right)\mathtt{PCM.LATHEAT} \qquad (\mathtt{PCM.CTC} < T < \mathtt{PCM.ATC}) \quad 8\text{-}46$$

You can initialize the resistivity of a PCM material to a user-specified value by setting the RESISTIVITY parameter of the **MATERIAL** statement. If unspecified, the initial resistivity will be set to the value of PCM.CRHO.

To simplify the numerics of PCM simulation, set the PCM parameter of the **METHOD** statement. This parameter effectively sets the **METHOD** parameters as described in Table 8-12.

Table 8-12 PCM Method Settings

Parameter	Setting
MIN.TEMP	1e-9
MAX.TEMP	1e9
DVMAX	1e9
DIRECT	TRUE
CARRIERS	0
NEWTON	TRUE
TAUTO	FALSE

If you want to simulate a mixture of semiconductors and PCM materials, then set both PCM and CARRIERS=2 on the **METHOD** statement.

You can redefine the resistivity relations given by Equations 8-36 through 8-45 with a user-specified function using the C-interpreter. You can assign the F.PCM parameter of the **MATERIAL** statement to the file name of a valid C interpreter function expressing the variation of resistivity with temperature and history.

8.2.10 C-Interpreter Defined Scattering Law Exponents

KSN and KSP are exponents in the relationship between relaxation time and carrier energy. In the case for electrons:

$$\tau_n \sim (E - E_c)^{\mathrm{KSN}} \qquad 8\text{-}47$$

They are generally calculated from the dependence of mobility on carrier temperature, see Equations 3-145 and 3-146 in Section 3.4.1 "The Energy Balance Equations".

The value of KSN and KSP therefore depend on the scattering mechanisms, which are determining the relaxation time. Generally, KSN and KSP will themselves depend on Carrier Energy. The C-Interpreter functions F.KSN and F.KSP are available in Atlas to model this dependence. The Energy input is 1.5 k_B $T_{n,p}$ if HCTE is specified, or 1.5 k_B T_L if only LAT.TEMP is specified. To use these C-Interpreter functions, the syntax is

```
MODELS F.KSN=<filename> F.KSP=<filename>
```

where the <filename> parameter is an ASCII file containing the C-Interpreter function. See Appendix A "C-Interpreter Functions" for more information regarding the C-interpreter and its functions.

8.3 Applications of GIGA

8.3.1 Power Device Simulation Techniques

This section contains a series of techniques, which you may find useful when simulating typical power device structures. Not all of the features described below are specific to Giga and are common to the Atlas framework.

Floating Guard Rings

No special syntax is needed for the simulation of uncontacted doping areas used in floating guard rings. The program is able to simulate guard ring breakdown with the standard impact ionization models. In some extreme cases, convergence may be slow due to poor initial guesses. If convergence is slow, both GUMMEL and NEWTON should be used in the **METHOD** statement.

Floating Field Plates

You should use the **ELECTRODE** statement to specify the field plate regions as electrodes. If these plates do not contact any semiconductor, then you can set these electrodes to float in the same manner as EEPROM floating gates. The following statement line specifies that the field plate region PLATE1 is a floating field plate.

```
CONTACT NAME=PLATE1 FLOATING
```

If the plates do contact the semiconductor, then do not use this syntax. Instead, current boundary conditions are used at the electrode with zero current. See the "Floating Contacts" on page 70 for more information about floating electrodes.

External Inductors

Inductors are commonly used in the external circuits of power devices. You can use the **CONTACT** statement to set an inductor on any electrode. The following statement sets an inductance on the drain electrode of 3 μH/μm.

```
CONTACT NAME=DRAIN L=3E-3
```

The next statement is used to specify a non-ideal inductor with a resistance of 100 Ω/μm.

```
CONTACT NAME=DRAIN L=3E-3 R=100
```

8.3.2 More Information

Many examples using Giga have been provided on the distribution package. These include power device examples but also SOI and III-V technologies. You can find more information about the use of Giga by reading the text associated with each example.

Chapter 9
Laser: Edge Emitting Simulator

9.1 Overview

Laser performs coupled electrical and optical simulation of semiconductor lasers. Laser works with Blaze and allows you to

- Solve the Helmholtz equation to calculate the optical modes, their fields and intensity patterns.
- Calculate optical gain, depending on the photon energy and the quasi-Fermi levels.
- Calculate the carrier recombination due to light emission (i.e., stimulated emission).
- Solve Photon Rate equations to calculate modal photon densities.
- Calculate laser light output power, light-voltage and current voltage characteristics.

To perform a Laser simulation, you need to know Atlas and Blaze first. If you don't, read Chapters 2 "Getting Started with Atlas" and 6 "Blaze: Compound Material 2D Simulator".

9.2 Physical Models

To simulate semiconductor lasers, the basic semiconductor equations (see Equations 3-1 to 3-4) are solved self-consistently with an optical equation that determines the optical field intensity distribution. Laser uses the following coordinate system:

- The X axis is perpendicular to the laser cavity and goes along the surface (from left to right).
- The Y axis is perpendicular to the laser cavity and goes down from the surface.
- The Z axis goes along the laser cavity.

The X and Y axes are the same as in other Atlas products. The electrical and optical equations are solved in the X and Y plane (i.e., perpendicular to the laser cavity).

9.2.1 2D Vector Helmholtz Equation

In general, components of electromagnetic field are coupled together into a system of equations, which is represented by vector Helmholtz equation:

$$\nabla \times \frac{1}{\varepsilon} \nabla \times \vec{H} = \frac{\omega^2}{c^2} \vec{H} \tag{9-1}$$

In Cartesian coordinates, assuming that laser waveguide is homogeneous in z-direction, dielectric permittivity tensor is diagonal and material in non-magnetic, this equation can be written as follows:

$$\begin{cases} \dfrac{\partial^2 H_x}{\partial x^2} + \varepsilon_{yy} \dfrac{\partial}{\partial y}\left(\dfrac{1}{\varepsilon_{zz}} \dfrac{\partial H_x}{\partial y}\right) + \left(1 - \dfrac{\varepsilon_{yy}}{\varepsilon_{zz}}\right) \dfrac{\partial^2 H_y}{\partial x \partial y} - \varepsilon_{yy} \dfrac{\partial}{\partial y}\left(\dfrac{1}{\varepsilon_{zz}}\right) \dfrac{\partial H_y}{\partial x} + \dfrac{\omega^2}{c^2} \varepsilon_{yy} H_x = \beta^2 H_x \\ \varepsilon_{xx} \dfrac{\partial}{\partial x}\left(\dfrac{1}{\varepsilon_{zz}} \dfrac{\partial H_y}{\partial x}\right) + \dfrac{\partial^2 H_y}{\partial y^2} + \left(1 - \dfrac{\varepsilon_{xx}}{\varepsilon_{zz}}\right) \dfrac{\partial^2 H_x}{\partial x \partial y} - \varepsilon_{xx} \dfrac{\partial}{\partial x}\left(\dfrac{1}{\varepsilon_{zz}}\right) \dfrac{\partial H_x}{\partial y} + \dfrac{\omega^2}{c^2} \varepsilon_{xx} H_y = \beta^2 H_y \end{cases} \tag{9-2}$$

This is a non-Hermitian complex eigenvalue problem for the square of propagation constant β and eigen vector $\xi=(H_x, H_y)$. The system is discretized with finite difference method and solved with a sparse iterative eigenvalue solver, using frequency ω as a parameter. To launch vector Helmholtz solver for laser simulation, set `V.HELM` parameter on the **LASER** or **MODELS** statements. The total number of transverse modes to be computed is set by the `NMODES` parameter.

After the eigenvalue problem is solved, all other components of optical field can be found:

$$H_z = \frac{i}{\beta}\left(\frac{\partial H_x}{\partial x} + \frac{\partial H_y}{\partial y}\right) \tag{9-3}$$

$$E_x = \frac{i}{\omega \varepsilon_{xx} \varepsilon_0}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \tag{9-4}$$

$$E_y = -\frac{i}{\omega\varepsilon_{yy}\varepsilon_0}\left(\frac{\partial H_z}{\partial x} - \frac{\partial H_x}{\partial z}\right) \quad 9\text{-}5$$

$$E_z = \frac{i}{\beta\varepsilon_{zz}}\left(\varepsilon_{xx}\frac{\partial E_x}{\partial x} + \varepsilon_{yy}\frac{\partial E_y}{\partial y}\right) \quad 9\text{-}6$$

The intensity pattern for each transverse optical mode is then computed using

$$I_m(x,y) = \left(\left[\overrightarrow{E_m} \times \overrightarrow{H_m}^*\right] + \left[\overrightarrow{E_m}^* \times \overrightarrow{H_m}\right]\right) \cdot \hat{z} \quad 9\text{-}7$$

and normalized by area:

$$\int I_m(x,y)\,dS = 1 \quad 9\text{-}8$$

Effective refractive index and effective absorption for each mode are defined as

$$n_{eff,\,m} = \Re e[\beta_m] \cdot \frac{c}{\omega}. \quad 9\text{-}9$$

$$\alpha_{eff,\,m} = 2\Im m[\beta_m] \quad 9\text{-}10$$

9.2.2 2D Scalar Helmholtz Equation

It is often the case that for each mode, one of the components is much larger than the other. This allows you to neglect the coupling between H_x and H_y components in Equation 9-2. We therefore obtain a system of two independent scalar equations for each component:

$$\begin{cases} \dfrac{\partial^2 H_x}{\partial x^2} + \varepsilon_{yy}\dfrac{\partial}{\partial y}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial H_x}{\partial y}\right) + \dfrac{\omega^2}{c^2}\varepsilon_{yy}H_x = \beta^2 H_x & \text{(TM)} \\ \varepsilon_{xx}\dfrac{\partial}{\partial x}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial H_y}{\partial x}\right) + \dfrac{\partial^2 H_y}{\partial y^2} + \dfrac{\omega^2}{c^2}\varepsilon_{xx}H_y = \beta^2 H_y & \text{(TE)} \end{cases} \quad 9\text{-}11$$

To use this approach, set S.HELM parameter on the **MODELS** statement instead of V.HELM. In addition, this method allows you to solve for only TE or only TM polarization. Thus, removing unnecessary modes. This is achieved by switching off HELM.TE or HELM.TM parameters. By default both parameters are true.

9.2.3 1D Scalar Helmholtz Equation

More simplification to Equation 9-2 may occur when dielectric permittivity is one-dimensional: $\varepsilon(x,y)=\varepsilon(x)$ or $\varepsilon(x,y)=\varepsilon(y)$. If we assume that the field components are also one-dimensional: $H_{x,y}(x,y)=H_{x,y}(x)$ or $H_{x,y}(x,y)=H_{x,y}(y)$ (i.e., we are dealing with infinite slabs). In this case, equation for each component will also become one dimensional. A 1D scalar solver can be launched by changing parameter HELM.GEOM on **LASER** or **WAVEGUIDE** statements, which has a default value HELM.GEOM=2DXY.

In the case of $\varepsilon(x,y)=\varepsilon(x)$, set the HELM.GEOM=1DX parameters and the following equations will be solved:

$$\begin{cases} \dfrac{\partial^2 H_x}{\partial x^2} + \dfrac{\omega^2}{c^2}\varepsilon_{yy}H_x = \beta^2 H_x & \text{(TM)} \\ \varepsilon_{xx}\dfrac{\partial}{\partial x}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial H_y}{\partial x}\right) + \dfrac{\omega^2}{c^2}\varepsilon_{xx}H_y = \beta^2 H_y & \text{(TE)} \end{cases} \quad 9\text{-}12$$

In the case of $\varepsilon(x,y)=\varepsilon(y)$, set the HELM.GEOM=1DY parameter and the following equations will be solved:

$$\begin{cases} \varepsilon_{yy}\dfrac{\partial}{\partial y}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial H_x}{\partial y}\right) + \dfrac{\omega^2}{c^2}\varepsilon_{yy}H_x = \beta^2 H_x & \text{(TM)} \\ \dfrac{\partial^2 H_y}{\partial y^2} + \dfrac{\omega^2}{c^2}\varepsilon_{xx}H_y = \beta^2 H_y & \text{(TE)} \end{cases} \quad 9\text{-}13$$

To solve the 1D non-Hermitian problems, Laser uses LR algorithm, which scales as $O(N^2)$, enabled by the parameter HEML1D=LR on the **METHOD** statement and is a default. An alternative is more stable but a more time consuming QR algorithm, which scales as $O(N^3)$ and is set by HEML1D=QR.

9.2.4 1.5D Effective Index Solver

Effective index solver attempts to solve a 2D scalar Helmholtz equation by combining 1D solutions in X and Y directions. It is suitable for quasi-2D structures, such as slab waveguides and stripe lasers, where the variation of dielectric constant along one of the directions is slow. Specifically, in the case of $\varepsilon(x,y) \approx \varepsilon(y)$, using a variable separation ansatz $H_x(x,y)=X(x)Y(y;x)$, Equation 9-11 for TM modes can be written as:

$$\begin{cases} \dfrac{\partial^2 Y}{\partial y^2} + \dfrac{\omega^2}{c^2}\varepsilon_{xx}Y = \left(\dfrac{\omega}{c}n_{eff}(x)\right)^2 Y \\ \varepsilon_{xx}\dfrac{\partial}{\partial x}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial X}{\partial x}\right) + \left(\dfrac{\omega}{c}n_{eff}(x)\right)^2 X = \beta^2 X \end{cases} \quad \text{(TM)} \quad 9\text{-}14$$

The first 1D eigen problem is solved at each X-slice in Y direction to find effective index $n_{eff}(x)$, while the second equation is solved once in the middle of the device to find propagation constant β. Similarly, for TE modes $H_y(x,y)=X(x)Y(y;x)$ and the equations read:

$$\begin{cases} \varepsilon_{yy}\dfrac{\partial}{\partial y}\left(\dfrac{1}{\varepsilon_{zz}}\dfrac{\partial Y}{\partial y}\right)+\dfrac{\omega^2}{c^2}\varepsilon_{yy}Y = \left(\dfrac{\omega}{c}n_{eff}(x)\right)^2 Y \\ \dfrac{\partial^2 X}{\partial x^2}+\left(\dfrac{\omega}{c}n_{eff}(x)\right)^2 X = \beta^2 X \end{cases} \quad \text{(TE)} \qquad 9\text{-}15$$

To launch this method, set HELM.GEOM=15DY. In the case of $\varepsilon(x,y)\approx\varepsilon(x)$, set HELM.GEOM=15DX. For this solver, in addition to the total number of modes, you may also choose the number of modes to be taken in Y direction in the first equation, by setting EFFMODES parameter on **LASER** or **WAVEGUIDE** statements. By default, only fundamental mode is taken in Y direction.

9.2.5 2D Cylindrical Helmholtz Solver

When V.HELM parameter is switched on and CYLINDRICAL parameter is present on **MESH** statement, a 2D cylindrical Helmholtz solver will be launched. In cylindrical coordinates, assuming that Laser is cylindrically symmetric, dielectric permittivity is isotropic, and material is non-magnetic. The vector Helmholtz equation can be written for H_r and H_z components as follows. (Equations for the case of $m=0$ or anisotropic dielectric constant look somewhat different but are not shown here for the sake of brevity.)

$$\begin{cases} -\dfrac{1}{\varepsilon}\cdot\dfrac{\partial^2 H_r}{\partial r^2}-\dfrac{1}{\varepsilon}\cdot\dfrac{\partial^2 H_r}{\partial z^2}+\left(-\dfrac{3}{\varepsilon r}+\dfrac{\partial}{\partial z}\left(\dfrac{1}{\varepsilon}\right)\right)\cdot\dfrac{\partial H_r}{\partial r}+\dfrac{(m^2-1)}{\varepsilon r^2}H_r+\dfrac{\partial}{\partial z}\left(\dfrac{1}{\varepsilon}\right)\cdot\left(\dfrac{\partial H_z}{\partial r}-\dfrac{\partial H_r}{\partial z}\right)-\dfrac{2}{\varepsilon r}\cdot\dfrac{\partial H_z}{\partial z} = \dfrac{\omega^2}{c^2}H_r \\ -\dfrac{1}{\varepsilon}\cdot\dfrac{\partial^2 H_z}{\partial r^2}-\dfrac{1}{\varepsilon}\cdot\dfrac{\partial^2 H_z}{\partial z^2}-\left(\dfrac{1}{\varepsilon r}+\dfrac{\partial}{\partial r}\left(\dfrac{1}{\varepsilon}\right)\right)\cdot\dfrac{\partial H_z}{\partial r}+\dfrac{m^2}{\varepsilon r^2}H_z+\dfrac{\partial}{\partial z}\left(\dfrac{1}{\varepsilon}\right)\cdot\dfrac{\partial H_r}{\partial z} = \dfrac{\omega^2}{c^2}H_z \end{cases} \qquad 9\text{-}16$$

Here, m is an orbital (azimuthal) number, which can be set by ORBIT.NUM on the **LASER** or **WAVEGUIDE** statement. This is a non-Hermitian complex eigenvalue problem for the square of eigen frequency ω and eigen vector $\xi=(H_r, H_z)$. The intensity of light is directed perpendicular to the R-Z plane.

9.2.6 Dielectric Permittivity

For anisotropic relative dielectric permittivity, Laser uses the following model [146] (i=x,y,z)

$$\varepsilon_i(x,y) = n_i^2(x,y) = (n_{MAT,i}(x,y) + \Delta n_{GAIN,i}(x,y) + \Delta n_{ABS}(x,y) + \Delta n_{FC}(x,y) + \Delta n_{TEMP}(x,y))^2 \quad 9\text{-}17$$

where:

- $n_{MAT,i}(x,y) = \sqrt{\varepsilon_{MAT,i}(x,y)}$ is the material high frequency complex refractive index. You can specify the material refractive index or corresponding material dielectric permittivity using the parameters on the **MATERIAL** statement shown in Table 9-1.
- change of complex refractive index due to gain is given by the following expression, where $j = \sqrt{-1}$, $g_i(x,y)$ is the local optical gain in i=x,y,z direction and the ALPHAR parameter on the **MATERIAL** statement represents a Kramers-Kronig relation between changes of real and imaginary parts:

$$\Delta n_{GAIN,i}(x,y) = -(\mathrm{ALPHAR} + j g_i(x,y)) \cdot \frac{c}{2\omega} \quad 9\text{-}18$$

- change of complex refractive index due to bulk absorption outside of active region is given by the ALPHAA parameter specified in the **MATERIAL** statement. Additionally or instead of this, you can specify the ABS.BULK parameter on the **MODELS** statement to compute bulk absorption coefficient at each node using multiband k.p band-structure
- change of complex refractive index due to free-carrier absorption is turned on by ABS.FCARRIER parameter on the **MODELS** statement and is given by the following expressions, where FC.RN, RC.RP, FC.EXPRN, FC.EXPRP, FC.AN, FC.AP, FC.EXPAN, and FC.EXPAP are parameters on the **MATERIAL** statement:

$$\Re e\{\Delta n_{FC}(x,y)\} = -(\mathrm{FC.RN} \cdot n^{\mathrm{FC.EXPRN}} + \mathrm{FC.RP} \cdot n^{\mathrm{FC.EXPRP}}) \quad 9\text{-}19$$

$$\Im m\{\Delta n_{FC}(x,y)\} = (\mathrm{FC.AN} \cdot n^{\mathrm{FC.EXPAN}} + \mathrm{FC.AP} \cdot n^{\mathrm{FC.EXPAP}}) \cdot \frac{c}{2\omega} \quad 9\text{-}20$$

- change of complex refractive index due to change of lattice temperature is given by the following expression, where DINDEXDT is the parameter on the **MATERIAL** statement and $T(x,y)$ is the local temperature:

$$\Delta n_{TEMP}(x,y) = \mathrm{DINDEXDT} \cdot (T(x,y) - T_0) \quad 9\text{-}21$$

Table 9-1 Rule Hierarchy for Determination of High Frequency Material Permittivity
EPS.XX, EPS.YY, and EPS.ZZ specify the real parts of XX, YY and ZZ components of dielectric permittivity constant. EPSIM.XX, EPSIM.YY, and EPSIM.ZZ specify the imaginary parts.
EPS.ISO and EPSIM.ISO specify real and imaginary parts of dielectric constant in isotropic case or for unspecified tensor components in anisotropic case. EPSINF can be used instead of EPS.ISO.
REAL.INDEX and IMAG.INDEX specify real and imaginary parts of refractive index, which may be more convenient than dielectric permittivity constant.
If F.INDEX is defined in the **MATERIAL** statement, then the bulk permittivity is calculated from the square of the index returned from the C-Interpreter function defined in the file specified by F.INDEX.
If INDEX.FILE is defined in the **MATERIAL** statement, then the bulk permittivity is calculated as the square of the index interpolated from the table specified in the file pointed to by INDEX.FILE.
If the material is listed in Table B-31, the permittivity is calculated as the square of index interpolated from the built-in tables for that material.

After complex refractive index is calculated, local absorption can be found from the following equation:

$$\alpha(x,y) = \frac{2\omega}{c}\Im m[n(x,y)] \qquad 9\text{-}22$$

9.2.7 Local Optical Gain and Spontaneous Emission

Laser has four gain/spontaneous emission models set by the GAINMOD parameter. GAINMOD can take the value of 1 for Standard model (Equations 3-618 and 3-621), 2 for Empirical model (Equations 3-617 and 3-623), 3 for Takayama model (Equations 3-617 and 3-624) and 5 for multiband k.p model.

Multiband k.p model #5 can be used for bulk regions (Equations 3-634 and 3-635) and for quantum wells (Equations 14-25 and 14-26). For multiband k.p model parameter ZB.ONE, ZB.TWO, ZB.THREE, or WZ.THREE should also specified on the **MODELS** statement. There is already a ABS.BULK parameter on **MODELS** statement, which computes absorption, using exactly the same mechanism. ABS.BULK is disabled in regions, where gain model #5 is used.

A user defined gain can also be used by specifying the name of a C-Interpreter function F.LGAIN on the **Laser** statement.

The C-interpreter file must define a function lgain() with the following protype:

```
int lgain(double hbar_omega, double omega,
     double electron_concentration, double hole_concentration,
     double electron_quasi_fermi_level, double hole_quasi_fermi_level,
     double electron_current_density, double hole_current_density,
     double x_coordinate, double y_coordinate, double z_coordinate,
     int region_ID_number,
```

```
 double x_composition, double y_composition,
  double* gain_to_be_calculated,
  double* gain_derivative_wrt_electron_concentration_to_be_calculated,
  double* gain_derivative_wrt_hole_concentration_to_be_calculated,
  double* gain_derivative_wrt_electron_quasi_fermi_level_to_be_calculated,
  double* gain_derivative_wrt_hole_quasi_fermi_level_to_be_calculated,
  double* unused1, double* unused2)
{
  return 0; /* 0 = success */
}
```

where `omega` is the angular frequency in units of 2π Hertz, `hbar` is Planck's reduced constant $\hbar$ in units of eV-s (6.582173483e-16), electron and hole con-centrations are both in units of cm^{-3}, quasi-Fermi levels are in electron volts (eV), and x and y heterostructure material compositions are values between 0 and 1. Note that derivatives of gain with respect to quasi-Fermi levels will only be used if the values provided for the derivatives of gain with respect to electron and hole concentrations are both zero. The last two input arguments are unused.

9.2.8 Stimulated Emission

Carrier recombination due to stimulated light emission is modeled as follows:

$$R_{st}(x,y) = \sum_{j=x,y,z} \sum_{m} \frac{c}{n_{eff,m}} g_j(x,y) \Gamma^j_m(x,y) \cdot S_m \qquad 9\text{-}23$$

where

- $R_{st}(x,y)$ is the recombination rate due to stimulated light emission.
- The first summation is performed over all directions.
- The second summation is performed over all laser modes.
- $n_{eff,m}$ is the effective refractive index.
- S_m is the modal photon density.
- $g_j(x,y)$ is local gain for electric field polarized in j direction
- $\Gamma^j{}_m(x,y)$ is given by

$$\Gamma^j{}_m(x,y) = 2 \cdot \Re e[n_j(x,y)] \cdot \frac{\left|E^j{}_m(x,y)\right|^2}{\int I_m(x,y)dS} \qquad 9\text{-}24$$

Gain confinement factor can be defined as the integral of $\Gamma^j{}_m(x,y)$ over active region [200, 332]. Laser can calculate gain confinement factor either by integrating over all quantum well regions:

$$\gamma^j{}_m = \int_{QWELL} \Gamma^j_m(x,y)dS \qquad 9\text{-}25$$

or integrating over the area with positive gain:

$$\gamma^j_m = \int_{gain>0} \Gamma^j_m(x,y)dS \quad (9\text{-}26)$$

Total confinement factor can be defined as sum of confinement factors over all directions:

$$\Upsilon_m = \sum_{j=x,y,z} \varphi^j_m \quad (9\text{-}27)$$

9.2.9 Photon Rate Equations

Optical gain provides a link between optical and electrical models. The optical gain depends on the quasi-Fermi levels and in turn impacts dielectric permittivity (see Equation 9-11), and by the coupling between the stimulated carrier recombination rate (R_{st}) and the density of photons (S) as described by Equation 9-23.

To determine S_m, Laser solves the system of photon rate equations:

$$\frac{dS_m}{dt} = \left(\frac{c}{n_{eff}}G_m - \frac{1}{\tau_{ph,m}}\right)S_m + R_{sp,m} \quad (9\text{-}28)$$

where the modal gain G_m is given by:

$$G_m = \sum_{j=x,y,z} \iint g_j(x,y,\omega_m)\cdot\Gamma^j_m(x,y)\cdot dS \quad (9\text{-}29)$$

and the modal spontaneous emission rate $R_{sp,m}$ is given by:

$$R_{sp,m} = \beta\cdot\sum_{j=x,y,z} \iint r^j_{sp}(x,y,\omega_m)\cdot\Gamma^j_m(x,y)\cdot\Delta E_m\cdot dS \quad (9\text{-}30)$$

where β is the spontaneous emission coupling factor, defined by parameter EMISSION.FACTOR on **MATERIAL** statement (default is 1), $r^j_{sp}(x,y,\omega_m)$ is spontaneous emission rate per unit energy per unit volume, $\Delta E_m = \hbar/\tau_{ph,m}$ is the line width of the laser mode and $\tau_{ph,m}$ is the modal photon lifetime, which represents laser losses and is given by [228, 281].

$$\frac{1}{\tau_{ph,m}} = \frac{c}{n_{eff}}(\alpha_{mir} + \alpha_{abs} + \mathtt{LOSSES} + \mathtt{EXTRACTION}) \quad (9\text{-}31)$$

Here, α_{abs} is the absorption loss. To include absorption loss into photon rate equation, specify the ABSORPTION parameter on the **LASER** statement. It is defined as:

$$\alpha_{abs} = \sum_{j=x,y,z} \iint \alpha_j(x,y)\cdot\Gamma^j_m(x,y)dS \quad (9\text{-}32)$$

where total absorption $\alpha_j(x,y)$ is given by Equation 9-22.

The parameter LOSSES is defined on **LASER** statement and corresponds to internal laser losses without regards to their origin. The parameter EXTRACTION is defined on **LASER** statement

and is used to phenomenologically account for extraction of light from the cavity if mirror losses are zero (as in the case of cylindrical cavity).

The mirror loss, α_{mir}, can be defined in two ways. One way is to define the percentage reflectivity of both mirrors. The other way is to define the individual facet mirror reflectivities. If the former is chosen, then set the MIRROR.LOSS parameter on the **LASER** statement to the percentage reflectivity for the facet cavity mirrors. This assumes that both front and back mirrors are identical and gives a mirror loss calculated by:

$$\alpha_{mir} = \frac{1}{2\ \mathtt{CAVITY.LENGTH}} ln\left(\frac{1}{\mathtt{MIRROR.LOSS}^2}\right) \quad 9\text{-}33$$

where CAVITY.LENGTH is set to the length of the laser cavity on the **LASER** statement.

If you choose the latter model for α_{mir}, set the RR and RF parameters on the **LASER** statement instead. These parameters are the rear and front facet reflectivities respectively. The mirror loss is then calculated by:

$$\alpha_{mir} = \frac{1}{2 \cdot \mathtt{CAVITY.LENGTH}} ln \frac{1}{\mathtt{RF} \cdot \mathtt{RR}} \quad 9\text{-}34$$

The user-specified parameters for the loss models in Equation 9-31 are given in Tables 9-2 and 9-3.

Table 9-2 User Specifiable Parameters for Equation 9-28

Statement	Parameter	Default	Units
LASER	LOSSES	0	cm^{-1}
LASER	EXTRACTION	0	cm^{-1}

Table 9-3 User-Specifiable Parameters for LASER Loss Models

Statement	Parameter	Default	Units
LASER	MIRROR.LOSS	90.0	%
LASER	CAVITY.LENGTH	100.0	μm

9.2.10 Optical Power

The optical modal power emitted from the front mirror is calculated by [313]

$$P_{f,m} = \frac{h\omega S_m c}{2n_{eff}} \cdot \frac{ln(1/(R_f R_r))}{1 + \sqrt{R_f/R_r}(1-R_r)/(1-R_f)} \quad 9\text{-}35$$

where S_m is the photon density of mode m, ω is the frequency of emitted light, R_f and R_r are the front and rear mirror reflectivities. When light is extracted through a coupling to a nearby waveguide, as in the case of cylindrical cavities, you can set mirror reflectivities to 100%, and use the parameter EXTRACTION instead. In this case, the optical power emitted is given by

$$P_{f,m} = \frac{h\omega S_m c}{2n_{eff}} \cdot \texttt{EXTRACTION} \quad (9\text{-}36)$$

The total optical intensity emitted from the front mirror is given by

$$I(x,y) = \sum_m P_{fm} I_m(x,y) \quad (9\text{-}37)$$

9.2.11 Gain Saturation

To simulate non-linear gain saturation in Laser, use the simple model described by [37]

$$g'(x,y) = \frac{g(x,y)}{1 + I(x,y)/\texttt{GAIN.SAT}} \quad (9\text{-}38)$$

where GAIN.SAT is a user-specifiable parameter on the **MATERIAL** statement, $g'(x,y)$ is the local gain. To enable this model, specify GAIN.SAT on the **LASER** statement. Non-linear, absorption loss is similarly modeled using the following expression.

$$\alpha'(x,y) = \frac{\alpha(x,y)}{1 + I(x,y)/\texttt{ABSORPTION.SAT}} \quad (9\text{-}39)$$

where $\alpha'(x,y)$ is the local absorption and ABSORPTION.SAT is a user-definable parameter on the **MATERIAL** statement. To enable this model, specify ABSORPTION.SAT on the **LASER** statement.

9.3 Edge Emitting Laser with Non-uniform Cavity and External Air Gap

9.3.1 Laser Cavity Model

With the exception of the cylindrical model, the physical models described above solve for the transverse modes of the cavity. Therefore, they implicitly assume that the laser cavity is uniform in the direction of propagation, and this direction is perpendicular to the plane over which the 2D Cartesian mesh is defined. The field in the propagation direction, z, is then of the form $e^{j\beta z}$. This section describes a model to simulate rectangular laser cavities that are non-uniform in the propagation direction and one of the directions transverse to it. The model is developed to facilitate lasers with Distributed Bragg Reflectors or Distributed Feedback structures. You can activate this model by specifying the parameter HYBRID on the **LASER** statement.

In this model, the propagation direction is assumed to be *y*. The laser cavity is assumed uniform in the *z* direction with PEC boundary conditions leading to the electric field profile given by $\boldsymbol{E}(\boldsymbol{\rho})\sin(k_z z)$, where $\boldsymbol{E}(\boldsymbol{\rho})$ is the three-dimensional field vector as a function of the position vector $\boldsymbol{\rho} = (x, y)$ in xy plane. $k_z = \frac{\pi}{\text{CAVITY.DEPTH}}$ is the wavenumber for the standing wave in the z-direction. Normally, the CAVITY.DEPTH parameter is set to a large value ($k_z \to 0$) to disregard the effects of optical confinement in the *z* direction.

Atlas uses the effective medium method, similar to the **VCSEL** model (see Chapter 10 "VCSEL Simulator"), in which the vector field $\boldsymbol{E}$ satisfies the following two-dimensional Helmholtz equation,

$$\nabla^2 \boldsymbol{E} + k_0^2 n^2(\boldsymbol{\rho}, \omega_0)\boldsymbol{E} = \nu k_0^2 n(\boldsymbol{\rho}, \omega_0) n_g(\boldsymbol{\rho}, \omega_0)\boldsymbol{E}, \quad 9\text{-}40$$

where $k_0 = \omega_0 / c$ and ν is a dimensionless eigenvalue, which we will discuss below. This Helmholtz equation is obtained after linearizing the frequency dependence of the dielectric function in the Helmholtz equation, and analytic continuation into the complex domain. The end result of this linearization is the expansion,

$$\omega^2 n^2(\boldsymbol{\rho}, \omega) \approx \omega_0^2 n^2(\boldsymbol{\rho}, \omega_0) + 2(\omega - \omega_0)\omega_0 n(\boldsymbol{\rho}, \omega_0) n_g(\boldsymbol{\rho}, \omega_0), \quad 9\text{-}41$$

where the second term defines the group velocity index,

$$n_g(\boldsymbol{\rho}, \omega_0) = \left[\frac{\partial}{\partial \omega}\omega n(\boldsymbol{\rho}, \omega)\right]_{\omega_0}. \quad 9\text{-}42$$

The Helmholtz equation transforms into a linear eigenvalue problem for a complex eigenvalue,

$$\nu = 2\frac{\omega_0 - \omega}{\omega_0} \quad 9\text{-}43$$

To solve the above Helmholtz equation, Atlas makes two approximations. The first approximation is that CAVITY.DEPTH is much larger than the width of the cavity in the x direction, so that modes can be defined approximately as having TE or TM character. The second approximation is a separation of variables for the field in the form $Y(y)X(x;y)$, so that the *y* dependence of X is only parametric (i.e., it is neglected in differentiation).

As mentioned above, Atlas uses the effective medium method to calculate these two functions. For TE modes, the field vectors and the boundary conditions for $X(x;y)$ are

$$E_z(\rho) = X(x;y)Y(y)$$
$$H_z = E_x = E_y = 0$$
$$X(a;y) = X(b;y) = 0. \quad 9\text{-}44$$

Here, a and b are the x-coordinates at the left and right boundaries of the laser mesh respectively. For the TM modes, they are

$$H_z(\rho) = X(x;y)Y(y)$$
$$E_z = H_x = H_y = 0$$
$$X'(a) = X'(b) = 0. \quad 9\text{-}45$$

In both cases, the $Y(y)$ function satisfies the outgoing plane wave boundary conditions, $Y' = \mp jkY$, where $k = (\Re e\omega)/c$. The plus and minus signs apply at the upper and lower boundaries respectively (see Figure 9-1). The sign convention adopted here is opposite to that in the transfer matrix method (see Section 11.3 "Matrix Method").

Atlas uses the 1D Helmholtz solver (see Section 9.2.3 "1D Scalar Helmholtz Equation") to solve for transverse eigenmodes $X(x;y)$ in each y slice of the Laser mesh, with y-dependent eigenvalue $\eta^2(y)$

$$\left[\frac{d^2}{dx^2} + k_0^2 n_{avg}^2(x, y, \omega_0)\right] X(x;y) = -k_0^2 \eta^2(y) X(x;y) \quad 9\text{-}46$$

In this equation the refractive index $n_{avg}(x, y, \omega_0)$ is obtained from local average of the refractive index over the Atlas mesh. The averaging is performed over all 4 rectangular mesh elements connected to each node, and it results in a smooth variation of $X(x;y)$ with respect to the "parameter" y. This averaging is compensated when solving for $Y(y)$ so that it cancels out in the resulting solution. The equation for Y follows after substituting Eq. (X) in the Helmholtz equation (H), ignoring the derivative of X with respect to y, and integrating out the x-dependence,

$$\left[\frac{d^2}{dy^2} + k_0^2 n_{eff}^2(y, \omega_0)\right] Y(y) = \nu k_0^2 \langle nn_g(y, \omega_0) \quad 9\text{-}47$$

Here, the integration with respect to x results in the transverse eigenmode defining an effective medium for the mode propagation in the longitudinal direction. The effective medium is defined via,

$$n_{eff}^2(y, \omega_0) = \frac{\int \left\{ n^2(x, y, \omega_0) - n_{avg}^2(x, y, \omega_0) - \eta^2(y) \right\} X^2(x;y) dx}{\int X^2(x;y) dx} \quad 9\text{-}48$$

$$\langle nn_g(y, \omega_0) \rangle = \frac{\int n(x, y, \omega_0) n_g(x, y, \omega_0) X^2(x;y) dy}{\int X^2(x;y) dy} \quad 9\text{-}49$$

Atlas uses the transfer matrix method to solve Eq.(Y) for $Y(y)$. In the transfer matrix method, each slice along the *y* direction is divided into intervals defined by the **LY.MESH** commands. Within each interval, n_{eff} and $\langle nn_g \rangle$ are assumed to be constant, and in the interval, $y \in [y_i, y_{i+1}]$, the solution takes the form:

$$Y(y) = A_i e^{j\beta_i y} + B_i e^{-j\beta_i y} . \quad 9\text{-}50$$

where the wavenumber is defined as

$$\beta_i = k_0 \sqrt{n_{eff}^2(x, y_i, \omega_0) - \nu \langle nn_g(y, \omega_0) \rangle} . \quad 9\text{-}51$$

The real part of this wavenumber describes the rapidly oscillating standing wave intensity pattern in space, while the imaginary part accounts for the mode growth or decay due to gain, absorption, as well as the finite lifetime due to open boundary conditions imposed on *Y*. Thus, $A_i e^{(-\Im m \beta_i) y}$ defines a slowly varying envelope function of the forward propagating wave in the resonator, while $B_i e^{\Im m \beta_i y}$ describes the backward propagating envelope function.

9.3.2 Front and Rear Facet Interface Model

The solution method for Y yields a transfer matrix, Q_L , which relates the *A* and *B* coefficients just below the top boundary to those just above the lower boundary of the laser mesh. The coefficients at the outer side of the interfaces are then related by transfer matrices Q_t and Q_b for the top and bottom interfaces

$$\begin{bmatrix} A_t \\ B_t \end{bmatrix} = Q_t Q_L Q_b \begin{bmatrix} A_b \\ B_b \end{bmatrix} , \quad 9\text{-}52$$

Currently, the top interface is always assumed to be air/semiconductor. In the case when this interface is polished with a coating, you can specify the power reflectivity of this coating using the RR parameter on the **LASER** statement. The top transfer matrix is then defined as

$$Q_t = \frac{1}{\sqrt{1 - r_t^2}} \begin{bmatrix} 1 & -r_t \\ -r_t & 1 \end{bmatrix} \begin{bmatrix} 1 + \frac{\beta_t}{k_t} & 1 - \frac{\beta_t}{k_t} \\ 1 - \frac{\beta_t}{k_t} & 1 + \frac{\beta_t}{k_t} \end{bmatrix} , \quad 9\text{-}53$$

where β_t is the wavenumber in the top slice of the laser mesh, $r_t = \sqrt{\text{RR}}$ and $k_t = \Re e(\omega / c)$ if RR is zero or unspecified, and $k_t = \beta_t$ if the interface is defined by a a coating with finite reflectivity. In the former case, Q_t reduces to the Fresnel reflection matrix, while in the latter to the conventional mirror reflection matrix.

When no external feedback is present, the bottom interface is treated in the same manner as the rear facet. You can specify the power reflectivity of any coating present at the bottom facet with the parameter RF,

$$Q_b = \frac{1}{\sqrt{1-r_b^2}} \begin{bmatrix} 1+\frac{\beta_b}{k_b} & 1-\frac{\beta_b}{k_b} \\ 1-\frac{\beta_b}{k_b} & 1+\frac{\beta_b}{k_b} \end{bmatrix} \begin{bmatrix} 1 & r_b \\ r_b & 1 \end{bmatrix}, \quad \text{9-54}$$

where β_b is the wavenumber in the top slice of the laser mesh, $r_b = \sqrt{RF}$, and $k_b = \Re e(\omega/c)$ if RF is zero or unspecified, and $k_b = \beta_b$ if the interface is defined by a a coating with finite reflectivity.

9.3.3 External Air Gap Cavity Model

Atlas allows you to make the bottom interface more complex than a dielectric step plus a coating. You can specify an air gap cavity by defining the location of a reflector for the bottom interface to be at an arbitrary location beyond the lower boundary of the laser (and Atlas) mesh. The reflector is specified by the parameter FRONTMIRROR.LOC, which can be specified on **LASER** as well as the **WAVEGUIDE** statements (see Section 9.4 "WAVEGUIDE Statement"). For a strongly confined mode in the transverse directon within the laser cavity, free space propagation of light within the air gap may result in significant diffraction. In general, the diffracted beam reflected back into the waveguide would couple to the entire spectrum of waveguide modes. However, only the coupling of this beam back into the incident mode is retained. The sum of couplings to all other modes then becomes an additional loss mechanism.

Atlas uses fully coherent beam propagation within the air gap to compute the transfer matrix within its bounds. The modes in this region are characterized by the transverse component, q, of the wavevectors $q\hat{x} + \sqrt{\Re e(\omega/c)^2 - q^2}\hat{y}$. Atlas first computes reflection and transmission coefficients for each q. It then computes the corresponding scattering matrix from the weighted average of these coefficients with respect to the magnitude squared of the Fourier components, $\hat{X}(q;L)$, of the transverse mode profile at the front facet of the waveguide.

This allows you to account for diffraction loss due to the reduced overlap between the transverse mode of the waveguide and the beam entering it after a round-trip in the external gap. You can include this loss in the rate equations by setting LOSS.DIFFRACTION to true (the default is false). If LOSS.DIFFRACTION is left as false, then Atlas treats all propagation within the air gap as normal to the front facet by computing reflection transmission coefficients at q=0 only.

9.3.4 Simulation Process and Parameters

To activate this model, set the HYBRID parameter on the **LASER** statement. You must specify E.INIT and E.FINAL to specify the energy range within which Atlas looks for the modes of the structure. Atlas computes the transfer function $1/|Q_{11}(\omega_0)|$ and resolves up to MAX.LMODES modes for each transverse mode by adaptively refining the peaks in this function. The resolution is specified by HYBRID.SPECRES in units of eV. The locations of the peaks are taken as reference frequencies to setup the effective refractive indices in the eigenvalue equation for each mode. Once a reference frequency is identified, a complex valued Newton method is applied to find ν and the electric and magnetic fields.

The electric and magnetic fields are normalized so that each mode carries a unit energy density within the bounds of the laser mesh. The energy density is then used to compute the

modal gain, loss, and spontaneous emission rate for use in the photon rate equations. At present, spatial averaging is performed over the wavelength scale rapid oscillations of standing wave patterns of the modes. Thus any effects of these oscillations are not yet included in effects such as spatial hole burning.

9.3.5 Structure Meshing and Output

When a DBR is present at one end of the cavity, Atlas automatically replaces the laser mesh over the DBR to properly include the region boundaries of the DBR structure as defined by the default Atlas mesh. You must define CAVITY.START parameter in the **LASER** statement to mark the boundary of the cavity excluding the DBR region at the top. You can exclude the CAVITY.END parameter to make it automatically coincide with the bottom of the laser mesh.

In the structure file, the *y*-dependence of the field amplitudes and modal intensity are given by $A_i e^{\mathrm{Im}(\beta_i)y}$ and not the full standing wave pattern. This choice is different from cylindrical edge emitting and VCSEL models that output the full field pattern. The choice in the HYBRID model allows you to visualize the modal profile without requiring an extremely large mesh to cover both the cavity length and the wavelength of the optical mode.

When you specify a filename using the SPEC.NAME on the **LASER** statement, Atlas outputs stick spectra of gain and photon densities at modal frequencies. When the HYBRID model is set, it also outputs another file with the extension .spec that contains the spectrum of $1/|Q(\omega_0)|$, so that you can study the evolution of the spectra from one bias point to the next.

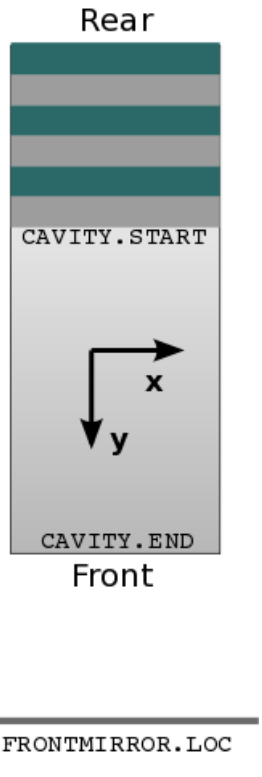


Figure 9-1: Typical Setup for a DBR Laser

9.4 WAVEGUIDE Statement

It is often useful to know and to be able to optimize the optical intensity patterns in a laser waveguide before doing a full-scale laser simulation. For that reason, vector Helmholtz solver can also be used as a standalone optical mode solver. To find optical modes at any point during a device simulation, use the **WAVEGUIDE** statement and specify the V.HELM, OMEGA, or PHOTON.ENERGY parameters and the number of transverse modes NMODE. If any of the regions has a gain model turned on, it will be included in the imaginary part of the dielectric constant. Thus, you can check how the optical modes are modified by the increase of gain in the system or by changing temperature profile. The **WAVEGUIDE** statement and its parameters are completely independent of the **LASER** statement. Therefore, it can be used by itself or simultaneously with the laser simulation.

If you want to solve for optical modes after each bias point as a post-processing, you may use just one **WAVEGUIDE** statement in the beginning of voltage sweep and include the TRACE parameter. A useful output characteristics would be real and imaginary parts of propagation constants and modal effective indexes, which can be stored into the IV log file by setting WVGD.PROP and WVGD.REFR on the output statement.

Additionally, the **WAVEGUIDE** statement also allows you to get a dispersion of the propagation constant $\beta(\omega)$. To run a dispersion calculation, specify DISPERSION, the range of photon frequencies OMEGA.INIT (or E.INIT) and OMEGA.FINAL (or E.FINAL), the number of samples NDISP, and the log file name DISP.NAME, where dispersion will be stored.

9.5 Simulation Parameters

Simulation of lasers includes three steps:

1. Defining a domain
2. Defining a type of laser simulation
3. Analyzing the output.

Numerical parameters are used to regulate convergence and computation time.

9.5.1 Specifying Simulation Domain

Laser uses rectangular mesh. If Atlas mesh is rectangular, it will be used by default. Alternatively, you can set up a separate laser mesh with LX.MESH and LY.MESH statements. By default, the solution domain spans over the whole device. Use the parameters LX.MIN, LX.MAX, LY.MIN, and LY.MAX to limit the domain. Additionally, you may switch off HELMHOLTZ parameter on the **REGION** or **ELECTRODE** statement to exclude the region or electrode from the domain.

For devices with mirror symmetry at X=0, specify only half of the device and use parameter REFLECT on **LASER** or **WAVEGUIDE** statement. The optical modes will be either symmetric or anti-symmetric with respect to H_x, E_y, E_z (and anti-symmetric or symmetric with respect to H_y, H_z, E_x). You can block either type by switching off HELM.SYM or HELM.ASYM parameters. By default, both symmetries are calculated.

The default boundary conditions on all boundaries are Perfect Electric Conductor (PEC), which is set by HELM.PEC parameter on the **LASER** or **WAVEGUIDE** statements. Alternatively, you can use HELM.PMC to set Perfect magnetic Conductor (PMC) boundary conditions on all boundaries. If both HELM.PEC and HELM.PMC are switched off, magnetic field components will be set to zero at the boundaries.

9.5.2 Physical Parameters

Relative dielectric permittivity is given by Equation 9-11. In many cases, say for index-guided lasers, imaginary part of dielectric permittivity can be neglected and the system of equations become pure real. This is regulated by the parameter INDEX.MODEL=0 in either the **LASER** or **WAVEGUIDE** statements and is a default. If imaginary part is important, set INDEX.MODEL=1. As usual, the trade-off for complex refractive index model is a slightly longer computation time.

Laser will find only certain number of modes with the highest propagation constant. The number of transverse modes can be specified by the parameter NMODE in the **LASER** or **WAVEGUIDE** statement.

The simplest way to run Laser is to do a single lasing frequency simulation. In this case, specify PHOTON.ENERGY or OMEGA in the **LASER** statement, which will be fixed throughout the simulation and used as a parameter to Helmholtz equation and finding gain. Note that in this case, the number of laser modes is equal the number of transverse modes set by NMODE parameter.

A FLOAT parameter (false by default) on the **LASER** statement will do a slight modification by letting the lasing frequency to float to the energy where gain, averaged over the device, is maximum.

If multiple transverse modes need to be included in simulation, specify LMODES parameter and set the range of photon energies with parameters E.INIT and E.FINAL or OMEGA.INIT and OMEGA.FINAL. In this case, for each transverse mode, Laser will find a number of longitudinal modes, which will have different frequencies but the same field and intensity pattern. The longitudinal modes correspond to eigen modes of Fabry-Perot resonator. To limit the number of longitudinal modes, use the MAX.LMODES parameter (default is 20). If a large range of lasing modes is required, but the number of lasing modes has to be kept low to save computation time, you can use DELTAE (same as ESEP) parameter to specify energy separation between laser modes. In the case of multiple transverse mode simulation, the total number of laser modes is less or equal to NMODE*MAX.LMODES.

Note: If multiple longitudinal modes are to be accounted for, use frequency dependent gain models by setting GAINMOD equal to 1 or 5 for the active lasing region.

The CAVITY.LENGTH parameter specifies is the length of the laser cavity in the Z direction. This will affect mirror losses and longitudinal mode energies.

Laser simulation can be done self-consistently with lattice temperature model (Equation 8-1). By default, stimulated emission is not used as a generation-recombination heat source. To use it as a heat source, specify STIM.HEAT together with GR.HEAT on the **MODELS** statement.

9.5.3 Numerical Parameters

Once the number of modes is known and mode energies are found, modal characteristics are calculated and included into photon rate equations for each laser mode. These equations are solved self-consistently with the rest of Atlas equations (e.g, Poisson, electron and hole continuity) using the Newton scheme. Atlas also has an older version of Laser, which uses 1.5D effective index solver together with relaxation method of solving photon rate equations. The older version works only when neither V.HELM or S.HELM is specified.

Note: Newton scheme works only when V.HELM or S.HELM is present on the **LASER** statement.

You have a choice of including not all, but only certain number of laser modes with strongest amplitudes in the total Jacobian. This is done using the parameter NMOD.NEWTON, which has the default value of 50 for longitudinal mode simulation and NMODE for single frequency simulation.

When the laser simulation starts, only electrical characteristics and local gain everywhere in the device is calculated. If at some bias point, usually just below laser threshold, local gain becomes positive, photon rate equations are added to the Jacobian and photon densities will be computed. If subthreshold behavior is of interest, you can enforce solution of photon rate equations by setting the START parameter on the **LASER** statement any time during the simulation.

To limit the update of the photon density of each laser mode at each Newton iteration, use parameter MAXCH. It enforces that photon density does not increase or decrease by more than MAXCH times.

TOLER parameter sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.

Once the Newton iterations have converged, refractive indexes will be updated, the Helmholtz equation will be re-solved, and new modal quantities will be calculated. Such iterative scheme of consecutive solutions of Helmholtz and all other equations usually converges in a couple of steps. You can limit the number of times the Helmholtz equation is solved for one bias point using parameter ITMAX.

9.5.4 Simulation Output

The output structure file will contain normalized intensity patterns for each transverse mode and total optical intensity given by Equation 9-37. Additionally, components squared of normalized optical fields, such as $|E_x(x,y)|^2$ and $|H_z(x,y)|^2$, can be stored in the structure file by specifying EX.OPT, EY.OPT, EZ.OPT, HX.OPT, HY.OPT, and HZ.OPT on the **OUTPUT** statement.

To store the spatial distribution of real and imaginary refractive index defined in Equation 9-11 at the lasing frequency into the structure file, set REFR.INDEX on the OUTPUT statement.

IV log file will contain current, modal photon density, modal power and modal gain and loss. You can also store the following quantities:

- Real and imaginary parts of refractive index can be probed at a specific location and stored into IV log file. For this, specify REFR.REAL and REFR.IMAG on the **PROBE** statement and also set a location and the name of the probe.
- In order to output eigenvalues (propagation constants or eigen frequencies) of the Helmholtz equation on the screen, specify PRT.EVAL on **LASER** or **WAVEGUIDE** statement.
- If real and imaginary parts of propagation constants and modal effective indexes are of interest, they can be stored into the IV log file by setting LAS.PROP (same as WVGD.PROP) and LAS.REFR (same as WVGD.REFR) on the **OUTPUT** statement.
- In order to output gain confinement factor defined in Equation 9-25 and Equation 9-27 into IV log file, use the parameters LAS.WELLCONX, LAS.WELLCONY, LAS.WELLCONZ, and LAS.WELLCONT. In order to output gain confinement factor defined in Equation 9-26 and Equation 9-27, use the parameters LAS.GAINCONX, LAS.GAINCONY, LAS.GAINCONZ, and LAS.GAINCONT.

If SPEC.NAME="filename" is specified, the modal spectrum file "modal_filename_X.log" will be generated after each bias point, where X=1,2,3,.... Alternatively, modal spectrum can also be saved using the SPECTRUM="filename" parameter on the **SAVE** statement, which will generate a "modal_filename.log" file. (In the latter case, if quantum wells are present, the additional file "filename.log" will be stored, which will contain gain and spontaneous emission of each quantum well).

9.5.5 Generation of Near-Field and Far-Field Patterns

Laser can generate the far-field pattern if you specify the PATTERNS parameter of the **SAVE** statement. The PATTERNS parameter specifies the file name prefix of two output files. The first of these files, with the .nfp suffix, contains the near-field pattern. This is essentially a copy of the near field pattern contained in the structure file output. The second of these files, with the .ffp suffix, contains the far-field pattern.

The far-field pattern is essentially a 2D fast Fourier transform (FFT) of the near-field pattern. The sampling for this transform is set by the minimum spacing in the laser mesh and the overall size of the laser mesh. The output samplings for the near-field and far-field patterns are controlled by the NEAR.NX, NEAR.NY, FAR.NX, and FAR.NY parameters of the **LASER** statement. These specify the numbers of samples in the X and Y directions in the near and far field. By default, the values are set to 100.

To specify the range of angles, where far field pattern is requested, use the parameters FAR.SPANX and FAR.SPANY with the defaults of 180°. The angles will span from -FAR.SPANX/2 to FAR.SPANX/2.

Chapter 10
VCSEL Simulator

10.1 Overview

VCSEL (Vertical Cavity Surface Emitting Lasers) performs electrical, thermal, and optical simulation of vertical cavity surface emitting lasers using accurate, robust, and reliable fully numerical methods and non-uniform meshes. With VCSEL, you can simulate advanced structures containing multiple quantum wells and oxide apertures including the effects, such as gain guiding, lattice heating, current crowding, and spatial hole burning. VCSEL works with Blaze and Giga and allows you to do the following:

- Test whether the structure complies to general VCSEL device requirements by simulating cold cavity reflectivity experiments. Calculate the reflectivity of a VCSEL cavity as a function of the incident light wavelength in a user-specified range. Then, determine the resonant frequency of the cold cavity.
- Solve the Helmholtz equation (see Equation 10-1) in cylindrical coordinates to calculate optical intensities of the multiple transverse modes for index guiding and gain guiding structures.
- Calculate the optical gain in multiple quantum well systems depending on the photon energies, quasi-Fermi levels, temperature, and optical intensity distribution. Calculate the carrier recombination due to the spontaneous and stimulated emission.
- Solve the photon rate equations (see Section 10.2.4 “Photon Rate Equations”) for multiple transverse modes to calculate the photon density in each mode and the total photon density.
- Calculate the light output power and the wavelength for each transverse mode.
- Speed up solution of the Helmholtz equation by using perturbational treatment.
- Simulate more general cavities that support multiple longitudinal modes. This obtains modal gain spectra and determines the dominant mode with maximum gain.

To perform a VCSEL simulation, you need be familiar with Atlas and Blaze first. Please see Chapters 2 “Getting Started with Atlas” and 6 “Blaze: Compound Material 2D Simulator” for more information.

Note: To avoid the confusion in terms, we refer to a VCSEL device as VCSEL and to a VCSEL simulator as VCSEL:Atlas in this chapter.

10.2 Physical Models

When modeling VCSELs, it is essential to take into account interaction of optical, electrical, and thermal phenomena that occur during the VCSEL operation. Modeling only optical properties is already an involved task. Different methods developed recently to improve the accuracy of optical solution produce results that vary from method to method [26]. The variation is especially strong between vectorial models that tend to produce a spread in results much larger than simpler scalar models.

On the other hand, the complexity of the vectorial methods makes the self-consistent electro-thermo-optical simulation impractical from the point of view of the calculation time required. For VCSEL simulation in **VCSEL:Atlas**, we adopted an approach that can account for mutual dependence of electrical, optical, and thermal phenomena.

Basic semiconductor equations (see Equations 3-1 to 3-4) are solved self-consistently with the Helmholtz Equation, Lattice Heat Flow Equation (see Equation 8-1), and the Photon Rate Equation (see Equation 10-12). Since we are using a scalar method, we neglect polarization effects and reduce Maxwell equations to scalar Helmholtz equation. The method used for the solution of Helmholtz equation is based on a well developed effective index model, which showed its validity for a much wider range of problems than originally expected [347]. This method fine tuned for simulation of various VCSEL structures is often referred to as Effective Frequency Method (EFM). EFM is a fast and flexible method that allows further development to include dispersion, diffraction losses, and graded interfaces.

VCSEL:Atlas uses a cylindrical coordinate system to take advantage of the cylindrical symmetry of the VCSEL devices. Figure 10-1 shows the relationship between Atlas's XY coordinate system and the r-θ-z cylindrical system used in **VCSEL:Atlas**. Note that the X coordinate axis is equivalent to the radius r, and the Y coordinate axis is equivalent to the Z axis in the cylindrical system.

10.2.1 Reflectivity Test Simulation

Reflectivity experiment is an important test conducted to evaluate the quality of a VCSEL device. In this experiment, the light is normally incident on a VCSEL device and its reflectivity is measured as a function of wavelength. Most manufactured VCSEL resonators are tested this way to ensure adequate optical performance. Numerical simulation of this experiment enables you to calibrate the material parameters used in the optical model and eliminate possible specification errors at the early stage of numerical analysis of a VCSEL device.

Another purpose of the numerical reflectivity test is to ensure that the analyzed device complies with general VCSEL requirements. These requirements include the following:

1. Presence of a high reflectivity band in the reflectivity spectrum of a VCSEL cavity.
2. The reflectivity over 99% in the high reflectivity band.
3. Presence of a drop in reflectivity within the high reflectivity band. The drop in reflectivity (increase in absorptance) occurs at the resonant wavelength of the cavity.

VCSEL:Atlas calculates reflectivity and absorptance using transfer matrix approach (Section 11.3 "Matrix Method"). In the simulation, the incident light propagates along the axis of symmetry of the device.

The VCSEL.INCIDENCE parameter specifies the position of the light origin with respect to the device (top or bottom). Specific output associated with the reflectivity test includes reflectivity and absorptance spectra plotted in the corresponding files, reflectivity(absorption)_top.log and reflectivity(absorption)_bottom.log.

To run the reflectivity test simulation, specify the VCSEL.CHECK parameter and other parameters in Table 10-1 on a **VCSEL** statement. Section 10.3.3 "Enabling VCSEL Solution" will discuss parameter specification in detail.

Table 10-1 User-Specifiable Parameters for the Reflectivity Test

Statement	Parameter	Default	Units
VCSEL	VCSEL.CHECK	False	
VCSEL	VCSEL.INCIDENCE	1	
VCSEL	OMEGA		s^{-1}
VCSEL	EINIT		eV
VCSEL	EFINAL		eV
VCSEL	PHOTON.ENERGY		eV
VCSEL	INDEX.TOP	1.0	
VCSEL	INDEX.BOTTOM	1.0	
VCSEL	NSPEC	100	

If the structure does not meet requirements 1-3, **VCSEL:Atlas** will close with an error. Otherwise, **VCSEL:Atlas** uses Brent's minimization procedure to find the resonant frequency of the cold cavity. **VCSEL:Atlas** uses the obtained value as a reference frequency in the solution of the Helmholtz equation.

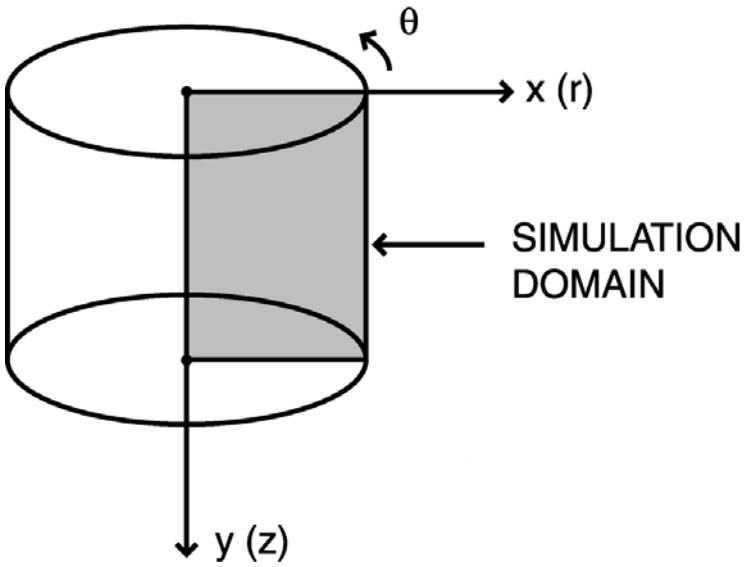


Figure 10-1: The relationship between rectangular and cylindrical coordinate systems used by VCSEL

10.2.2 Helmholtz Equation

VCSEL:Atlas solves the Helmholtz Equation in cylindrical coordinates *r*, *z*, and φ using effective frequency method [347].

$$\nabla^2 E(r, z, \varphi) + \frac{\omega^2}{c^2}\varepsilon(r, z, \varphi, \omega)E(r, z, \varphi) = 0 \quad \text{10-1}$$

where ω is frequency, ε(*r*, *z*, φ, ω) is the complex dielectric permittivity, *E*(*r*, *z*, φ) is the optical electric field, and *c* is the speed of light in vacuum.

Using the expansion around a reference frequency ω_0:

$$\frac{\omega^2}{c^2}\varepsilon(r, z, \varphi, w) \approx \frac{\omega_0^2}{c^2}\varepsilon(r, z, \varphi, \omega_0) + 2\frac{\omega_0}{c^2}\varepsilon(r, z, \varphi, \omega_0)(\omega - \omega_0) \quad \text{10-2}$$

we transform the Helmholtz equation into:

$$[\nabla^2 + k_0^2\varepsilon(r, z, \varphi)]E(r, z, \varphi) = \nu k_0^2\varepsilon(r, z, \varphi)E(r, z, \varphi) \quad \text{10-3}$$

where $k_0=\omega_0/c$ and ν is a dimensionless frequency parameter given by:

$$\nu = 2\frac{\omega_0 - \omega}{\omega_0} = 2\frac{\lambda - \lambda_0}{\lambda} - j\frac{2 \cdot Im(\omega)}{\omega_0} \quad \text{10-4}$$

We assume that the field is separable:

$$E(r, z, \varphi) = f(r, z, \varphi)\Phi(r, \varphi) \quad \text{10-5}$$

This assumption of separability is the main approximation of the method. The applicability of the method can be extended beyond this approximation by adding a non-separable component of the field on the right hand side of Equation 10-5. For dispersive materials, you can make another modification of the effective frequency method by taking into account material dispersion in Equation 10-2. For separable fields, we obtain the longitudinal wave equation for function f:

$$\left[\frac{\partial^2}{\partial z^2} + k_0^2\varepsilon(r, z, \varphi)\right]f(r, z, \varphi) = \nu_{eff}(r, \varphi)k_0^2\varepsilon(r, z, \varphi)f(r, z, \varphi) \quad \text{10-6}$$

at each lateral position (*r*, φ). Due to the cylindrical symmetry, we can ignore the azimuthal variations. In this case, we only need to solve Equation 10-6 for each cylindrically symmetric region characterized by a particular distribution of ε in the vertical direction z. Complex Newton method combined with the characteristic matrix approach (Section 11.3 “Matrix Method”) yields eigenvalues ν_{eff} and eigenfunctions *f*(*z*) of the longitudinal wave equation.

The transverse wave equation takes the form:

$$\left[\frac{1}{r}\frac{d}{dr}\left(r\frac{d}{dr}\right) - \frac{l^2}{r^2} + \nu_{eff}(r)k_0^2\langle\varepsilon\rangle_r\right]\Phi_{lm}(r) = n_{lm}K_0^2\langle\varepsilon\rangle_r\Phi_{lm}(r) \quad \text{10-7}$$

where

$$\langle \varepsilon \rangle_r = \int_0^L \varepsilon(r,z) f^2(r,z) dz \quad \text{10-8}$$

VCSEL:Atlas uses standard eigentechniques to solve an ordinary differential Equation 10-7. Solutions of this equation are LP_{lm} modes.

For each mode, the imaginary part of the eigenvalue determines the overall mode gain(loss) factor, which is negative below the lasing threshold. The lasing occurs when:

$$Im(n_{lm})\big|_{th} = 0 \quad \text{10-9}$$

The dielectric permittivity is given by:

$$\varepsilon(r,z) = \varepsilon_0 + (-\mathrm{ALPHAR} + j)\frac{\sqrt{\varepsilon_0} g(r,z)}{k_0} - j\frac{\sqrt{\varepsilon_0}(\mathrm{ALPHAA} + \mathrm{FCN} \cdot n + \mathrm{FCP} \cdot p)}{k_0} \quad \text{10-10}$$

where:

- ε_0 is the bulk high frequency permittivity. You can specify the bulk high frequency permittivity using the EPSINF parameter of the **MATERIAL** statement. If unspecified, the rule hierarchy in Table 10-2 will be used to define the high frequency permittivity.
- ALPHAR is a line width broadening factor.
- $k_0 = \omega_0 / c$
- $g\ (r, z)$ is the local optical gain.
- ALPHAA is the bulk absorption loss and is specified in the **MATERIAL** statement (specify ABSORPTION in the VCSEL statement to include absorption loss).
- FCN and FCP are the coefficients of the free-carrier loss and are set using the **MATERIAL** statement (specify FCARRIER in the VCSEL statement to include this loss mechanism).

Table 10-2 User-Specifiable Parameters for Equation 10-10

Statement	Parameter	Default	Units
MATERIAL	ALPHAR	4.0	
MATERIAL	ALPHAA	0.0	cm^{-1}
MATERIAL	EPSINF		
MATERIAL	FCN	3.0×10^{-18}	cm^2
MATERIAL	FCP	7.0×10^{-18}	cm^2
VCSEL	ABSORPTION	FALSE	
VCSEL	FCARRIER	FALSE	

10.2.3 Local Optical Gain

For a discussion of gain models, see Sections 3.9.3 “The Standard Gain Model” through 3.9.5 “Takayama's Gain Model”.

Stimulated Emission

Carrier recombination due to stimulated light emission is modeled as follows:

$$R_{st}(r,z) = \sum_m \frac{c}{\mathrm{NEFF}} g_m(r,z)\left|E_m(r,z)\right|^2 \cdot S_m \qquad 10\text{-}11$$

where R_{st} is the recombination rate due to stimulated light emission, NEFF is the group effective refractive index, and S_m is the photon number. The m subscript in this equation and all subsequent equations refers to a modal quantity. For example, S_m in Equation 10-11 is the photon number for mode m. The NEFF parameter is user-defined but has a default value of 3.57 (see Table 10-3).

Table 10-3 User-Specifiable Parameters for Equation 10-11

Statement	Parameter	Default	Units
VCSEL	NEFF	3.57	

10.2.4 Photon Rate Equations

Optical gain provides the link between optical and electrical models. The optical gain depends on the quasi-Fermi levels and in turn impacts dielectric permittivity (see Equation 10-12), and by the coupling between the stimulated carrier recombination rate (R_{st}) and the density of photons (S) as described by Equation 10-11.

To determine S_m, VCSEL solves the system of photon rate equations:

$$\frac{dS_m}{dt} = \left(\frac{c}{\mathrm{NEFF}} G_m - \frac{1}{\tau_{ph_m}} - \frac{c\ \mathrm{LOSSES}}{\mathrm{NEFF}}\right) S_m + R_{sp_m} \qquad 10\text{-}12$$

where the modal gain G_m is given by

$$G_m = \iiint g_m(r,z) \cdot \left|E_m(r,z)\right|^2 r\,d\theta\,dr\,dz \qquad 10\text{-}13$$

and the modal spontaneous emission rate R_{sp_m} is given by

$$R_{sp_m} = \iiint r_{sp}(r,z)_m r\,d\theta\,dr\,dz\,. \qquad 10\text{-}14$$

$E_m(r,z)$ is the normalized optical field.

Table 10-4 User Specifiable Parameters for Equation 10-12

Statement	Parameter	Default	Units
VCSEL	LOSSES	0	cm^{-1}

The modal photon lifetime, τ_{ph_m}, in Equation 10-12 represents the losses in the laser. The losses per mode are given by

$$\frac{1}{\tau_{ph_m}} = \frac{c}{\mathrm{NEFF}}(\alpha_{a_m} + \alpha_{fc_m} + \alpha_{mir}) = \frac{c}{\mathrm{NEFF}} G_m - \omega_0 \cdot \nu_{lm} \quad \text{10-15}$$

α_a is the bulk absorption loss, α_{fc} is the free-carrier loss, α_{mir} is the mirror loss, and ν_{lm} is a dimensionless frequency parameter. These are defined as:

$$\alpha_{a_m} = \iiint \mathrm{ALPHAA} \cdot |E_m(r,z)|^2 \times r d\theta dr dz \quad \text{10-16}$$

$$\alpha_{fc_m} = \iiint (\mathrm{FCN}\ n + \mathrm{FCP}\ p) \cdot |E_m(r,z)|^2\ r d\theta dr dz \quad \text{10-17}$$

$$\alpha_{mir} = G_m - \alpha_{a_m} - \alpha_{fc_m} - \omega_0 \cdot \nu_{lm} \frac{\mathrm{NEFF}}{c} \quad \text{10-18}$$

Spontaneous Recombination Model

For a discussion of spontaneous recombination models, see Sections 3.9.1 “The General Radiative Recombination Model” and 3.9.2 “The Default Radiative Recombination Model”.

Optical Power

The optical power emitted by the front facet is given by Equation 10-19.

$$P_f = \frac{h\omega S_m c}{\mathrm{NEFF}} \frac{\alpha_{mir}}{1 + \sqrt{R_f / R_r}(1 - R_r)/(1 - R_f)} \quad \text{10-19}$$

10.3 Simulating Vertical Cavity Surface Emitting Lasers

The key to simulating a Vertical Cavity Surface Emitting Laser is the input deck. The input deck describes the physical device structure, the physical models to be used, and the specific measurements to be performed. The deck itself can be roughly divided into three sections: structure specification, model definition, and obtaining solutions. The following sections will describe these concepts in detail.

10.3.1 Specifying the Device Structure

VCSEL devices are the most complicated devices to be addressed by device simulation. This is due to the many layers involved in making distributed Bragg reflectors (see "Specifying Distributed Bragg Reflectors" on page 601) and the possible quantum well (see "Specifying Quantum Wells" on page 602) active layers. This complexity is mitigated by the fact that these devices are mostly epitaxial. Also, many of the layers are periodic. Recognizing these simplifications, we've designed a custom syntax for defining VCSEL devices.

This syntax makes specification of periodic structures (e.g., super-lattices) simple, which completely avoids problems of aligning various layers to each other. This new syntax also simplifies meshing the structure.

Despite this simplification, specifying a VCSEL device can still be complex. We recommend that you look at the standard examples. See Section 2.4 "Accessing The Examples" for more information about these examples.

The order of statements is important when specifying the device structure. The following order of statements should be strictly followed.

1. Specify the **MESH** statement. This initializes the structure definition and allows specification of cylindrical symmetry.
2. Set up the mesh in the X direction by using the **X.MESH** statements. The mesh in the Y direction is usually specified along with the device layers in the **REGION** statements. You can, however, introduce the Y mesh lines by using the **Y.MESH** statements after the **X.MESH** statements.
3. Specify the device regions using **REGION** and **DBR** statements. These statements specify the layer thickness, composition, doping, order of placement, and meshing information in the Y direction. The order of **REGION** and **DBR** statements is up to you and is usually defined by the order the layers appear through the structure.
4. You can use quantum well models for the gain and spontaneous recombination (see Sections 3.9.7 "Unstrained Zincblende Models for Gain and Radiative Recombination" and 3.9.10 "Strained Wurtzite Three-Band Model for Gain and Radiative Recombination") by specifying QWELL on the **REGION** statement for the appropriate regions (see Section 14.6 "Parabolic Quantum Well Model").
5. Specify the locations of the electrodes using **ELECTRODE** statements.
6. Use **DOPING** statements to specify additional doping or composition fraction information. This usually isn't necessary since most, if not all doping and composition information can be specified during the specification of regions.

Setting Up Cylindrical Symmetry

VCSELs use cylindrical symmetry to account for the 3D nature of VCSEL devices. To specify cylindrical symmetry in the **MESH** statement, use the CYLINDRICAL parameter. An example of specifying cylindrical symmetry is down below.

Example

```
MESH CYLINDRICAL
```

The axis of symmetry for cylindrical is the Y axis at X=0. When using cylindrical symmetry, no X coordinates should be specified with negative values since X represents radius.

The usual abbreviation rules for parameter names can be used. See Section 2.5 “The Atlas Syntax” for more information.

Specifying the Mesh

To specify the mesh in the X direction, use the **X.MESH** statement. Each **X.MESH** statement specifies the location of one mesh line in the X direction and the spacing between mesh lines at that location.

The location is specified by the LOCATION parameter in microns. The spacing is specified by the SPACING parameter, which is also in microns. The number of mesh lines specified is limited to built-in limits in the overall size of the mesh. There should be, however, at least two **X.MESH** statements. There shouldn’t be **X.MESH** specifications with negative values in the LOCATION parameter. Such specification would be ill-defined due to the cylindrical symmetry.

You should also specify mesh lines at any defining edges of regions, electrodes, or other geometrical aspects in the X direction Also for similar reasons, use the Y.MESH statements to specify mesh lines in the Y direction.

To have mesh lines inserted automatically at defined region edges, use the **REGION** and **DBR** statements. To enable this feature, specify AUTO in the **MESH** statements

Specifying Regions

A region is a volume (in cylindrical coordinates a disk or annulus) that has a uniform material composition. The region is specified by a **REGION** statement. In the **REGION** statement, the material composition is specified by the MATERIAL parameter.

The MATERIAL parameter can take on values of any material name as described in Appendix B “Material Systems”. For Ternary and Quaternary materials, you can also specify composition fractions X or Y or both, using the X.COMPOSE or Y.COMPOSE parameters or both. You can use both the DONORS and ACCEPTORS parameters to specify uniform densities of ionized donors or acceptors or both.

The thickness of the region in the Y direction in microns is specified by the THICKNESS parameter. You can also use the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters to specify the location and extent of the region. Except in special cases, we don’t recommend this because of the difficulties aligning the numerous layers involved.

Unless it’s specified, the region will encompass the entire range of mesh in the X direction as specified in the **X.MESH** statements. Mesh lines in the Y direction will then be added at the upper and lower edges of the region. The mesh spacing at the edges is specified in microns by the SY parameter.

The TOP/BOTTOM parameters specifies the relative ordering of regions in the Y direction. Specifying TOP indicates that the region starts at the top of all previously specified regions (i.e., at the minimum previously specified Y coordinate) and extends to the thickness in the negative Y direction. Remember for device simulation, the Y axis conventionally extends down into the wafer for increasing values of Y. If no previous regions are specified, TOP indicates the region starts at Y=0 and extends to the thickness in the negative Y direction. BOTTOM indicates that the region starts at the bottom of all previously specified regions (i.e., at the maximum previously specified Y coordinate) and extends to the thickness in the positive Y direction. If no previous regions are specified, BOTTOM indicates the region starts at Y=0 and extends to the thickness in the positive Y direction.

Conventionally, we have found it convenient to start the active layers at the Y=0 location (as done in the standard examples). In ordering, **REGION** statements:

1. Start with the upper **DBR** by using the TOP parameter.
2. Use the TOP parameter on the top side contact layers.
3. Use the BOTTOM parameter on the active layers.
4. Use the BOTTOM parameter on the lower DBR.
5. Use the BOTTOM parameter on the substrate/lower contact regions.

Specifying Distributed Bragg Reflectors

A Distributed Bragg Reflector (DBR) is a periodic structure composed of regions of two alternating material compositions. A DBR can be specified by a series of **REGION** statements alternating between two different material composition. DBRs, however, typically consist of many of such layers and specifying them with REGION statements can be tedious, which could be prone to errors. You can use the **DBR** statement to simplify the specification instead. Generally, the DBR statement (alias SUPERLATTICE) can be used to specify any superlattice composed of layers of two alternating material compositions. For more information about this statement, see Section 22.9 “DBR”.

The MAT1 and MAT2 parameters specify the material names of the two materials, which is used the same way the MATERIAL parameter of the **REGION** statement is used to specify a single material.

The X1.COMP, Y1.COMP, X2.COMP, and Y2.COMP parameters are used to specify the X and Y composition fractions of the two materials for ternary and quaternary materials.

The thicknesses of the two layers are specified by the THICK1 and THICK2 parameters of the DBR statement. The doping for the layers are specified by the NA1 and NA2 parameters for acceptors. The ND1 and ND2 parameters for donors.

The mesh spacing for the DBR can be specified using the SPA1 and SPA2 parameters, just like the SY parameter of the **REGION** statement. You can also use the N1 and N2 parameters to specify the integer number of the mesh divisions in the introduction of Y direction for each of the layers.

The total number of layers is specified by the integer HALF.CYCLE parameter. The HALF.CYCLE parameter specifies the total number of layers. Thus, an odd values specifies that one more layer of the material 1 is to be used than of material 2.

The TOP and BOTTOM parameters of the DBR statement are used just like the **REGION** statement, except the first material to be added to the top/bottom of the device is always of material 1.

DBR and **REGION** statements can be intermingled indiscriminately.

Specifying Oxide Apertures

Once you specify all the epitaxial layers of the VCSEL, you can then specify an oxide aperture by using a **REGION** statement with the MATERIAL parameter assigned to an insulator and specify the geometry by using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters.

Specifying Quantum Wells

You can implement single and multiple quantum wells as described in Section 14.6 "Parabolic Quantum Well Model".

Specifying Electrodes

Device electrodes are specified using the **ELECTRODE** statement. Note that electrodes in device simulation are treated as boundary conditions. Therefore, there's no advantage to resolve the full outline of the electrode. Usually, the easiest way is to place electrodes at the top and bottom of the structure. To do this, specify the TOP or BOTTOM parameters of the **ELECTRODE** statement. The extent along the top or bottom of the device can be described by the X.MIN or X.MAX or both parameters. If you omit these parameters the electrode will extend all the way along the corresponding surface of the device.

Use the NUMBER parameter to assign electrodes to a number. Electrodes should be assigned consecutive numbers starting at 1. They can also be given a name by using the NAME parameter. They can also be addressed later by their name or number when modifying the electrode characteristics, using the **CONTACT** statement or assigning biasing information in the SOLVE statements.

10.3.2 Specifying VCSEL Physical Models and Material Parameters

Much of the simulation results depend on how the physical models are described. For more information about physical models, see Section 3.6 "Physical Models".

Selecting Device Models

For most laser devices including VCSELs, we advise that you at least enable predominant recombination mechanisms, since these usually define the threshold characteristics and compete with stimulated emission for carrier recombination. Thus, directly impact the efficiency of the device.

Typically, three bulk recombination mechanisms should be addressed: Schockley-Read-Hall (SRH), Auger, and radiative recombination. You can enable these mechanisms by specifying SRH, AUGER, and OPTR parameters on a **MODELS** statement.

The parameters of these models are specified on **MATERIAL** statements. These include TAUN and TAUP, which specify the SRH lifetimes. AUGN and AUGP, which specify the Auger coefficients. And COPT, which specifies the radiative recombination rate coefficient.

These models and associated parameters are discussed in more detail in Section 3.6.3 "Carrier Generation-Recombination Models".

Specifying Material Parameters

For Laser or VCSEL simulation there are several other material parameters that should be considered. First and foremost, specify the high frequency dielectric relative permittivity of each material region. This can be specified as given by Table 10-2.

The high frequency dielectric enters in the Helmholtz equation and directly affects which laser mode may be present. In particular, try to consider the design of the DBRs carefully. Typically, the objective of the design of the DBRs is to choose the layer thicknesses so that they are one quarter wavelength thick relative to the local dielectric. During this design process, you need to also carefully consider the band-gap of the active region, since this affects the energy and wavelength of the maximal gain. A good design will try to operate near the peak gain, while satisfying the conditions for oscillation between the DBR mirrors.

Another important consideration for choosing material parameters is the selection of band edge parameters such as band gap and electron affinity. Although defaults do exist for many material systems it is almost always better for you to specify the best estimate. Also, consider the mobilities since they directly affect series resistance.

Section 3.6.1 “Mobility Modeling” has more information about these parameters.

10.3.3 Enabling VCSEL Solution

VCSEL Solution Mesh

The solution to Helmholtz equation is performed on a spatially discrete domain or mesh. Use of matrix method for the solution of the longitudinal wave equation allows the reduction in the number of mesh points in y (z in VCSEL) to the number of material boundaries. This significantly improves calculation time while maintaining the accuracy. The mesh points in the transverse direction *x* (r in VCSEL) coincide with the device mesh.

Specifying VCSEL Parameters

Specify the **VCSEL** statement to enable the VCSEL simulator. Once you enable it, the semiconductor device equations are solved self-consistently with the photon rate equations and the Helmholtz equation.

To enable reflectivity test simulation, do the following:

1. Specify the VCSEL.CHECK parameter in a VCSEL statement.
2. Specify the VCSEL.INCIDENCE parameter to monitor cold cavity reflectivity for light incident on a structure from either top or bottom of the structure.
   - VCSEL.INCIDENCE = 1 - The light incident from the top.
   - VCSEL.INCIDENCE = 0 - The light incident from the bottom.
   - VCSEL.INCIDENCE = 2 or >2 - Both directions of light incidence are considered. The program compares cavity resonant frequencies obtained for each direction of incidence. If results do not agree within a certain tolerance, VCSEL:Atlas will close with an error.

   By default, light is incident from the top of the structure.
3. Specify INDEX.TOP for the refractive index of the medium above the structure.
4. Specify INDEX.BOTTOM for the refractive index of the medium below the structure. Default medium above and below the structure is air.

5. Specify the PHOTON.ENERGY or OMEGA parameters for (initial) photon energy or frequency.
6. Specify the EINIT and EFINAL paprameters. These parameters set the photon energy range in reflectivity test. If not specified, they take the following default values: EINIT=0.8 PHOTON.ENERGY and EFINAL=1.2 PHOTON.ENERGY.
7. Specify NSPEC for the number of sampling points between EINIT and EFINAL. The default value is NSPEC=100.

You can specify the following optional parameters in the VCSEL simulation:

- NMODE specifies the number of transverse modes of interest. Default value is NMODE=1.
- PROJ enables faster solution of the photon rate equations below threshold. You should disable this solution scheme when the bias reaches lasing threshold. Specify ^PROJ in a VCSEL statement to do that.
- PERTURB enables faster solution of the Helmholtz equation. When specified, the program solves the longitudinal wave equation only once. Perturbational approach is used to account for the refractive index changes [347].
- LOSSES specifies any additional laser losses.

10.3.4 Numerical Parameters

The following numerical parameters control the performance of the VCSEL simulation.

- TOLER sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.
- ITMAX sets maximum number of external VCSEL iterations during photon density calculation. The default value is 30.
- TAUSS is an iteration parameter used in the calculation of photon densities. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.
- MAXCH is the maximum allowed relative change of the photon density between VCSEL iterations. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.

10.3.5 Alternative VCSEL Simulator

If you need to model a vertical cavity laser that supports multiple longitudinal modes, use the **LASER** statement to enable VCSEL solution.

Specify the **LASER** statement to enable the laser simulator. Once you enable it, the semiconductor device equations are solved self-consistently with the laser rate equations. The major difference between this simulator and the one discussed above is in the approach to the solution of the Helmholtz equation.

The solver used in this case allows multiple longitudinal modes, but it is not as efficient in dealing with structural variations in the transverse direction.

There are several kinds of parameters that appear in the **LASER** statement. Most of the parameters available are discussed above with respect to VCSEL statement.

Specify laser physical parameters and models, and specify the DBR1.START, DBR1.FINAL, DBR2.START, and DBR2.FINAL parameters in the **LASER** statement.

The mirror loss is defined as:

$$\alpha_{mir} = \frac{1}{2L} ln\left(\frac{1}{R_f R_r}\right) \quad \text{10-20}$$

where L is the cavity length given as:

```
L=DBR2.START-DBR1.FINAL
```

R_f and R_r are the front and rear mirror reflectivities and are calculated from the solution of Helmholtz equation in the range specified by DBR1.START, DBR2.START, and DBR2.FINAL.

Laser simulates single or multiple longitudinal modes as well as single or multiple transverse modes. Also with Laser, you can simulate index guiding only or gain guiding. Simulation of gain guiding takes more computation time but should be considered when spatially varying gain effects are anticipated.

If you to calculate the laser spectrum, specify the multiple longitudinal modes model and additional parameters. Do the following:

1. Specify the LMODES parameter in the statement. This enables the multiple longitudinal mode model.
2. Specify the EINIT and EFINAL parameters. These parameters set the photon energy range within, which Laser will take into account multiple longitudinal modes. Make sure initial photon energy is within this range and is specified for the active lasing region.
3. Specify the photon energy separation (ESEP parameter). If this isn't specified, Laser will automatically calculate the number of longitudinal modes based on the cavity length and the energy range. We recommend that you allow Laser to choose the photon energy separation.
4. Specify the spectrum file name (SPEC.SAVE). Laser will produce a structure file containing spectrum data after calculation of each bias point. Laser will automatically append _dcN.log to the specified file name (where N is the number of the bias point) for steady-state solutions or _trN.log for a transient simulation. The first bias point where Laser is active will have N=1. This is rarely the first bias point during simulation. Use TonyPlot to examine these files. If you specify the SPECSAVE parameter, it will only save the spectrum files on every las.SPECSAVE solution.

The index in the spectrum file name will still increase by one each time. The spectrum data can be stored in a single file for the transient simulation, only if the MULTISAVE-LASER statement is set to FALSE. To enable multiple transverse mode solutions, specify the number of transverse modes by using the NMODE parameter from the LASER statement.

By default, VCSELs simulate using the index guiding model. To simulate using the gain guiding, specify INDEX.MODEL=1.

10.3.6 Far Field Patterns

Sometimes it is convenient to describe the beam as a diffraction free beam [84, 318]. We first model the near field pattern as a Gaussian by fitting to the near-field pattern calculated by VCSEL as described above. The equation for the Gaussian pattern is given in Equation 10-21.

$$E(r) = exp\left[-\left(\frac{r}{w}\right)^2\right] \quad \text{10-21}$$

Here, r is the radial coordinate and w is the fitting parameter. You can model the far-field pattern as a Gaussian by specifying GAUSS on the **VCSEL** statement. The Bessell beam far-field pattern described by Equation 10-22 can be selected by specifying BESSEL on the **VCSEL** statement.

$$E(r) = J\phi(\mathrm{A.BESSEL} \times r)\, exp\left[-\left(\frac{r}{w}\right)^2\right] \quad \text{10-22}$$

Here $J\phi$ is the zero order Bessel function and A.BESSEL is a user specifiable parameter with units of 1μ on the **VCSEL** statement.

You can choose a combination Gaussian/Bessel funciton by specifying GBESSEL on the **VCSEL** statement. This function is described by Equation 10-23.

$$E(r) = J\phi(\mathrm{A.BESSEL} \times r) \quad \text{10-23}$$

You can save near and far field patterns by specifying a value for the PATTERNS parameter in the **SAVE** statement. The value of the PATTERNS parameter is a character string representing the root name of a file for saving the near and far field patterns. The near field pattern is saved to a file with the string .nfp appended to the root name. The far field pattern is saved to a file with the string .ffp appended to the root name. Use TonyPlot to examine these files.

10.4 Semiconductor Laser Simulation Techniques

The most common technique for simulating laser diodes is to simulate forward characteristics with a gradually increasing bias. The forward voltage is usually specified. You can use, however, current boundary conditions and include external elements.

To save computational time, we recommend that you use the **VCSEL** statement. The effective frequency method employed for the solution of the Helmholtz equation in this case allows you to consider structures with oxide apertures or other structural variations in transverse direction. We also recommend that you test the structure using reflectivity experiment simulation prior to applying the bias.

If you use the multiple longitudinal mode model, computational time will be longer and strongly dependent on the number of longitudinal modes involved in the calculation.

Simulating multiple transverse modes takes more computation time and should only be preformed when the higher order modes are of interest.

Chapter 11
Luminous: Optoelectronic Simulator

11.1 Overview

Luminous is a general purpose light propagation and absorption program integrated into the Atlas framework. Luminous and Luminous 3D calculate optical intensity profiles within the semiconductor device. The intensity profiles are then converted into photogeneration rates, which are directly integrated into the generation terms in the carrier continuity equations. This unique coupling of tools allows you to simulate electronic responses to optical signals for a broad range of optical detectors. These devices include but are not limited to pn and pin photodiodes, avalanche photodiodes, Schottky photodetectors, MSMs, photoconductors, optical FETs, optical transistors, solar cells, and CCDs.

The following sections address various types of optoelectronic devices. Go to the sections that are most relevant to your application. We strongly recommend, however, that you read the other sections too.

Note: You should be familiar with Atlas and either S-Pisces or Blaze before you can use Luminous. If not, read Chapter 2 "Getting Started with Atlas" and either Chapter 5 "S-Pisces: Silicon Based 2D Simulator" or Chapter 6 "Blaze: Compound Material 2D Simulator".

The propagation of light can be described by any one of 4 physical models or several user-defined approaches. The physical models for light propagation include the following.

- Ray tracing (RT), which is described in Section 11.2 "Ray Tracing". Ray tracing is a general method of resolving 2D and 3D non-planar geometries but ignores coherence and diffraction effects.
- The transfer matrix method (TMM), which is described in Section 11.3 "Matrix Method". The matrix method is a 1D method that includes interference effects. This method is recommended for large area devices such as thin film solar cells.
- The beam propagation method (BPM), which is described in Section 11.4 "Beam Propagation Method in 2D". BPM is a general 2D method that includes diffraction effects at an increased computational expense.
- Finite difference time domain (FDTD) methods, which are described in Section 11.5 "Finite Difference Time Domain Analysis". FDTD is the most general 2D and 3D method and accounts for both diffraction and coherence by direct solution of Maxwell's equations. This is the most computationally expensive approach.

The user-definable methods are described in Section 11.6 "User-Defined Photogeneration". The remaining sections in this chapter discuss the various other common topics in photodetector simulation.

11.1.1 Hybrid Source Specifications

In some cases, it may be useful to specify different propagation models for different portions of the optical spectrum. You can do this by specifying the same beam index (specified by NUMBER on the BEAM statement) for more than one **BEAM** statement. In doing so, you should be careful to ensure that the portions of the spectrum specified in subsequent **BEAM** statements by WAVEL.START and WAVEL.END or E.START and E.END do not overlap.

Note: The overlap conditions of the spectral sampling is not currently checked in the Luminous software as we anticipate other applications for the general capability of hybrid beams.

When the solution for the intensity of a specified beam on the **SOLVE** statement is requested, the intensity applies to all parts of the hybrid total. This allows for application of for example ray tracing at longer wavelengths and FDTD at short wavelengths for a multispectral source. This can produce significant advantages in flexbly optimizing input decks for total run time.

11.2 Ray Tracing

Optoelectronic device simulation is split into two distinct models that are calculated simultaneously at each DC bias point or transient timestep.

1. Optical ray trace using real component of refractive index to calculate the optical intensity at each grid point
2. Absorption or photogeneration model using the imaginary component of refractive index to calculate a new carrier concentration at each grid point.

This is followed by an electrical simulation using S-Pisces or Blaze to calculate terminal currents.

Note: Luminous assumes that the refractive indices, both real and imaginary, are constant within a particular region.

11.2.1 Ray Tracing in 2D

Defining The Incident Beam

An optical beam is modeled as a collimated source using the **BEAM** statement. The origin of the beam is defined by parameters `X.ORIGIN` and `Y.ORIGIN` (see Figure 11-1). The `ANGLE` parameter specifies the direction of propagation of the beam relative to the X axis. `ANGLE=90` is vertical illumination from the top. `MIN.WINDOW/MAX.WINDOW` parameters specify the illumination window. As shown in Figure 11-1, the Illumination Window is “clipped” against the device domain so that none of the beam bypasses the device.

The beam is automatically split into a series of rays so that the sum of the rays covers the entire width of the illumination window. When the beam is split, Atlas automatically resolves discontinuities along the region boundaries of the device.

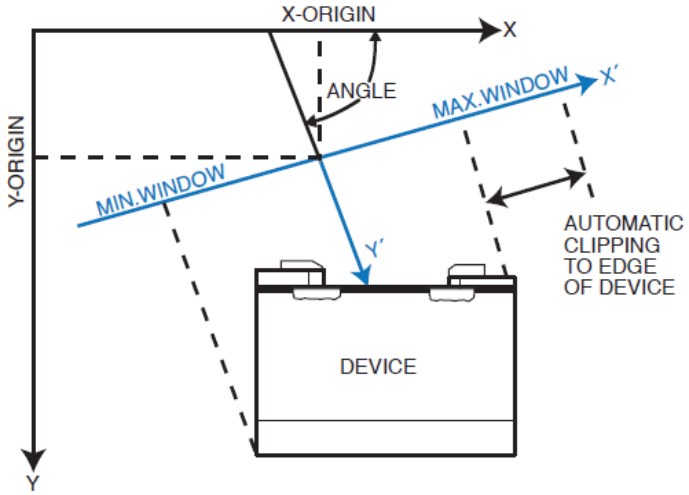


Figure 11-1: Optical Beam Geometry

Although the automatic algorithm is usually sufficient, you can also split the beam up into a number of rays using the RAYS parameter. Each ray will then have the same width at the beam origin and the sum of the rays will cover the illumination window. Even when you specify the RAYS parameter, Atlas will automatically split the rays to resolve the device geometry.

Ray Splitting At Interfaces

Rays are also split at interfaces between regions into a transmitted ray and a reflected ray. Figure 11-2 illustrates the difference between rays that are split to resolve the geometry and transmitted/reflected rays split at a region interfaces.

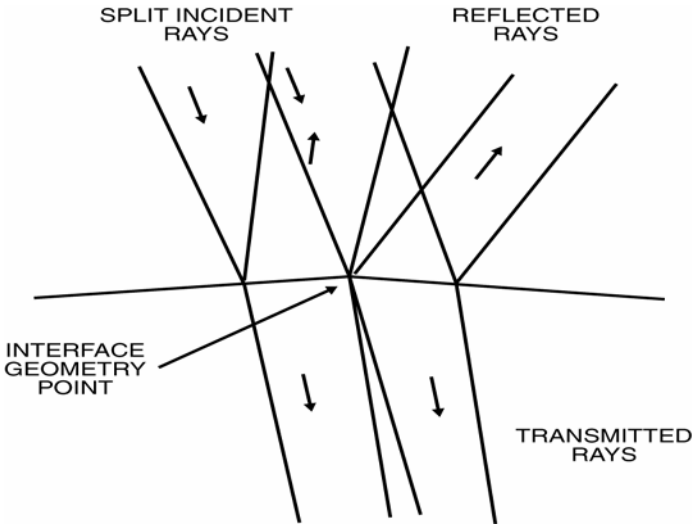


Figure 11-2: Reflected and Transmitted Rays

In Figure 11-2, the incident rays come in from the top left. They intersect a interface between two material regions with differing refractive indices. Within this interface lies a geometric point where the normal to the interface changes. This implies that the angles of reflection and transmission will be different for light incident to the left of the point from light incident on the right. Thus, the incident rays are split to resolve the interface point. The second level of splitting occurs at the interface itself. Here, the incident rays are split into reflected and transmitted rays.

11.2.2 Ray Tracing in 3D

In 3D, a simpler algorithm is used for ray tracing because the 2D algorithms become intractable in 3D. Therefore, the algorithm doesn't automatically split rays to resolve topological features of the device. You should specify enough rays to resolve such features to the desired accuracy. But there's a trade off between the computation time and accuracy in specifying the number of rays.

The 3D source geometry is very similar to the geometry shown in Figure 11-1. In 3D, the source origin is specified by three coordinates: X.ORIGIN, Y.ORIGIN, and Z.ORIGIN. Unlike Figure 11-1, there are two angles that describe the direction of propagation.

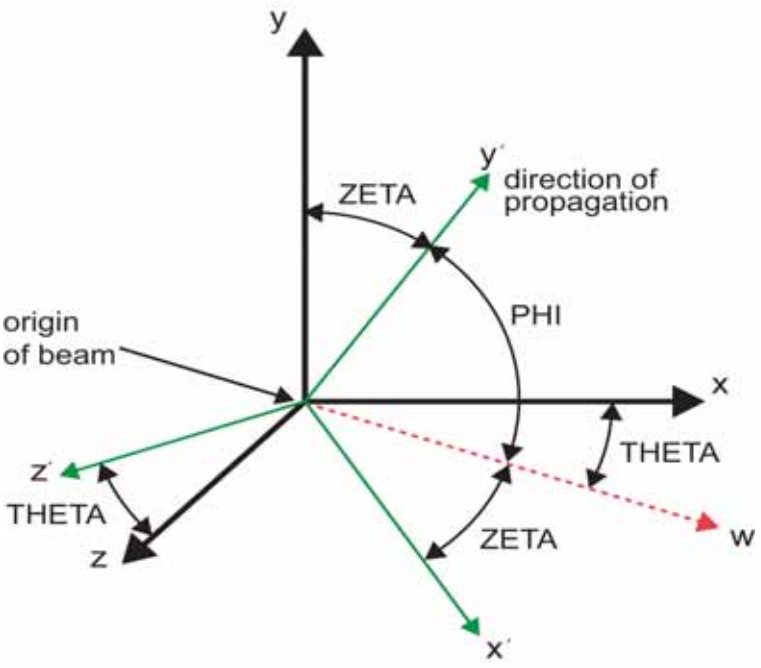


Figure 11-3: Propagation in 3D

Here, the angle PHI (aliased to ANGLE) describes the angle or propagation with respect to the XZ plane. The angle THETA describes the angle of rotation specified about the Y axis. These angles are on the **BEAM** statement.

In 3D, the clipping of the window of propagation is done in two directions. In the X direction, this is described by XMIN and XMAX. XMIN and XMAX are aliases of MIN.WINDOW and MAX.WINDOW. In the Z direction, the window is described by ZMIN and ZMAX.

The numbers of rays in the X and Z directions are described by NX and NZ. Ray samples are taken at regular intervals over the source window.

The discrete sampling of the source beam into rays in Luminous 3D is unlike that done in Luminous. In Luminous, the source beam is automatically broken up into a set of rays that resolve the device topology and variations in the interior of the device. In Luminous 3D, this process is more complex and is a computationally intractable. As such in Luminous 3D, specify a discrete sampling of the source beam (see Figure 11-4).

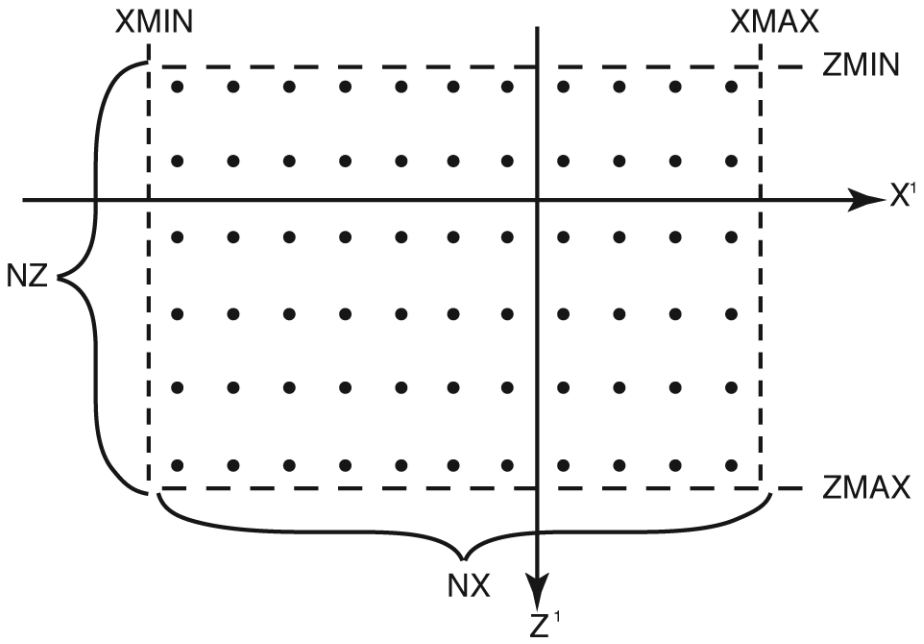


Figure 11-4: Source beam sampling

In Figure 11-4, the extent of the source sampling is specified by the XMIN, XMAX, ZMIN and ZMAX parameters of the **BEAM** statement. Even samples are taken along each of the beam front principal axes. The number of samples in the X and Z directions are given by the NX and NZ parameters of the **BEAM** statement.

11.2.3 Reflection and Transmission

Figure 11-5 shows the relationship between the angles of incidence (θ_i), reflection (θ_r), and transmission (θ_t) at the interface between two media. These coefficients are calculated as a function of the refractive indices in the two media.

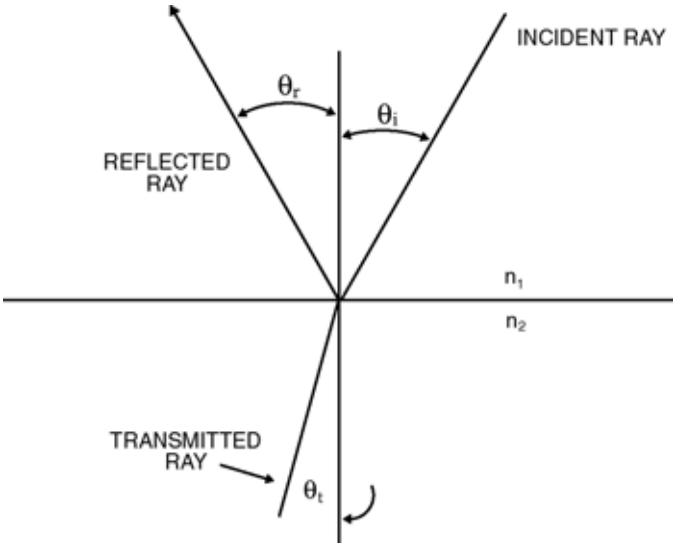


Figure 11-5: Angles of incidence, reflection and transmission

The reflection and transmission coefficients of the light for parallel and perpendicular polarization are calculated as shown in Equations 11-1 to 11-6.

$$E_r = \frac{n_2 \cos\theta_i - n_1 \cos\theta_t}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel p or TE polarization)} \tag{11-1}$$

$$E_t = \frac{2 n_1 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel p or TE polarization)} \tag{11-2}$$

$$E_r = \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular s or TM polarization)} \tag{11-3}$$

$$E_t = \frac{2 n_1 \cos\theta_i}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular s or TM polarization)} \tag{11-4}$$

$$R = \left(\frac{E_r}{E_i}\right)^2 \quad \text{(perpendicular s or TM polarization)} \tag{11-5}$$

$$T = \left(\frac{E_t}{E_i}\right)^2 \frac{n_2}{n_1} \frac{\cos\theta_t}{\cos\theta_i} \quad \text{(perpendicular s or TM polarization)} \tag{11-6}$$

where E_i is electric field of the incident wave, E_r is the field of the reflected wave, E_t is the field of the transmitted wave. R is the reflection coefficient, T is the transmission coefficient, n_1 is the refractive index on the incident side and n_2 is the refractive index on the transmission side.

Note: Here, we have adopted Yee's notation for TM (Transverse Magnetic) and TE (Transverse Electric) fields [366].

The angles of reflection and transmission are given in Equations 11-7 and 11-8.

$$\theta_r = \theta_i \tag{11-7}$$

$$n_1 \sin\theta_i = n_2 \sin\theta_t \tag{11-8}$$

where θ_i is the angle of incidence, θ_t is the angle of transmission, and θ_r is the angle of reflection.

Specifying Reflections

By default, no reflected rays are traced. The transmission coefficients are properly calculated for the transmitted rays. The REFLECTS=<i> parameter is used to set an integer number of reflections to consider. Note that setting a very large number of reflections can lead to extremely long simulation times for the ray trace.

One convenient way to overcome the long CPU times is to use the MIN.POWER parameter. This terminates each ray when the optical power falls to the fraction of the original power defined by this parameter.

Front Reflection

By default, reflection and refraction at the first interface (the initial interface with the device) are ignored. The first reflection coefficient is zero and the transmission coefficient is one. The polarization and angle of the transmitted ray at the first interface is identical to the polarization and angle of the incident beam.

If the FRONT.REFL parameter of the **BEAM** statement is specified, the transmission coefficient is calculated using Equations 11-1 to 11-6. When the transmission coefficient is calculated, it's assumed that the material outside the device domain is a vacuum. The transmitted rays are attenuated by the transmission coefficient but the reflected ray is not traced.

Back Reflection

By default, the reflection at the back of the device are ignored. No reflected ray is traced once the back of the device is reached. If the BACK.REFL parameter is specified, the backside reflection coefficient is calculated (again assuming a vacuum outside the device) and the back-side reflected ray is traced.

Sidewall Reflection

By default, the reflection from the sides of the device are ignored. No reflected ray is traced back into the structure. As above, BACK.REFL is used to enable the sidewall reflections assuming a vacuum outside the device.

User Specified Reflection

You can specify the value of the reflection coefficient at an interfaces by specifying the REFLECT parameter of the **INTERFACE** statement. The value of REFLECT must be between 0.0 and 1.0 inclusive, and the OPTICAL parameter must also be specified on the statement. You should also specify the values of P1.X, P1.Y, P2.X, and P2.Y.

11.2.4 Periodic Boundaries

In certain cases, you may find it convenient to take advantage of periodicity to reduce computation. You can do so in ray tracing by specifying PERIODIC on the **BEAM** statement. When you specify PERIODIC, all rays intersecting the extreme edges of the device in the x and z directions perpendicular to these axes will continue at the opposite edge of the device without reflection or refraction.

Discontinuous Regions

You can simulate devices electrically where a single region is defined as two or more separated areas. The ray tracing algorithm, however, doesn't support such structures. If a structure has two separate areas with the same region number, we recommend you using DevEdit to renumber the regions, perhaps even creating a new region number for each area.

Note: This limitation is only for two separated areas with the same region number and not for two regions with different region numbers of the same material. This latter case can be simulated.

11.2.5 Diffusive Reflection [372]

For optical ray-tracing semiconductor interfaces may be modelled as having diffusive reflections. To enable this model, specify the DIFFUSIVE parameter on the **INTERFACE** statement. When using this model, you must also specify the region of incidence. To do this, use the REGION parameter of the **INTERFACE** statement.

You can also specify the fraction of the incident power that experiences specular reflection using the SPECULAR parameter of the **INTERFACE** statement. The fraction of the incident power that is lost to absorption can be specified using the ABSORPTION parameter of the **INTERFACE** statement.

By default, the diffusive reflection has a Lambertian distribution (having constant luminance with respect to angle). You can also choose a Gaussian or Lorentzian distribution by specifying either GAUSS or LORENTZ respectively on the **INTERFACE** statement. The Gaussian function is described by Equation 11-9.

$$F(\phi) = \frac{1}{\mathtt{DISPERSION}} \; e^{-\left(\frac{\phi^2}{2\mathtt{DISPERSION}^2}\right)} \qquad 11\text{-}9$$

Here, $F(\phi)$ is the scattering function. ϕ is the dispersion angle and DISPERSION is a user definable parameter on the **INTERFACE** statement specifying the relative spread of the dispersion.

The Lorentz function is described by Equation 11-10.

$$F(\phi) = \frac{\texttt{DISPERSION}}{\phi^2 + \texttt{DISPERSION}^2} \quad \text{11-10}$$

When the DISPERSION parameter is set to zero no dispersion takes place.

The SCATTERED parameter on the **INTERFACE** statement specifies the number of rays per scattering event.

11.2.6 Light Absorption and Photogeneration

The cumulative effects of the reflection coefficients, transmission coefficients, and the integrated loss due to absorption over the ray path are saved for each ray. The generation associated with each grid point can be calculated by integration of the Generation Rate Formula (Equation 11-11) over the area of intersection between the ray and the polygon associated with the grid point.

$$G = \eta_0 \frac{P\lambda}{hc} \alpha e^{-\alpha y} \quad \text{11-11}$$

where:

- P is the ray intensity factor, which contains the cumulative effects of reflections, transmissions, and loss due to absorption over the ray path.
- η_o is the internal quantum efficiency, which represents the number of carrier pairs generated per photon observed.
- y is a relative distance for the ray in question.
- h is Planck's constant
- λ is the wavelength.
- c is the speed of light.
- α is the absorption coefficient given by Equation 11-12.

$$\alpha = \frac{4\pi}{\lambda} k \quad \text{11-12}$$

where k is the imaginary part of the optical index of refraction.

Photogeneration on a Non-uniform Mesh

The photogeneration algorithm used integrates the optical intensity around each node point. This is done to ensure that the total photogeneration rate isn't grid sensitive. A uniform photogeneration rate is defined as a constant value of (photogeneration rate at any node)*(element area around the node). In TonyPlot, a uniform photogeneration rate may appear to vary across a non-uniform mesh density.

Photogeneration at Contacts

The photogeneration associated with nodes that are also defined as electrodes is a special case. The electrical boundary conditions require that the carrier concentration at electrode nodes equals the doping level. This means that photogeneration at nodes that are electrodes must be zero. But just setting these nodes to zero photogeneration will typically cause an apparent drop in quantum efficiency.

The photogeneration rate at the contact nodes is calculated as usual. This photogeneration rate, however, is applied to the neighboring node inside the semiconductor. This means for a uniform mesh and photogeneration rate, if the photogeneration rate is 1.0×10^{17} pairs/cm^{-3}s, then the nodes at the contacts will have zero photogeneration and the next node into the semiconductor will have 2.0×10^{17} pairs/cm^{-3}s.

11.2.7 Outputting the Ray Trace

For example, in Luminous the rays generated during ray tracing are stored in all subsequent structure files. In Luminous 3D, this isn't the case. To save the rays in Luminous 3D, set the RAYTRACE parameter of the **BEAM** statement to the name of a file where the results of the ray trace are to be stored.

11.2.8 Monte Carlo Ray Trace

The Monte Carlo ray trace algorithm is a means of simulating random textured interfaces found in photodetectors, such as solar cells. The idea behind the algorithm is to characterize interactions between light and structural interfaces using probabilistic estimation through random number generation.

The algorithm is described as follows:

First, you should specify MONTE.CARLO on the **BEAM** statement. You can then specify the number of rays using the RAYS parameter. Remember that the more rays specified the larger the statistical sampling and presumably the better the characterization. You should try increasing the number of rays until you see convergence in the value of available photocurrent to a value invariant with the number of rays.

In 3D, you may also use the NX and NZ parameters to specify the number of rays, but this approach simply forms the product NX*NZ to obtain the total number of rays.

Individual rays are started at uniformly random locations over the source domain usually specified by the MIN.WINDOW and MAX.WINDOW parameters or defaulted by clipping to the device (see Figure 11-1). In 3D, the source domain is defined by the XMIN, XMAX, ZMIN, and ZMAX parameters.

Each ray is traced in the direction of propogation until it is incident with a structural interface between regions or a region and the outside of the device. When an interface is encountered, the reflection and transmission probabilities are calculated using Equations 11-5 and 11-6.

Rather than trace both the transmitted and reflected ray as is done in standard ray tracing, a uniform random number is generated and compared against the transmission (or reflection) probability to determine which ray to trace. Only one ray is traced per interaction at an interface.

Interfaces can also be given textural characteristics using the angular distribution functions, ADFs, as described by the **INTERFACE** statement (see Equations 11-40 through 11-45). When a ray is incident on an interface with an assigned ADF, the surface normal is randomly modified to obey the probability distribution of the ADF. The modified angle is used to calculate the transmission and reflection probabilities as well as the transmission and reflection angle. The transmission and reflection probabilities are used in the standard way described above to determine which ray is traced.

The key to improved collection efficiency of textured surfaces is that light reflecting from a angled facet may be reflected back into the textured surface and improve the net transmission of light. These multi-reflected rays are accounted for in the MC ray trace algorithm by examining the direction of the transmitted or reflected ray relative to the surface normal. If a reflected ray is reflected in the direction of the interface or a transmitted ray is transmitted in the direction of the source, the algorithm is recursively reapplied until the ray is reflected from or transmitted through the interface.

11.3 Matrix Method

Matrix method presents a fast and accurate way to simulate electromagnetic wave propagation through a layered medium. Among a variety of approaches to matrix theory, the most commonly used are the characteristic matrix and the transfer matrix descriptions.

Using the matrix method is not limited to Luminous only. It also finds its application in VCSEL and LED modules. Atlas uses the characteristic matrix approach that relates total tangential components of the electric and magnetic fields at the multilayer boundaries. More often used transfer matrix, which relates tangential components of the electric field of left-travelling and right-travelling waves, has several disadvantages that limit its applicability.

The structure of a multilayer completely determines the characteristic matrix of this multilayer. The transfer matrix also contains information about the media on both sides of the multilayer. The characteristic matrix approach is more general and can be expanded to deal with graded interfaces, in which case, the transfer matrix method is inappropriate to use.

11.3.1 Characteristic Matrix

First, we describe the characteristic matrix of a single layer. As mentioned earlier, the matrix relates tangential components of the electric $E(z)$ and magnetic $H(z)$ fields at the layer boundaries $z=0$ and $z=d$.

$$\begin{bmatrix} E(0) \\ H(0) \end{bmatrix} = M \begin{bmatrix} E(d) \\ H(d) \end{bmatrix} \quad \text{11-13}$$

The matrix itself is

$$M = \begin{bmatrix} \cos\varphi & j\sin\varphi / Y \\ jY\sin\varphi & \cos\varphi \end{bmatrix} \quad \text{11-14}$$

where

$$\varphi = \frac{2\pi}{\lambda} nd\cos\Theta \quad \text{11-15}$$

is the phase shift for the wave propagating through the layer, n is the complex refractive index, and Θ is the angle of wave propagation in the layer (Figure 11-6). Y is the optical admittance of the layer, which is for parallel (p or TE) and perpendicular (s or TM) polarizations, is given by:

$$Y^{(s)} = \sqrt{\frac{\varepsilon_0}{\mu_0}} n\cos\Theta \quad \text{11-16}$$

$$Y^{(p)} = \sqrt{\frac{\varepsilon_0}{\mu_0}} n / \cos\Theta \quad \text{11-17}$$

where ε_0 and μ_0 are permittivity and permeability of free space.

The characteristic matrix of a multilayer is a product of corresponding single layer matrices. If m is the number of layers, then the field at the first ($z=z_0$) and the last ($z=z_m$) boundaries are related as follows:

$$\begin{bmatrix} E(Z_0) \\ H(Z_0) \end{bmatrix} = M_1, M_2 \ldots M_m \begin{bmatrix} E(Z_m) \\ H(Z_m) \end{bmatrix}, M_i = \begin{bmatrix} \cos\varphi_i & j\sin\varphi_i / Y_i \\ jY_i \sin\varphi_i & \cos\varphi_i \end{bmatrix} \quad 11\text{-}18$$

11.3.2 Reflectivity, Transmissivity, and Absorptance

The detailed derivation of amplitude reflection (r) and transmission (t) coefficients is given in [257]. The resulting expressions are shown in Equations 11-19 and 11-20.

$$r = \frac{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} - Y_s M_{22}}{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} + Y_s M_{22}} \quad 11\text{-}19$$

$$t = \frac{2Y_0}{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} + Y_s M_{22}} \quad 11\text{-}20$$

where Y_0 and Y_s are the characteristic admittances of the media on both sides of the multilayer. M_{ij} are the elements of the characteristic matrix of the multilayer. Light is incident from the medium with admittance Y_0. The energy coefficients (reflectivity, transmissivity, and absorptance) are given by:

$$R = |\mathrm{r}|^2 \quad 11\text{-}21$$

$$T = \frac{Re(Y_s)}{Re(Y_0)} |t|^2 \quad 11\text{-}22$$

$$A = (1-R)\left[1 - \frac{Re(Y_s)}{Re[(M_{11} + Y_s M_{12})(M_{21} + Y_s M_{22})^*]}\right] \quad 11\text{-}23$$

These expressions take into account imaginary part of refractive index. Therefore, they are more general than Fresnel formulae (Equations 11-1 through 11-6). Luminous automatically uses these expressions when appropriate.

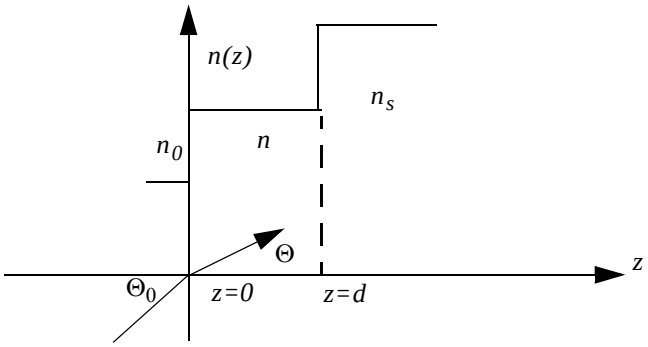


Figure 11-6: Refractive index profile of a single-layer coating

11.3.3 Transfer Matrix and Standing Wave Pattern

The characteristic matrix of a multilayer structure fully describes the optical properties of the multilayer. It is not completely, however, independent of external parameters. If the angle of incidence of light on a multilayer structure changes, the characteristic matrix changes too. Equations 11-15, 11-16, and 11-17 show the dependence of the single layer matrix elements on the angle of wave propagation in the layer. This implicit dependence on the angle of light incidence makes the characteristic matrix approach similar to the transfer matrix method for most practical purposes. In fact, when the multilayer structure does not contain any graded layers, the two methods are equivalent.

The transfer matrix directly relates the electric field amplitudes of transmitted and reflected waves to the amplitude of the incident wave. This property of the transfer matrix makes it the method of choice in many optics applications.

The transfer matrix elements are related to the characteristic matrix elements by the following expressions:

$$Q_{11} = (M_{11} + M_{21}/Y_i + Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad \text{11-24}$$

$$Q_{12} = (M_{11} + M_{21}/Y_i - Y_o M_{12} - Y_o/Y_i M_{22})/2 \quad \text{11-25}$$

$$Q_{21} = (M_{11} - M_{21}/Y_i + Y_o M_{12} - Y_o/Y_i M_{22})/2 \quad \text{11-26}$$

$$Q_{22} = (M_{11} - M_{21}/Y_i - Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad \text{11-27}$$

where Y_i and Y_o are the admittances of the input and output media defined according to Equations 11-16 and 11-17 for (*s*) and (*p*) polarization components respectively.

The amplitude reflection and transmission coefficients are:

$$r = Q_{21}/Q_{11} \quad (11\text{-}28)$$

$$t = 1/Q_{11} \quad (11\text{-}29)$$

In some devices that use multilayer structures, it is sufficient to know the transmittance and reflectance of the structure in order to fully characterize the device response to incident light. Other devices, however, such as thin film detectors, require the knowledge of light intensity distribution within the multilayer structure. Therefore in addition to the light transfer properties of such a structure, you need to be able to calculate the standing wave pattern formed by interference of traveling wave components in each layer. Standard intensity based ray-tracing calculations do not account for coherent effects and are unable to simulate standing wave patterns. Beam Propagation Method and full-wave Maxwell solvers are computationally expensive and are not useful for structures containing large number of layers. In this case, the numerical solution of the Helmholtz equation using the transfer matrix method is an attractive alternative.

Although this approach is inherently one-dimensional, it is known to produce highly accurate results for practical devices in two dimensions. Important requirements are:

- Lateral feature sizes in the device are much larger than typical layer thicknesses in a multilayer.
- Incident light is normal to the surface or is at small angles to the surface normal. When these requirements are met, the device can be divided into several lateral regions in which the structural variations occur in vertical direction z only.

For each lateral region, the one-dimensional Helmholtz equation is solved using the transfer matrix method.

First, we find reflected component of electrical field at z=0:

$$E_r = rE_i$$

where E_i and E_r are the amplitudes of incident and reflected waves respectively. The electric field and magnetic field at z=0 are:

$$E(0) = E_i + E_r \quad (11\text{-}30)$$

$$H(0) = (E_r - E_i)Y_i \quad (11\text{-}31)$$

Then, the fields at layer interfaces can be found step by step using one layer characteristic matrices. The electric field at any point within a certain layer is given by:

$$E(z) = E(z_j)\cos(\phi) - iH(z_j)\sin(\phi)/Y(j) \quad (11\text{-}32)$$

$$H(z) = H(z_i)\cos(\phi) - iE(z_i)\sin(\phi)/Y(i) \quad (11\text{-}33)$$

where $E(z_j)$ and $H(z_j)$ are electric and magnetic fields at layer interface between layers (j) and (j-1), $Y(j)$ is the optical admittance of the layer (j) and phase is given by:

$$\phi = 2\pi n \cos(\theta)(z - z_j)/\lambda \quad \text{11-34}$$

This way, intensity profile is generated throughout the device structure. The photogeneration rate is given by:

$$G(z) = \frac{\lambda}{hc}\alpha\frac{|E(z)|^2}{2\eta} \quad \text{11-35}$$

To enable transfer matrix method for calculation of intensity distribution and photogeneration profiles in thin film detectors, specify TR.MATRIX parameter on a **BEAM** statement.

For convenience, the parameters are shown in Table 11-1. The SUB.INDEX and SUB.EXTINCT parameters allow specification of the complex index of refraction at the exit side of the layer stack. Similarly, you can use the AMBIENT.INDEX parameter of the **BEAM** statement to specify the real index on the input side of the layer stack.

Table 11-1 External Index Parameters

Statement	Parameter	Default
BEAM	AMBIENT.INDEX	1.0
BEAM	SUB.INDEX	1.0
BEAM	SUB.EXTINCT	0.0

11.3.4 Transfer Matrix with Diffusive Interfaces

Frequently, thin-film detectors are constructed with rough or irregularly textured interfaces between materials. These interfaces exhibit partly specular and partly diffused reflection and transmission characteristics.

An approach has been proposed [162, 163, 377] that models both coherent (specular) and incoherent (diffused) light propagation and absorption. With this approach, the specular light is modeled using the transfer matrix method while the diffuse light is modeled using ray tracing.

The connection between the specular and diffused portions of the light is described by haze functions defined at each irregular interface as shown in Figure 11-15.

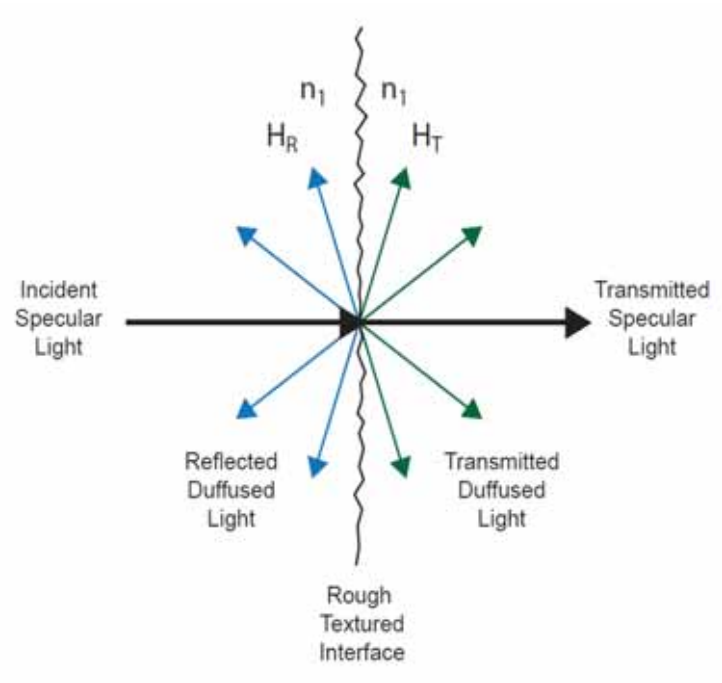


Figure 11-7: Relationship between the haze functions and the specular and diffused light at a textured interface

Equations 11-36 and 11-39 give the transmissive haze function H_T and the reflective haze function H_R.

$$H_T = 1 - exp\left[-\left(\frac{4\pi \texttt{SIGMA} \cdot \texttt{CT} \cdot \left|n_1 \cos\phi_1 - n_2 \cos\phi_2\right|}{\lambda}\right)^{NT}\right] \quad 11\text{-}36$$

$$H_R = 1 - exp\left[-\left(\frac{4\pi \texttt{SIGMA} \cdot \texttt{CR} \cdot \cos(\phi) n_1}{\lambda}\right)^{NR}\right] \quad 11\text{-}37$$

Here:

- λ is the optical wavelength.
- ϕ is the angle of incidence.
- n_1 is the index of refraction on the incident side of the interface.

- n_2 is the index of refraction on the transmitted side of the interface.
- CR, CT, NR, NT, and SIGMA are user-defined parameters described in Table 11-2.

Here, CR, CT, NR, and NT can be considered tuning parameters. SIGMA characterizes the mean variation of rough features on the interface.

Table 11-2 Default Values for User Specifiable Parameters of Equations 11-47 and 11-48

Parameter	Statement	Default	Units
CR	**INTERFACE**	1.0	
CT	**INTERFACE**	0.5	
NR	**INTERFACE**	2.0	
NT	**INTERFACE**	3.0	
SIGMA	**INTERFACE**	20.0	nm

You can describe the CT and CR parameters as a wavelength function in tabular form using the TCT.TABLE and TCR.TABLE parameters of the **INTERFACE** statement. These parameters can be names of the files containing ASCII descriptions of the dependencies. The first row of each file is a single number describing the number of samples. Subsequent rows contain pairs of numbers containing specifying the wavelength in microns and the value of CT or CR at that wavelength.

The hazefunctions, H_T and H_R, give the fraction of transmitted or reflected light that is diffused as given in Equations 11-38 and 11-39.

$$H_T = \frac{I_{dT}}{I_{dT} + I_{sT}} \quad \text{11-38}$$

$$H_R = \frac{I_{dR}}{I_{dR} + I_{sR}} \quad \text{11-39}$$

Here:

- I_{dT} is the transmitted diffused intensity.
- I_{sT} is the transmitted specular intensity.
- I_{dR} is the reflected diffused intensity.
- I_{sR} is the reflected specular intensity.

The transmissive haze function, H_T, gives the ratio of the transmitted diffused light intensity to the specular light intensity. The reflective haze function, H_R, gives the ratio of the reflected diffused light intensity to the specular light intensity. Figure 11-8 shows some example evaluations of the haze functions.

The transmissive and reflective hazefunctions can be written to log files by assigning the OUT.HT and OUT.HR parameters of the **INTERFACE** statement to the desired output file names. When outputting the haze functions, you should also specify the range of wavelengths and number of samples using the LAM1, LAM2, and NLAM parameters on the **INTERFACE** statement.

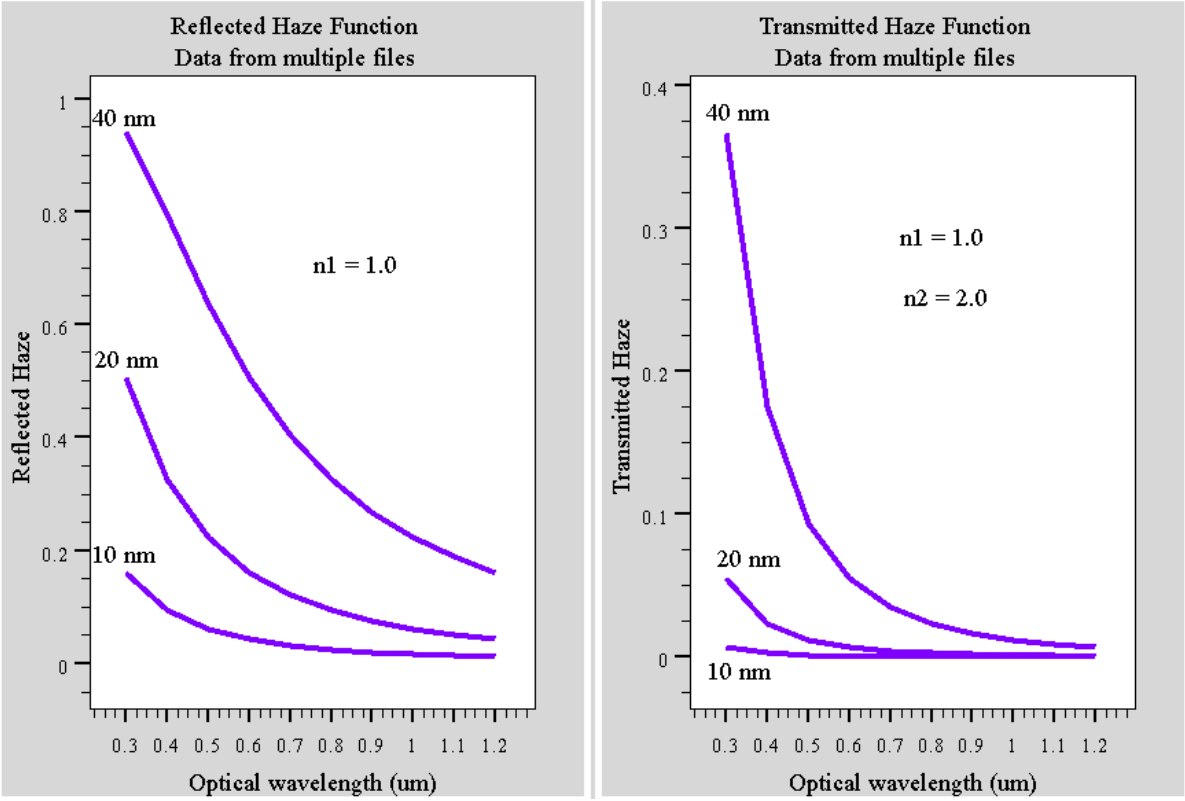


Figure 11-8: Haze Functions as a Function of Wavelength Example

The transmitted and reflected incoherent (diffused) light is also described by an Angular Distribution Function (ADF). The ADF is specified on the **INTERFACE** statement as either CONSTANT, TRIANGLE, GAUSS, LORENTZ, LAMBERT, or ELLIPSE. Equations 11-40 through 11-45 show the functional forms for the various ADFs.

CONSTANT: $\mathrm{ADF}(\phi) = 1$ 11-40

TRIANGLE: $\mathrm{ADF}(\phi) = 1 - 2\phi/\pi$ 11-41

GAUSS: $\mathrm{ADF}(\phi) = exp\left(\frac{-\phi^2}{2\mathrm{DISPERSION}}\right)$ 11-42

LORENTZ: $\mathrm{ADF}(\phi) = exp\frac{1}{\phi^2 + \mathrm{DISPERSION}^2}$ 11-43

LAMBERT: $\mathrm{ADF}(\phi) = \mathrm{COS}(\phi)$ 11-44

ELLIPSE: $\mathrm{ADF}(\phi) = \frac{\mathrm{COS}(\phi)}{\mathrm{COS}^2(\phi) + \left(\frac{0.5}{\mathrm{SEMIMINOR}}\right)^2 \mathrm{SIN}^2\phi}$ 11-45

Here, the angles are in radians and the DISPERSION and SEMIMINOR parameters are specified on the **INTERFACE** statements.

You can also specify the ADFs in tabular form. You can assign the ADF.TABLE parameter to the file name of the file containing an ASCII description of the angular dependence of ADF. The first row of this table contains the number of samples. Subsequent rows contains two numbers. The first number is the angle in degrees from 0° to 90°, and the second number is the optional unnormalized value of the ADF at that angle.

To enable the diffusive transfer matrix model, specify DIFFUSE on the **BEAM** statement.

The ADFs are normalized in each case for evaluation of diffusive generation by photoabsorption as given in Equation 11-46.

$$G(y) = 2\int_{o}^{\pi/2} P_o \mathrm{ADF}(\phi)\alpha e^{\frac{-\alpha y}{\mathrm{COS}\phi}} d\phi \qquad 11\text{-}46$$

Here:

α is the absorption coefficient.

P_o is the intensity at the interface (either I_{dT} or I_{dR}).

y is the distance from the interface.

To enable the model specify DIFFUSIVE on the **BEAM** statement and specify the diffusive characteristics of each interface as appropriate using **INTERFACE** statements.

The ADFs can be output by assigning the OUTADF parameter of the **INTERFACE** statement to the name of the output file. When you output the ADF, you should also specify the range of wavelengths and number of samples using the LAM1, LAM2, and NLAM parameters of the **INTERFACE** statement.

11.3.5 TMM Green's Function Method

The TMM Green's Function method (enbled by the TMM.GF parameter of the SAVE statement) is implemented based on the paper: **Danz et. al,** *"Dipole lifetime in stratified media"***, J. Opt. Soc. Am. B, Vol. 19, pg. 412 (2002)**. This method implements a 3D Green function calculation for a layered stack of materials. The main application is in LED and OLED. The method solves for the 3D field of a point dipole inside the stack. The dipole is Hertzian and is by default taken to be randomly oriented. Specific directions for the dipole can be specified using the parameters THETA and POLAR, and isotropic averaging can be modified using HORIZONTAL parameter as before.

11.4 Beam Propagation Method in 2D

You can simulate most optoelectronic devices using ray tracing based on geometrical optic principles. But when diffraction or coherent effects are important, ray tracing methods are no longer sufficient. In this case, you need a method that takes into account the wave nature of light.

A method that does this is called the Beam Propagation Method (BPM). BPMs such as [327] rely on paraxial appoximation. The BPM in Luminous, however, has been extended to solve a more general Helmholtz Wave Equation (Equation 11-47).

$$\nabla^2 E(\vec{r}, t) - \frac{n}{c}^2 \frac{\partial^2 E(\vec{r}, t)}{\partial t^2} = 0 \qquad 11\text{-}47$$

Here, E is the electric field of an optical wave, n is the complex refractive index of the material, and c is the speed of light in vacuum.

The Helmholtz Wave Equation is consistent with the Rayleigh-Sommerfeld formulation of scalar diffraction theory [119]. It can also be derived from Maxwell's equations assuming homogeneous and isotropic media and neglecting non-linear effects.

These assumptions put certain restrictions on the applicability of BPM. For example, material regions with gradual change in refractive index or photodetector saturation modeling aren't supported.

11.4.1 Using BPM

The BPM in Luminous isn't designed to completely replace the fast and robust ray tracing algorithm. In fact, we recommend you start the analysis of your particular application using the ray tracing method (see Sections 11.2.1 "Ray Tracing in 2D" to 11.2.3 "Reflection and Transmission").

Only use BPM when you think light diffraction or coherent effects are important. Here are the following guidelines when to use it:

- The source is monochromatic (no coherent effects are observed for multispectral sources).
- The application is 2D (currently, a 3D BPM isn't implemented due to calculation time considerations).
- Spatial distribution of irradiance or photogenerated carrier density is important.

Note: Carrier diffusion lengths should be considered.

- The structure isn't periodic in the X direction.

BPM should be useful for:

- Multilayer structures with layer thicknesses comparable to wavelength.
- Narrow incident beams (when transverse beam size is comparable to wavelength).

Currently, BPM doesn't support either user-specified reflection coefficient models (F.REFLECT) or photogeneration rate vs. position or time. The SCAN.SPOT parameter of **SOLVE** statement and anti-reflection coatings (see "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method" on page 662) are also not included.

If you need to use BPM, include the BPM parameter in the **BEAM** statement. BEAM parameters related specifically to ray tracing (such as RAYTRACE, RAYS, THINEST, MAX.WINDOW, and MIN.WINDOW) are ignored when BPM is specified.

Unlike ray tracing, the incident beam is specified on the top of the device. The origin of the beam is then assumed to be at the X.ORIGIN in the **BEAM** statement, where Y.ORIGIN parameter is ignored. This is done to avoid unnecessary beam propagation in the material surrounding the device.

You can specify additional parameters (LONGIT.STEP and TRANSV.STEP) in the BEAM statement to set respective step sizes for the internal BPM grid. If you don't define these step sizes, the default values are used. The default step size in both directions is equal to $\lambda/16.0$, where λ is the wavelength of light in vacuum. Don't use step sizes larger than $\lambda/2.0$, especially if coherent effects are of interest. Also, we don't recommend extremely small step sizes because it can result in unnecessarily long computation times. The default values are adequate for most cases.

11.4.2 Light Propagation In A Multiple Region Device

Since Equation 11-47 doesn't describe light propagation through the region's boundary, each region has to be considered separately. Each region's boundary is composed of straight line segments (edges) that form a polygon. Light propagating from one of the segments could be reflected back into the region from other segments of the boundary, while transmitted light enters other regions. Reflected and transmitted optical fields are calculated using Fresnel formulae described in Equations 11-1 to 11-6 in Section 11.2.3 "Reflection and Transmission".

Once the field reflected from a certain boundary is known, it then propagates back into the region. The light then reaches all edges of the polygon. At this point, the field incident upon each boundary is saved (i.e., added to the total field incident on each boundary). Once all polygon edges are addressed like this, the algorithm then proceeds to the next region.

When the pass through the device is complete, one reflection of light on each boundary has been accounted. The REFLECTS parameter specifies the number of passes to be done.

Like in ray tracing, if the REFLECTS parameter is large (e.g., 10), the optical part of the simulation might take considerable amount of time. To limit the number of propagation routines, use the MIN.POWER parameter. But unlike in ray tracing, where the default value of this parameter is zero, the default is MIN.POWER=1e-6. This limit helps avoid unnecessary computations.

Since only propagation through convex polygons can be properly addressed, concave polygons are automatically split to form two convex regions. If necessary, air polygons are added at the top of the device structure to fill the empty regions, where the propagation of light also needs to be considered.

11.4.3 Fast Fourier Transform (FFT) Based Beam Propagation Algorithm

Field propagation from a boundary segment through a region is considered in the coordinate system of that segment. The X axis is along the boundary and Z axis is along the normal (i.e., the Z axis corresponds to the usual Atlas Y axis, see Figure 11-9).

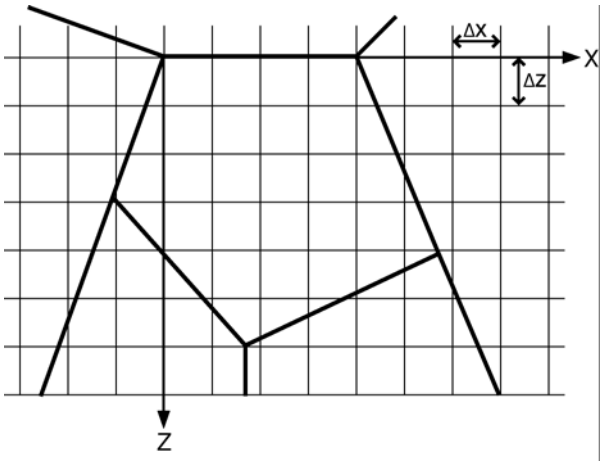


Figure 11-9: Internal Coordinate System

We assume the optical field and the device structure are uniform in the Y direction (perpendicular to the plane of the Figure 11-9). The rectangular grid covering the whole region is chosen according to the step sizes defined by `LONGIT.STEP` (ΔZ) and `TRANSV.STEP` (ΔX) parameters. The Helmholtz Equation in a 2D cartesian coordinate system (see Equation 11-48) describes field propagation through the region.

$$\frac{\partial^2 E(x,z)}{\partial z^2} + \frac{\partial^2 E(x,z)}{\partial x^2} + k^2 E(x,z) = 0 \qquad 11\text{-}48$$

Provided that the field on the input plane Z=0 is known, Equation 11-48 allows you to find the field on any subsequent plane Z=ΔZ. The numerical solution of Equation 11-48 is based on the transformation of input complex field into superposition of plane waves. This superposition represents the direction cosine spectrum of plane waves, which is obtained by the FFT of the original field (see Equation 11-49).

$$F(k_x, z=0) = \int_{-\infty}^{\infty} E(x, z=0)\exp(-ik_x X)dx \qquad 11\text{-}49$$

Each plane wave propagates at a certain angle to the Z axis. The phase accumulated by each plane wave component before reaching the plane Z=ΔZ is given by:

$$F(k_x, z=\Delta Z) = F(k_x, z=0)\exp(i\sqrt{k^2-k_x^2}\,\Delta Z) \qquad 11\text{-}50$$

where $K=2\pi n/\lambda$, n is the complex refractive index. We then can include diffraction and absorption upon propagation simultaneously. The direction cosine spectrum given by Equation 11-50 at Z=ΔZ is transformed back into the spatial domain by applying an inverse FFT.

$$E(x, z = \Delta Z) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(k_x, Z = \Delta Z)\exp(ik_x X)dk_x \quad \text{11-51}$$

The field distribution at Z=ΔZ is then found.

These steps are repeated until the field in the whole region is found. The field incident on each boundary segment and the field at each node within the region are obtained using bi-linear interpolation.

When the propagation through the same region is considered, the resulting field values at each node are added to the values saved upon the previous propagation. Since the phase information is retained in our propagation scheme, adding the field ensures the coherent effects are taken into account.

The numerical scheme to solve Equation 11-48 requires the rectangular grid in the transverse direction (X) to cover an area significantly larger than the region of interest.

Zero-padding of the input field distribution is done automatically to specify the input field on the grid. Super-gaussian filters are used in the spatial domain to suppress possible reflections from numerical boundary.

Evanescent waves are included in the propagation scheme to avoid discontinuities in the spectral domain [119].

11.5 Finite Difference Time Domain Analysis

The finite-difference time-domain (FDTD) method [366] can be used to model optical interactions between a directed optical source and a semiconductor device in much the same way as is done with geometric ray tracing. The difference is that the FDTD solves for the electric and magnetic field propagation directly from the fundamental wave equations. The FDTD approach correctly models the wave propagation with reflection, diffraction, and interference effects.

Note: It is important to remember that in these applications, the carrier diffusion lengths are usually much greater than the wavelength of the light stimulus. Ray tracing is usually an adequate description. But in certain applications, the physical scale of details of the semiconductor device are too small for ray tracing to describe the optical interaction. One such case would be a nonplanar antireflective structure. In such cases, you should consider the FDTD approach.

11.5.1 Physical Model

Maxwell's equations of light propagation are given by:

$$\frac{\partial H}{\partial t} = \frac{-1}{\mu}\nabla \times E - \frac{1}{\mu}(M_{source} + \sigma^* H) \quad 11\text{-}52$$

$$\frac{\partial E}{\partial t} = \frac{-1}{\varepsilon}\nabla \times H - \frac{1}{\varepsilon}(J_{source} + \sigma E) \quad 11\text{-}53$$

Here, E is the electric field, H is the magnetic field, ε is the permittivity, μ is the permeability, M_{source} is the equivalent magnetic current density, J_{source} is the electric current density, σ is the electric conductivity, and σ^* is the equivalent magnetic loss.

In two dimensions Equations 11-52 and 11-53 can be rewritten for transverse electric (TE) polarization as:

$$H_x = H_y = E_z = 0 \quad 11\text{-}54$$

$$\frac{\partial Hz}{\partial t} = \frac{1}{\mu}\left[\frac{\partial Ex}{\partial y} - \frac{\partial Ey}{\partial x} - (M_s + \sigma^* Hz)\right] \quad 11\text{-}55$$

$$\frac{\partial Ex}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial Hz}{\partial y} - (J_s + \sigma Ex)\right] \quad 11\text{-}56$$

$$\frac{\partial Ey}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial Hz}{\partial x} - (J_s + \sigma Ey)\right] \quad 11\text{-}57$$

For transverse magnetic (TM), Equations 11-52 and 11-53 can be written as:

$$E_x = E_y = H_z = 0 \quad 11\text{-}58$$

$$\frac{\partial Hx}{\partial t} = \frac{1}{\mu}\left[\frac{\partial Ez}{\partial y} - (M_s + \sigma^* Hx)\right] \quad 11\text{-}59$$

$$\frac{\partial Hy}{\partial t} = \frac{1}{\mu}\left[\frac{\partial Ez}{\partial x} - (M_s + \sigma^* Hy)\right] \quad 11\text{-}60$$

$$\frac{\partial Ez}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial Hy}{\partial x} - \frac{\partial Hx}{\partial y} - (J_s + \sigma Ez)\right] \quad 11\text{-}61$$

For three dimensions, Equations 11-52 and 11-53 can be written as:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x\right] \quad 11\text{-}62$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y\right] \quad 11\text{-}63$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right] \quad 11\text{-}64$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \sigma^* H_x\right] \quad 11\text{-}65$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \sigma^* H_y\right] \quad 11\text{-}66$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma^* H_z\right] \quad 11\text{-}67$$

You can specify the parameters of these equations as described in Table 11-3.

Table 11-3 User Specifiable FDTD Material Parameters

Statement	Parameter	Default	Units
MATERIAL	MAG.LOSS	0.0	mho/cm
MATERIAL	E.CONDUC	0.0	Ω/cm
MATERIAL	PERMITTIVITY	1.0	
MATERIAL	PERMEABILITY	1.0	
MATERIAL	J.ELECT	0.0	A/cm^2
MATERIAL	J.MAGNET	0.0	V/cm^2

Rather than using the parameters listed in Table 11-3, it is usually more convenient to use the complex index of refraction. The implementation of FDTD allows you to use the complex

index of refraction exactly as it is used in ray tracing and the appropriate conversions to the parameters of Table 11-3 are done automatically.

In FDTD analysis, Equations 11-54 through 11-61 are solved using a finite-difference approximation in space and finite-difference in time with suitable boundary conditions to obtain static predictions of optical intensity and photogeneration rate as a function of the optical source characteristics.

11.5.2 Beam and Mesh

To initiate FDTD analysis in Luminous and Luminous 3D, you should specify FDTD on the **BEAM** statement. The X.ORIGIN, Y.ORIGIN, Z.ORIGIN, ANGLE, X.MIN, Y.MIN, Z.MIN, and Z.MAX of the **BEAM** statement are defined for FDTD exactly as for ray tracing (see Figures 11-1 and 11-3). In FDTD, an auxiliary mesh is set up to solve the electro-magnetic equations (see Equations 11-54 through 11-61). The FDTD mesh, as we will refer to it, is defined in the primed coordinate system (x′, y′, z′) as shown in Figures 11-1 and 11-3. The FDTD mesh is a tensor product (rectangular) mesh that is not necessarily uniform in the x′, y′, and z′ directions.

Note: For the remainder of this section, we will refer to the FDTD coordinate system using the unprimed x, y, and z.

The easiest way to handle mesh generation is to use the supplied automatic mesh generator. To enable automatic mesh generation, specify FD.AUTO on the **BEAM** statement. The automatic meshing algorithm creates a minimal mesh that satisfies two conditions. First, the local mesh spacing must be less than the wavelength divided by the value of the parameter TD.SRATE. If you also specify the BIG.INDEX parameter, the local spacing will be given by the wavelength divided and by the parameter value of TD.SRATE divided by the local index or refraction.

In some cases, the prescribed mesh spacing may not be sufficient to resolve small features like quantum wells. In such cases, you may specify the FD.AUTO.MIN parameter. This specifies the minimum number of mesh lines introduced into any region in any direction (*x*, *y*, or z). The default value of the FD.AUTO.MIN parameter is 2, so you are guaranteed at least 2 mesh lines in any direction to resolve any device region.

You can specify values of TD.SRATE and BIG.INDEX without specifying FD.AUTO. In such a case, a single value of mesh spacing is calculated for the whole mesh depending on the largest index of refraction in the device.

You can also specify the FDTD mesh using the mesh line statements: FDX.MESH and FDY.MESH. These statements are similar to the X.MESH, Y.MESH, and Z.MESH statements and allow you to specify location and spacing using the LOCATION and SPACING parameters. When using FDX.MESH, FDY.MESH, and FDZ.MESH, the locations and spacings are relative to the FDTD mesh and are specified in microns. This method offers the advantage of non-uniform meshing and allows for local mesh refinement. In this approach, all FDX.MESH, FDY.MESH, and FDZ.MESH lines must be stated before the associated BEAM statement.

You may specify a uniform mesh using the SX, SY, and SZ, or SPACING parameters of the BEAM statement. The limits of the FDTD mesh are chosen by the minimum and maximum device coordinates relative to the FDTD coordinate system or X.MIN, X.MAX, Z.MIN, and Z.MAX parameters (if specified) depending on coordinate clipping. The actual spacings in x

and y will be adjusted to fit an integer number of uniform mesh lines within the FDTD mesh domain.

You can also define a rectangular uniform mesh in x, y, and z by specifying the desired number of mesh lines in the x′, y′, and z′ directions using the NX, NY, and NZ parameters on the **BEAM** statement. TM polarization is the default.

You can specify the polarization of the beam by using the POLARIZATION parameter on the **BEAM** statement (POLARIZATION=0.0 implies TE polarization, POLARIZATION not equal to 0.0 implies TM polarization). Conversely, you can specify TM or TE directly on the **BEAM** statement.

To initiate FDTD analysis, specify the FDTD parameter on the **BEAM** statement.

11.5.3 Boundary Conditions [308]

As mentioned previously, Equations 11-54 through 11-61 are solved in a 2D rectangular domain. You must specify boundary conditions to complete the problem description. In Luminous 2D and Luminous 3D, FDTD has 3 types of boundary conditions: Perfect Electrical Conductor (PEC) boundaries, Perfectly Matched Layer (PML) boundaries, and source boundaries (plane and point)

Two algorithms are implemented in the FDTD simulator to treat the fields at the periodic boundaries:

- The Cosine-Sine algorithm in 2D and 3D [293] is suitable for harmonic (sine or cosine) sources in oblique incidence onto a periodic structure.
- The split-field algorithm in 2D and 3D [293] is implemented to simulation the propagation of an oblique Gaussian pulse incident on a periodic structure. The subsequent Luminous analysis provides angle- and wavelength-resolved scattered power data.

The appropriate alogrithm is chosen automatically based on the source waveform.

Perfect Electrical Conductor (PEC) Boundary

Currently, all boundaries are by default perfect electrical conductor (PEC) boundary conditions. This implies that field values vanish at the boundaries and incident waves are completely reflected. This is convenient when simulating periodic problems, however, it may present difficulties when simulating non-periodic domains. In such cases, the perfectly matched layer boundary is preferred.

Periodic Boundaries

In many cases, the physical reality can be characterized as periodic in space. The obvious case is that of the photonic crystal. Here, simulation of the most primitive cell while still accounting for periodicity is one of the finer points of using finite difference time domain.

Also, certain special cases are conveniently described assuming some degree of periodicity. For example, you can consider angled incident light. Here, shadowing at the edges of the device can best be eliminated using periodic boundaries.

In our implementation of FDTD, we allow periodic boundaries in the X and Z directions only. This is in keeping with the general Atlas standard of Y being directed into the device.

The periodic boundaries are modeled by algorithmically connecting the solutions from one edge of the mesh to the coresponding solutions on the other edge. This algorithm produces truly periodic solutions.

When using this algorithm, you should consider using the device periodic boundaries enabled by the PERIODIC parameter of the **MESH** statement. You should be very careful to ensure that your structure is really periodic. This means that each point along one edge must have exactly the same compositional characteristics as the corresponding point on the other side. Everywhere in between arbitrary variation is allowed.

To enable periodic handling of boundaries in the X direction, you can specify X.PERIODIC on the **BEAM** statement. Similarly, the Z.PERIODIC parameter will set periodicity in the Z directions. The parameter PERIODIC on the **BEAM** statement sets both.

Periodicity is particularly important in evaluation of the far field patterns since these require two dimensional evaluation of Fourier transforms which also assume periodicity.

Perfectly Matched Layer (PML) Boundary [29]

The perfectly matched layer boundary is used to absorb outgoing light. This is useful when we want to simulate an unbounded domain. In other words, we try to absorb (rather than reflect) all outward bound waves. There are several main concepts for PMLs you should know.

- PMLs are not really boundary conditions in the conventional sense. These boundary conditions have a thickness associated with them and are included in the mesh. We hope to select the absorption coefficient of the layer to allow absorption of the outgoing light to a specified minimum.
- A PMLs is terminated by a PEC. Thus, all light passing through the PML, after accounting for absorption, is reflected back into the simulation domain after making two trips through the PML.
- PMLs are designed to be impedance matched to the simulation domain so that there is no reflection at the interface between the PML and the simulation domain.
- By default, PMLs absorb only in one direction x or y depending on their location. Therefore, light propagating parallel to a PML is not affected by the PML. A transverse value of imaginary index can be specified by the IMAG.INDEX of the **BEAM** statement.

The absorption coefficient normal to the PML is specified by Equation 11-68.

$$\alpha(x) = \mathrm{ALPHA}\left|\frac{x - x_0}{\mathrm{WIDTH}}\right|^{\mathrm{DEGREE}} \qquad 11\text{-}68$$

Here, x_0 is the coordinate of the interface between the PML and the simulation domain, and x-x_0 represents the depth into the PML. The ALPHA parameter is user-definable on the **PML** statement and specifies the maximum absorption coefficient in the layer. The WIDTH parameter specifies the width of the layer

Note: The PML is added to the simulation domain. WIDTH should be chosen appropriate to the relative dimensions of the simulation domain.

The DEGREE parameter allows absorption to be increased with the depth into the PML. Theoretically, DEGREE=0 should produce no reflection at the interface between the PML and the simulation domain. In practice, however, it is found that such abrupt transitions produce reflections due to discretization.

Table 11-4 shows the default values for the PML statement.

Table 11-4 PML Definition Parameters

Statement	Parameter	Default	Units
PML	ALL	False	
PML	ALPHA	0.0	1/cm
PML	BACK	False	
PML	BEAM	(all)	
PML	BOTTOM	True	
PML	DEGREE	2	
PML	FRONT	False	
PML	LEFT	False	
PML	NSAMP	10	
PML	PERMEABILITY	1.0	
PML	PERMITTIVITY	1.0	
PML	R90	0.0	
PML	SIGMAX	0.0	mho/cm
PML	TOP	False	
PML	RIGHT	False	
PML	WIDTH	0.0	microns
PML	XDIR	False	
PML	YDIR	False	

As mentioned, the PML is terminated by a PEC. Therefore, the reflection coefficient can be calculated by Equation 11-69.

$$R(\Theta) = exp\left(\frac{2 - \mathtt{ALPHA} \cdot \mathtt{WIDTH}\cos\Theta}{\mathtt{DEGREE} + 1}\right) \quad \text{11-69}$$

Here, R(Θ) is the reflection coefficient of the PML as a function of Θ, the angle of incidence. For convenience, you can specify the reflection coefficient at normal incidence using the R90 parameter of the PML statement. Given a selected value of DEGREE, Luminous calculates the value of one of R90, WIDTH, and ALPHA if the remaining two parameters are specified.

The placement of PMLs using the PML statement is described in Figure 11-10.

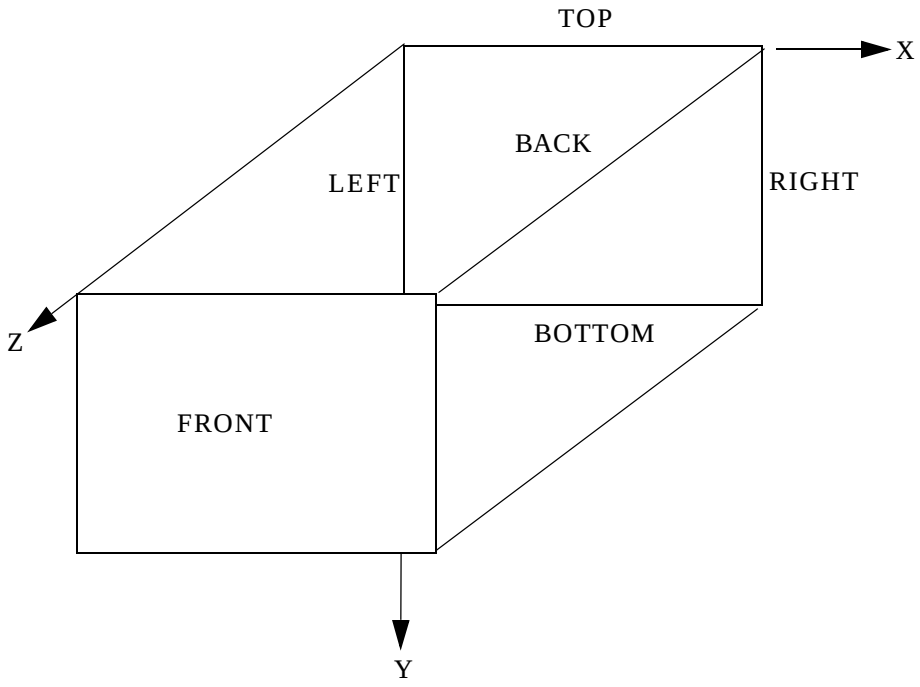


Figure 11-10: PML Placement

The ALL parameter specifies that the PML will apply to TOP, BOTTOM, RIGHT, LEFT, FRONT, and BACK.

You must also specify the number of grid points used to resolve the PML with the NSAMP parameter of the **PML** statement. As usual, the number of grid points used to represent the PML directly impacts computation time.

You can use the BEAM parameter of the PML statement to associate the PML with a previously defined optical source by beam number. This means that the **BEAM** statement must precede any associated **PML** statements. If the BEAM parameter is not specified, the PML will apply to all previously defined BEAMs.

Plane Source Boundary

The planar sources in Luminous FDTD use the transmitted field scattered field (TFSF) formulation. In this formulation, fields scattered from the target in the direction of the source are allowed to pass through the source. This enables absorbing boundaries, such as PMLs, behind the source to absorb the scattered fields. Thus, preventing them from further interaction with the target. This is important to avoid secondary interactions with re-scattered fields reflecting from the source that might occur if only a PEC boundary (default) is used. Therefore, it is strongly recommended that you always define a PML on the same edge as a plane source to avoid difficult to interpret results, such as "available photocurrents" exceeding the "source photocurrent".

By default, the source boundary is located along the X axis (see Figure 11-1). The source is simulated as a plane sinusoidal wave. Since the source originates at one of the edges of the simulation domain, the source is a TE wave if TE is specified on the **BEAM** statement and TM otherwise.

You can define if the stimulus as a cosine or a sine wave by specifying COSINE or SINE on the BEAM statement. In addition, you can specify a offset phase angle on the PHASE parameter of the BEAM statement.

Outside of the context of the beam propagation from the FDTD coordinate system origin in the positive Y direction, the plane wave may originate from any edge of the FDTD domain. The origin of the plane source is located using the S.TOP, S.BOTTOM, S.LEFT, S.RIGHT, S.FRONT, and S.BACK parameters of the BEAM statement.

By default, the S.TOP is true and the beam propagates in the same direction as for ray tracing. The alternate origins are not suggested for general usage but may be useful for optimizing PML boundaries.

Point Source Boundary

Another convenience that is usually not suggested for general usage is a point source. To enable a point source, specify POINT on the BEAM statement.

The location of the point source is defined by the X.POINT, Y.POINT, and Z.POINT parameters of the BEAM statement relative to the X, Y coordinate system in units of microns.

Device Spatial Extention

The user may add vacuum regions in selected directions of the device in order to simulate the scattering of EM field into free space. These vacuum regions are named "scattering regions" and allow:

- A proper separation of the total-field and the scattered-field region in the case of an external
- TFSF plane wave source.
- The construction of the near-to-far-field contour (2D) or surface (3D) around the device to compute the FFP.
- The visualization the scattering fields outside the device.

The length of the scattering region is specified via the SCAT.TOP, SCAT.BOTTOM, SCAT.LEFT, SCAT.RIGHT, SCAT.FRONT, SCAT.BACK, SCAT.LENGTH paramters of the BEAM statement.

Moreover, the user can extend the device limits into an asymptotic stratified structure when using FDTD.

This is useful for simulation of infinite/semi-infinite devices and for computation of the far-field patterns in these systems, both is 2D and 3D. The parameters EXTEND.TOP, EXTEND.BOTTOM, EXTEND.LEFT, EXTEND.RIGHT, EXTEND.FRONT, EXTEND.BACK and EXTEND.ALL of the BEAM statement are used to specify the length of the spatial extention in the corresponding direction.

11.5.4 Static Solutions and Error Estimation

The FDTD method is frequently used to analyze problems that are inherently transient. In the context of Luminous and Luminous 3D the problems are usually static.

To approximate static solutions, we allow the optical source to propagate from a static source until all transients are settled out. To help automate the detection of this "steady-state", the error estimator given by Equation 11-70 is defined.

$$Error = \frac{\sqrt{\sum_{in}\left(\frac{E_x'' - E_x' + E_y'' - E_y' + E_z'' - E_z'}{n_x n_y n_z}\right)}}{\frac{E_x' + E_x' + E_z'}{n_x n_y n_z}} \qquad 11\text{-}70$$

Here, E_x', E_y', and E_z' represent the field envelope over the last full cycle of the source. E_x'', E_y'', and E_z'' represent the field envelope over the previous full cycle. If you specify the parameter TD.ERRMAX on the **BEAM** statement, you can use this parameter to terminate the time domain simulation when "steady-state" is achieved.

11.5.5 Inputs and Outputs

During FDTD analysis various specific information can be extracted. These are described in the following paragraphs.

Run-Time Output

The run-time output first prints out the coordinates of the FDTD mesh and a mesh summary. Next, the run-time output prints out the specified time step, the time step limit for Courant-Friedrich-Levy stability, the time corresponding to the reciprocal of the optical frequency, and the transition time for the minimum space step. Then, the run-time output contains a summary of the optical constants for each region. Region 0 is outside the device mesh domain. This table includes regions corresponding to electrodes.

Finally, the run-time output contains iteration values. The first column is the iteration number. The second is the simulation time in seconds. The third is the number of wavelengths of the source. The fourth is the percent of the current wavelength. The fifth is the current error estimate, and the last is the estimated time to complete (if TD.LIMIT or PROP.LENS is specified).

Saving FDTD Simulation State

During development of the FDTD capability, we found that saving and loading the current state of the FDTD domain (i.e., fields) to either file or memory then restoring that state later for either

- using the same mesh thus saving some computation time in interpolating from the device simulation domain
- or offering convenience in data extraction for post display (i.e., extraction of cutplanes or facets).

We are not sure what other possibile uses there may be for saving and restoring state. It is not of practical importance for providing an initial guess for a subsequent solution. Any difference between the initial guess and the solution must propogate everywhere through the FDTD domain. Thus, an incorrect albeit educated initial guess is no better than no initial guess at all.

To save the "state" to disk, you should assign the parameter OUTSTATE of the **BEAM** statement to the root of file names of the files where the domain state is to be saved on disk. Seven files

are actually written with seven unique file names based on the root name. Seven meshes are required to represent the solution data since solutions are obtained on each face and edge of the primitive element (rectangle in 2D or 6 sided right-parallelapiped in 3D). The seventh mesh is for the primitive mesh that describes all the material compositions for the other 6 meshes. Once the OUTSTATE parameter is specified the state is written to disk each time a FDTD solution processed.

To restore the state, specify the root name of the files that the state with the root name you specified to OUTSTATE above. You specify the input state on the INSTATE parameter of the **BEAM** statement. When you also specify the logical parameter MESHONLY on the **BEAM** statement, the simulator will load the mesh from the saved state and not load the fields. If this parameter is omitted, the fields will be loaded as well. The loading of the saved state takes place at the very beginning of the FDTD solution process when the FDTD domain is created.

The current simulation state can also be saved to internal memory. This operation is very fast since it only requires a associating a user token with a pointer in memory. The disadvantage to buffering FDTD domains in memory is that they are very big so you may need to consider your available memory and swap space. Use the Linux "top" function to see how much memory you are using. Another disadvantage to buffering is that the state is lost between Atlas runs.

To save the domain in memory, you should assign the **BEAM** statement parameter MEMORIZE to a word of your choosing. After each FDTD solution, the current state is saved off in memory and associated with the word you assigned to the MEMORIZE parameter.

To restore the state from memory, you assign the RESTORE parameter to the word you had previously assigned to the MEMORIZE parameter. Like in the case of the INSTATE parameter the FDTD domain is restored from memory at the beginning of the FDTD solution process.

Disk Management of FDTD File Sizes

The file sizes in FDTD analysis are critical since you can very easily generate enormous volumes of data quite easily with FDTD. And since most files written by FDTD are only for visual postprocessing, we provide you the capability to tailor the individual file sizes by controlling the number of digits of precision maintained in the output files. The STR.PRECIS parameter controls the precision output to structure files including 2D near and far field patterns and cut planes. STR.PRESIS specifies the number of digits to maintain after the decimal place in the structure file. The default value of STR.PRESIS=0 has special significance since this implies the values will be written to full machine precision.

Time Domain Output

Output structure file "snapshots" can be saved by assigning a root file name to the TD.FILE parameter of the BEAM statement. This specifies that outfiles will be written periodically with incremented names to a file with the specified root name. These files can be examined by TonyPlot to view the field distributions as a function of time. TD.MANY specifies how many file names will be used in total before files are written over. TD.EVERY specifies how many time domain solutions are calculated between each output file.

Typically, the FDTD mesh is much larger/denser than the typical Atlas device simulation mesh. This implies that saving the "snapshots" and loading them into TonyPlot can be lengthy. To improve the efficiency of visualization, you can subsample the domain on output. To do this, specify the parameters SUBNX, SUBNY, and SUBNZ or SUBDX, SUBDY, and SUBDZ on the BEAM statement. The parameters SUBNX, SUBNY, and SUBNZ specify the numbers of evenly

spaced samples output in the X, Y, and Z directions. You can use the parameters SUBDX, SUBDY, and SUBDZ to specify the sample spacing in X, Y, and Z.

Also to expedite file transfers for post-processing in TonyPlot, you can crop the simulation domain using the parameters X.CROP.MIN, X.CROP.MAX, Y.CROP.MIN, Y.CROP.MAX, Z.CROP.MIN, and Z.CROP.MAX. These parameters are specified in units of microns and by default encompass the entire device domain including lenselets and PMLs.

Photogeneration

Once "steady-state" is achieved, the FDTD simulation is stopped and the resultant intensities are exported (interpolated) back onto the device simulation mesh. These intensities are used to calculate photogeneration rates used directly in the drift-diffusion equations exactly as it is done in ray tracing. The photogeneration rate is given by:

$$G = \frac{\lambda}{hc}\alpha\frac{|E|^2}{2\eta} \qquad 11\text{-}71$$

Near and Far Field Patterns

The FDTD analysis is supported by a very general purpose far field pattern extraction capability. This capability allows you to extract near and far field patterns from any plane normal to a coordinate axis in the FDTD domain. Typically, this can be used to extract output coupling from internal sources since it might be useful for LED analysis, or to extract reflected sources since it might be useful in radar cross-section analysis.

You can initiate the extraction of near and far field patterns by assigning the FACET parameter to the root file name for the patterns. The output files will be the root name with the extensions ".nfp" for near field and ".ffp".

The near and far field patterns are extracted along a plane contained in the FDTD domain normal to one of the principal axes. The near field pattern is simply the collection of E_x, E_y, E_z, H_x, H_y, and H_z interpolated at the intersections between the facet and the FDTD mesh. The plane is defined by one of three parameters X.FACET, Y.FACET, or Z.FACET. X.FACET defines the location along the X axis of the intersection of the extraction plane with the X axis. Y.FACET and Z.FACET are similarly defined. Remember, you can only specify one of X.FACET, Y.FACET, and Z.FACET at a time. The values of X.FACET, Y.FACET, and Z.FACET are expressed in microns relative to the FDTD coordinate system.

Certain logical placement parameters are provided to simplify pattern extraction along the extreme faces of the FDTD domain. These parameters are TOP.FACET, BOTTOM.FACET, LEFT.FACET, RIGHT.FACET, FRONT.FACET, and BACK.FACET. Since each of these parameters imply a location and direction, they may also only be used one at at time and may not be specified in conflict with any of the parameters X.FACET, Y.FACET, or Z.FACET. To clarify TOP.FACET means the same as the assignment Y.FACET=<minimum y coordinate in microns>.

The Luminous FDTD simulator provides a comprehensive computation of far-field patterns (FFP) in free space in different device boundary conditions (PMLs, PEC, periodic and combinations thereof) and for different source types (harmonic, broadband, plane wave, point source, etc.). The FFP are computed via a near-to-far-field integration using a suitable contour (2D) or surface (3D) constructed by the simulator taking into account the device boundary conditions [293].

The output data is saved into the following files:

- For harmonic sources:

- `.FFP file` : shows the angle resolved scattered/radiated power by the device. Polar angles convention used. For periodic devices, both scattering Bragg peaks and the envelope shape of the pattern are shown.

- For broadband sources (e.g. pulses):

- `.FFP file` : shows the same data as above but for each frequency included in the source spectrum as well as the sum of all frequency contributions.
- `.SPC file` : shows the power spectrum of the scattered/radiated fields from the device as a function of source frequency (in THz). For plane wave sources, the total cross section as a function of source frequency is also shown.

The FDTD 2D simulation using the POLAR parameter in the BEAM statement with non zero value. In this case, both TE and TM are simulated successively. The output files saved at the end of each FDTD (2D) simulation (.str, .ffp, etc.) now have a suffix _te and _tm indicating the simulation was run in the TE or TM more respectively.

For FDTD 3D simulation, when using the parameter HORIZONTAL in the BEAM statement (values between 0 and 1), three consecutive runs are performed for dipole orientation along x (TM(p) parallel polarization), y (TM(p) normal polarization) and z (TE(s) polarization). The respective output files associated with each polarization now have the suffixes _tm_para, _tm_norm and _te.

Energy Dissipation and Interference

To close the discussion of FDTD, we will mention a couple of things about energy dissipation and interference effects.

Suppose we illuminate a layer 1 micron thick with a TE sine wave of wavelength 1 micron. Further assume the wave propagates from the left of the layer against a PEC at the right of the layer. Figure 11-11 shows the resultant envelope of the light intensity after 1, 2, 3 and 4 wavelengths. The effects of the coherent reflection should be obvious. It is also notable, however, that the peak of the envelope increases as the square of the number of wavelengths. This is because there is no power dissipation in the system. The wave amplitude increases as the number of wavelengths and the power increases as the square of the amplitude. This illustrates an important consideration in FDTD analysis. That is that if projected waves are not completely absorbed by the test structure and the PML boundary conditions, the envelope of intensity increases with time. This has an important impact on the error estimation. The error estimate that is proportional to the change in intensity envelope divided by the value of the intensity envelope will be bounded by the square of the simulation time.

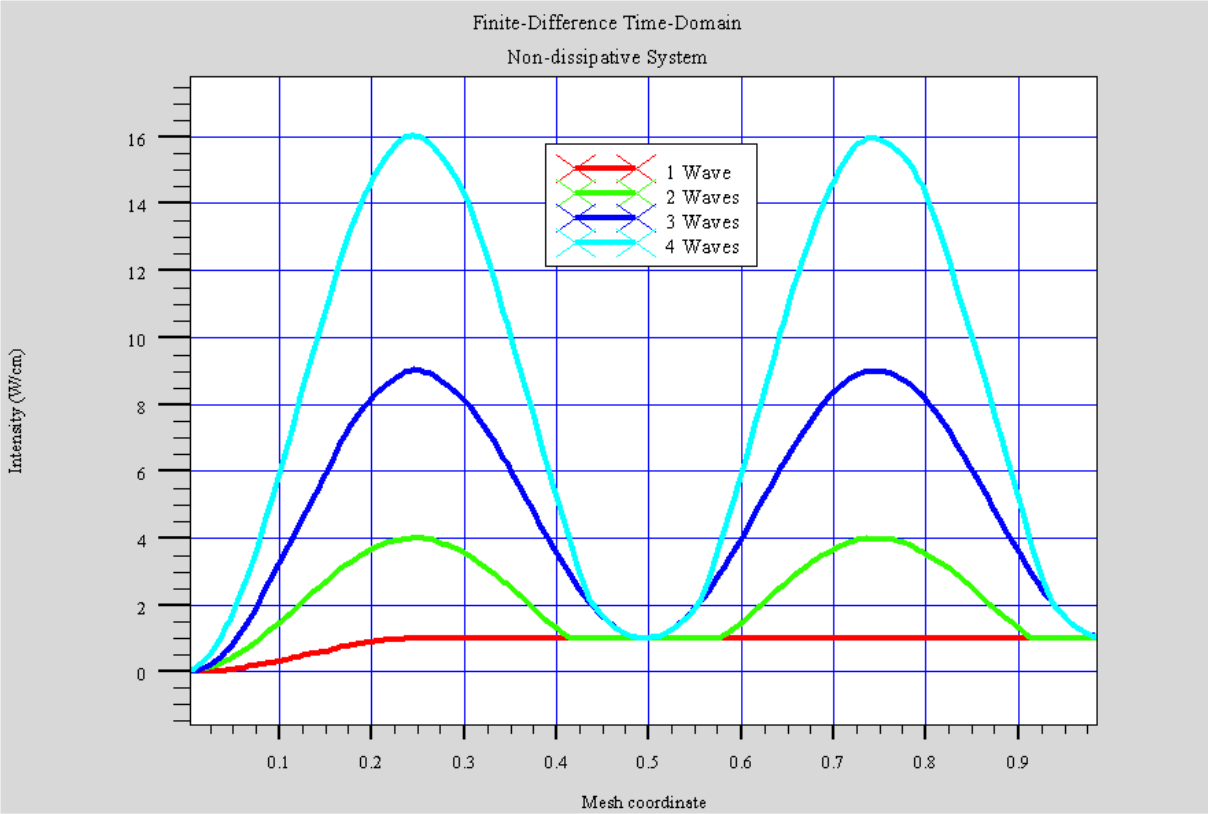


Figure 11-11: Finite-Difference Time Domain Non-Dissipative System

Another interesting aspect of Figure 11-11 is that the minimum of the envelope is not at 0 as expected for such interference but 1. This is because the initial wave does not experience interference and thus contributes a unit offset to the minimum.

FDTD Output Summary

The FDTD Output Summary performed at the end of FDTD simulations prints out the following quantities:

- Percentage of losses in all PMLs defined and in the device.
- Total power absorbed in the device.
- Total power scattered by/radiated from the device.
- Source photocurrent
- Available photocurrent

*For plane waves (harmonic or broadband)

- Reflectance,
- Specular reflectance
- Transmittance
- Total cross section

*For dipole sources (harmonic or broadband)

- Purcell factor

For broadband sources, the quantities output are the sum of the contribution from all the frequencies contained in the injected source.

11.5.6 Accuracy, Stability, and Simulation Time

There are clear trade-offs between speed and accuracy. For spatial sampling, you should specify `FD.AUTO` since this best insures that all layers will be resolved, and it will insure that spatial sampling will obey the rule that the step size is always less than the specified wavelength divided by the local real index of refraction divided by the parameter `TD.SRATE`.

The relation between `TD.SRATE` and error can best be understood by considering the expected difference between a sine wave and a discrete representation given by `TD.SRATE` evenly spaced samples as shown in Figure 11-12.

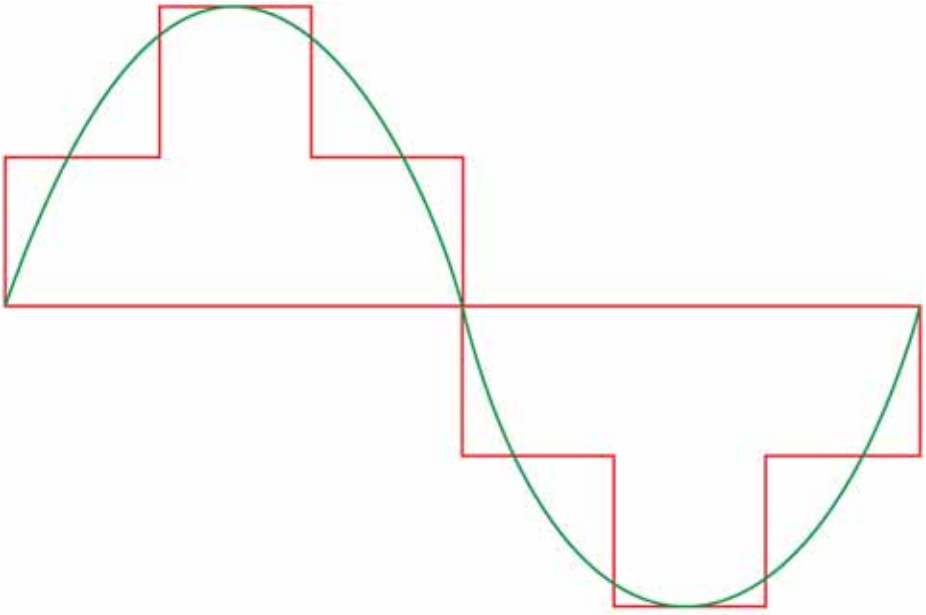


Figure 11-12: Discretization Errors TD.SRATE=6

Numerically, you can go all the way down to Nyquist rate (TD.SRATE=2). A value of TD.SRATE=10 produces about a 10% error. Taflove recommends a value of TD.SRATE=20 [308]. The simulation time increases roughly as the square of TD.SRATE for 2D simulation and the cube of TD.SRATE for 3D. The best way to overcome this is to reduce the size of the problem by appropriate use of periodic, mirror (PEC), and absorbing (PML) boundary conditions.

If you are using PML boundaries, we suggest that you make them a similar dimension to the overall problem. We feel that this gives PMLs equal weight numerically, which is important since the absorption in PMLs is used in all subsequent postprocessing steps (e.g., available photocurrent, R, T, A, and optical coupling for LEDs). The automatic meshing (FD.AUTO) works properly in the PML so you only need specify the width (WIDTH), the reflectivity (R90) and the degree (DEGREE). We will discuss R90 and DEGREE shortly. Keep in mind each PML adds mesh to the structure increasing simulation time proportional to the width (WIDTH).

In some cases, PMLs can be removed entirely and replaced with a mirror boundary to reduce simulation time. This is common when simulating an aluminum mirror. In reality, reflectivity commonly exceeds 98% and if for example the difference between reflectance of the real mirror and a perfect mirror is less than the expected accuracy of simulation, you can justify removal of the PML in favor of a mirror boundary condition.

Another trick is to reduce the local index of refraction in regions that are not important to the optical problem (e.g., contacts and metals). This should only be done in cases where the index in that region is actually dominant for meshing.

The problem size in terms of time discretization is proportional to the total simulation time divided by the constant time step size.

The step size can be directly set by the parameter DT. It is advisable that the the time step not exceed the CFL stability limit. This is considered so important that a fail-safe mechanism has been introduced to place a upper limit on the time step size. This limit can be disabled by

specifying ^DT.SAFE. Specifying time steps above the CFL stability limit is risky in the sense the simulation may "blow up". The symptoms of this catastrophic condition is that the residual errors will increase rapidly before stabilizing at some number much greater than one. This is usually accompanied by huge increases in the fields and absorption dominant in a single PML or the main simulation domain. Once this state is reached, there is no recovery.

If you experience stability problems, you can simply decrease the time step thus increasing simulation time. You may also try to change the problem a bit to perhaps save a marginal situation. The best place to look is in the PMLs. Adjust DEGREE, R90, TD.STATE, or WIDTH until you reach your desired settings.

You should know that selection of time steps below the CFL stability limit does not insure stability. The CFL does not consider the effects of the imaginary part of the complex index of refraction, *k*. Experience has found that stability is adversely affected by increased *k*. This would indicate good stability will favor lower numbers of both the DEGREE and the R90 parameter. For example, you may want to select DEGREE=1 and R90=0.01 in the case that TD.SRATE=10 since the anticipated discretization errors will exceed the reflectance of the PML.

As for picking very large time steps, you are eventually limited by Nyquist rate. Sampling below Nyquist will cause your simulated frequency to alias into a lower frequency. The static frequency model approach used by FDTD will keep the complex index of the actual frequency but the propogation frequency will be the remainder of the actual frequency divided by the Nyquist rate. The results may still be interesting but will certainly be non-physical.

The last part of the problem size is the total simulation time. This is controlled by a distance parameter (PROP.LENG) since such a parameter is more user-friendly in the sense that it relates to intuitively obvious features of the simulation.

The propagation length (PROP.LENG) is the spatial distance in the simulation domain you expect the light to meaningfully go. This information can be taken to be the maximum dimension from the source to the furthest meaningful PML measured in microns. If a PML is important, it should not lie outside the horizon of the simulation time. If you choose a propogation length that is too short, it should be obvious in the log files and or structure files produced by the FDTD simulator (TD.LOG and TD.FILE). The log files will show unconverged behavior. Structure files will show abrupt edges to the simulation horizon.

When you specify PROP.LENG, you should specify BIG.INDEX. The logical parameter BIG.INDEX specifies that during the conversion from distance to time. The velocity of light should be scaled by the largest index in the simulation domain. This helps insure that the PMLs are reached.

You should also try to consider if reflections may be critical in proper simulation. Reflections increase optical path length in predictable ways.

The size of the simulation problem is defined by the above considerations. The computational burden is defined by CPU time and memory space.

A wall clock time estimate is conveniently sent to the run-time output periodically, so you can tell how long any simulation is likely to take. These estimates are fairly accurate but are subject to many things that are out of our control. You will, however, be able to decide whether to abort a simulation based on such projections.

The only way to reduce the estimated completion time of a problem of a given size is to apply multiple processors or reduce the precision of the simulation. If you have multiple processors available, you can set the number of processors using the -P option. If you are running in a non-standard precision, you will see speed-up with standard precision.

The symptoms of excessive memory use are generally sluggish performance of the platform in question. Process monitors will show that memory is fully consumed and sometimes the platform will require rebooting. In such cases, you can specify NO.ENV on the BEAM statement to reduce the memory. This causes the envelope information to no-longer be kept. The envelope function is used to calculate the residual error. It is also output to structure files.

When you specify NO.ENV error residual is estimated by the change in intensity, which is saved instead of the envelopes. This saves roughly 50% on memory. The followng table gives a summary of memory usage estimates for various cases.

Table 11-5 Approximate Memory Usage for Various Options

	3D	2D
^NO.ENV(default)	NX*NY*NZ*P*24.5	NX*NY*P*8.5
NO.ENV	NX*NY*NZ*P*14.5	NX*NY*P*6.5

Here, P is precision. Standard precision is 64 bit or 8 bytes.

For the default mode in standard precision in 3D, you have about 200 bytes per mesh point. A 3D 1000×1000×1000 mesh occupies about 200 GB. This is probably about the limit of what is reasonable from current commonly available resources.

Assuming TD.SRATE=10 and an average index of 4, 1000 mesh points translates to about 25 wavelengths. For a wavelength of 0.5 microns, this is about 12.5 microns.

The good news is that problems larger than several wavelengths can usually be reduced since coherence effects do not usually (or should not) occur over distances of several wavelengths.

This also implies that large problems should sometimes be reduced by dimension or left to higher order methods (e.g., ray tracing). Also keep in mind, you may use hybrid optical sources described in Section 11.1.1 “Hybrid Source Specifications”.

As always, you are free to separate the full problem into smaller more manageable parts. For example in many cases, you can simulate the effects of a photonic crystal using FDTD over a small area and apply those effects into larger area devices without applying FDTD to the entire domain.

11.5.7 Anisotropic Index Materials and Liquid Crystals

Materials with anisotropic real index of refraction such as liquid crystals can be simulated using the finite difference time domain method in 3D. By nature, anisotropy is 3D and is only supported in Luminous 3D.

To permit anisotropic simulation, you will first need to specify anisotropy on the **BEAM** statement by enabling the ANISO flag. When running in anisotropic mode, you may expect slower run times and larger memory requirement.

To specify a region as anisotropic, you need to specify N.ANISO on the **MATERIAL** statement associated with that region and specify the indices along the principal axes and the rotation matrix. The parameters N.XX, N.YY, and N.ZZ are used to specify the indices along the principal crystalline axes and the Euler angles N.ALPHA, N.BETA, and N.GAMMA are specified also on the MATERIAL statement to give the angular difference between the crystalline and optical axis.

To simulate twisted-nematic liquid-crystal (TNLC) materials, we must allow for continuous spatial variation of the index tensor. Full flexibility in the definition of the index tensor as a funciton of position is very expensive from a memory allocation point of view. As such we limit the variation in index to occur only along one axis. The selection of which axis is specified by the N.VS.X, N.VS.Y, or N.VS.Z paramters of the **BEAM** statement. Keep in mind that the axes refered to by these parameters are relative to the beam referenced coordinate system. In this system, the light propogates in the positive Y direction.

Simulation of TNLCs can now be done by assigning a tensor index with a twist angle imparted as a function of distance through a region. This model is enabled by the TNLC parameter of the MODEL statement. The Twist angle is given by the relation:

$$\alpha = \text{A0.TNLC} + d*\text{TR.TNLC} \qquad 11\text{-}72$$

where d is the distance along the optical axis in microns specified by the N.VS.X, N.VS.Y, or N.VS.Z parameter of the **BEAM** statement. This distance is measured from the nearest extent of the region in the direction of the optical source. The parameters A0.TNLC and TR.TNLC are the initial angle in degrees and the twist rate in degrees per micron. The parameters A0.TNLC and TR.TNLC are specified on the **MODELS** statement.

Anisotropic PMLs can also be specified by first enabling anisotropic analysis in the **BEAM** statement by setting the ANISO flag. Once that is done, we can simply specify the principal indices using the NXX, NYY, and NZZ parameters of the PML statement.

You should be careful to make sure that the PML anisotropic indices match the adjacent material indices along the entire interface.

As an alternative to PMLs, we have provided a capability to specify a matching layer that smoothly varies the magnitude of the index tensor from for example an isotropic layer or PML to the anisotropic liquid crystal. To specify such a layer, you must first enable it by specifying GIAR on the **MODELS** statement. GIAR stands for graded-index anti-reflective [224]. The variation of real index of refraction is governed by the expression:

$$n = n1 + (n2 - n1)*(10 - 15*t + 6*t*t)*t*t*t \qquad 11\text{-}73$$

Here, t is the distance through the region as discussed above divided by the wavelength. $n1$ is the index at points nearest the optical origin, and $n2$ is the index at the points farthest from the optical origin.

The values of the extreme indices can be specified for an isotropic layer using the parameters N1.GIAR and N2.GIAR. For TNLC layer, you need to specify TNLC on the **MODELS** statement as well as the values for the extreme values for the ordinary and extraordinary indices using the parameters NO1.GIAR, NO2.GIAR, NE1.GIAR, and NE2.GIAR.

Currently, absorption is not supported in anisotropic simulations so the electrical simulation afforded by, for example drift diffusion, is only coincidental to the analysis of the optical properties using FDTD.

11.6 User-Defined Photogeneration

11.6.1 User-Defined Beams

Standard beam input syntax in Luminous and Luminous 3D allows specification of plane waves with Gaussian or flat-top (top-hat) irradiance profiles. Although many practical applications require only plane wave illumination, some optoelectronic devices rely on certain focusing or collimating optical systems that collect incident light. Some of these optical systems are implemented as lens arrays deposited on top of the device structure, which can be directly included in Luminous 3D. Still, a number of applications that require complex collimating or focusing optics cannot be approached in this way. To accommodate these applications, we enabled input of incident beams with complex wavefronts. Currently, the ray-tracing is not restricted to particular beam properties and allows specification of an arbitrary collection of rays as an input beam.

Theoretically, a number of ways to specify a beam of arbitrary shape exists. The one chosen in Atlas aims to allow you an easy interface with optical system design software. Thus, collimators or focusing optics designed with conventional optical system tools can be represented in Atlas by a ray bundle obtained on the output of the optics tool. Atlas treats the ray bundle specified in a file as one beam. You need to set the input file name on the **BEAM** statement as a value of INPUTRAYS parameter.

The file INPUTRAYS must have the following format.

```
NUMBER OF RAYS

NUMBER OF CHANGING PARAMETERS

PARAMETER1 PARAMETER2 PARAMETER3 ...

RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3
VALUE ...

RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3
VALUE ...

...             ...                  ...                    ...

RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3
VALUE ...
```

NUMBER OF RAYS stands for the total number of rays specified in the file. It should be the same as the last RAY NUMBER in the table.

NUMBER OF CHANGING PARAMETERS gives the total number of ray parameters given in the table. Most often, some parameters are the same for all the rays. For example, wavelength or ray power can be the same. In this case, there is no need to keep the table column for this constant parameter. Instead, you can set this parameter directly in the **BEAM** statement. NUMBER OF CHANGING PARAMETERS is equal to the number of columns in the table (not counting the first column of ray numbers).

PARAMETER1, PARAMETER2, PARAMETER3, and so on are numerical codes of parameters. Their order specifies the order of columns in the table. Numerical codes correspond to the following parameters.

- 1 - X coordinate of ray origin (in device coordinate system).
- 2 - Y coordinate of ray origin (in device coordinate system).

- 3 - Z coordinate of ray origin (in device coordinate system).
- 4 - ray angle phi in XY plane (0 - along X and 90 - along Y).
- 5 - ray angle theta in XZ plane (in 3D only, 0 - along X and 90 - along Z).
- 6 - ray polarization angle (relative to the plane of incidence, 0 - TE, and 90 - TM).
- 7 - ray relative power.
- 8 - ray wavelength.
- 9 - ray area (μm^2 in 3D) or thickness (μm in 2D).

The first column of the table where ray parameters are specified lists ray numbers (integer values) in ascending order. The first ray number should be 1. The last ray number should be equal to the NUMBER OF RAYS. Often, it is the case that ray parameters change continuously from ray to ray (at least in parts of the table). The INPUTRAYS file in Atlas allows a shorthand input for these parts. You can omit ray numbers where possible to allow for automatic fitting of parameter values for these rays.

Linear approximation is used for the omitted rays based on the values of parameters given in the table. This is better illustrated in the following example.

```
21
3
1 4 7
1   0.0    60.0   0.2
11  300.0  90.0   1.0
21  600.0  120.0  0.2
```

Here, NUMBER OF RAYS=21, NUMBER OF PARAMETERS=3, and the order of parameter columns: X coordinate (1), angle phi (4), and relative ray power (7).

The rays from 2 to 10 are specified using linear interpolation with end values given by rays 1 and 11. The rays from 12 to 20 are also defined similarly. This shorthand input is equivalent to the following.

```
21
3
1 4 7
1   0.0    60.0   0.2
2   30.0   63.0   0.28
3   60.0   66.0   0.36
4   90.0   69.0   0.44
5   120.0  72.0   0.52
6   150.0  75.0   0.6
7   180.0  78.0   0.68
8   210.0  81.0   0.76
9   240.0  84.0   0.84
```

```
10  270.0  87.0   0.92
11  300.0  90.0   1.0
12  330.0  93.0   0.92
13  360.0  96.0   0.84
14  390.0  99.0   0.76
15  420.0  102.0  0.68
16  450.0  105.0  0.6
17  480.0  108.0  0.52
18  510.0  111.0  0.44
19  540.0  114.0  0.36
20  570.0  117.0  0.28
21  600.0  120.0  0.2
```

When you use the INPUTRAYS file to define the beam, some of the parameters on the **BEAM** statements are disabled because they become incompatible with the user-defined beam. The parameters that are ignored in this case are CIRCULAR, ELLIPTICAL, GAUSSIAN, all LENS.* parameters, MAX.WINDOW, MIN.WINDOW, MEAN, RAYS, SIGMA, X.CENTER, X.GAUSSIAN, X.MEAN, X.RADIUS, X.SIGMA, X.SEMIAXIS, XMAX, XMIN, Y.SEMIAXIS, Z.CENTER, Z.GAUSSIAN, Z.MEAN, Z.RADIUS, Z.SIGMA, Z.SEMIAXIS, ZMAX, and ZMIN.

The X.ORIGIN, Y.ORIGIN, Z.ORIGIN, PHI, THETA, POLARIZE, REL.POWER, WAVELENGTH, RAYAREA parameters in the **BEAM** statement specify values for all rays if the corresponding parameters are not defined explicitly in the INPUTRAYS file. The parameters Z.ORIGIN and THETA and the corresponding parameters in INPUTRAYS are for Luminous 3D applications only. They are ignored in Luminous 2D.

RAYAREA is a parameter that specifies area in μm^2 (thickness in 2D in µm) of each ray in the **BEAM** statement. This parameter is used when the specified INPUTRAYS file does not contain ray area information.

One difficulty that arises when dealing with arbitrary beams has to do with specification of beam intensity (W/cm^2) in the **SOLVE** statement. For a general ray bundle, illuminating the device the intensity distribution is not uniform. Thus, intensity cannot be specified by a single number. There are two ways to avoid this problem. Instead of setting intensity for a ray bundle obtained on the output of an optical system (collimator or focusing system), you can go back to the original beam that produced this ray bundle. According to the intensity law of geometrical optics [36], you can define ray area (thickness) and overall beam intensity for that input beam. Alternatively, if power (W) carried by each ray is known, you can treat beam intensity parameter B<n> in the **SOLVE** statement as a scaling factor. Then, set all ray power values using the ray relative power parameter in the INPUTRAYS file (or REL.POWER on the BEAM statement if it is the same for all rays). In this case, you can treat RAYAREA as another scaling parameter for unit conversions. In any case, make sure you define the ray area because it's important for obtaining correct values of photogeneration rates. If you don't specify ray area, Atlas uses an estimate based of coordinate values of ray origins. You should not rely, however, on the accuracy of such an estimate.

You can use INPUTRAYS with POWER.FILE to define multi-spectral complex beams. In this case, Atlas ignores ray wavelength specified in INPUTRAYS and uses the wavelength values from POWER.FILE for each ray. Ray power is then a product of relative ray power from INPUTRAYS and spectral power density defined in POWER.FILE.

11.6.2 User-Defined Arbitrary Photogeneration

An option exists for you to define the photogeneration rate. A C-Interpreter function written into a text file can be supplied to the program using the F.RADIATE parameter of the **BEAM** statement. For example, if a file, myoptics.c, was developed using the template C-Interpreter functions supplied, it can be referenced by using:

```
BEAM NUM=1 F.RADIATE=myoptics.c
.
.
SOLVE B1=1.0
SOLVE B1=2.0
```

The file, myoptics.c, returns a time and position dependent photogeneration rate to the program. This returned value is multiplied at every node point by the value of B1. With this option, you override all other parameters of the **BEAM** statement and all the material refractive indices.

11.6.3 Exponential Photogeneration

Another option is available if distribution of photo-injected carriers is uniform, linear, or has an exponential dependence on depth. Use the **PHOTOGENERATE** statement to specify the desired dependence on depth along a line segment.

The X.ORIGIN, Y.ORIGIN and X.END, Y.END parameters set the coordinates of the beginning and the end of the line segment. The default values correspond to the top left and bottom left corners of the device. Other parameters CONSTANT, EXPONENT, FACTOR, LINEAR, and RADIAL specify the photogeneration rate according to the following formula:

$$G(l,r) = (\mathtt{CONSTANT} + \mathtt{LINEAR} \cdot l + \mathtt{FACTOR} exp(-\mathtt{EXPONENT} \cdot l)) exp\left(-\left(\frac{r^2}{\mathtt{RADIAL}^2}\right)\right) \quad \text{11-74}$$

where l is the distance along the line segment from the origin point and r is the radial distance from the line segment (Figure 11-13). **PHOTOGENERATE** statement ensures TMA compatibility with PHOTOGEN statement. See Section 22.44 "PHOTOGENERATE" for the list of TMA compatible parameter names.

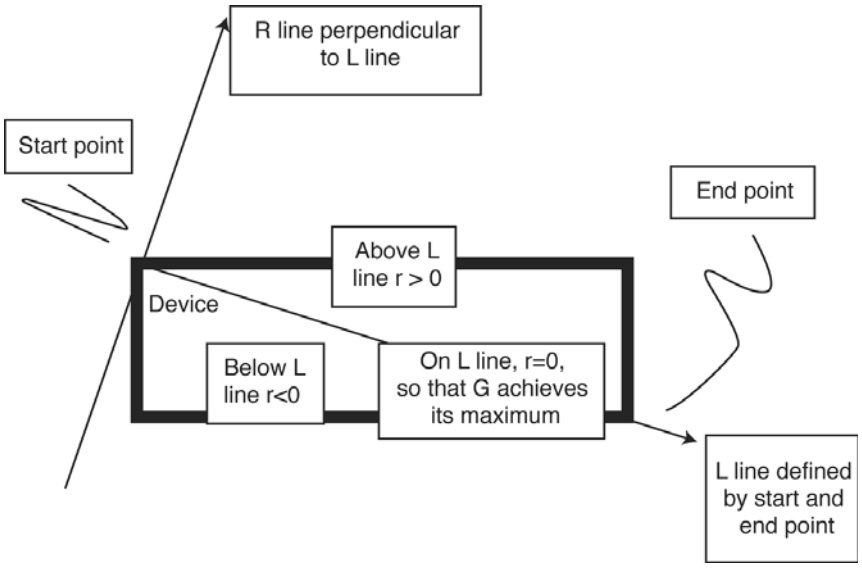


Figure 11-13: Specification of the Photogeneration Rate

11.6.4 Tabular Photogeneration (Luminous 2D only)

You may also specify the photogeneration rate as a function of location through a simple tabular interface. This is often useful to importing data from some other generic optical propagation tool. The values of photogeneration rate are contained in a file. You specify the file name on the FILE.PHOTOGEN parameter of the **BEAM** statement.

The file contains ASCII values of the photogeneration rate in units of per second per cubic cm. These samples are made on a regular mesh in X and Y rastered in increasing values of X and then Y (i.e., row - column order). The values are space separated in X (along each row) and separated by carriage returns in Y (between rows along each column).

The spacings in X and Y in microns are specified in the DX.PHOTOGEN and DY.PHOTOGEN parameters of the **BEAM** statement. You can specify the origin of the photogeneration in microns using the X0.PHOTOGEN and Y0.PHOTOGEN parameters of the **BEAM** statement. If you do not specify X0.PHOTOGEN and Y0.PHOTOGEN, then the origin will be assumed to coincide with the minimum X and Y coordinates in the device structure.

For example, the BEAM statement might read:

```
BEAM X0.PHOTOGEN=0 Y0.PHOTOGEN=0 DX.PHOTOGEN=1 DY.PHOTOGEN=1 \
     FILE.PHOTOGEN=generation.table
```

The contents of the file named "generation.table" might be simply:

```
1e22 1e22
1e20 1e20
```

In this case, the photogeneration rate is uniform in the X direction range of X=0 to X=1 micron but varries linearly between 1e22 to 1e20 in the Y direction.

11.7 Photocurrent and Quantum Efficiency

One of the important figures of merit of a photodetector is quantum efficiency. Here, quantum efficiency is defined as the ratio of the number of carriers detected at a given photodetector electrode divided by the number of incident photons on the detector. Ideally, this ratio should be 1.0 in detectors without any positive feedback mechanisms, such as avalanche gain. Luminous doesn't directly calculate quantum efficiency. It does, however, calculate two useful quantities printed to the run-time output and saved to the log file. These quantities are source photocurrent and available photocurrent, which can be viewed in TonyPlot from log files produced by Luminous.

Definition of Source Photocurrent

The source photocurrent for a monochromatic source is given in Equation 11-75. Here, B_n is the intensity of the beam number *n*, which is user-definable in the SOLVE statement. λ is the source wavelength specified by the WAVELENGTH parameter in the **BEAM** statement. $\hbar$ is Planck's constant. *c* is the speed of light. W_t is the width of the beam including the effects of clipping (see Section 11.2.1 "Ray Tracing in 2D"). This can be considered as a measure of the rate of photons incident on the device expressed as a current density.

$$I_s = q\frac{B_n\lambda}{hc}W_t \qquad 11\text{-}75$$

Definition of Available Photocurrent

The available photocurrent for a monochromatic source is given by Equation 11-76. Here, all the terms in front of the summation have the same definitions as for the source photocurrent. The sum is taken over the number of rays traced, N_R. W_R is the width associated with the ray. The integral is taken over the length, Y_i, associated with the ray. P_i accounts for the attenuation before the start of the ray due to non-unity transmission coefficients and absorption prior to the ray start. α_i is the absorption coefficient in the material that the ray is traversing.

The available photocurrent can be thought of as a measure of the photo absorption rate in the device expressed as a current density. This should be similar but somewhat less than the source photocurrent. The losses are due to reflection and transmission of light out of the device structure.

$$I_A = q\frac{B_n\lambda}{hc}\sum_{i=1}^{N_R} W_R\int_o^{Y_i} P_i\alpha_i e^{-\alpha_i y}dy \qquad 11\text{-}76$$

Depending how you define it, you can calculate quantum efficiency by dividing the current from one of the device electrodes by either the source photocurrent or the available photocurrent.

11.8 Defining Optical Properties of Materials

For ray tracing, the complex index of refraction of the various material regions in the structure must be specified. For many of the more common semiconductors and insulators, there are built-in tables of index versus wavelength. You can specify the index for the materials lacking reasonable default complex index of refraction.

Note: You can add the INDEX.CHECK parameter to any **SOLVE** statement to print out the refractive indices being used for that bias step. The indices are only printed when you perform the ray trace at the first reference to a given beam on the **SOLVE** statement.

11.8.1 Setting Single Values For The Refractive Index

The REAL.INDEX and IMAG.INDEX parameters of the **MATERIAL** statement can be used to set the real and imaginary indices of a specified material, region or regions. For example, the statement:

```
MATERIAL MATERIAL=Air REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all material regions composed of “Air”

The statement:

```
MATERIAL REGION=1 REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for region number 1.

The statement:

```
MATERIAL REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all regions.

11.8.2 Setting A Wavelength Dependent Refractive Index

The preceding examples set the complex index of refraction for a material regardless of wavelength. This is probably adequate for monochromatic sources. For multispectral simulations, the index of refraction should be modeled as having a dependence on wavelength. There are two ways to do this.

ASCII File Input

The first way is to specify index versus wavelength in a file. This is a text file that contains ordered triplets of wavelength, real index, and imaginary index. The first entry in the table is the number of samples. If you set the INDEX.FILE parameter of the **MATERIAL** statement to the name of the index file, the linear interpolation from this table will complete to obtain the index of refraction as a function of wavelength.

A valid index file is shown below.

```
2
0.5 2.0 0.0
0.6 3.0 0.02
```

In this example, a real index of 2.5 and an imaginary index of 0.015 would be used for a wavelength of 0.55 microns.

You may also use ASCII file inputs to input real index and complex index independently. You can assign INDX.REAL to the name of an index file containing ordered pairs representing the wavelength and real index of refraction. Similarly, you can assign INDX.IMAG to the name of an index file containing ordered pairs representing the wavelength and the imaginary index. In all cases the table is preceded by the integer number of samples.

In each case, you may also choose to imbed comment lines delimited by a "#" character at the beginning of the line that will not be processed when read in.

By default, the units of wavelength are microns and the index is represented by real and imaginary parts. You can change this interpretation by including two character codes as the first non-comment line in the file. The following table explains these interpretations.

Table 11-6 ASCII Index File Interpretation Codes.

Code	Interpretation
none	Wavelength in microns, real index, and imaginary index
"ae"	Energy in eV, real index, and absorption coefficient in cm^{-1}
"an"	Wavelength in nm, real index, and absorption coefficient in cm^{-1}
"am"	Wavelength in um, real index, and absorption coefficient in cm^{-1}
"nm"	Wavelength in nm, real index, and imaginary index
"ev"	Energy in eV, real index, and imaginary index

C-Interpreter Function

The second way to specify index versus wavelength is by using the C-Interpreter function. The syntax is:

```
MATERIAL NAME=Silicon F.INDEX=myindex.c
```

The file myindex.c is an external file conforming to the template supplied with the program. It returns wavelength dependent real and imaginary indices. Instructions for using the C-Interpreter and finding the template functions are described in Appendix A "C-Interpreter Functions".

SOPRA Database

You can choose to use refractive index data from the SOPRA database. To do this, you must specify the appropriate index file from Table B-40 on the SOPRA parameter of the **MATERIAL** statement (see Section B.13.1 "SOPRA Database").

Outputing Wavelength Dependent Complex Index for TonyPlot

The complex indices of refraction can be written to files suitable for display in TonyPlot by specifying the file name root without extension on the OUT.INDEX parameter of the **MATERIAL** statement. The files created will use this root appended by "n.log" for real indices and "k.log" for imaginary indices. For indices input by the SOPRA database, the root will be appended by "N-n.log" and "N-k.log" where N is a sequence number (e.g., 1, 2, 3 ...) to accommodate the capability of the SOPRA format to allow dependence on composition fraction and temperature.

The Absorption Edge

For various reasons, you may see absorption (non-zero imaginary index) at energies less than the bandgap. You can ensure that simulation use zero absorption below the band gap by specifying ABS.EDGE on the **BEAM** statement.

11.8.3 Quantum Well Absorption

The complex index of absorption can self-consistently include the effects of quantum confinement. To enable this model, first you should specify the QWELL parameter on the **REGION** statements of regions where you want to consider quantum confinement. This will simply force the calculation of bound state energies in the region.

Second, you need to specify ABS on the same **REGION** statements to inform the simulator that you want to use the confined state energies in the calculation of the complex index or refraction.

Next, you can either use the confined "bandgap" represented by the difference between the lowestconduction band bound state energy and the highest valence band bound state energy in the calculation of complex index of refraction using the C interpreter function F.NKEG. This is specified by the F.NKEG parameter on the **MATERIAL** statement.

If you do not specify the F.NKEG parameter, the model will use the negative of the gain function for the quantum well. In this case, you should also specify the gain proper gain function parameters on the **MODELS** statement (see Section 3.9 "Optoelectronic Models").

11.9 Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method

In some cases, it is desirable to add optical elements that are considered outside of the device simulation domain. In Luminous, we provide such a capability for adding various lenslets that are accounted in the simulation of light propogation but have no direct interaction with the device simulation phases.

You can define spherical, ellipsoidal, composite, aspheric, pyramidal, random textured, or user definable surfaces. The surfaces may be optionally coated with a single layer of uniform composition and thickness. You can specify periodicity in any of the fundimental shapes to create photonic crystals.

You can visualize these shapes by defining OUTFILE on the **BEAM** statement using the SUBNX, SUBNY, SUBNZ parameters to define the resolution in which you want to visualize these continuous forms.

Luminous enables you to model and design complicated AR coatings optimized for your specific application. AR coatings are simulated in Luminous using an efficient transfer matrix method. This method enables you to deal with multilayer AR coatings with virtually no overhead in computation time. Section 11.3 “Matrix Method” discusses matrix method and its application to wave propagation in a stratified medium. AR coatings can be modeled using this approach for Matrix Method analysis (see Section 11.3 “Matrix Method”) or ray tracing (see Section 11.2 “Ray Tracing”).

Typically, AR coatings affect only optical properties of the device. Therefore, specifying the coating as a part of the device structure will unnecessarily increase the complexity of the electrical simulation. In Atlas, coatings are associated with the interfaces of the device instead. This allows you to specify certain optical properties of any material boundary while not affecting electrical properties of the structure.

To define optical properties of a coating associated with that interface, use the **INTERFACE** statement and include the OPTICAL parameter. Using this statement, you can define an AR coating, a dielectric mirror, or an ideally reflecting surface. Note that the coating is absent in the structure defined for Atlas.

The AR.THICK parameter defines the thickness of the coating layer. The AR.INDEX parameter defines the refractive index of the layer. For a single-layer coating or for the first layer of a multilayer coating, specify the coordinates of the points defining the coated surface P1.X, P1.Y, P2.X, and P2.Y.

Note: These coordinates should define a line, not a rectangular box of thickness (AR.THICK).

Figure 11-14 shows an example of a single-layer coating. The following shows the syntax used for a single-layer coating.

```
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.0634 P1.X=0.0 P1.Y=0.0
P2.X=10.0 P2.Y=0.0
```

This defines a 63.4µm layer of real refractive index 2.05 at Y=0.0 µm in a structure.

For a single-layer coating or for the first layer of a multilayer coating, you can also specify coordinates of the bounding box XMIN, XMAX, YMIN, YMAX. All interfaces within the bounding box obtain optical properties of the coating. If you don't set the bounding box coordinates, the default bounding box containing the whole device will be used.

The REGION parameter gives you additional flexibility in the description of an AR coating. REGION specifies the number of a region adjacent to the coating. REGION=0 is the default value, corresponding to the ambient region outside the device boundaries.

When you define a bounding box for an interface, it could contain other interfaces, which you do not intend to be AR coated. When you use REGION, you can select only those interfaces adjacent to that region to have AR coating properties.

You can specify any number of coatings for each device. The COATING parameter in the **INTERFACE** statement refers to the number of the coating. If COATING parameter is not set, the first coating will be used. Coatings should be specified in order (i.e., COATING=3 cannot be set before COATING=2 is defined).

Different coatings should not overlap. If this occurs, the later coating will be used in the overlapping part.

Typically, the purpose of a single-layer AR coating is to minimize reflectivity of normally incident monochromatic light at the design wavelength λ.

Equations 11-77 and 11-78 define the parameters of choice for a single-layer AR coating between materials with refractive indexes n_1 and n_2:

$$\mathrm{AR.INDEX} = \sqrt{n_1 n_2} \qquad 11\text{-}77$$

$$\mathrm{AR.THICK} = \frac{\lambda}{4 \cdot \mathrm{AR.INDEX}} \qquad 11\text{-}78$$

According to Equations 11-77 and 11-78, the coating in the example considered above has optimal properties for light at 520μm normally incident from air on a silicon detector.

You need several **INTERFACE** statements to specify a coating composed of multiple layers. Each statement defines only one layer of a coating. The syntax of the first **INTERFACE** statement for a multilayer coating is identical to the specification of a single-layer coating. Each subsequent layer must have the number of the coating and the number of the layer specified by COATING and LAYER parameters. Specify the same COATING parameter for all **INTERFACE** statements that refer to the layers of the same AR coating. For example, the following statements specify a two-layer coating:

```
INTERFACE  OPTICAL  AR.INDEX=1.5  AR.THICK=0.06  P1.X=0.0  P1.Y=0.0
P2.X=10.0 \
                    P2.Y=0.0 REGION =2

INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.06 COATING=1 LAYER=2
```

Here, COATING=1 and LAYER=2 show the second **INTERFACE** statement describing a second layer of a two-layer coating number one. Use the LAYER parameter to describe the order of layers in a coating. If LAYER is not specified, the first (top) layer will be used. Again, you cannot specify LAYER=3 before LAYER=2. You only need to specify the coordinates of the interface for the first layer of each coating.

In this example, the first coating layer with refractive index 1.5 is adjacent to the region number 2 on all boundaries within the specified bounding box. The second layer is outside of the first layer relative to region number 2, along the same boundaries. You only need to specify the bounding box and REGION only for the first layer of each coating.

Note: You also should also the COATING parameter to a unique index for each device in a MixedMode simulation.

If a coating is made out of an absorbing material, you can use AR.ABSORB parameter to take into account absorption. You can also use totally reflective coatings. The coating will behave as an ideal reflector if you set the AR.INDEX parameter to a value > 1000.

Another way to specify the refractive index of the coating is to use the MATERIAL parameter on the **INTERFACE** statement. This enables you to apply any refractive index model supported by **MATERIAL** statement in Atlas. This includes default wavelength dependent refractive index for supported materials, user-specified index in INDEX.FILE, C-interpreter function in F.INDEX, or the REAL.INDEX and IMAG.INDEX parameters on the corresponding **MATERIAL** statement. When you use the **MATERIAL** parameter on the **INTERFACE** statement, the AR.INDEX and AR.ABSORB parameters will be disabled.

An alternative way to set the properties of the top interface is to use C-Interpreter function with F.REFLECT specified in the **BEAM** statement. Using this function, you can specify the reflection coefficient, angle of transmission, and transmitted polarization as a function of position, wavelength, angle of incidence, and incident polarization.

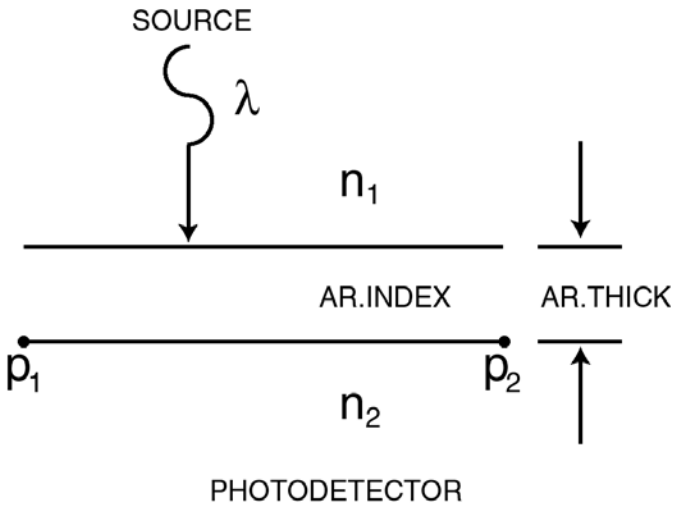


Figure 11-14: Single Layer AR Coating Under Normal Incidence

11.9.1 Anti-Reflective Coatings in Luminous 3D

Luminous 3D enables you to model and design multilayer AR coatings optimized for your specific 3D application.

As in Luminous, you can define properties of an AR coating using the **INTERFACE** statement with OPTICAL parameter in Luminous 3D. The AR.THICK, AR.INDEX, AR.ABSORB, LAYER, REGION, and COATING parameters retain their meaning and functionality in 3D. The only difference in description of an AR coating in 3D pertains to specification of the coating location. The coating location cannot be set by specifying point coordinates. For a single-layer 3D coating or for the first layer of a multilayer 3D coating, you need to specify coordinates of the bounding box XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX.

The following example

```
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.0634 REGION=0
```

specifies that a 63.4nm layer of real refractive index 2.05 covers the entire outside boundary of a 3D device. All possible internal boundaries (heterojunctions) are not coated.

In a multilayer coating, the first layer is assumed to be adjacent to the region specified by REGION.

```
INTERFACE OPTICAL AR.INDEX=1.5 AR.THICK=0.06 XMIN=0.0 XMAX=2.0
YMIN=0.0 \
          YMAX=1.0 ZMIN=0.0 ZMAX=6.0 REGION=2

INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.06 COATING=1 LAYER=2
```

Here, COATING=1 and LAYER=2 show the second **INTERFACE** statement describing a second layer of a two-layer coating number one. Use the LAYER parameter to describe the order of layers in a coating. If LAYER is not specified, the first (top) layer will be used. Again, you cannot specify LAYER=3 before LAYER=2. You only need to specify the coordinates of the interface for the first layer of each coating.

In this example, the first coating layer with refractive index 1.5 is adjacent to the region number 2 on all boundaries within the specified bounding box. The second layer is outside of the first layer relative to region number 2, along the same boundaries. You only need to specify the bounding box and REGION only for the first layer of each coating.

You can also specify the material composition of the AR layer by assigning the AR.MATERIAL parameter of the **INTERFACE** statement to any Atlas material. You can then use the **MATERIAL** statement to define the layer refraction characteristics. See also Section 11.9 “Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method”.

11.10 Specifying Lenslets, Texturing, and Photonic Crystals

You can specify one of four lenslets to focus the light into the device. The lenslet has no direct effect on the device simulation mesh. It is only used in the ray trace. Lenslets are described by user defined parameters of the **LENS** statement. All lenslet specifications apply to the last previously specified beam.

Lenslets can be used in either ray tracing or FDTD. If you want to visualize lenslets used in ray tracing, you will need to assign the OUTFILE parameter of the **BEAM** statement. The size of the output file is controlled by the SUBNX, SUBNY, and SUBNZ parameters.

Spherical Lenslets

The spherical lenslet is described by the intersection of a plane (perpendicular to the Y axis and a sphere). See Figure 11-15. The Y coordinate of the plane is given by PLANE. The origin of the sphere is described by X.LOC, Y.LOC, and Z.LOC. The sphere radius is described by RADIUS.

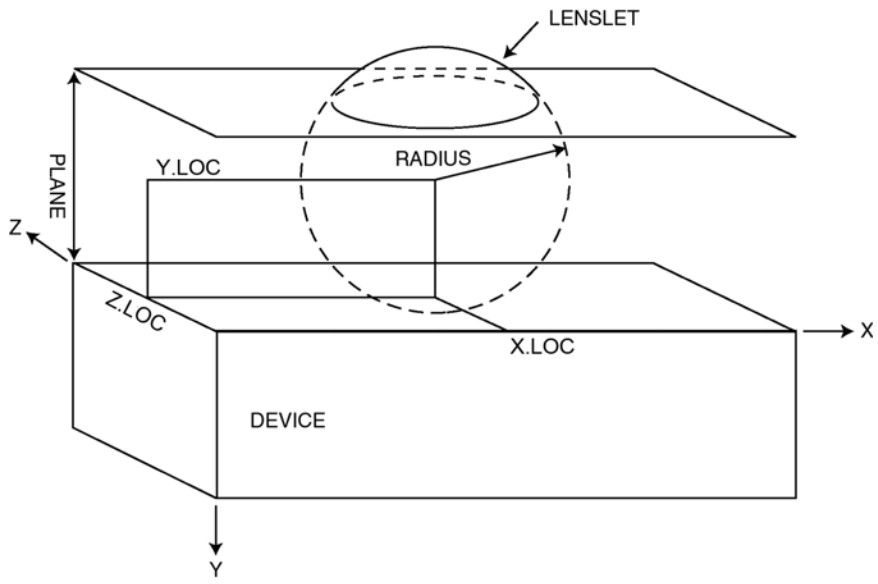


Figure 11-15: Luminous 3D Lenslet Specification

For ray tracing purposes, the space between the plane and the device surface is uniformly of a single index of refraction specified by INDEX. The spherical section projecting above the plane in the negative Y direction is also considered to consist of the same index of refraction. The plane and the spherical section form the lenslet. To enable a spherical lens, you should specify SPHER on the **LENS** statement.

Ellipsoidal Lenslets

To specify an ellipsoidal lenslet, define the axial half lengths using the X.SEMI, Y.SEMI and Z.SEMI parameters of the **LENS** statement. The ellipsoidal lenslet center is specified by the X.LOC, Y.LOC, and Z.LOC parameters. The PLANE parameter defines the location of the lens plane. The INDEX parameter gives the lens real index of refraction. Equation 11-79 gives the lenslet surface above the lens plane.

$$\frac{(x - \mathrm{X.LOC})^2}{\mathrm{X.SEMI}^2} + \frac{(y - \mathrm{Y.LOC})^2}{\mathrm{Y.SEMI}^2} + \frac{(z - \mathrm{Z.LOC})^2}{\mathrm{Z.SEMI}^2} = 1 \qquad 11\text{-}79$$

To enable the ellipsoidal lens, you need to specify ELLIP on the **LENS** statement.

Composite Lenslets

You can also specify a composite lenslet. The composite lenslet is composed of a central planar section, four cylinder sections and four sphere sections, which in shown in Figure 11-16. The composite lenslet is defined by the XMIN, XMAX, ZMIN, ZMAX, WIDTH, HEIGHT, PLANE and INDEX parameters. Figure 11-16 shows the meanings of these parameters.

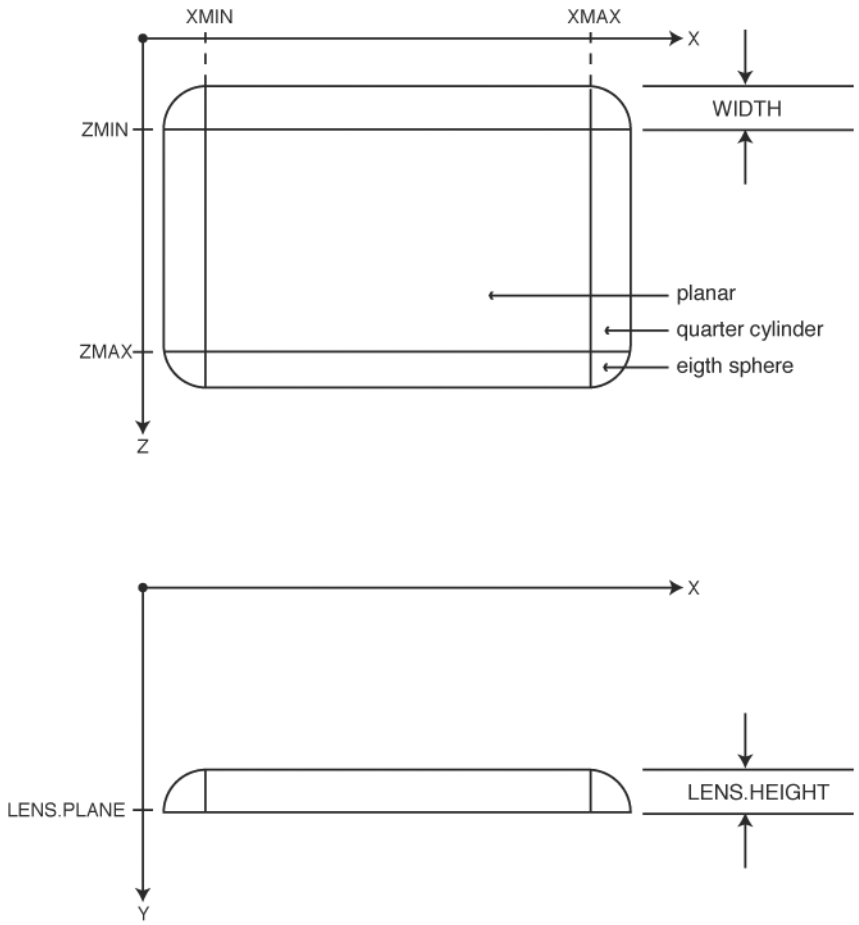


Figure 11-16: Composite Lenslet

To enable the composite lens, you need to specify COMPO on the **LENS** statement.

Aspheric Lenslets

Aspheric lenslets are characterized in the normal directions x and z by the following two equations:

$$Y_x = \frac{X^2}{R_x + (R_x^2 - X^2)^{1/2}} + C_{4_x}X^4 + C_{6_x}X^6 + C_{8_x}X^8 + C_{10_x}X^{10} \quad \text{11-80}$$

$$Y_z = \frac{Z^2}{R_z + (R_Z^2 - Z^2)^{1/2}} + C_{4_z}Z^4 + C_{6_z}Z^6 + C_{8_z}Z^8 + C_{10_z}Z^{10} \quad \text{11-81}$$

To enable the aspheric lens, you need to specify ASPHER on the **LENS** statement.

To arrive at these equations, you must provide tabulated data in the form of Sag-h pairs describing the surface of the lenslet in the *x* and the *Z* direction. You should specify the name of the file containing the data for the *X* direction in the X.SAGS parameter in the **BEAM** statement. You should also specify the name of the file containing the data for the *Z* direction in the Z.SAGS parameter in the **LENS** statement.

These files should first contain one line with the integer number of samples. The file should then contain a number (equal to the number of samples) of additional lines each containing a pair of real numbers representing the ordered pair of Sag-h data. These pairs must be arranged by increasing *h*. A sample at *h=0* is implicit and corresponds to *Sag=1*.

From this data, a Levenberg-Marquardt non-linear least squares algorithm is applied to obtain the constants *r*, c_4, c_6, c_8, and c_{10}. This algorithm is iterative. You can specify the number of iterations used in the SAGITS parameter on the **LENS** statement with a default value of 10.

To have the results of the least squares fit written to files to be displayed in TonyPlot, specify the file names using the X.SOUT and Z.SOUT parameters of the **LENS** statement.

Other parameters on the **LENS** statement describing the aspheric lenslet that should be specified include: X.LOC, Z.LOC, INDEX, and PLANE.

Pyramid Lenslets

To enable a pyramid shaped lenslet, specify PYRAMID on the **LENS** statement. The remainder of the specification is described in Figure 11-17. The level of the lenslet is specified by the PLANE parameter. The perimeter is specified by the X.MIN, X.MAX, Z.MIN, and Z.MAX parameters. The peak of the pyramid is located by the X.LOC and Z.LOC parameters. The height of the pyramid is given by the APEX parameter.

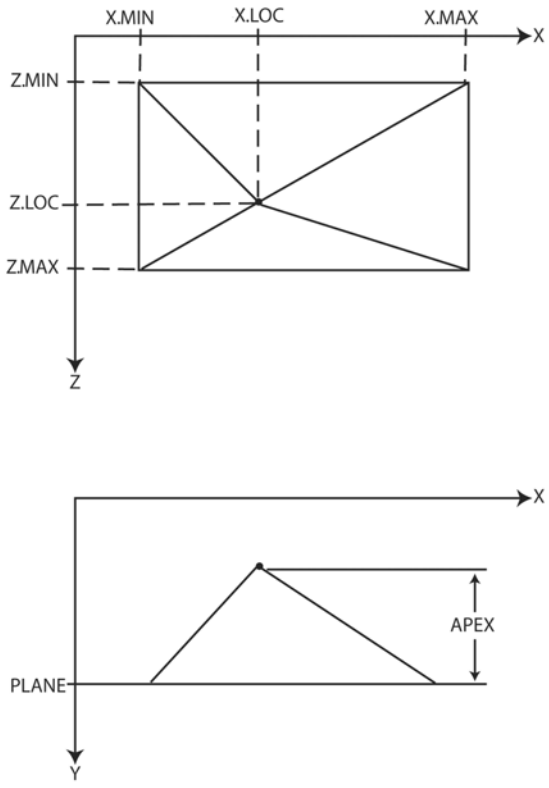


Figure 11-17: Pyramid Lenslet

Random Textured Lenslets

Random textured lenslets provide the capability to simulate random textured surfaces. The texturing can be best described by its algorithm.

First, a set of random steps are generated across the device domain in the Z direction. These random steps start at the minimum Z coordinate and progress until the maximum Z coordinate is reached. Each random step is made by first generating a Gaussian distributed random number with mean given by the value of the Z.MEAN parameter and a standard deviation given by the Z.SIGMA parameter of the **LENS** statement. This random step size is added to the previous Z location to yield the next Z location. Step sizes are restricted to positive numbers. This process is repeated until the Z location reaches the maximum Z coordinate.

Next, for each Z location the same process is applied in the X direction, where the mean step size is given by the X.MEAN parameter and the standard deviviation X.SIGMA. This being done, we now have a complete set of X, Z locations. At each X, Z location, we next generate a Gaussian distributed random number. This number has a mean given by the value of the PLANE parameter and a standard devication given by the Y.SIGMA parameter of the **LENS** statement. This random number gives the locations of the lenslet surface in the Y direction.

For convenience, the parameter MEAN can be used to specify a single value of both X.MEAN and Z.MEAN. Similarly, the parameter SIGMA can be used to specify a single value for X.SIGMA, Y.SIGMA, and Z.SIGMA.

Figure 11-18 shows an example surface generated using a value of MEAN=0.1 and SIGMA=0.1.

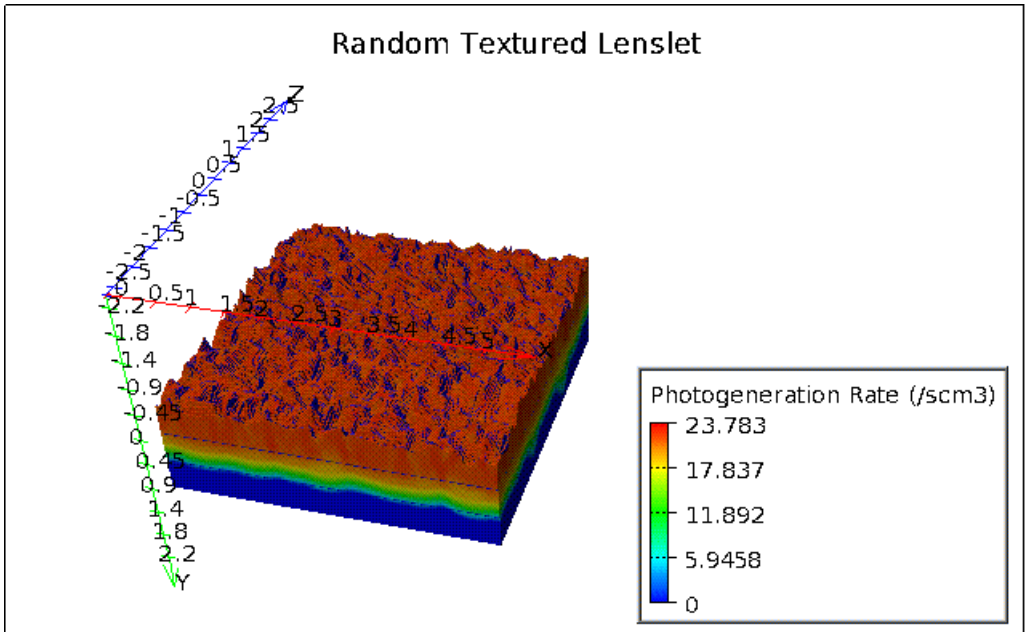


Figure 11-18: An Example Random Textured Lenslet (MEAN=0.1, SIGMA=0.1)

You can create a random textured lenslet by specifying the RANDOM parameter on the LENS statement. You can vary the seed value of the random number generation using the SEED parameter of the **LENS** statement. For a fixed value of the SEED parameter, the same random surface will be generated.

The random lenslet can also be used for 2D ray tracing.

User-Definable Lenslets

User-definable lenslets allow you to define the lenslet surface in a C-Interpreter function (see Appendix A "C-Interpreter Functions"). The function has a simple interface where given the X and Z coordinates, you define the y coordinate of the lens surface. To enable the user-definable lenslet, you need to specify the USER parameter of the **LENS** statement. You also need to assign the F.LENS parameter of the **LENS** statement to the name of the file containing the C interpreter function. Finally, you will need to specify the index of refraction of the lenslet using the N and K parameters or the MATERIAL parameter of the **LENS** statement.

Lenslet Anti-reflective Coatings

It is sometimes useful to specify anti-reflective coatings on lenslets. For FDTD, this is simple to accomplish by nesting two lenslet specifications a small distance apart. For ray tracing, this is ineffectual since ray tracing does not directly account for the coherence effects necessary for proper modeling of reflective properties for thin coatings. In these cases, you should use the lenslet built-in AR coating capability. This works much like the AR coating capability described in Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method".

To specify an AR coating on a lenslet for ray tracing, you need to specify the AR.INDEX and AR.THICKNESS parameters of the **LENS** statement. The AR.INDEX parameter is set to the real index of refraction of the coating. The AR.THICKNESS parameter is set to the coating thickness in microns.

Photonic Crystals

You can define photonic crystals by defining periodic repititions of the primitive lens types described above by specifying the periodicity in the X and Z directions. The parameters PC.DX and PC.DZ specify the spacing between duplications is the X and Z directions. These parameters are in microns.

When simulating photonic crystals, make sure that you size the device domain to an integer number of PC.DX and PC.DZ in the X and Z directions respectively. Also, you should specify PERIODIC on the **BEAM** statement to simulate a semi-infinite array.

In the next section, we will describe a practical example of setting up a hexagonal photonic crystal.

Hexagonal Photonic Crystals (A Practical Example)

We thought it would be instructive to describe how to specify a hexagonally symmetric photonic crystal as are often described in literature.

In this discussion, consider a simple equal sided hexagonal array of equal sized cylindrical elements. First, we look at the symmetries involved to determine the lens description which only allows rectangular arrays.

The simplest solution is that we must use six **LENS** statements one for each of the vertices of the primitive hexagonal cell. Further inspections show that a minumum of four lenses are needed to describe the hexagonal mesh as described in Figure 11-19.

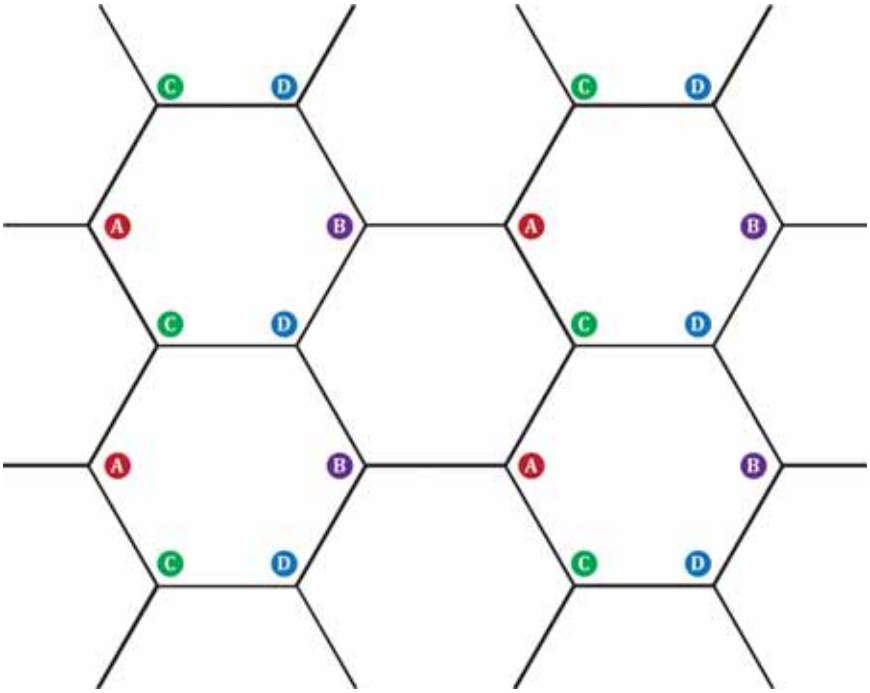


Figure 11-19: Hexagonal points of translational symmetry

The hexagonal mesh could just as well be represented using six lenses (one for each vertex on the primitive). The software will not misinterpret the duplicated points, but the lesson here is that you should always first consider the periodicities involved in whatever mesh is being analyzed. It is usually best to sketch the situation as in Figure 11-19.

Next, if you desire to use periodic boundaries, you need to consider the planes of symmetry in the mesh. Ideally, your (symmetric) device boundaries need to coincide with planes of symmetry in the mesh. For this example, the mesh planes of symmetry are labeled in Figure 11-20.

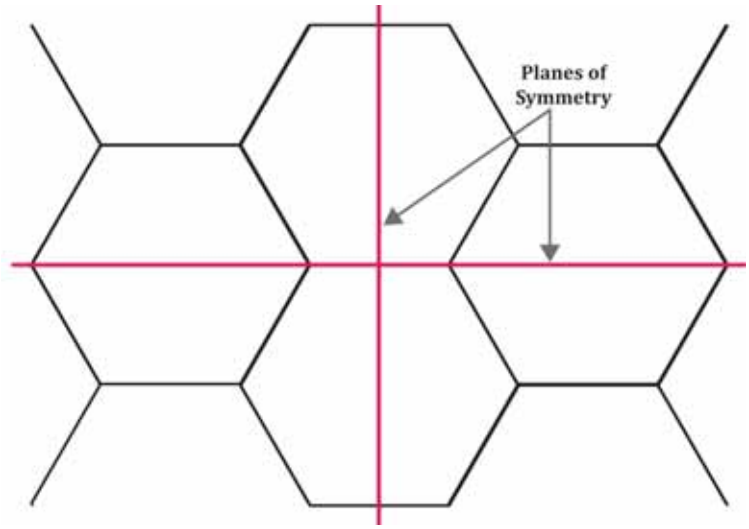


Figure 11-20: Hexagonal mesh planes of symmetry

With these considerations in hand, we are now prepared to look at the case of equal sided hexagons with a side length of 1 micron with cylindrical primitives at each vertex. Figure 11-21 is useful to construct the primitive lens center points PC.X0 and PC.Z0 and the translational periods PC.DX and PC.DZ.

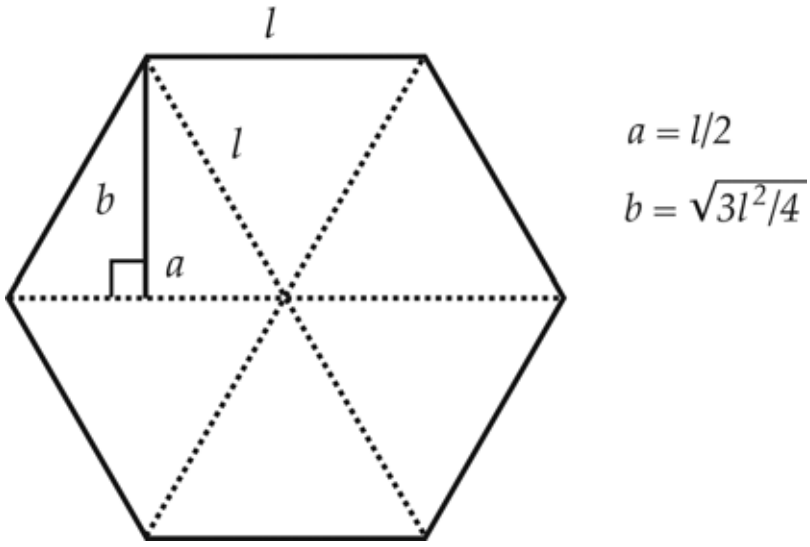


Figure 11-21: Hexagonal unit cell constructions

In this case, the values *a* and *b* are 0.5 and 0.866.

The following lines describe such a photonic crystal:

```
LENS CYLINDER PLANE=0.0 RADIUS=0.2 INDEX=1.5 HEIGHT=0.5 \
        PC.X0=-1.0 PC.DX=3.0 PC.Z0=0.0 PC.DZ=1.732
LENS CYLINDER PLANE=0.0 RADIUS=0.2 INDEX=1.5 HEIGHT=0.5 \
        PC.X0=1.0 PC.DX=3.0 PC.Z0=0.0 PC.DZ=1.732
LENS CYLINDER PLANE=0.0 RADIUS=0.2 INDEX=1.5 HEIGHT=0.5 \
        PC.X0=-0.5 PC.DX=3.0 PC.Z0=-0.866 PC.DZ=1.732
LENS CYLINDER PLANE=0.0 RADIUS=0.2 INDEX=1.5 HEIGHT=0.5 \
        PC.X0=0.5 PC.DX=3.0 PC.Z0=-0.866 PC.DZ=1.732
```

Here, we centered the primitives at X=0 and Z=0. You should be able to understand the coordinate selections by lookinging at Figure 11-21.

We defined the X and Z mesh limits as follows considering the planes of symmetry as described in Figure 11-20.

```
X.MESH LOC=-4.5 SPAC=0.5
X.MESH LOC=4.5  SPAC=0.5


Z.MESH LOC=-4.33 SPAC=0.5
Z.MESH LOC=4.33  SPAC=0.5
```

The resulting structure is shown in Figure 11-22.

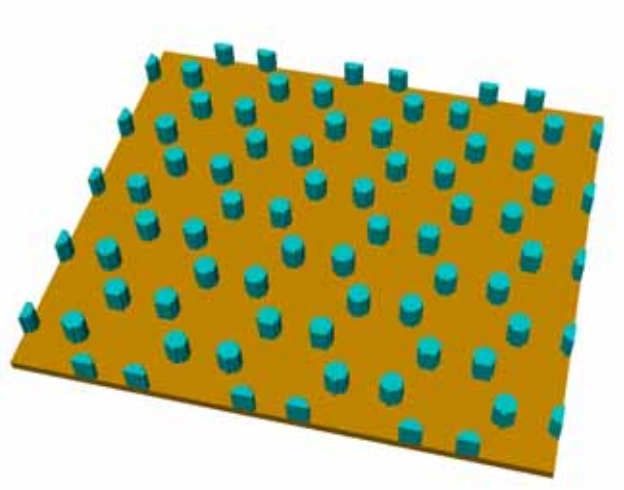


Figure 11-22: Hexagonal photonic crystal structure

11.11 Frequency Conversion Materials (2D Only)

Frequency conversion materials have the property that light absorbed in the material can be re-emitted at a single wavelength. This property can be used to either up or down convert light to improve spectral response of photodetectors, for example in solar cells.

In Luminous, these materials are modeled electrically as insulators. The light conversion capability is enabled by specifying FREQ.CONV on the **MODELS** statement.

The re-emission process is modeled at discrete locations in the frequency conversion material. These locations are described by even sampling in the X and Y directions. You can specify the numbers of samples in the respective directions using the EMISS.NX and EMISS.NY parameters of the **MATERIAL** statement. The sampling of emission centers is described in Figure 11-23.

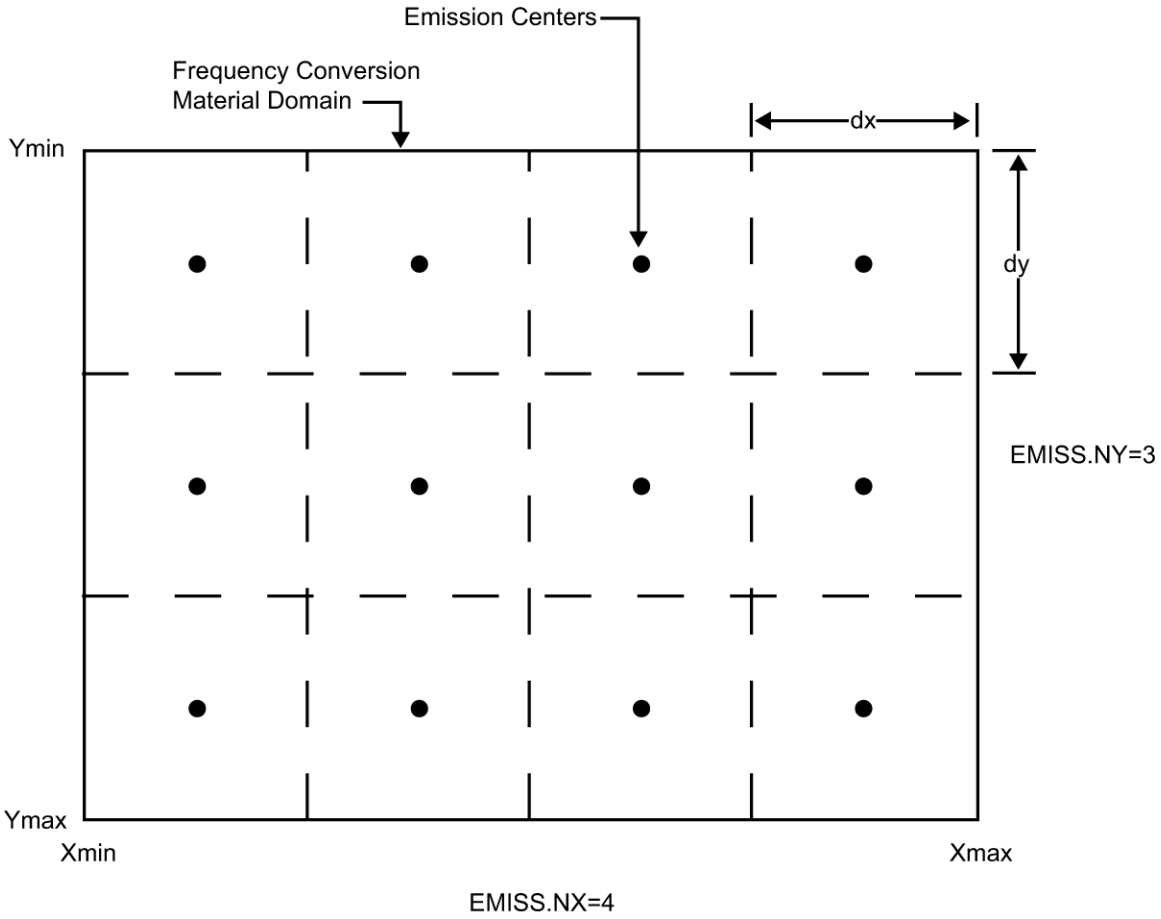


Figure 11-23: Frequency Conversion Material Emission Center Sampling

Physically, the remission process occurs in random directions. This process is modeled by performing ray traces from the emission centers in all directions 0 to 360° over an even sampling. You can specify the number of samples in angle space by the parameter EMISS.NANG on the **MATERIAL** statement. The angular sampling is described in Figure 11-24.

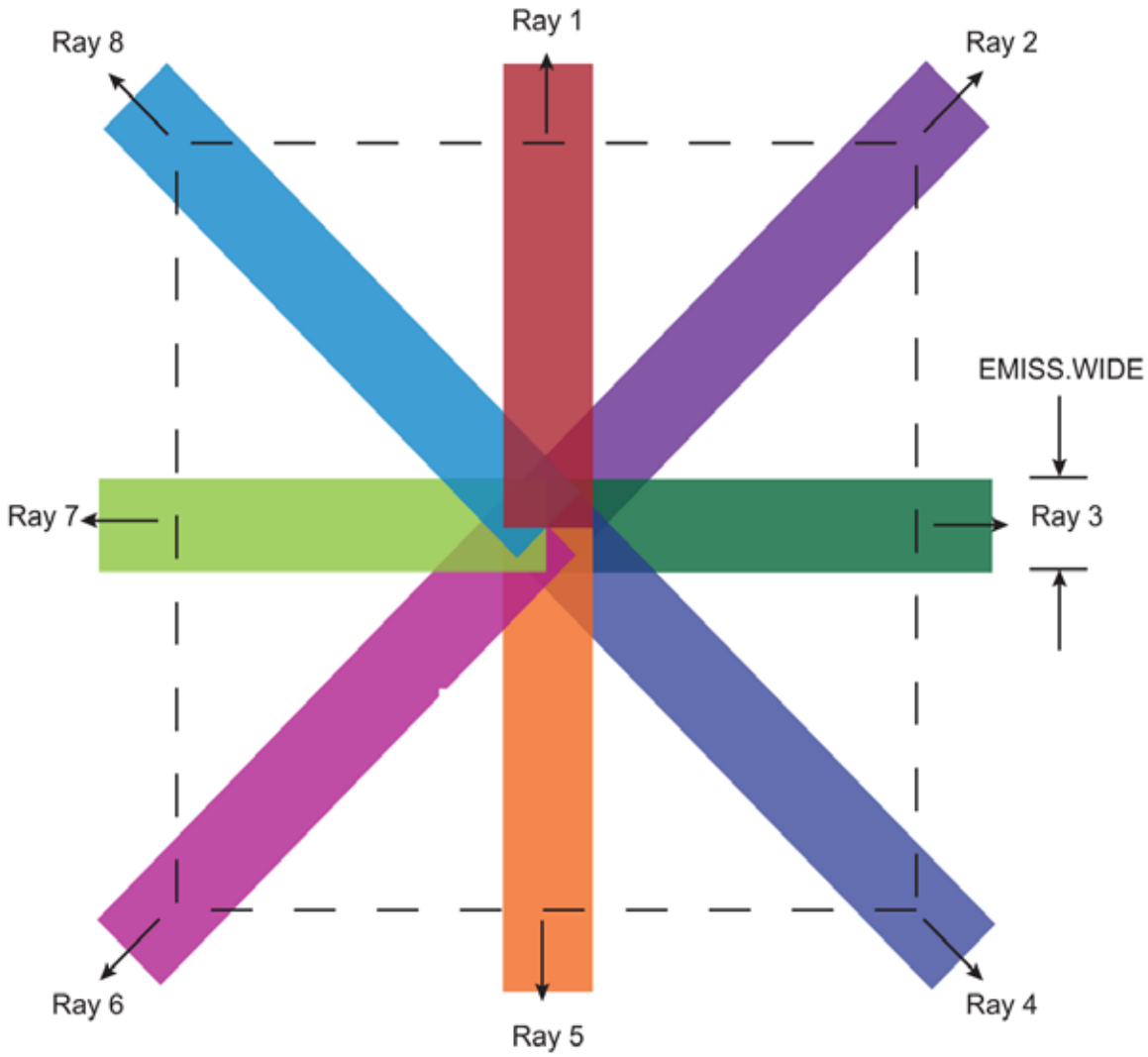


Figure 11-24: Frequency Conversion Material Angular Sampling

The rays are traced using the 2D trapezoidal ray tracing described in Section 11.2 “Ray Tracing”. The ray tracing automatically resolves internal and surface topologies. You can specify that the device domain will be treated as periodic with respect to the re-emitted rays by specifying PFREQ.CONV on the **MODELS** statement. By default, the value of PFREQ.CONV is TRUE, so the periodicity can be turned off by specifying ^PFREQ.CONV on the **MODELS** statement.

You can specify the width of each trapezoid as described in Figure 11-24 using the EMISS.WIDE parameter of the **MATERIAL** statement. The width of the trapezoids is limited, however, by the minimum of the spacing between emission centers (the minimum of *dx* and *dy* in Figure 11-23).

The rays are emitted in monochromatic light at a wavelength specified by the EMISS.LAMB parameter of the **MATERIAL** statement.

The intensity of each ray is calculated by integrating the photogeneration rate due to absorption of primary and re-emitted light in the frequency conversion material. The emission rate is given by Equation 11-82.

$$P_{ray} = \int_{\substack{\text{Emission}\\\text{Center}\\\text{Area}}} G(x,y)dA \frac{hc}{\texttt{EMISS.LAMB}} \frac{\texttt{EMISS.EFFI}}{\texttt{EMISS.LAMB} \cdot \texttt{EMISS.NANG}} \quad \text{11-82}$$

where:

- P_{ray} is the initial intensity of each ray.
- $G(x,y)$ is the photogeneration due to absorption in the area of the emission center.
- EMISS.LAMB is the user-specified emission wavelength.
- EMISS.WIDE is the user-specified emission ray width.
- EMISS.NANG is the user specified number of angles for emission.
- EMISS.EFFI is the user specified emission efficiency (number of photons emitted per electron hole pair generated by absorption).

The emission rate per ray is recalculated iteratively since the generation rate includes the generation due to re-emitted rays. Convergence of this iterative sequence is controlled by the FC.ACCURACY parameter of the **MODELS** statement, which specifies the maximum change in emission intensity as a divided by the most recently calculated emission intensity.

The user-controllable parameters of the frequency conversion model are given in Table 11-7.

Table 11-7 Frequency Conversion Material Parameter Summary

Parameter	Statement	Default	Units
FREQ.CONV	**MODELS**	False	
PFREQ.CONV	**MODELS**	True	
FC.ACCURACY	**MODELS**	1.0e-6	
EMISS.LAMB	**MATERIAL**	0.5	microns
EMISS.EFFI	**MATERIAL**	1.0	
EMISS.NX	**MATERIAL**	1	
EMISS.NY	**MATERIAL**	1	
EMISS.NANG	**MATERIAL**	1	
EMISS.WIDE	**MATERIAL**	1.0	microns

11.12 Simulating Photodetectors

This section describes techniques simulate photodetectors. This section applies to the simulation of any of the following devices: p-n and p-i-n photodiodes, avalanche photodiodes, Schottky photodetectors, CCDs, MSMs, photoconductors, optical FETs and optically triggered power devices.

11.12.1 Defining Optical Sources

Identifying an Optical Beam

You can define up to ten optical sources. Optical sources are described by using the **BEAM** statement. All **BEAM** statements must appear somewhere after the **MESH**, **REGION**, **DOPING**, and **ELECTRODE** statements and before any **SOLVE** statement. The parameters in the **BEAM** statement describe a single optical source.

The NUM parameter is used to uniquely identify one optical source. Values between 1 and 10 are valid for the NUM parameter. Optical sources are subsequently referred to by the value of their NUM parameter. The power of the optical beam is set by using the B<**n**> parameter of the **SOLVE** statement, where n is the beam number defined by NUM.

Origin Plane of the Beam for 2D Optical Sources

To specify the origin of the optical source, use the X.ORIGIN and Y.ORIGIN parameters. These parameters describe the origin of the optical beam relative to the device coordinate system. Currently, it is required that the origin lie outside any device region. The ANGLE parameter specifies the angle of the direction of propagation of the beam with respect to the device coordinate system. ANGLE=90 specifies vertical (normal) illumination from above.

The width of the optical beam is specified using the MIN.WINDOW and MAX.WINDOW parameters. These parameters specify the limits of the source beam relative to the beam origin. If you omit either of the limits, that limit will be clipped to the edge of the device domain.

Note: It's extremely important that no section of the origin plane of the beam intersects or is inside the simulation grid. Otherwise, you'll get incorrect results. This is important to check in cases where the ANGLE isn't 90^{o} or 270^{o}.

Origin Plane of the Beam for 3D Optical Sources

To specify the origin of a 3D source, use the X.ORIGIN, Y.ORIGIN, and Z.ORIGIN parameters. These parameters describe the location of the origin relative to the device coordinate system. The THETA and PHI parameters describe the direction of propagation relative to the beam origin. The THETA parameter describes rotation of the X axis about the Z axis. The PHI parameter describes rotation of the Z axis about the Y axis. As with the 2D for normal illumination, specify a negative Y.ORIGIN and THETA=90.

Specify the window of illumination using the XMIN, XMAX, ZMIN, and ZMAX parameters. Then, specify the numbers of samples in the X and Z directions using the NX and NZ parameters. These values should be set to numbers large enough to resolve any salient features of the topology.

To save the rays into a file for visualization, specify the RAYTRACE=<filename> parameter. The components (u_x, u_y and u_z) of the direction vector representing the direction of the beam propagation are given by the following:

$$u_y = cos(\mathtt{PHI} - 90) \quad (11\text{-}83)$$

$$r = -sin(\mathtt{PHI} - 90) \quad (11\text{-}84)$$

$$u_x = r\,cos(\mathtt{THETA}) \quad (11\text{-}85)$$

$$u_z = r\,sin(\mathtt{THETA}) \quad (11\text{-}86)$$

Reflections

You can also specify whether to ignore the first reflection using the FRONT.REFL or the backside and sidewall reflection using the BACK.REFL. You should activate the backside reflections for devices, which use a back side reflector to improve collection efficiency. To set the number of reflections solved, use the REFLECTS parameter.

Typically, BACK.REFL should be used if the structure simulated is equivalent to the complete photodetector geometry as in a discrete device. If the simulation structure is a section of a larger substrate as in CCD simulation then don't use BACK.REFL.

Since the ray trace for arbitrary angles of incidence uses reflection and transmission coefficients, specify the polarization by using the POLARIZATION parameter.

In complex structures, you should limit the ray tracing to trace only those rays with significant optical power. The MIN.POWER parameter is used to terminate ray traces that drop below MIN.POWER* (optical source power).

11.12.2 Specifying Periodicity in the Ray Trace

By default, the unspecified boundary conditions for device simulation are mirror or periodic boundary conditions. For ray tracing, by default, the edges of the device are considered interfaces with a vacuum. Thus, at the edges of the device, by default, reflections are calculated. In some cases, it is convenient to consider the edges of the device to be periodic with respect to ray tracing. This is particularly true when the beam is not normal to the plane. You should specify PERIODIC in the **BEAM** statement to obtain such boundaries for ray tracing.

11.12.3 Defining Luminous Beam Intensity in 2D

When a beam is defined in Luminous, it is, by default, assumed to have a uniform light intensity across the width of the beam. The intensity is defined in the **SOLVE** statement with the B<n> parameter and is in the units of W/cm^2.

The beam intensity distribution can also be defined with a Gaussian profile.

$$f(x) = \frac{1}{\mathtt{SIGMA}\sqrt{2\pi}} exp\left(-\frac{(x - \mathtt{MEAN})^2}{2(\mathtt{SIGMA})^2}\right) \quad (11\text{-}87)$$

The location of the peak of the Gaussian distribution with respect to the beam coordinates is specified by the MEAN parameter of the **BEAM** statement. The standard deviation of the distribution is specified with the SIGMA parameter.

x is the cumulative distance over the beam width where the distribution will be formed. Define the RAYS parameter to get smooth distribution of intensity profile. This determines the increment of *x*. Therefore, the RAYS parameter should be set to a large number as the spacing between rays is always a constant (width of the beam /RAYS).

The resultant distribution has a unity peak value, which is then scaled by the B<n> value specified on the **SOLVE** statement. For example, the following **BEAM** statement will define a beam window 2 μm wide centered at X.ORIGIN and Y.ORIGIN, which has a Gaussian peak of 0.01 W/cm^2 and a standard deviation of 0.05 μm.

```
beam   num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=0.6 \
xmin=-1   xmax=1 GAUSSIAN   MEAN=0   SIGMA=0.05 RAYS=200 \

SOLVE B1=1E-2
```

11.12.4 Defining Luminous 3D Beam Intensity

A beam defined in Luminous 3D is assumed by default to have a uniform light intensity across the width of the beam. The intensity is defined in the **SOLVE** statement with the *B,n*. parameter and is in the units of W/cm^2.

In Luminous 3D, you can also specify a circular or elliptical source. To specify an elliptical source, specify the appropriate origin location using X.ORIGIN, Y.ORIGIN, and Z.ORIGIN. This will specify the center of the elliptical source. The major and minor axes of the ellipse are specified by the XRADIUS and ZRADIUS parameters. Remember, you must also specify the NX and NZ parameters. These will sample uniformly across the diameters of the ellipse.

You may also specify Gaussian profiles in Luminous 3D. To do this, specify the location of the beam origin. You can specify the Gaussian(s) in the X and Z direction. These can be done independently or together. These Gaussians are specified by the XMEAN, XSIGMA, ZMEAN, and ZSIGMA parameters of the **BEAM** statement.

11.12.5 Monochromatic or Multispectral Sources

The optical source can be either monochromatic or multispectral. For monochromatic sources, you can use the WAVELENGTH parameter to assign the optical wavelength. WAVELENGTH uses the units microns to be more consistent with the rest of Atlas. Note this if you're accustomed to the optoelectronic engineering preference for nanometers.

For multispectral sources, spectral intensity is described in an external ASCII file indicated by the POWER.FILE parameter. This is a text file that contains a list of pairs defining wavelength and spectral intensity. The first line of the file gives the integer number of wavelength-intensity pairs in the file. An example of such a file is shown below.

```
4

0.4 0.5

0.5 1.0

0.6 1.2

0.7 1.1
```

This example specifies that there are four samples and that at a wavelength of 0.4 μm, the intensity is 0.5 Watts per square cm per μm of wavelength, and so on. With multispectral sources, specify a discretization of the interpolated information. Values must be specified for the WAVEL.START, WAVEL.END, and WAVEL.NUM parameters. These values specify the starting and ending wavelengths and the number of wavelengths to sample. Luminous uses wavelengths at equal intervals over a specified range of wavelengths.

If you don't specify the values of WAVEL.START, WAVEL.END, and WAVEL.NUM, these parameters take on the corresponding values from the specified POWER.FILE. For the example file shown above, Luminous uses the following default values of these parameters: WAVEL.START=0.4, WAVEL.END=0.7, and WAVEL.NUM=4.

Luminous performs an independent ray trace at each of the sample wavelengths. For example:

```
WAVEL.START=0.4 WAVEL.END=0.6 WAVEL.NUM=2
```

causes ray traces at wavelengths of 0.45 and 0.55. Luminous obtains the intensity associated with each sample by integrating the values of the spectral intensity file using a piece wise linear approximation. Each integral is performed over the range between successive midpoints. In the preceding example, the integration for the sample at 0.45 would be performed over the range of 0.4 to 0.5.

For a multispectral source, the generation rate (like Equation 11-11) is given by:

$$G = \eta_0 \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} \frac{P(\lambda)L\lambda}{hc}\alpha e^{-\alpha y} d\lambda \qquad 11\text{-}88$$

where:

- η_0 is the internal quantum efficiency.
- $P(\lambda)$ is the power spectral density of the source.
- L is a factor representing the cumulative loss due to reflections, transmissions, and absorption
  over the ray path.
- λ is the wavelength.
- $\hbar$ is Planck's constant.
- c is the speed of light.
- α is the absorption coefficient given by Equation 11-12.
- y is the depth of the device, where x,y forms the two-dimensional mesh.

WAVEL.START and WAVEL.END are the spectral limits specified on the **BEAM** statement.

Note: The integral in Equation 11-88 may be inexact due to the discrete sampling in conjunction with wavelength dependence of the absorption coefficient. For constant absorption coefficient, the integral is exact of the number of discrete wavelength samples specified by WAVEL.NUM on the **BEAM** statement.

The source photocurrent (like Equation 11-75) is given by:

$$I_s = q\frac{B_n}{hc}W_t \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} P(\lambda)\lambda d\lambda \qquad 11\text{-}89$$

where $P(\lambda)$ is the power spectral density of the source. The other parameters have the same definition as in Equation 11-75.

The available photo current (like Equation 11-76) is given by:

$$I_A = q\frac{B_n}{hc}\sum_{i=1}^{N_R} W_R \int_0^{X_i} \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} P(\lambda)\alpha_i e^{-\alpha_i y} P_i \lambda dy d\lambda \qquad 11\text{-}90$$

where $P(\lambda)$ is the power spectral density of the source. The other parameters have the same definition as in Equation 11-76.

Luminous allows you to chose the units of spectral intensity in an input spectrum file. By default, the units of spectral intensity in the POWER.FILE are $W/(cm^2\ \mu m)$. In this case, Luminous carries out the integration according to Equations 11-88, 11-89, and 11-90. If the units of spectral intensity in the POWER.FILE are W/cm^2, then you need to use the flag ^INTEGRATE on the **BEAM** statement. In this case, each value $P(\lambda)$ in the second column of the file is a total intensity for a corresponding spectral interval. Integrals over spectrum in Equations 11-88, 11-89, and 11-90 are reduced to summations of $P(\lambda)$ values. In both cases, Luminous treats the subsequently specified value of B<n> on the **SOLVE** statement as a unitless multiplier or scale factor for intensity.

This is different from monochromatic case where this parameter has units of W/cm^2. To preserve consistency with monochromatic case, you can also do the same in the multispectral applications. If you specify the total beam intensity for multispectral sources using B<n> parameter on the **SOLVE** statement, you need to set NORMALIZE flag on the **BEAM** statements. This ensures that the intensity spectrum is normalized and B<n> has units of W/cm^2.

For either the monochromatic or multispectral sources, you can uniformly scale the wavelength(s) and intensities by using the WAVEL.SCALE parameter and the POWER.SCALE parameter respectively. These parameters are useful if the intensities or wavelengths are specified in units other than the default units. To output the samples from the input spectrum file to a file suitable for display in TonyPlot, specify the output file name on the OUT.POWER parameter of the **BEAM** statement.

Remember that these are built-in tabular spectra and include many samples. You are therefore advised to use subsample using the WAVEL.START, WAVEL.END, and WAVEL.NUM parameters as described above.

The run-time output includes a couple of headings to represent the integrated intensity as input and as subsampled. "Iinput" is the integrated input spectrum. This is the integral over the samples contained in the file specified by the POWER.FILE parameter (or contained in the built-in AM0 or AM1.5 spectra). "Isampled" is the integral of the sub-sampled spectrum. You sub-sample the input spectrum when you specify WAVEL.START, WAVEL.END, and WAVEL.NUM. If you do not specify this sub-sampling, the raw input samples are used and "Isampled" is identical to "Iinput".

If you sub-sample, "Isample" is always less than or equal to "Iinput". If you chose the start and end samples to match those contained in the input spectrum, you will get "Iinput" equal to "Isampled" due to the way we integrate power when we sub-sample.

You can look at the input spectrum in TonyPlot by specifying a log file name in the value of the OUT.POWER parameter of the **BEAM** statement and loading that file into TonyPlot.

Sub-sampling has advantages to save on time computing optical propagation, such as ray tracing or FDTD solutions for each specified wavelength. Usually, sub-sampling introduces only moderate quantization error if the index of refraction does not vary too quickly.

11.12.6 Black Body Sources

You may specify a black body radiation source as described by

$$P\lambda(T) = \frac{2hc^2}{\lambda^5}\frac{1}{e\frac{hc}{\lambda kT}-1} \qquad 11\text{-}91$$

where λ is wavelength and T is temperature. To enable the model, you should specify the black body temperature using the TBLACK parameter of the **BEAM** statement. The units of equation $P\lambda(T)$ are W/cm2/steradian. For calculation, we normalize to 1 steradian since we cannot predict the light gatherign characteristics of your optical system. The black body radiation spectrum will be most useful in distinguished temperature of equivalent targets.

11.12.7 Solar Sources

Luminous and Luminous 3D provide built-in definitions of solar spectra for AM0 [261] and AM1.5 [262] conditions. To specify these spectra, define AM0 or AM1.5 on the appropriate BEAM statement.

11.12.8 Extracting Dark Characteristics

One of the first tasks in analyzing a new detector design is to examine dark current, device capacitance, and possibly other unilluminated characteristics. This can normally be done without using Luminous. Extraction of the characteristics is adequately covered in the chapters on S-Pisces or Blaze.

The extraction of reverse bias leakage currents for diodes presents some difficult numerical problems for device simulators. These problems are associated with limitations on numerical precision. Atlas, as well as most other available device simulators, use double precision arithmetic to evaluate terminal currents. Double precision arithmetic provides roughly 16 decimal digits of precision. Internal scaling allows the measurement of currents down to a level of between about 10^{-12} A/micron to 10^{-16} A/micron. Unfortunately, photodiode leakage currents are often around or below this level. This means that the currents printed are subject

to significant numerical noise and don't provide an accurate estimate of the device leakage currents. There are two ways to estimate reverse leakage current.

Integrated Recombination

From a theoretical standpoint, the reverse behavior of diodes can be dominated by one of two effects: diffusion currents in the neutral regions or recombination currents inside the depletion region [307]. Atlas can provide insight into both of these contributing mechanisms. To estimate recombination current, use the **MEASURE** statement to calculate the integrated recombination rate. The following statement can be used:

```
MEASURE U.TOTAL
```

When this statement is executed, it prints out the total integrated recombination rate. Multiply this value by the electron charge (1.6023×10^{-19} coulombs) to obtain an estimate of the recombination current contribution to the reverse diode leakage current.

Extrapolation from High Temperatures

The diffusion current contribution can be estimated by taking advantage of the non-linear relationship between the diffusion current and temperature. Referring to the expression for the "Ideal Diode" current given by Equation 11-92, the dominant temperature dependency arises from the variation of the intrinsic concentration.

$$J = \left(\frac{qD_p p_{n0}}{L_p} + \frac{qD_n n_{p0}}{L_n}\right)\left(\exp\left(\left(\frac{qV}{kT_L}\right) - 1\right)\right) \quad 11\text{-}92$$

where n_{p0} and p_{n0} are thermal-equilibrium minority carrier densities on either side of the junction. This gives an exponential variation of the diffusion current with temperature given in Equation 11-93.

$$J \approx \exp\left(\frac{-E_g}{kT_L}\right) \cdot \left(\exp\left(\frac{qV}{kT_L}\right) - 1\right) \quad 11\text{-}93$$

This relation can be used to estimate the diffusion current contribution at the operating temperature. The basic idea is to calculate the current at a high temperature where the problem of numerical precision does not arise. Then, scale the current to the operating temperature using Equation 11-93. For example, if the device is to operate at 300K, you can set the temperature to 450K using the TEMPERATURE parameter of the **MODELS** statement. Disable any temperature dependence of the energy gap by specifying the band gap using the EG300 parameter and setting EGALPHA and EGBETA parameters to zero, which are all on the **MATERIAL** statement. The following statement illustrates this approach as it might apply to a silicon diode:

```
MODEL TEMPERATURE=450

MATERIAL EG300=1.12 EGALPHA=0.0 EGBETA=0.0
```

Atlas can then be used to obtain the reverse bias current at the elevated temperature. Equation 11-94 can be applied to obtain the depletion current contribution at the operating temperature:

$$J = J_e \cdot \exp\left(\frac{E_g}{kT_e} - \frac{E_g}{kT_L}\right) \cdot \left(\frac{\exp\left(\frac{qV}{kT_L}\right) - 1}{\exp\left(\frac{qV}{kT_e}\right) - 1}\right) \quad \text{11-94}$$

where J_e is the current measured at the elevated temperature, E_g is the bandgap, T_e is the elevated temperature, T_L is the operating temperature, V is the operating bias voltage, and J is the current estimate at the operating temperature. Once you've obtained estimates of the recombination and diffusion contributions, you can obtain the total leakage current by summing the two contributions.

Numerical Solution Parameters

Atlas uses a cut-off value of carrier concentration below, which solutions are not required to converge. This limit is set by the CLIM.DD. parameter. See Chapter 21: "Numerical Techniques" for more details on CLIM.DD. For photodetectors, you often need to reduce CLIM.DD to 10^5 in order to resolve carrier concentrations in depleted regions before illumination.

11.12.9 Extracting Detection Efficiency

One of the simpler tasks in characterizing a photodetector design is to measure DC detection efficiency. This will be done typically as a function of bias voltage, optical intensity, or wavelength. Each of these analyses can be performed using the **SOLVE** statement. The Bn parameter can be used to set the optical intensity of the optical sources described in the previous section. The following example illustrates obtaining a solution with a specified optical intensity:

```
SOLVE B1=1.0
```

This specifies that a solution is to be obtained for an optical intensity in the beam numbered "1" of 1.0 Watt/cm^2. If this were the first **SOLVE** statement specified, the ray trace in Luminous would be initiated. At the start of the solution, the optical intensities of each optical source with a positive intensity is printed. In addition, the available photocurrent and source photocurrent are printed. See the prior section on Photocurrent for a definition of these two quantities.

Internal and External Quantum Efficiency

The available photocurrent divided by the source photocurrent is a measure of the external quantum efficiency of the detector. The calculated terminal current can be divided by the source or available photocurrents is used to evaluate the internal quantum efficiency of the device.

11.12.10 Obtaining Quantum Efficiency versus Bias

The intensities specified in the **SOLVE** statement apply until another **SOLVE** statement changes the intensity of the beam. Sequences of **SOLVE** statements can be used to vary the optical intensity at arbitrary intervals. The simple linear ramps of optical intensity can be abbreviated using the LIT.STEP and NSTEP parameters of the **SOLVE** statement. The LIT.STEP parameter specifies the size of the DC step and NSTEP specifies how many steps are desired.

Another option for analyzing DC quantum efficiency is to fix the optical intensity and vary bias voltages. The bias voltages can be varied in arbitrary discrete steps using several **SOLVE** statements, or in a linear ramp using individual **SOLVE** statements. This is useful for determining the optimum operating bias of devices such as avalanche detectors and photo transistors.

11.12.11 Obtaining Transient Response to Optical Sources

It is sometimes desirable to examine the time domain response of a detector to time-dependent (e.g., ramped or pulsed) optical sources. Luminous provides this capability with the RAMP.LIT parameter. When the RAMP.LIT parameter is specified in a SOLVE statement, the optical intensity is changed linearly from the most recently set intensity to the intensity set in the B parameter. If a particular source intensity is not set using the corresponding B parameter, its intensity is not varied during the transient.

The period of the linear ramp is specified by the RAMPTIME parameter. The transient simulation stops after the time specified by the TSTOP parameter. If the time given by TSTOP is greater than that given by RAMPTIME, the source intensities remain constant until the time given by TSTOP. For transient ramps, the TSTEP parameter should also be set. TSTEP is typically set to allow several samples within the RAMPTIME. After the first time step, subsequent time step sizes will be chosen automatically based on estimates of truncation error. The following is an example of the specification of an optical impulse transient:

```
SOLVE B1=1.0 RAMPTIME=1E-9 TSTOP=1E-9 TSTEP=1E-11

SOLVE B1=0.0 RAMPTIME=1E-9 TSTOP=20E-9 TSTEP=1E-11
```

In this example, a triangular impulse in the intensity of the optical source is simulated. The peak intensity is 1.0 and the impulse width is 2 ns. The response of the device is simulated for an additional 18 ns. An initial time step of 10 ps is chosen for both parts of the impulse.

11.12.12 Obtaining Frequency Response to Optical Sources

You can also simulate small signal response to optical sources. To obtain a solution for small signal response, set the SS.PHOT parameter in the **SOLVE** statement. The **BEAM** parameter must also be assigned to the specific optical source index for small signal response of that source. A single source small signal frequency can be specified using the FREQUENCY parameter. You can vary the frequency within a single **SOLVE** statement by using the NFSTEP and FSTEP parameters. The NFSTEP indicates how many frequency steps are to be simulated, while the FSTEP indicates the step size. If the MULT.F parameter is specified, the start frequency is multiplied by the step size for the specified number of steps. If not, the step size will be added to the start frequency for the specified number of steps. For example:

```
SOLVE SS.PHOT BEAM=1 FREQUENCY=1e6 NFSTEP=5 FSTEP=10 MULT.F
```

will invoke solutions as optical frequencies at every decade from 1MHz to 100 GHz.

If the small signal parameters are specified in the same **SOLVE** statement as a DC bias ramp, the small signal response is extracted for each bias voltage for each small signal frequency. This is a useful strategy for analyzing the frequency response of the device as a function of bias voltage.

Note: When simulating spectral response you should try to sample enough to resolve variations in complex index of refraction, variations in input spectrum (e.g., AM1.5), and variations due to coherence effects in layers with thicknesses on the order of the wavelength. Such coherence effects are of no consequence in ray tracing but may be important in the transfer matrix or finite difference time domain methods.

11.12.13 Obtaining Spatial Response

To obtain spatial response, move an optical spot along a line segment perpendicular to the direction of propagation of the source. Each incremental step is equal to the width of the spot. The total distance over which the source is scanned is defined by the MIN.WINDOW and MAX.WINDOW parameters of the **BEAM** statement. The number of steps is defined by the RAYS parameter of the **BEAM** statement. The spot width is defined by the ratio:

```
(MAX.WINDOW - MIN.WINDOW)/RAYS
```

The spot scan is started by the SCAN.SPOT parameter of the **SOLVE** statement. This parameter is set to the beam index of the optical source to be scanned. In other words, the beam defined by the **BEAM** statement whose NUMBER parameter is set to the beam index. During the spot scan, Atlas obtains solutions and outputs, such as terminal currents, as well as the relative beam location at each incremental spot location. This information can be used by TonyPlot to produce plots of photoresponse as a function of position.

In Luminous 3D, the SCAN.SPOT parameter will cause the simulator to power each ray defined by the NX and NZ parameters sequentially.

11.12.14 Obtaining Spectral Response

The spectral response, defined as device current as a function of the wavelength of the optical source wavelength, can be obtained. The LAMBDA parameter of the **SOLVE** statement sets the source wavelength of the beam in microns. Since the wavelength can be set in each SOLVE statement, successive solutions can be obtained as a function of wavelength. Each time the LAMBDA parameter is specified, a new ray trace is run for that new wavelength and the electrical solution recalculated.

The following statements could be used to extract terminal currents at a series of discrete wavelengths:

```
SOLVE B1=1 LAMBDA=0.2
SOLVE B1=1 LAMBDA=0.3
SOLVE B1=1 LAMBDA=0.4
SOLVE B1=1 LAMBDA=0.5
SOLVE B1=1 LAMBDA=0.6
SOLVE B1=1 LAMBDA=0.7
SOLVE B1=1 LAMBDA=0.8
```

In this example, the spectral response is obtained for wavelengths from 0.2 microns to 0.8 microns. When using LAMBDA, the WAVELENGTH parameter of the **BEAM** statement will overridden.

Equivalently, you may find it more convenient to specify a spectral ramp. To specify a ramp of wavelength, you should assign the **BEAM** parameter to the beam index of the optical source to be ramped. Also, you should specify the initial wavelength with the LAMBDA parameter, the final wavelength with the WFINAL parameter, and the wavelength step size with the WSTEP parameter as illustrated in the following.

```
SOLVE B1=1 BEAM=1 LAMBDA=0.2 WSTEP=0.1 WFINAL=0.8
```

This is completely analogous to the seven statements described above.

Note: If you specify POWER.FILE on the **BEAM** statement, the spectral intensities will be scaled by the value interpolated from the power file.

11.12.15 Obtaining Angular Response

The response of photodetectors to angle of incidence can be obtained in a manner analogous to the method used to obtain spectral response.

In this case, the ANGLE parameter of the **SOLVE** statement specifies an angle added to the value of the **BEAM** parameter PHI (ANGLE). This allows collection of response versus angle. The following examples statements demonstrate the syntax.

```
SOLVE B1=1 ANGLE=0
SOLVE B1=1 ANGLE=10
SOLVE B1=1 ANGLE=20
SOLVE B1=1 ANGLE=30
SOLVE B1=1 ANGLE=40
SOLVE B1=1 ANGLE=50
SOLVE B1=1 ANGLE=60
```

Equivalently, you may find it more convenient to specify a angle ramp. To specify a ramp of angle, you should assign the **BEAM** parameter to the beam index of the optical source to be ramped. Also, you should specify the initial angle with the ANGLE parameter, the final angle with the AFINAL parameter, and the angle step size with the ASTEP parameter as illustrated in the following.

```
SOLVE B1=1 BEAM=1 ANGLE=0.0 ASTEP=10.0 AFINAL=60.0
```

This is completely analogous to the seven statements described above.

11.13 Simulating Solar Cells

The Luminous simulators are well suited to the simulation of solar cells. Particularly, the variety of robust optical propagation models in combination of the capabilities of the device simulators in the Atlas framework makes accurate solar cell simulation a possibility. The following paragraphs focus on problems and solutions specific to the simulation of solar cells.

11.13.1 Solar Spectra

As discussed in Section 11.12.5 “Monochromatic or Multispectral Sources”, Luminous 2D and Luminous 3D provide flexible means for defining multispectral sources. In addition, you can directly access built-in public domain [261, 262] solar spectra for am0 and am1.5 conditions by specifying AM0 or AM1.5 on the BEAM statement. These spectra consist of numerous samples and we therefore suggest you carefully consider subsampling as discussed in Section 11.12.5 “Monochromatic or Multispectral Sources”.

The following is an example of how you might define the BEAM statement for analysis of am1.5 conditions with subsampling.

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 WAVEL.NUM=50
```

Here, we are choosing to simulate am1.5 conditions sampled between 0.3 microns and 1.2 microns at 50 samples. We do not have to worry about sampling the many peaks and valleys in the solar spectrum as it is automatically integrated into each subsample. The subsampling effects only the number of wavelengths for which the propagation model is calculated not the integrated intensity.

11.13.2 Solar Optical Propagation Models

The choice of optical propagation model for your purposes depends largely on the nature of your solar cell devices. In some cases, you may want to apply several of the different models to compare the results against measurement for your particular analysis.

In all cases, accurate knowledge of the complex index of refraction of the materials comprising the solar cell versus wavelength is key in accurate light propagation modeling. Luminous and Luminous 3D provide built-in defaults for most materials of importance to solar cell technology (see Section B.13 “Optical Properties” for more details). Additionally, Luminous and Luminous 3D provide flexible means for you to introduce user-supplied data in the form of ASCII tables or C-Interpreter functions as discussed in Section 11.8 “Defining Optical Properties of Materials”.

If you have little knowledge of the optical properties of the materials in your solar cell, or if you are less interested in the propagation problem, you may choose to simply define the photogeneration rate using the C-Interpreter function F.RADIATE. This function allows you to directly specify the photogeneration rate as a function of position in the device. The following illustrates the use of the F.RADIATE C-Interpreter function for constant generation rate.

First, you need to identify the C function defining the generation rate to the simulator. This is done in the input deck on the BEAM statement using the F.RADIATE parameter as follows:

```
BEAM NUM=1 F.RADIATE=solar.lib
```

Here, the C-Interpreter function is contained in a separate file named "solar.lib". For example, the contents of the file "solar.lib" may be as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
#include <malloc.h>
#include <string.h>
#include <template.h>

/*
 * Generation rate as a function of position
 * Statement: BEAM
 * Parameter: F.RADIATE
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * z          location z (microns)
 * t          time (seconds )
 * *rat       generation rate per cc per sec.
 */
int radiate(double x, double y, double z, double t, double *rat)
{
*rat = 1e21;
return(0);                /* 0 - ok */
}
```

Here, we simply define a constant photogeneration rate everywhere in the semiconductor of 1e21 carrier pairs per second per cubic centimeter.

You can easily define this function to specify any arbitrary generation rate as a function of position and time. You should refer to Appendix A "C-Interpreter Functions" for more information on the C-Interpreter.

Ray tracing is the principal propagation model for cases where we are not concerned about interference or diffraction effects in the bulk of the device. This is most useful for relatively thick bulk devices, such as crystalline silicon devices. It is useful for planar or textured devices. For textured devices, the texturing must be described explicitly in the device structure. Details of the ray tracing models are given in Section 11.2 "Ray Tracing".

The following gives an example of the specification of ray tracing for the optical propagation model for Luminous.

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 WAVEL.NUM=50
```

In this case, we will perform 50 ray traces. The omission of other model keywords, such as TR.MATRIX, FDTD, or BPM, signifies that ray tracing will be performed.

For Luminous, the algorithm automatically traces exactly the right number of rays to completely resolve all discontinuities at the surface and deep into the device. In 3D, you must specify enough rays to resolve the geometry in the NX and NZ parameters.

With ray tracing, you can also specify anti-reflection coatings using the **INTERFACE** statement as discussed in Section 11.9 “Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method”.

We recommend the transfer matrix method, as discussed in Section 11.3 “Matrix Method”, for simulation planar of (1D) devices where bulk interference effects may be of consequence. In these cases, the transfer matrix method provides fast accurate solutions for 1D analysis problems only.

To enable transfer matrix analysis, you need to add the TR.MATRIX parameter to the BEAM statement as in:

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 WAVEL.NUM=50
TR.MATRIX
```

We stress that the transfer matrix method is a one dimensional method and can only address variations in the Y direction. This should be adequate for much of the simulation problems addressed for planar thin film devices.

In the same way, you define anti-reflection coatings in ray tracing, you can also use the **INTERFACE** statement to define anti-reflection coatings (see Section 11.9 “Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method”) for the transfer matrix method.

For textured thin film devices, we have implemented a hybrid transfer matrix– 1D ray tracing method. In this case, interfaces can be characterized by haze functions defining the proportion of light transmitted and reflected as diffuse versus specular light and angular distribution functions describing the angular intensity of diffuse light. These are bulk characteristics and you should define the device as a planar structure as if you were applying the 1D transfer matrix method. These haze and angular distribution functions are described in the **INTERFACE** statement as illustrated in the following example:

```
INTERFACE OPTI P1.X=0 P2.X=1 P1.Y=0 P2.Y=0 DIFF ELLIPSE SEM=.31
SIG=20
```

Here, the optical interface, OPTI, defined along two points –P1.X, P1.Y and P2.X, P2.Y– is treated as a diffusive (textured) interface as signified by the DIFF parameter. The angular distribution function is elliptical as indicated by the ELLLIPSE parameter with a semi-major axis of 0.31.

The interface haze function is characterized by a mean roughness length of 20 nm as specified by the SIG parameter. This parameter can be correlated to measured surface variations from SEM photomicrographs.

To enable the diffusive transfer matrix model, you should specify DIFFUSE on the BEAM statement as in the following:

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 WAVEL.NUM=50 TR.MAT
DIFFUSE
```

The diffusive transfer matrix method is discussed in detail in Section 11.3.4 “Transfer Matrix with Diffusive Interfaces”.

For cases requiring full 2D or 3D analysis of coherence or diffraction effects, the finite difference time domain (FDTD) analysis should be used. This method provides the most accurate solutions for optical propagation problems but at the greatest computational expense. The FDTD method performs direct solutions to the electromagnetic equations of light propagation. To enable the FDTD model, you need to specify FDTD on the **BEAM** statement as in the following example:

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 WAVEL.NUM=50 FDTD
```

There are many additional parameters which should be set for FDTD analysis. These parameters are described in detail in Section 11.5 “Finite Difference Time Domain Analysis”.

11.13.3 Solar Cell Extraction

Here, we will describe how to extract the most common figures of merit for solar cells. These extraction methods are solution strategies placed in your input deck after defining the device structure, physical models, and the optical source as described in the previous section.

Typically, the first step in solar analysis is to illuminate the device. You can turn on the illumination by assigning the beam intensity on a **SOLVE** statement as follows:

```
SOLVE B1=1.0
```

Here, we assign a value of one "sun" to the intensity of the optical source indexed as number "1" (B1).

As we had assigned AM1.5 on the **BEAM** statement, this corresponds to standard test conditions. In some cases, we may encounter convergence difficulties and may need to ramp the light intensity as shown in the following:

```
SOLVE B1=0.01

SOLVE B1=0.1

SOLVE B1=1.0
```

You may also choose to assign the beam intensity to a value greater than one to simulate solar concentrator effects. Since we have not yet assigned any bias voltages, the current in the run time output corresponds to the short circuit current (Isc).

The next step is to extract the illuminated, forward current-voltage characteristic. First, we should define an output file for capturing the data. For example:

```
LOG OUTFILE="IV.LOG"
```

This specifies that all current voltage characteristics extracted from any following SOLVE statement will be captured in the file IV.LOG in the current working directory. To capture the I-V characteristic, you need to ramp the voltage past the open circuit voltage (Voc) as in the following:

```
SOLVE VANODE=0 NAME=ANODE VSTEP=0.1  VFINAL=0.5

SOLVE NAME=ANODE VSTEP=0.01 VFINAL=0.7
```

Here we assume that the anode is P type and the characteristic is extracted on positive anode voltages. First, we ramp from 0 to 0.5 volts in 0.1 volt steps. Then, we ramp from 0.51 to 0.7 volts in 0.01 volt steps. We assume that the open circuit voltage is near to 0.7 volts. We chose the shorter voltage step in the second **SOLVE** statement to improve accuracy in obtaining Voc. Essentially, we will resolve Voc to an accuracy comparable to the final voltage step size.

The current-voltage characteristic is observable in TonyPlot using the following:

```
TONYPLOT IV.LOG
```

We can now use the DeckBuild Extract capability to capture the various solar cell figures of merit. We first initialize the Extract capability with the captured input file as in the following:

```
EXTRACT INIT INFILE="IV.LOG"
```

The following lines demonstrate the extraction of short circuit current (Isc), open circuit voltage (Voc), maximum power (Pm), voltage at maximum power (Vm), current at maximum power (Im), and fill factor (FF).

```
EXTRACT NAME="Isc" Y.VAL FROM CURVE(V."anode", I."cathode") WHERE
X.VAL=0.0

EXTRACT NAME="Voc" X.VAL FROM CURVE(V."anode", I."cathode") WHERE
Y.VAL=0.0

EXTRACT NAME="Pm"  MAX(CURVE(V."anode", (V."anode" * I."cathode")))

EXTRACT     NAME="Vm"          X.VAL     FROM     CURVE(V."anode",
(V."anode"*I."cathode") ) \

        WHERE Y.VAL=$"Pm"

EXTRACT NAME="Im"  $"Pm"/$"Vm"

EXTRACT NAME="FF"  $"Pm"/($"Jsc"*$"Voc")
```

Given that we know the area of the device in square cm, we can assign it to a variable and extract power efficiency as follows:

```
SET area=1e4

EXTRACT NAME="OPT_INT"   MAX(BEAM."1")

EXTRACT NAME="POWER_EFF" (ABS($"Pm")/($OPT_INT*$area))
```

Here, OPT_INT is the optical intensity in W/sqcm.

It is also common to extract spectral response of solar cells. In this case, the source is monochromatic so we can omit the inclusion of a source spectrum, such as AM1.5, and instead define a single wavelength as in the following:

```
BEAM NUM=1 WAVELENGTH=0.3
```

Here, the wavelength of choice is not important since we will define the extraction wavelength on the **SOLVE** statement. Again, we should open an output file to capture the data as in the following:

```
LOG OUTFILE="SPECTRUM.LOG"
```

Then, we can simply capture the spectral response using a wavelength ramp defined on the SOLVE statement as follows:

```
SOLVE B1=0.001 BEAM=1 LAMBDA=0.3 WSTEP=0.02 WFINAL=1.2
```

Here, the B1 parameter sets the intensity to 1 mW. The **BEAM** parameter identifies the index of the optical source that is going to be involved in the wavelength ramp. LAMBDA specifies the initial wavelength in microns, WFINAL specifies the final wavelength in microns, and WSTEP specifies the wavelength step in microns.

We can then observe the spectral response file using TonyPlot as follows:

```
TONYPLOT SPECTRUM.LOG
```

If you had specified a spectrum file (or AM0 or AM1.5) on the **BEAM** statement, then the value of the B1 parameter is multiplied by the intensity interpolated from the spectrum file at the specified wavelengths.

11.14 Optical Characterization

In many cases it is desirable to reduce the complexity of a problem by modeling the problem over a representative space in a simpler tabular or analytic form. Here, we will call this reduction of complexity "characterization".

Optical characterization is initiated by the **CHARACTERIZE** statement. The **CHARACTERIZE** statement is much like the **LOG** and **PROBE** statements in that it descrbes a set of actions to take place at each subsequent solution. In this case, we produce a characterization file at each solution.

The root file name for characterization of absorption is specified by the ABSORPTION parameter. File names are generated from the root name appended by a sequence number and ".log". These files can be subsequently examined in TonyPlot. These files contain absorption rate versus depth (Y). All transverse dependencies are captured in this simple form. This simple form is to map into large area devices.

The file names can be sequenced. The maximum number of files in the sequence can be specified by the SEQ.ABS parameter .

To characterize, you must also specify a source beam index on the BEAM parameter. This is the same number assigned to the same number as was assigned to the NUMBER parameter of the **BEAM** statement.

The range of samples for the characterization is specified by the DEP1 and DEP2 parameters. DEP1 and DEP2 specify minimum and maximum Y coordinates for subsampling in microns.

Subsampling will be taken uniformly over NDEP samples unless you specify S.DEP1 and S.DEP2, which specify the spacings in microns at the start and end of sampling.

In this case you do not need to specify NDEP.

Chapter 12
LED: Light Emitting Diode Simulator

12.1 Overview

LED is a simulator within the Atlas framework that provides general capabilities for simulation of light emitting diode (LED) devices. When used with the Blaze simulator, LED supports the simulation of DC and transient electrical and light emitting characteristics of LEDs. When used with the GIGA simulator, LED permits the self-consistent analysis of thermal effects on both electrical and optical emission characteristics. You can also use the LED simulator with the MixedMode simulator to analyze the circuit level performance of LED devices.

The LED simulator supports simulation of zincblende (e.g., AlGaAs/GaAs, InGaAsP/InP), wurtzite (e.g., GaN/AlGaN/InGaN), and organic/polymer material systems. It can also account polarization effects and the effects of strain on both emission spectra and piezoelectric polarization.

Extraction of electrical characteristics, such as DC, transient and small signal response, is augmented by the capability to extract light emitting characteristics, such as emission power versus current, emission spectra, emission wavelength, efficiency and output coupling (a function of emission angle).

Rigorous 2D reverse ray tracing modeling is used to characterize output coupling characteristics in including far field patterns.

The finite difference time domain (FDTD) method can alternatively be applied to determine output coupling in 2D or 3D particularly for devices with photonic features, such as photonic crystals or gratings.

The following sections describe how to specify LED devices, how to select and specify physical models including most importantly the optical models and how to extract the most pertinent device characteristics for device performance optimization.

12.2 Defining Light Emitting Devices (LEDs)

There are several ways to define an LED device structures in Atlas/LED. First, you can define a structure using the Athena process simulator. Athena simulates the "construction" of devices by simulating the process flow and the physical process steps. Second, you can define the device structure using the DevEdit device builder tool. The DevEdit tool allows you to specify the device structure in a simple visual point and click environment. Finally, you can define device structures using Atlas framework commands (statements). For LED devices, the Atlas command language approach is probably the most convenient since the device structures are somewhat simple due to their epitaxial nature.

In the following sections, we will concentrate on definition of the device structure using the Atlas command syntax. We recommend that you read Chapter 2 “Getting Started with Atlas” where we describe general features of the Atlas simulator, commands and structure definition before you proceed. Chapter 3 “Physics” also provides insight into the details of the electrical and optical physical models used by Atlas/LED. Chapter 6 “Blaze: Compound Material 2D Simulator” describes some general physics of simulation of compound semiconductors and heterojunctions.

Within the Atlas command syntax, there are two approaches for defining device structures. The general purpose structure definition approach (see Section 2.6.3 “Using The Command Language To Define A Structure”) requires more user-specification and requires that you describe the entire mesh. The auto-mesh approach (see Section 2.6.4 “Automatic Meshing (Auto-meshing) Using The Command Language”) is simpler to use, requires less user-specification and is particularly well-suited to specification of epitaxial (especially multiple-quantum well or super-lattice) devices. It is for these reasons that we recommend the auto-mesh approach when defining LED devices .

In the following sections, we will focus mainly on this approach. We will point out important differences with the other methods: Athena, DevEdit or manual-mesh as these differences may arise.

The first command in any Atlas input deck is the **MESH** command. When loading a structure from Athena, DevEdit or a previous Atlas simulation, you should assign the INFILE parameter of the **MESH** statement to the name of the file containing the device structure. For example:

```
MESH INFILE=led.in
```

would cause the simulator to load a device structure from the file named led.in in the current working directory. When defining the device structure in the Atlas auto-meshing syntax, the first statement should contain the AUTO parameter as shown in the following statement:

```
MESH AUTO
```

This will invoke the auto meshing capability.

Note: The INFILE parameter is not specified since this would indicate that the device was both being loaded as well as constructed in the Atlas synta.x Therefore, would produce an ambiguity or conflict.

When using auto-meshing, you must only specify the mesh in the X direction using **X.MESH** statements. The mesh spacing in the Y direction is specified in the **REGION** statements as described later. The following example shows how you would specify an X mesh 1 μm long with a uniform mesh spacing of 0.1 μm.

```
X.MESH LOCATION=0    SPACING=100

X.MESH LOCATION=1000 SPACING=100
```

Using auto-meshing, you should next specify the device regions using **REGION** statements. The **REGION** statement can be used to specify the material types of individual "regions", as characterized by a spatial extent in the X and Y directions, and their composition, doping, strain and certain other characteristics describing how the region is to be modeled.

In the following example, we will describe a simple GaN/InGaN diode structure suitable for LED device simulation.

```
REGION MATERIAL=GaN   THICKNESS=0.25  ACCEPTOR=1e19 BOTTOM NY=20

REGION MATERIAL=InGaN THICKNESS=0.003 DONOR=2e14  X.COMP=0.2 BOTTOM
NY=20 LED

REGION MATERIAL=GaN   THICKNESS=0.25  DONOR=1e18 BOTTOM NY=20
```

In this example, we describe a three layer LED. The top layer is composed of GaN (MATERIAL=GaN), it has a thickness of 0.25 microns (THICKNESS=0.25) and has an acceptor concentration of 1e19 (ACCEPTOR=1e19). The middle layer is composed of InGaN (MATERIAL=InGaN), it has a thickness of 0.003 microns (THICKNESS=0.003) cm^{-3}, has an In composition fraction of 0.2 (X.COMP=0.2), and has a donor concentration of 2e14 (DONOR=2e14) cm^{-3}. The bottom layer is also composed of GaN (MATERIAL=GaN), it has a thickness of 0.25 microns (THICKNESS=0.25), and has a donor concentration of 1e18 (DONOR=1e18) cm^{-3}.

The BOTTOM parameter in each of the **REGION** statements specifies that each region resides directly below the preceeding region. NY specifies the minimum number of Y grid lines used to resolve the region. The LED parameter in the second **REGION** statement specifies that region will be treated as a light emitting region during the post-processing stages of the simulation, and certain light emitting characteristics of that region will be extracted.

The final step in defining the LED using auto-meshing is to specify the electrodes. To specify electrodes, use the **ELECTRODE** statement. The following example specifies two electrodes extending all the way across the top and the bottom of the device.

```
ELECTRODE NUMBER=1 NAME=anode TOP

ELECTRODE NUMBER=2 NAME=cathode BOTTOM
```

Here, the NUMBER parameter is used to give the electrodes numerical tags they can be referred to in subsequent operations. Similarly, the NAME parameter is used for future references (see Section 22.58 “SOLVE”).

12.3 Specifying Light Emitting Diode Models

In this section, we discuss methods and models particular to LED analysis.

12.3.1 Specifying Polarization and Piezoelectric Effects

In some material systems (such as the GaN/InGaN system), the built-in fields due to polarization and strain (piezoelectric effect) can play significant roles in the LED emission characteristics. Atlas introduces these polarization fields as sheet charges at the top and bottom edges of regions. To include the effects of polarization on a region, you should specify the POLARIZATION parameter on the corresponding **REGION** statement.

This will only include spontaneous polarization. If you want to also include piezoelectric polarization, you can also specify the CALC.STRAIN parameter. If you specify CALC.STRAIN, the simulator will automatically calculate strain from the lattice mismatch and will calculate the piezoelectric polarization and apply it to the region. You can also specify the value of the STRAIN parameter, which specifies the axial strain in the region.

With STRAIN and POLARIZATION set, the simulator will apply a piezoelectric polarization calculated using the strain value assigned by the STRAIN parameter.

12.3.2 Choosing Radiative Models

The next step in simulating LED devices is the selection of appropriate radiative models. For LED devices, the most relevant radiative model is the model for spontaneous (radiative) recombination. For radiative recombination, Atlas/LED offers several options. Section 3.9 "Optoelectronic Models" discusses these options in more detail.

First, is the general radiative model (see Equation 3-385). To enable this equation, specify OPTR in a **MODELS** statement associated with the active region(s) of the LED device (i.e., those regions with the LED parameter specified in the **REGION** statement). For example:

```
MODELS MATERIAL=InGaN OPTR
```

will enable the standard radiative recombination model. The disadvantage of using the standard model is that it provides no information about the spectral content of the LED light emission. In fact, using this model to properly specify emission intensity, you must specify the emission wavelength in the **SOLVE** statement for extraction of LED luminous intensity. To specify the emission wavelength for the general radiative recombination model given by Equation 3-383, assign the value of the wavelength to the L.WAVE parameter as shown in the following SOLVE statement, where L.WAVE is in units of microns. For example:

```
SOLVE L.WAVE=0.51
```

The COPT parameter of the **MATERIAL** statement specifies the rate constant for the general radiative recombination model. For example:

```
MATERIAL COPT=1.25e-10
```

COPT assigned a value of 1.25×10^{-10} cm^3/s to the radiative rate constant. You can calibrate the value of this constant using the more physical radiative models discussed next. This calibration will be discussed at the end of this section.

Beyond the basic model just described, Atlas/LED provides three more physical models for radiative rates. For zincblende materials, Atlas provides a one band model [363] and a two band [183] model. For wurtzite materials, Atlas/LED provides a three band model [58].

The one band zincblende model by Yan et.al. [363] accounts for optical transitions between a single valence band and the conduction band. To activate this model, specify ZB.ONE in the **MODELS** statement. See Section 3.9.7 "Unstrained Zincblende Models for Gain and Radiative Recombination" for a discussion of the Yan model.

The two band zincblende model [183] accounts for optical transitions between light and heavy hole valence bands and the conduction band. To activate this model, specify ZB.TWO in the **MODELS** statement. See Section 3.9.7 "Unstrained Zincblende Models for Gain and Radiative Recombination" for a discussion of the ZB.TWO model.

The three band wurtzite model [58] is based on k.p modeling and accounts for optical transitions between the conduction band and heavy and light hole and the crystal field split-off valence bands. See Section 3.9.10 "Strained Wurtzite Three-Band Model for Gain and Radiative Recombination" for a discussion of the WZ.KP model. To activate this model, specify WZ.KP in the **MODELS** statement. For the on-going example, we chose the following WZ.KP model (since InGaN is a wurtzite material).

```
MODEL MATERIAL=InGaN WZ.KP
```

You should note that any of the ZB.ONE, ZB.TWO, or WZ.KP model can be used to analyze spectral characteristics, such as emission wavelength and emission spectra.

You can also choose to analyze LED efficiency degradation due to competing recombination mechanisms by enabling these other non-radiative mechanisms. Particularly, you can enable Schockley-Read-Hall or Auger mechanisms. To activate these models, specify SRH and AUGER in the **MODELS** statement as follows:

```
MODELS SRH AUGER
```

Make sure you do not enable two radiative mechanisms for the same region, such as OPTR and one or more of the ZB.ONE, ZB.TWO, or WZ.KP models. If you do, you may over account for the same radiative recombination. For more information about recombination models, see Section 3.6.3 "Carrier Generation-Recombination Models".

As mentioned before, you can calibrate the radiative rate constant (using COPT of the **MATERIAL** statement) of the basic radiative recombination model (enabled by the OPTR parameter of the **MODELS** statement) to the more physical models (ZB.ONE, ZB.TWO, or WZ.KP). To do this, first prepare a simulation with the activated physical model. In this simulation, save a structure file around the device operating conditions. You can then extract the proper value of COPT to use with the OPTR model by dividing the radiative recombination rate at any point by the product of the electron and hole concentrations.

12.3.3 Using k.p Band Parameter Models in Drift Diffusion

For the WZ.KP model, the band calculations performed in Equations 3-659, 3-660, 3-661, and 3-663 can be used in the drift diffusion part of the simulation. To use this capability, specify the K.P parameter on the **MODELS** statement as follows:

```
MODELS WZ.KP K.P
```

12.3.4 Computation of Output Power Spectral Density

You can specify light generation via two different methods in Atlas. The first is to select on the **MODELS** statement one of the radiative emission models available in Atlas, and specify the appropriate parameters on the **MATERIAL** statement. The second is to provide a file with wavelength-dependent spectral density. For all materials, the default radiative emission

models are used if QWELL or SPONT parameters are specified. If SINGLET exciton density exists in a region, radiative emission from this region is included by default as $N_x/\texttt{TAUS.EXCITON}$, where N_x is the exciton density and TAUS.EXCITON is the radiative lifetime.

Ouput PSD from radiative emission models

The radiative emission computed by emission models is the Power Spectral Density (PSD) on energy scale, i.e. the power generated by a dipole per unit volume, per unit engery (Watts/cm^3/eV). In the following discussion, we denote this PSD function by $P(E)$. The LED module integrates PSD inside each mesh element to compute Power emitted in Watts/cm/eV for 2D simulations (LED is not yet available in 3D). The integration is performed on a uniform mesh inside the entire box:

$$B \equiv [\texttt{XMIN},\texttt{XMAX}] \times [\texttt{YMIN},\texttt{YMAX}] \quad \text{12-1}$$

Note: If you specify only the emission location using the parameters X and Y, Atlas integrates power emission over the entire device, i.e. it assumes that *B* extends over the entire device.

When you specify NSAMP on the SAVE statement, Atlas creates a uniform energy grid with spacings

$$\Delta E = \frac{\texttt{EMAX} - \texttt{EMIN}}{\texttt{NSAMP}} \quad \text{12-2}$$

If you specify LMIN and LMAX instead, Atlas computes $\texttt{EMIN} = hc/\texttt{LMAX}$, and $\texttt{EMAX} = hc/\texttt{LMIN}$, where hc is in units of eVμm. It then applies Equation 12-2 to compute ΔE. Atlas then computes average power emitted within an energy interval $[E_n,E_{n+1}]$ as:

$$S_n = \int_{E_n}^{E_{n+1}} P(E)\ dE$$

S_n has units of Watts/cm, where “per cm” refers to uniformity in third direction as is the case for all 2D simulations. The values S_n are written as the quantity “SOURCE PL” to the log file specified via OUT.SPEC parameter on SAVE statement. You can use this output to verify that the source photoluminescence is correct according to your expectations.

After the computation of S_n, Atlas computes the output coupling from each emission location. In the case where a box is specified by $[\texttt{XMIN},\texttt{XMAX}] \times [\texttt{YMIN},\texttt{YMAX}]$, the emission locations are uniformly placed with XNUM divisions of the interval [XMIN,XMAX] and YNUM divisions of the interval [YMIN,YMAX]. The power generated is taken to be spatially uniform inside the box and is equal to S_n. The output coupling is computed for photon energy in the middle of the interval, i.e. at $(E_n + E_{n+1})/2$. This coupling, which we will denote as C_n, can be computed by any of the three methods: FDTD, TR.MAT (transfer matrix method), or reverse ray tracing. Atlas then computes the output PSD for energy interval $[E_n,E_{n+1}]$ as,

$$\text{Output PSD}\ \frac{\text{Watts}}{\text{eV}\mu m} = \frac{C_n S_n}{(E_{n+1} - E_n)}\frac{[\text{Watts/cm}]}{[\text{eV}]} \times \frac{10^{-4} cm}{\mu m} \quad 12\text{-}3$$

The above described computation is repeated for each energy interval to compute the PSD over the specified energy range.

Ouput PSD from user specified spectrum

Instead of specifying radiative emission model, you can also provide an input file specifying wavelength in the first column and spectral density in the second. You must specify this file on the MATERIALS statement for the relevant material using the USER.SPECT parameter. The first entry in the file is the number of rows of wavelength and spectral density pairs to be read. Since this PSD is specified on wavelength scale, Atlas converts this PSD to compute power emission in each energy interval. Note that Atlas still computes uniform energy grid specified by Equation 12-2, which corresponds to a non-uniform wavelength grid. In addition, the aforementioned caveat about the assumption made about the box B when only X and Y are specified also applies in this case.

We now describe how S_n is computed from the wavelength dependent PSD, which we denote as $W(\lambda)$. The function $W(\lambda)$ is taken to specify the power generated by the dipole per unit volume per unit wavelength. Atlas assumes $W(\lambda)$ to be in units of Watts/μm/cm^3 and it normalizes W so that $\int W(\lambda) d\lambda = 1$Watt/cm^3. In each interval, the "User PL" is first converted to ***power density*** U_n for interval $[E_n, E_{n+1}]$ as follows:

$$U_n = \frac{1}{\lambda_{mid}} \int_{\lambda_a}^{\lambda_b} d\lambda W(\lambda)\lambda, \quad 12\text{-}4$$

where:

$$\lambda_a = \frac{hc}{E_{n+1}} \quad 12\text{-}5$$

$$\lambda_b = \frac{hc}{E_n} \quad 12\text{-}6$$

$$\lambda_{mid} = \frac{\lambda_a + \lambda_b}{2} \quad 12\text{-}7$$

Thus, the units of U_n are Watts/cm^3, and it refers to the power generation in single mesh element. In the present case of file input PSD, U_n are uniform within each region. The reason that Equation 12-4 is an integral of $W(\lambda) \times \lambda$ and not just W is that U_n is computed as average photon energy multiplied by the average photon generation rate in the energy interval. To compute S_n, Atlas now integrates U_n by integrating over all nodes inside the box

defined B in Equation 12-1. At this point, Atlas recovers the power emission per unit length S_n that is then multiplied by the output coupling as described above.

12.4 Data Extraction

12.4.1 Extracting Luminous Intensity

When the LED parameter is specified on any **REGION** statement, the luminous intensity emitted by the LED device is automatically calculated and written to the log file. The file that this information is output to is specified by the OUTFILE parameter of the **LOG** statement. The following example outputs the log file to the file led.log.

```
LOG OUTFILE=led.log
```

You can then load led.log into TonyPlot for post-processing visualization, where you can plot characteristics, such as luminous intensity versus current (see Figure 12-1).

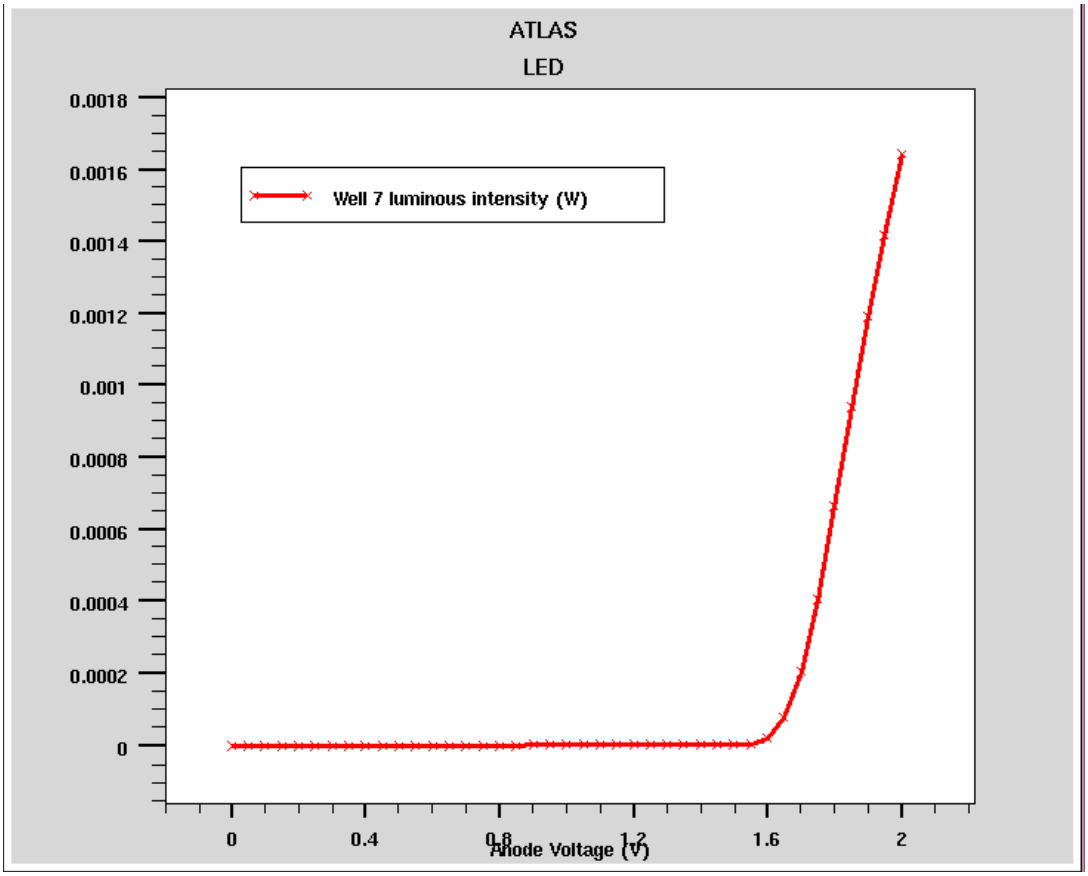


Figure 12-1: Example Luminous Intensity Plot

Note: Luminous intensity is calculated by integrating the luminous spectrum obtained by models containing spectral content (e.g., **MODELS** WZ.KP SPONT). This does not include other non-spectral models (e.g., **MODELS** OPTR).

12.4.2 Extracting Emission Spectra

The LED simulator can save multiple spectrum files from the **SOLVE** statement. To save a spectrum file, specify the SPECTRUM parameter after each solution. SPECTRUM is assigned to the file name prefix. Each subsequent file will be written to a file named by SPECTRUM and appended with a version number starting at 0.

The LED simulator can output spectral intensity as a function of energy and wavelength integrated over all wells specified as LEDs. To output spectral intensity, specify the SPECTRUM parameter of the **SAVE** statement. SPECTRUM is assigned to the name of a file that the LED spectrum is written.

You must also specify either LMIN and LMAX or EMIN and EMAX. LMIN and LMAX specify the minimum and maximum values of wavelength in microns to be captured. EMIN and EMAX specify the minimum and maximum values of energy in eV to be captured. Specify either energy range or wavelength range but not both. To capture the number of descrete samples, specify the NSAMP parameter of the **SAVE** statement.

Figure 12-2 shows an example spectrum file plotted in TonyPlot.

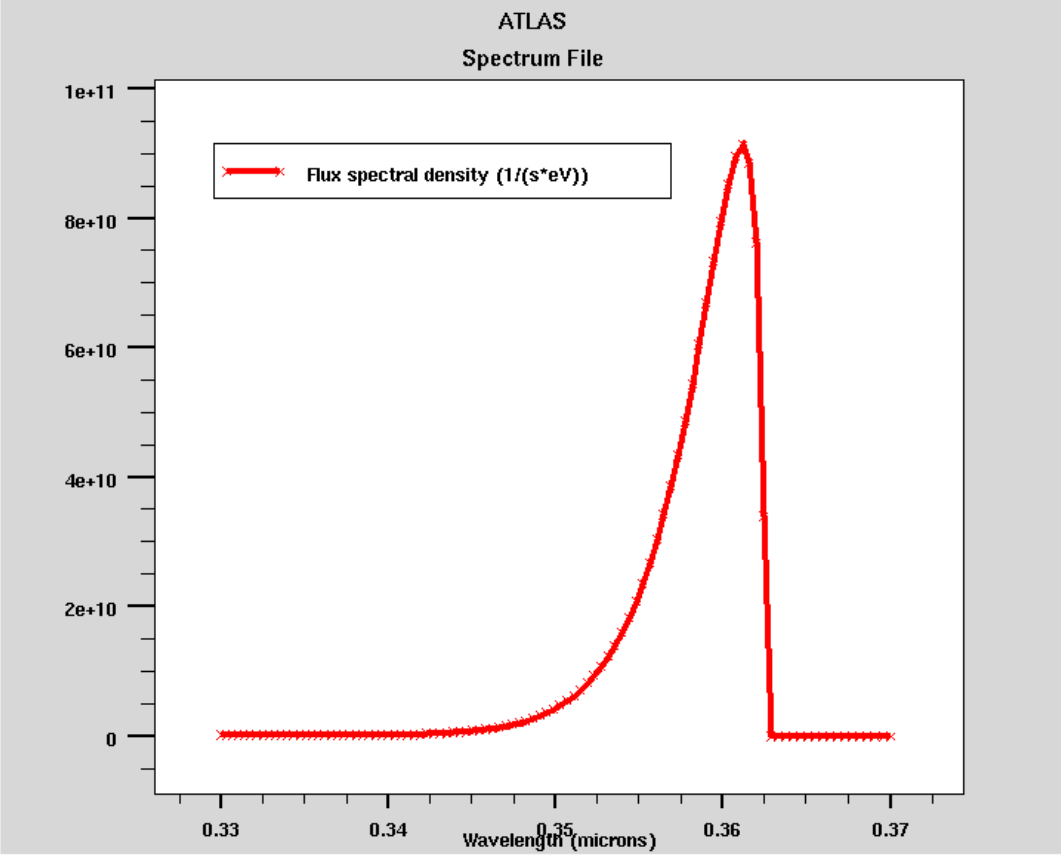


Figure 12-2: Plot of Spectrum File for LED

Note: Emission spectra can only be obtained using a spectral model (e.g., MODELS WZ.KP SPONT).

12.4.3 Extracting Emission Wavelength

In the LED simulator, you can use the probe to measure wavelength for LED devices. This will extract the wavelength from the peak of the spectral response over a specified range. To use this capability, specify WAVELENGTH in the **PROBE** statement. Use either EMIN and EMAX or LMIN and LMAX in the **PROBE** statement to specify the search range. EMIN and EMAX specify the range in terms of energy in eV, while LMIN and LMAX specify the range in terms of wavelength in microns.

12.5 Optical Scattering Matrix Method

Electromagnetic propagation can be described by power waves, so that the power being transmitted is given by

$$P = \alpha \cdot \alpha^* \quad \text{12-8}$$

Here, we define α waves as traveling in the positive direction and b waves as traveling in the negative direction. The optical scattering matrix describes how the power waves out of a region depend upon the power waves incident on the region

$$\begin{pmatrix} a_2 \\ b_1 \end{pmatrix} = \begin{pmatrix} S_{aa} & S_{ab} \\ S_{ba} & S_{bb} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ b_2 \end{pmatrix} \quad \text{with} \quad \begin{matrix} a_1 \rightarrow \\ \leftarrow b_1 \end{matrix} \left| \begin{pmatrix} S_{aa} & S_{ab} \\ S_{ba} & S_{bb} \end{pmatrix} \right| \begin{matrix} a_2 \rightarrow \\ \leftarrow b_2 \end{matrix} \quad \text{12-9}$$

This is a 1D method, the device is assumed to be planar, the primary direction of emission is perpendicular to this plane. Normally, the device is expected to be planar in the xz-plane and the primary direction of emission is parallel to the y-axis.

Light emission, such as exciton decay, is modeled as a damped, driven, Hertzian dipole.

$$\frac{d^2 p}{dt^2} + \beta \cdot \frac{dp}{dt} + \omega^2 p = \frac{e^2 E}{m} \quad \text{12-10}$$

The damping coefficient, β, is written as the sum of the damping due to radiative loss and the damping due to non-radiative loss (where κ is the quantum efficiency).

$$\beta = \kappa\beta + (1-\kappa)\beta \quad \text{12-11}$$

We expect the emission term to be affected by the interaction of the dipole to the electromagnetic waves being reflected inside the device, but the non-radiative loss to be unaffected.

$$\beta = \kappa_0\beta_0 D + (1-\kappa_0)\beta_0 \quad \text{12-12}$$

κ_0 and β_0 are the values for the dipole assuming no reflection (i.e., assuming the dipole were in an infinite medium). The scaling factor D is written as

$$D = \int_0^\infty (d(k_{xz}))dk_{xz} \quad \text{12-13}$$

where k_{xz} is the wavevector in the xz-plane (where when kxz>k we are looking at evanescent modes).

12.5.1 Using the Optical Scattering Method

The first parameter that needs to be calculated is the scaling factor D. This is done with a command of the form

```
SAVE X=0.5 Y=0.12 L.WAVE=0.5 OPDOS=ignore PARA.INTEGRAL=dkxz.dat
```

`X` and `Y` give the location (in um) of the dipole. `L.WAVE` gives the wavelength (in um) of the em-radiation we are simulating. The `OPDOS` parameter indicates we want to calculate the D factor. (Normally, the value of `OPDOS` is the filename to save the data to, but `ignore` means we don't want to save the data). The `INTEGRAL` parameter gives the file to save the `d(kxz)` data to. An example of this data is shown in Figure 12-3.

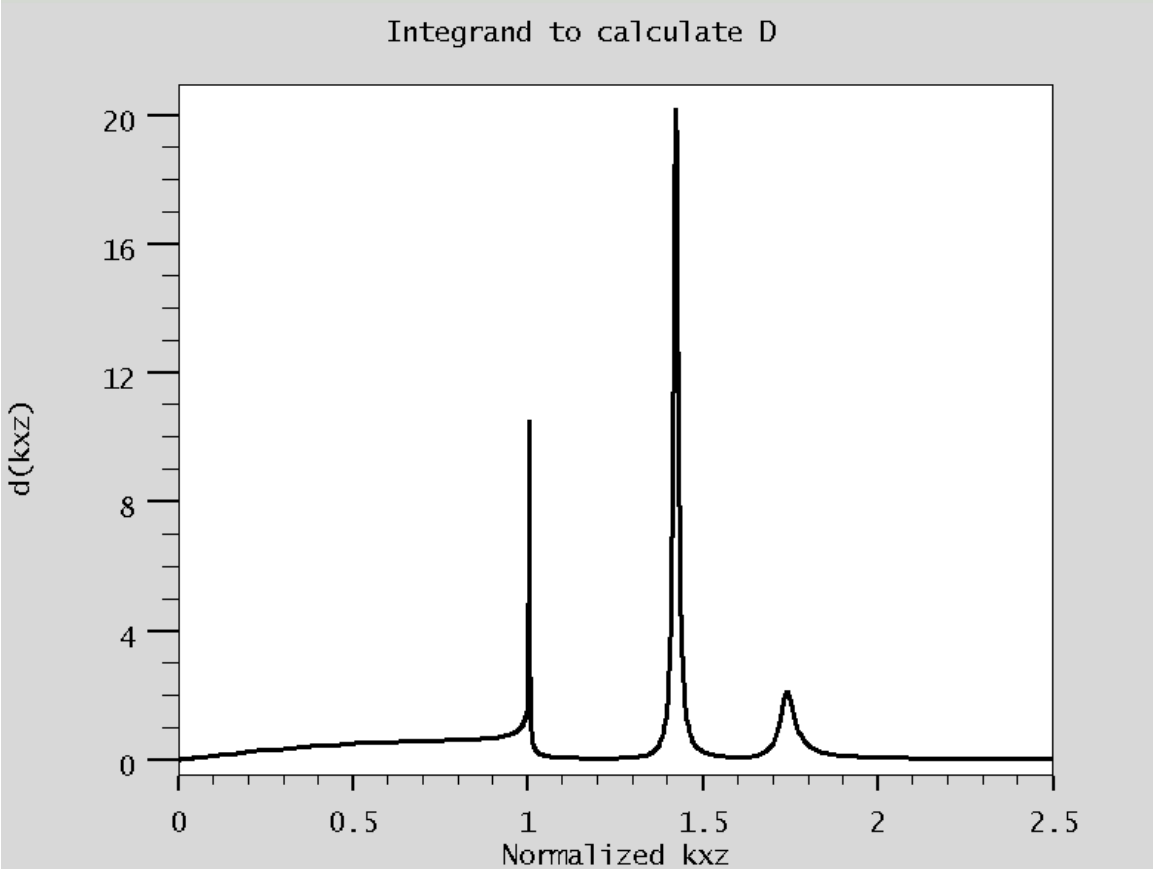


Figure 12-3: *d*(kxz) as a function of kxz

We are more interested in the integral of this data, so the more normal command would be something like

```
SAVE  X=0.5  Y=0.12  LMIN=0.36  LMAX=0.83  T.INIT=0.01  T.MIN=0.01
T.REL=0.02 OPDOS=d.dat
```

The `LMIN` and `LMAX` parameters give the range of wavelengths to calculate *D* over (the `T.INIT`, `T.MIN`, `T.REL` parameters define how many wavelengths to calculate *D* at, 10 nm steps in this instance, and will be omitted from many of the remaining examples), the results are saved to the `d.dat` file.

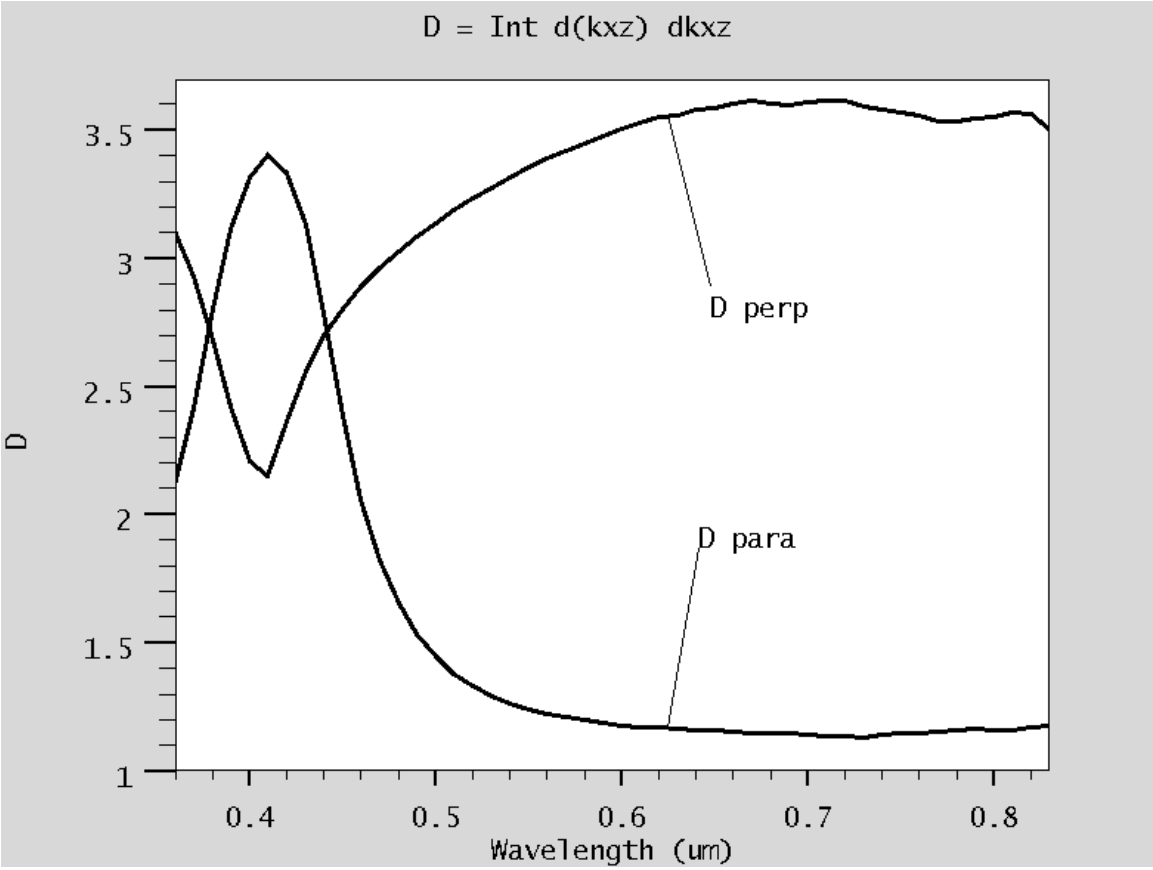


Figure 12-4: *D*(λ) as a function of λ

The data is calculated for dipoles parallel to the kxz plane, and for dipoles perpendicular to the kxz plane.

We are interested in the light that is emitted from the device. `1.YIELD` is the total power emitted by the device (assuming unity emission by the dipole) between 0 degrees and a user-defined angle. `1.FRAC.YIELD` is the total power emitted by the device in a 2 degrees cone centered on a user-defined angle. For example

```
SAVE X=0.5 Y=0.12 LMIN=0.36 LMAX=0.83 1.YIELD=1.yield.dat
ANGLE.OUT=45

SAVE X=0.5 Y=0.12 LMIN=0.36 LMAX=0.83 1.FRAC.YIELD=1.frac.yield.dat
ANGLE.OUT=45
```

Both the `1.YIELD` and `1.FRAC.YIELD` are calculated for dipoles both parallel and perpendicular to the xz-plane, but the emission from the perpendicular dipoles is generally small. The `1.YIELD` data is shown in Figure 12-5.

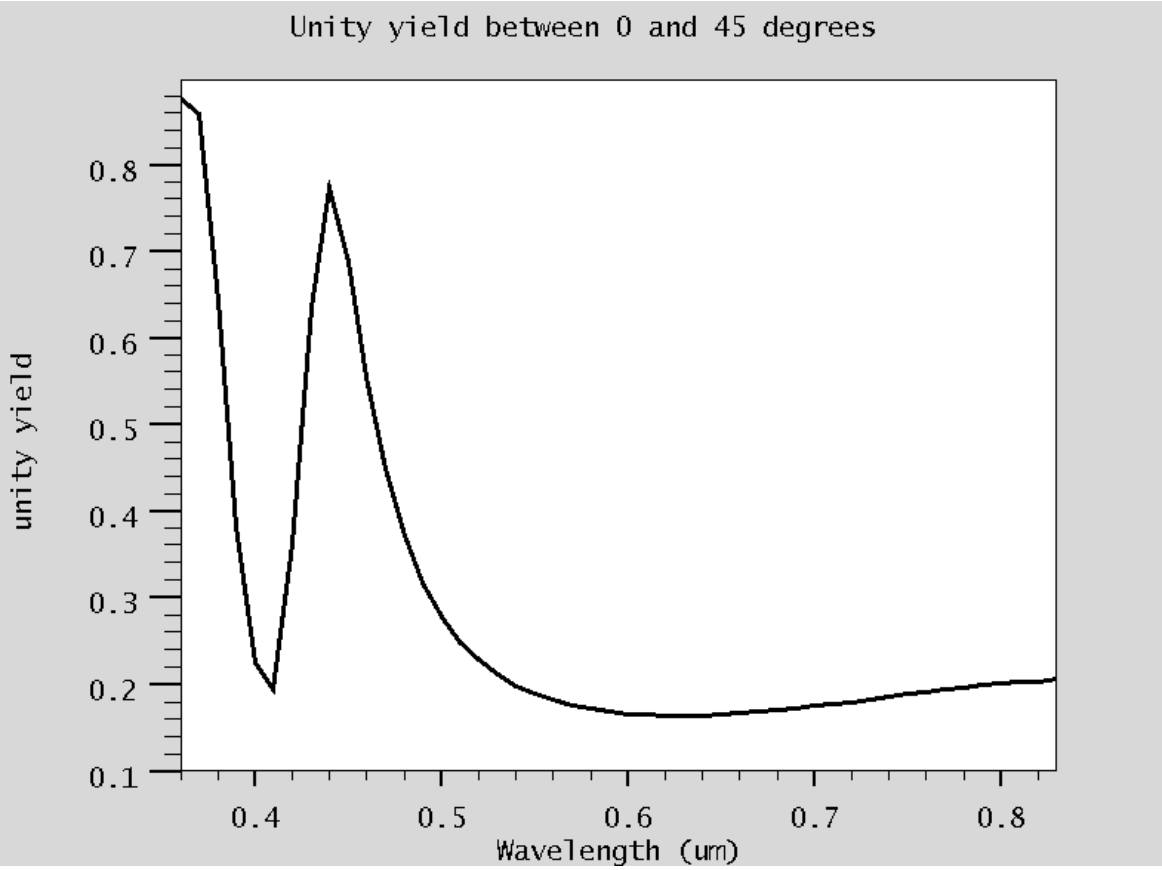


Figure 12-5: 1.YIELD data for dipoles parallel to the xz-plane as a function of λ

The OPDOS, 1.YIELD, and 1.FRAC.YIELD results are usually a function of the device structure but not the bias conditions. Therefore, this data can be calculated once (for each device, dipole position, and output angle) and can be read in for any subsequent simulations with the READ.OPDOS, READ.1.YIELD, and READ.1.FRAC.YIELD parameters on the save command.

The actual power emitted by the device as a function of bias is of most interest. This is calculated with

```
SOLVE VANODE=1
SAVE X=0.5 Y=0.12 LMIN=0.36 LMAX=0.83 ANGLE.OUT=45 \
     YIELD=yield_1.dat FRAC.YIELD=fyield_1.dat USER.SPECT=emission.dat
SOLVE VANODE=2
SAVE X=0.5 Y=0.12 LMIN=0.36 LMAX=0.83 ANGLE.OUT=45 \
     YIELD=yield_2.dat FRAC.YIELD=fyield_2.dat USER.SPECT=emission.dat
```

See Section 12.8.1 “User-Definable Emission Spectra” for information on the USER.SPECT parameter.

Normally, the layer that emits light has a finite thickness so we need to solve this data at several Y positions and integrate the results to get the total power emitted by the device. The usual form of these commands would be

```
SOLVE VANODE=1
SAVE X=0.5 YMIN=0.1 YMAX=0.15 YNUM=6 \
            LMIN=0.36 LMAX=0.83 T.INIT=0.01 Y.MIN=0.01 T.REL=0.02 \
            ANGLE.OUTPUT=45 USER.SPECT=emission.dat \
            INT.YIELD=int_yield_1.dat INT.FRAC.YIELD=int_frac_yield_1.dat
```

where INT.YIELD and INT.FRAC.YIELD save the YIELD and FRAC.YIELD data integrated over the calculated Y positions.

Table 12-1 Main Parameters Used For Optical S-Matrix Calculations.

Statement	Parameter	Type	Default	Units
SAVE	X	Real	0	um
SAVE	Y	Real	0	um
SAVE	YMIN	Real		um
SAVE	YMAX	Real		um
SAVE	YNUM	Integer		
SAVE	L.WAVE	Real	0.8	um
SAVE	LMIN	Real		um
SAVE	LMAX	Real		um
SAVE	OPDOS	Character		
SAVE	INT.OPDOS	Character		
SAVE	READ.OPDOS	Character		
SAVE	1.YIELD	Character		
SAVE	1.INT.YIELD	Character		
SAVE	READ.1.YIELD	Character		
SAVE	1.FRAC.YIELD	Character		
SAVE	1.INT.FRAC.YIELD	Character		
SAVE	READ.1.FRAC.YIELD	Character		
SAVE	YIELD	Character		
SAVE	INT.YIELD	Character		
SAVE	FRAC.YIELD	Character		
SAVE	INT.FRAC.YIELD	Character		

Table 12-1 Main Parameters Used For Optical S-Matrix Calculations.				
SAVE	PARA.INTEGRAL	Character		
SAVE	INT.PARA.INTEGRAL	Character		
SAVE	PERP.INTEGRAL	Character		
SAVE	INT.PERP.INTEGRAL	Character		
SAVE	ANGLE.OUTPUT	Real	30	
SAVE	F.RHO.DIPOLE	Character		
SAVE	T.RHO.DIPOLE	Character		
SAVE	USER.SPECT	Character		
SAVE	N.SURFACE	Real	1	
SAVE	K.SURFACE	Real	0	
SAVE	N.SUBSTRATE	Real	1	
SAVE	K.SUBSTRATE	Real	0	
SAVE	MAT.SURFACE	Character		
SAVE	MAT.SUBSTRATE	Character		
SAVE	D.ORIENT	Real	0	
SAVE	PRINT	Logical	False	
SAVE	PRINT.EVERY	Logical	False	
SAVE	REAL.RI	Logical	False	
SAVE	INT.QEFF	Real	1	

INT.QEFF is the internal quantum efficiency of the dipoles. This is the κ_0 parameter in Equation 12-12.

F.RHO.DIPOLE gives the file that contains a C-Interpreter function "rhodipole" that gives the dipole concentration as a function of position. This allows you to calculate the total optical output from a device without having to do an electrical simulation. If the C-Interpreter function is used the dipole profile from an electrical simulation are ignored.

T.RHO.DIPOLE gives a file that contains tabulated dipole concentration as a function of position. The file should contain two columns: the first column is a position in microns, and the second column is the dipole concentration in cm^{-3} at this position. This allows you to calculate the total optical output from a device without having to do an electrical simulation. If a table is used, the dipole profile from an electrical simulation is ignored.

N.SURFACE, K.SURFACE, or MAT.SUBSTRATE are ways to define the refractive index of the surface of the device. N.SUBSTRATE, K.SUBSTRATE, or MAT.SUBSTRATE are ways to define the refractive index of the substrate of the device.

D.ORIENT is a parameter to define the orientation of the dipoles in the calculation of YIELD and FRAC.YIELD. The final results are calculated with

$$Y = \frac{2}{2 + \text{D.ORIENT}} Y_{para} + \left(1 - \frac{2}{2 + \text{D.ORIENT}}\right) Y_{perp} \qquad 12\text{-}14$$

If D.ORIENT=1, then $Y=(2/3)Y_{para}+(1/3)Y_{perp}$ (which is equivalent to random dipoles). If D.ORIENT=0, then $Y=Y_{para}$ (which is equivalent to all dipoles in the xz-plane).

The PRINT parameter outputs the 1D structure being solved by the optical simulation. This gives the surface, substrate, and intermediate layers with their thicknesses and the material of each layer.

The PRINT.EVERY parameter outputs the 1D structure for each particular wavelength being solved. This outputs the refractive index for each layer.

The REAL.RI parameter forces the materials within λ/50 of the dipole to have a real refractive index.

Table 12-2 Tuning Parameters for Optical S-Matrix Calculations

Statement	Parameter	Type	Default	Units
SAVE	I.RELERR	Real	1.00E-005	
SAVE	I.MINI	Real	0	
SAVE	I.MAXDOUBLE	Real	10	
SAVE	I.DXMIN	Real	1.00E-005	
SAVE	DOS.MAXN	Real	100	
SAVE	T.INITSTEP	Real	0.01	um
SAVE	T.MINSTEP	Real	0.01	um
SAVE	T.RELERR	Real	0.02	
SAVE	OPSM.POW1	Logical	FALSE	
SAVE	OPSM.OP1	Integer	1	
SAVE	OPSM.OP2	Integer	0	
SAVE	OPSM.RESET	Logical	FALSE	

These are various parameters used to tune the behavior of the optical s-matrix calculations.

The I.RELERR, I.MINI, I.MAXDOUBLE, I.DXMIN, and DOS.MAXN parameters are used to control the integral in the calculation of OPDOS and 1.YIELD (at a fixed wavelength, dipole position, and output angle). DOS.MAXN is the upper limit of the integral to calculate *D* (in units of normalized wavevector). They are used to determine whether to add more kxy points to calculate the integral. The lower I.RELERR, the more points. I.DXMIN determines the minimum kxy step between points.

The T.INITSTEP, T.MINSTEP, and T.RELERR parameters are used to control how many wavelengths are calculated between LMIN and LMAX. T.INITSTEP is the initial step used in the table. T.RELERR gives the minimum allowed error (between the actual calculation and a linear interpolation between adjacent points). If the current error is bigger, then more points are added. However, T.MINSTEP is the minimum allowed step in the table. If the two wavelengths are already closer than this, then no extra points are added.

OPSM.POW1, OPSM.RESET, OPSM.OP1, and OPSM.OP2 are primarily debug parameters and can be ignored. OPSM.POW1 tells Atlas to ignore the power emitted by the device and to use a unity power when calculation YIELD and FRAC.YIELD. OPSM.RESET tells Atlas to delete the internal database of past results, so they will be recalculated if needed. OPSM.OP1 and OPSM.OP2 control the output. Normally, Atlas prints an output at every wavelength that is calculated. OPSM.OP1 gives the ordinal of the first calculation to output. OPSM.OP2 gives the number of calculations that should be skipped before the next output.

12.6 Reverse Ray-Tracing

Reverse ray-tracing is a technique that allows you to obtain optical output characteristics of an active optoelectronic device based on material properties and device geometry. Direct ray-tracing is commonly used for modeling of passive optoelectronic components, such as photo-detectors.

In direct ray-tracing, rays originating at the external light source are traced into the device and are absorbed to form electron/hole pairs, which are subsequently detected (using the ray-tracing method in Luminous is described in Sections 11.2.1 "Ray Tracing in 2D" and 11.2.2 "Ray Tracing in 3D"). Unlike direct ray-tracing, the rays in the reverse method originate inside the active region and are traced until they exit the device. One or multiple origin points (user specified) for rays are considered, and interference effects for rays originating at the common source point can be taken into account. When you enable interference, you can analyze the spectral selectivity of the device structure by performing ray-tracing at multiple wavelengths.

You can assess light extraction from the device by using the **SAVE** statement (after biasing). Five different sets of parameters for reverse ray-tracing are used in the examples throughout the section to demonstrate the features available for modeling LEDs. The parameter sets are given in the order of increasing complexity of simulation. Computation times also vary. It takes approximately 200 times longer to run the last **SAVE** line compared to the first one.

Set the ANGPOWER parameter in the **SAVE** statement to start the reverse ray-tracing algorithm. The name of the output file containing the angular power density vs. output angle dependence is specified as a value for ANGPOWER. The information in the angpower file includes the angular power density for TE- and TM- polarized light and total angular power density and total flux angular density vs. output angle.

Note: In TonyPlot, polar charts show the Y axis directed upward (in the opposite direction to the internal coordinate system used in Atlas). Therefore, the plots will appear to be flipped around X axis (top of the structure is at the bottom of the chart).

The RAYPLOT parameter specifies the name of the output file containing the information on each ray exiting the device. This file is only created when single origin for all rays is assumed. The information includes ray output angle, relative ray power (TE-, TM-polarization, and total), and initial internal angle at the origin (only if INTERFERE parameter is not specified). 0° angle corresponds to the rays in the X axis direction. 90° angle corresponds to the rays in the Y axis direction.

To start ray-tracing from one point of origin, specify the origin's coordinates for rays X and Y and the wavelength L.WAVE. It is important to choose the origin in the active region of the device. Rays are not traced if the radiative recombination is zero at the ray origin. All remaining parameters outlined below are optional and their default values are assumed if the parameters are not specified.

REFLECTS specifies a number of reflections to be traced for each ray originating at the point (X,Y) (similar to REFLECTS parameter in the **BEAM** statement). The default value (REFLECTS=0) provides for a quick estimate of the coupling efficiency. To obtain a more accurate result, use REFLECTS>0, especially if you specify MIR.TOP or MIR.BOTTOM. The choice of this parameter is based on a compromise between calculation time and accuracy. The maximum allowed value is REFLECTS=10. Number of reflections set to 3 or 4 is often a

good choice. Example 1 produces a simple ray-tracing analysis of an LED. Figure 12-6 shows the resulting angular distribution of the emitted light intensity.

Example 1

```
SAVE ANGPOWER=OPTOEX18ANG_1.LOG RAYPLOT=OPTOEX18RAY_1.LOG X=2.0
Y=1.01 L.WAVE=0.9 REFLECTS=4
```

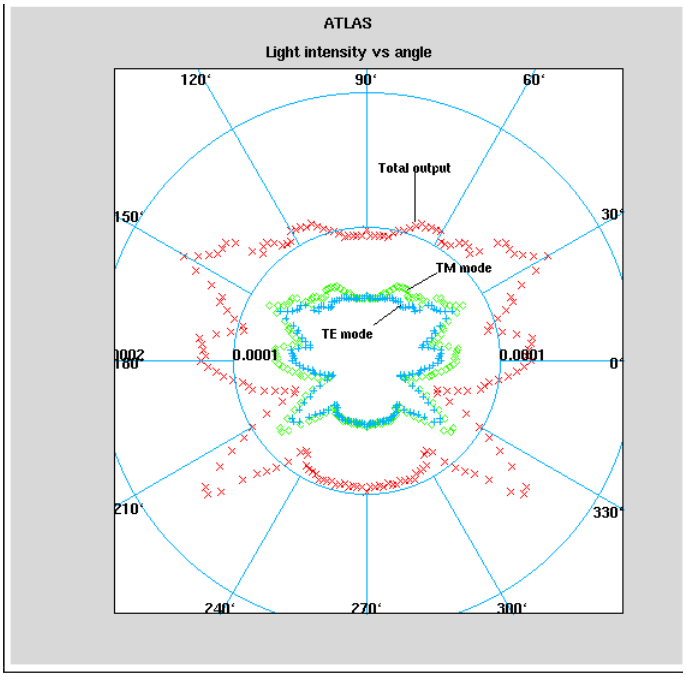


Figure 12-6: Emitted Light Intensity vs Angle for Example 1

Specify the DIPOLE parameter to turn on a particular angular distribution of the internal radiating field that corresponds to a preferred in-plane orientation of dipoles. This orientation is often found in polymer based OLEDs and results in higher optical output coupling. The default setting (DIPOLE=false) corresponds to isotropic distribution of emitting dipoles and spherically symmetric radiation pattern.

MIR.TOP specifies that the top surface of the device be treated as an ideal mirror.

MIR.BOTTOM specifies that the bottom surface of the device be treated as an ideal mirror.

SIDE specifies that the rays reaching the sides of the device terminate there and do not contribute to the total light output. This is often a good assumption for realistic LEDs as these rays tend to be either absorbed internally or blocked by the casing of the device.

TEMPER is the temperature (needed for using appropriate refractive indexes of the materials). The default setting of 300 K will be used if TEMPER is not specified.

POLAR specifies polarization of the emitted photons in degrees (linearly polarized light is assumed). Parallel (TM-mode, POLAR=0.0) and perpendicular (TE-mode, POLAR=90.0) polarizations result in significantly different output coupling values.

You should use the default value (POLAR=45.0) if there is no preferred direction of polarization of emitted photons (unpolarized light emission).

MIN.POWER specifies the minimum relative power of a ray (similar to MIN.POWER parameter in the BEAM statement). The ray is not traced after its power falls below MIN.POWER value. This is useful for limiting the number of rays traced. The default value is MIN.POWER=1e-4.

NUMRAYS specifies the number of rays starting from the origin. The default is 180. The acceptable range 36-3600.

Example 2 shows how some of the parameters described above are used for modeling of a realistic LED. The angular distribution of light power in Figure 12-7 exhibits almost a Lambertian pattern for this simple planar LED geometry. Note that optical coupling coefficient produced in this calculation reflects a 2D nature of the example (the light origin is not a point but rather an infinite line in Z direction).

Simultaneously, you can calculate the optical coupling efficiency for an axially symmetric 3D device. Normally, this is the value to be compared with experimental results. To do this, specify the COUPLING3D parameter (while SIDE is set). For this calculation, the light source is assumed to be a point located on the axis of symmetry.

Example 2

```
SAVE   ANGPOWER=OPTOEX18ANG_2.LOG   RAYPLOT=OPTOEX18RAY_2.LOG   X=2.0
Y=1.01 L.WAVE=0.9 SIDE MIR.BOTTOM NUMRAYS=360 REFLECTS=4
```

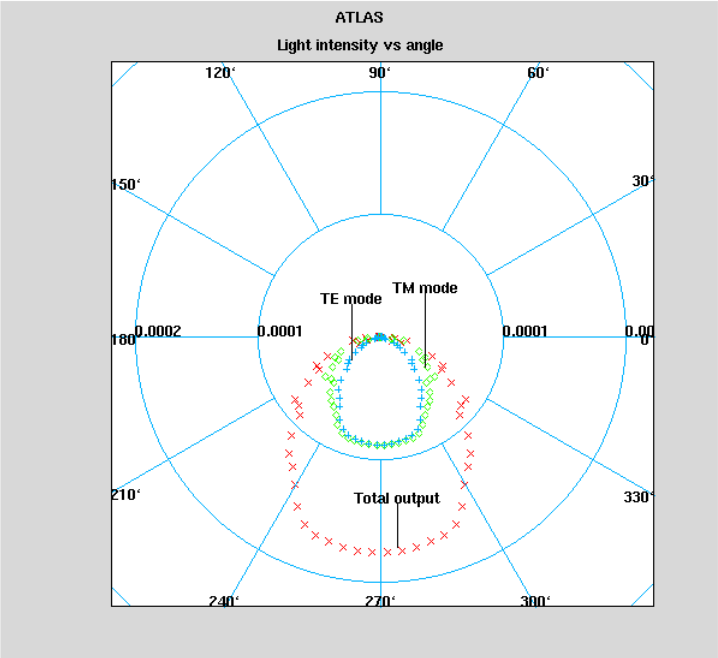


Figure 12-7: Emitted Light Intensity vs Angle for Example 2

The rays are assumed to be incoherent by default. This is a good approximation if the thickness of the active layer of the device is on the order of a wavelength/index. For layer thicknesses, smaller than the wavelength coherent effects might be important. When INTERFERE is set, the rays originating at the common source point are taken to be 100% coherent. In this case, the phase information upon propagation is preserved. Phase change upon reflection is also considered. Thus, interference of rays exiting the device at the same angle is taken into account. In this case, the internal angle information is not written to the output rayfile.

Example 3 takes into account interference. Figure 12-8 shows the result. You can take absorption of rays into account by setting the ABSORB parameter. The absorption is assumed to be specified for each material by the imaginary part of the refractive index. Carrier density dependent absorption and photon-recycling are not considered at this point.

Example 3

```
SAVE   ANGPOWER=OPTOEX18ANG_3.LOG   RAYPLOT=OPTOEX18RAY_3.LOG   X=2.0
Y=1.01 L.WAVE=0.9   INTERFERE SIDE MIR.BOTTOM REFLECTS=4
```

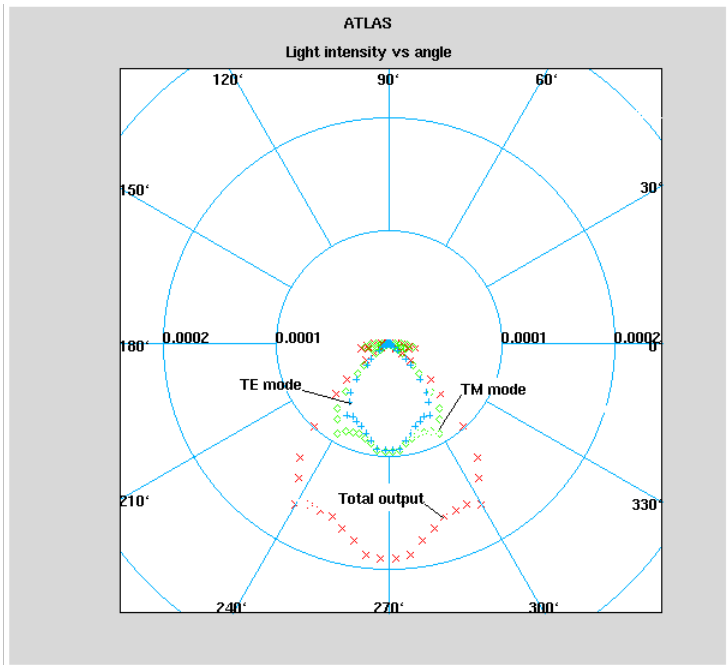


Figure 12-8: Emitted Light Intensity vs Angle for Example 3

Although ray-tracing from one point of origin can give a reasonable estimate of optical output coupling and angular distribution of light power, it is often desirable to consider multiple points within the active layer of the device to obtain more accurate results. To set multiple origin points, use the following parameters.

- XMIN, XMAX, YMIN, and YMAX define a rectangular area containing all origin points.
- XNUM and YNUM specify the number of points along x and y axes within the rectangular area. If XNUM=1, the X coordinate of all origin points will be set to XMIN so that the points are chosen along the line X=XMIN. Similarly, you can choose points along the specific y-line.

Ray-tracing from multiple origins is realized by repeating single origin algorithm for each point and by adding up the normalized angular power density values thus obtained. The luminous power assigned to each source (origin) is proportional to the radiative recombination at that point. The luminous power of all sources adds up to the value obtained by integration of radiative recombination over the entire device. Rays originating at different source points are completely incoherent (even when INTERFERE is set), which is consistent with the spontaneous character of the radiation produced by an LED. The rayplot file is not written for multiple origins (even if RAYPLOT parameter is set). Example 4 shows how multiple origin simulation can be done. Figure 12-9 shows angular distribution of light obtained in this case.

Example 4

```
SAVE ANGPOWER=OPTOEX18ANG_4.LOG XMIN=1.0 XMAX=3.0 XNUM=3 YMIN=1.01
YMAX=1.09 YNUM=9 INTERFERE SIDE L.WAVE=0.9 MIR.BOTTOM REFLECTS=4
```

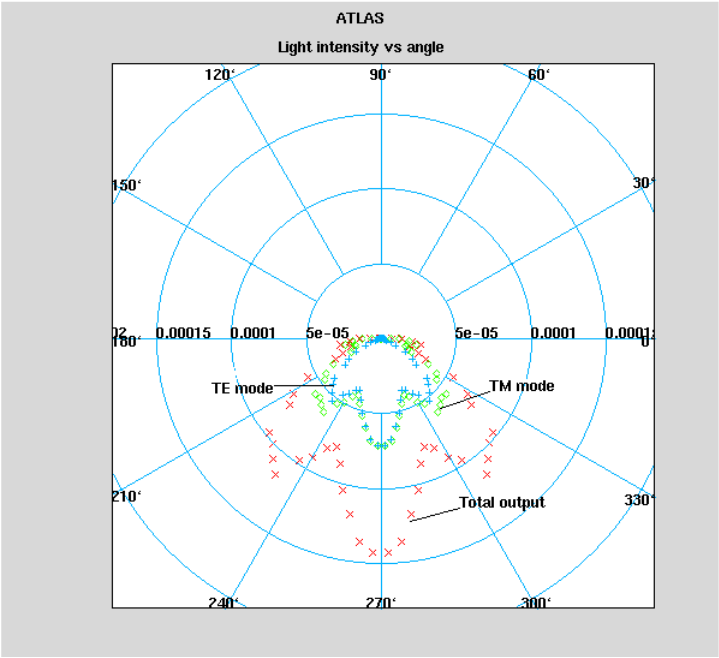


Figure 12-9: Emitted Light Intensity vs Angle for Example 4

You can also take spectral selectivity of optical output coupling for LEDs into account. Reverse ray-tracing at multiple wavelengths is considered if the SPECTRUM parameter specifies a filename for spectral selectivity output while the ANGPOWER parameter is set.

EMIN and EMAX, or LMIN and LMAX specify the energy or wavelength range respectively.

NSAMP specifies the number of spectral components to be considered. We suggest that you specify the SPECTRUM file only when INTERFERE is set. Otherwise, a warning will be issued during ray-tracing. When ANGPOWER and SPECTRUM are set in the **SAVE** statement, the resulting optical output coupling is averaged over the entire energy (wavelength) range from EMIN to EMAX (from LMIN to LMAX). The same applies to the quantities in the output angular distribution file. The spectrum file only shows how output coupling changes with wavelength. Currently, the shape of the gain curve is not taken into account (flat gain).

Example 5 shows how to use multiple spectral components. Figure 12-10 shows that the results obtained after averaging over spectral components and multiple origin points while taking interference into account are similar to the single point source model, where interference and multiple spectral content are ignored (Example 2). This demonstrates the applicability of a simpler model in this case.

Example 5

```
SAVE ANGPOWER=OPTOEX18ANG_5.LOG SPECTRUM=OPTOEX18SP_5.LOG XMIN=1.0
XMAX=3.0 XNUM=3 YMIN=1.01 YMAX=1.09 YNUM=5 INTERFERE SIDE LMIN=0.86
LMAX=0.94 NSAMP=15 MIR.BOTTOM REFLECTS=4
```

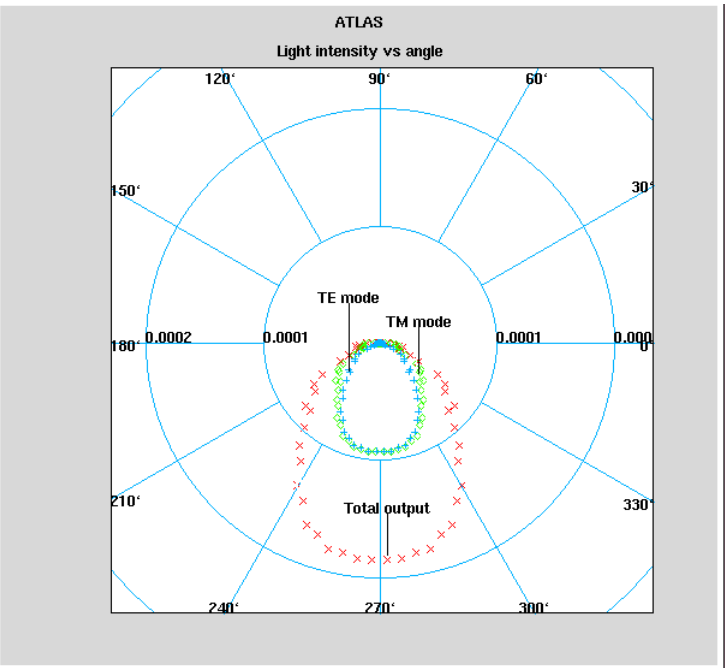


Figure 12-10: Emitted Light Intensity vs Angle for Parameter Set 5

12.7 FDTD Analysis of Output Coupling

Finite difference time domain analysis is well suited to calculation of output coupling from periodic structures, such as LED with photonic-crystal coupling. Performing such analysis is only a small deviation from the methods discussed in the previous section.

First, you must remember that you are going to simulate a periodic device by applying periodic boundary conditions. These boundary conditions appear in two places: the device mesh and the FDTD mesh. Therefore, you need to make sure that the two meshes both have the same periodicity. If you cannot define such a device, you cannot take advantage of periodicity.

To model devices as periodic in the device simulator, simply add the PERIODIC parameter to the **MESH** statement at the beginning of the simulation deck like this:

```
MESH PERIODIC
```

or

```
MESH PERIODIC THREE.D
```

Next, you can either load a device structure or syntactically create a structure. But in both cases, you need to make sure that the device and any optical elements defined above the semiconductor device are periodic in the X and Z directions if desired.

You also need to include a **BEAM** statement. The **BEAM** statement is used to specify the optical problem. The application of the FDTD method along with the **BEAM** statement is discussed fully in Section 11.5 “Finite Difference Time Domain Analysis”.

Your **BEAM** statement can be easily composed. First, you define the BEAM origin and direction of propagation. Currently, this is always done for the Luminous simulator.

The BEAM origin should have the Y coordinate above the device with enough space to fit a plane to evaluate near and far field patterns. Using a coordinate of 1/4 of the smallest wavelength is adequate for that space. Your BEAM statement should start as follows:

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
```

Here, the X and Z origins do not matter and the Y origin is set, assuming that your device starts at Y=0. The angle ensures the domain is properly oriented with Y pointing down.

Next, you need to include some parameters for defining the discretization. These parameters should be defined like this:

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
     FD.AUTO TD.SRATE=20 PROP.LENG=4 BIG.INDEX TD.ERR=0.01 \
```

Here, we are saying that the spatial mesh must be sample at least 20 times per wavelength (TD.SRATE=20). This is the number suggested in literature and we have found it works well. Please be aware that computation time goes up as the square of this number in 2D and the cube in 3D.

Next, we are going to shine the light for an amount of time needed to propagate 4 microns. This is done in a number of discrete time steps. This distance is very much related to the maximum dimension of your device. In this case, we are saying that our device is of the order of 4 microns in size. Actually, here we also need to consider our PMLs that should be of the same order in size as our device and account for multiple reflections. So a good number might be 4 or 5 times the maximum dimension of your device.

The next parameter accounts for the fact that the speed of light in media is slower than in vacuum. This means the calculations based on given by `TD.SRATE` and `PROP.LENG` are automatically multiplied by the largest index of refraction in the device at the wavelength of operation.

The parameters `TD.SRATE`, `PROP.LENG`, and `BIG.INDEX` control the absolute size of the problem. The `TD.ERR` parameter specifies a convergence error that may be met before `PROP.LENG` is reached and end the simulation early. You should start out with this set at 0.01 and look at the error log discussed next to determine if you should change that value.

The residual error is calculated as the integrated variation of the envelope of all fields between two successive wavelengths (cycles) normalized by the integrated envelope.

The next line shows how you output a error log.

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
     TD.SRATE=20 PROP.LENG=4 BIG.INDEX TD.ERR=0.01 \
     TD.LOG=example \
```

This will cause the error log to be output to the file `"example.log"`. The following figure shows an error log in TonyPlot.

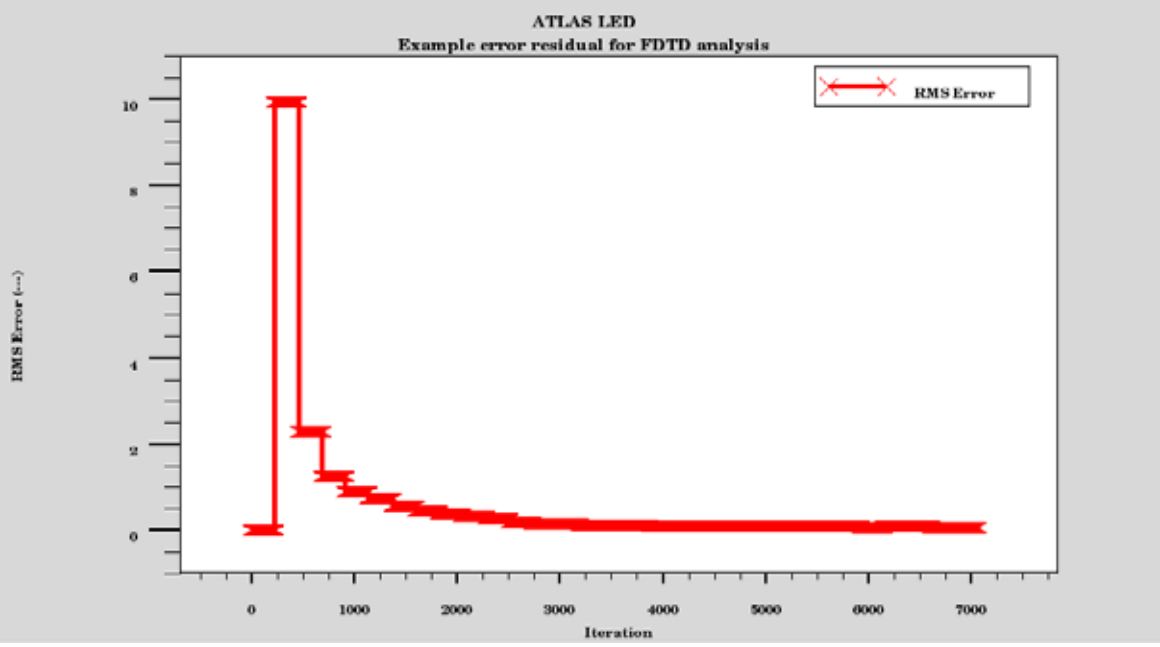


Figure 12-11: Error Log

The level steps are due to the evaluation only occurring every 1 wavelength (cycle). The first is zero since there is no previous envelope to compare with. The error log also can be used to display the absorption in the various PMLs and bulk vs. iteration. Plotting these data can be very useful in optimizing your experiment for speed and accuracy.

The next line shows you how you can extract structure files containing the complete field solutions produced from FDTD.

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
     TD.SRATE=20 PROP.LENG=4 BIG.INDEX TD.ERR=0.01 \
     TD.LOG=example \
     TD.FILE=example SUBNX=100 SUBNY=100 SUBNZ=100 \
```

Here, we make the simulator output a structure file to the name "example.str" subsampled to 200×200×200 samples. We highly recommend you subsample your output files until you see what your system can handle. This structure file contains all the field information, structural information including the optical elements such as lenslets and PMLs, and the optical intensity and photogeneration rate.

You can extract cutplanes through the structure file interface by specifying the TD.ZCUT or TD.XCUT parameters.

You can assign the FACET parameter to a file root name to save the near and far field patterns normal to an axis as defined by the X.FACET, Y.FACET, or Z.FACET parameters. Such an assignment may look like this:

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
     TD.SRATE=20 PROP.LENG=4 BIG.INDEX TD.ERR=0.01 \
     TD.LOG=example \
     TD.FILE=example SUBNX=100 SUBNY=100 SUBNZ=100 \
     FACET=example Y.FACET=-0.2/4/2 \
```

Here, we intend to write two files "example.nfp" and "example.ffp" containing the near field pattern and the far field pattern. The far field pattern is constructed from the 2D fft of the near field pattern for 3D and the 1D fft for 2D.

You will notice we placed the output facet in between the top of the device and the BEAM origin as defined earlier.

Finally, we need to specify where the output is to be evaluated. Usually, we consider light flowing out the top to be successfully coupled to the output. We specify this conditon as follows:

```
BEAM NUM=1 FDTD X.ORIGIN=0 Z.ORIGIN=0 Y.ORIGIN=-0.2/4 ANGLE=90 \
     TD.SRATE=20 PROP.LENG=4 BIG.INDEX TD.ERR=0.01 \
     TD.LOG=example \
     TD.FILE=example SUBNX=100 SUBNY=100 SUBNZ=100 \
     FACET=example Y.FACET=-0.2/4/2 \
     OUT.TOP
```

This basically means that all the energy absorbed in the TOP PML is compared against the total energy absorbed to calculate the output coupling. You may name any PML this way and can include more than one PML. That is all you need for the **BEAM** statement.

Next, we need to define the PMLs. PMLs should not intimidate you and they are used for three reasons:

- They absorb everything for a given wavelength. There is no reflection if the indexes are properly matched.
- The absorption in PMLs is integrated to evaluate how much energy leaves the device through that face. This is compared with the absorption integrated in the bulk to give output coupling.
- If your PMLs are not properly designed (i.e., index matched at the boundaries), you will see increasing errors. This usually means you are introducing more energy from the source than is allowed to escape through the PMLs or to be absorbed in the bulk.

For now, keep it simple and define your PMLS to have a width of about size of the device in that direction and assign the following two parameters:

```
DEGREE=1 R90=0.0001
```

You must define either the index or the material that the PML abuts (is coincident along the edge of) in the device domain. This index must match or the interface will produce reflections. The index is specified using the REAL.INDEX parameter or material is specified using the MATERIAL parameter. The index match must be exact over all wavelengths.

Next, we can assign various lenslets and or photonic crystal elements on top of the structure to augment output coupling. Such descriptions are described in Section 11.10 “Specifying Lenslets, Texturing, and Photonic Crystals”.

Finally, we specify the extraction of output coupling on the SAVE statement just like we do for reverse ray tracing (see Section 12.6 “Reverse Ray-Tracing”) except we replace the assignment of ANG.POWER with an assignment of REF.BEAM to the BEAM index specified by the NUMBER parameter in the previous **BEAM** definition

This triggers evaluation of the output coupling that can be examined in the file specified by the SPECTRUM parameter of the **SAVE** statement.

12.8 The LED Statement

The LED statement can be used as a convenient method to perform all of the above mentioned **SAVE** operations automatically after each solution step.

The parameter names are the same as the individual **SAVE** statements. By default, the files named by the SPECTRUM, ANGPOWER and SPECT.ANGLE parameters of the **LED** statement will given unique incremented names at each save. This may generate many files. If you want to save only the last version of each file, specify the ONEFILE parameter of the **LED** statement.

12.8.1 User-Definable Emission Spectra

For some materials (particularly organics), the radiative models for emission spectra may be inadequate. In these cases, you can specify up to two user-definable emission spectra. These emission spectra are defined using a simple tabular description and is stored in a file separate from the input deck. The file contains first a line containing one number representing the number of wavelength-intensity pairs to follow. The wavelength-intensity pairs are lines with two numbers in them, the wavelength in microns and the relative intensity in Watts. The absolute intensity is not important since the intensity values are normalized.

The following file fragment demonstrates the format:

```
12
0.4 .00142
0.42 0.00222
0.44 0.013
.
.
.
```

Here, the first line contains the number of samples, in this case 12. Then next line specifies that at 0.4 microns wavelength the relative intensity is 0.00142 Watts etc.

To use the user-defined spectra, assign the values of USER.SPECT or DOPE.SPECT or both to the appropriate file name. To output the user and dopant spectra to TonyPlot compatible log files, specify OUT.USPEC and OUT.DSPEC.

Chapter 13
MixedMode: Mixed Circuit and Device Simulator

13.1 Overview

MixedMode is a circuit simulator that can include elements simulated using device simulation and compact circuit models. It combines different levels of abstraction to simulate relatively small circuits where compact models for single devices are unavailable or sufficiently accurate. MixedMode also allows you to also do multi-device simulations. MixedMode uses advanced numerical algorithms that are efficient and robust for DC, transient, small signal AC and small signal network analysis.

MixedMode is typically used to simulate circuits that contain semiconductor devices for accurate compact models that don't exist or circuits where devices play a critical role must be modeled accurately. Applications of MixedMode include: power circuits that may include diodes, power transistors, IGBTs, and GTOs, optoelectronic circuits, circuits subject to single event upset, thin film transistor circuits, high-frequency circuits, precision analog circuits, and high performance digital circuits.

MixedMode circuits can include up to 200 nodes, 300 elements, and up to ten numerical simulated Atlas devices. These limits are reasonable for most applications. But, they can be increased in custom versions on request to Silvaco. The circuit elements that are supported include dependent and independent voltage and current sources as well as resistors, capacitors, inductors, coupled inductors, MOSFETs, BJTs, diodes, and switches. Commonly used SPICE compact models are available. The SPICE input language is used for circuit specification.

This chapter describes circuit simulation capabilities rather than device simulation capabilities. The first part of the chapter contains introductory and background information. Then, describes presents and explains MixedMode syntax. This is followed by some sample input decks. The final sections contain a statement reference and a detailed description of the provided electrical compact models for diodes, BJTs, and MOSFETs.

13.1.1 Background

Circuit simulators such as SPICE [11] solve systems of equations that describe the behavior of electrical circuits. The devices that are of interest to circuit designers are normally well characterized. Compact or circuit models are analytic formulae that approximate measured terminal characteristics. Advanced compact models provide high accuracy with minimum computational complexity. Device modeling, device characterization and parameter extraction are concerned with the development and use of accurate and efficient compact models.

Physically based device simulation solves systems of equations that describe the physics of device operation. This approach provides predictive capabilities and information about the conditions inside a device. It can, however, require significant amounts of CPU time. Information is usually transferred from device simulation to circuit simulation as follows: electrical characteristics are calculated using a physically-based device simulator. These calculated electrical characteristics are then used as input by a device modeling and parameter extraction package such as Utmost [319]. The extracted parameters are used to characterize a compact model used by the circuit simulator.

This approach is adequate for many purposes but has limitations. It requires that satisfactory compact models already exist. The use of compact models always introduces some error. Models that are adequate for digital circuit simulation may be inadequate for other applications. Applications and devices for which compact modeling is not always satisfactory include: precision low power, high power, high frequency circuit simulation, SOI, IGBT, GTO, TFT, and optoelectronic devices.

13.1.2 Advantages of MixedMode Simulation

The limitations of compact models can be overcome by using physically-based device simulation to predict the behavior of some of the devices contained in a circuit. The rest of the circuit is modeled using conventional circuit simulation techniques. This approach is referred to as mixed-mode simulation, since some circuit elements are described by compact models, and some by physically-based numerical models.

MixedMode simulation provides several worthwhile advantages. No compact model need be specified for a numerical physically-based device. The approximation errors introduced by compact models can be avoided particularly for large signal transient performance. You can also examine the internal device conditions within a numerical physically-based device at any point during the circuit simulation. But the cost is increased CPU time over SPICE as CPU time is comparable to a device simulation excluding the external circuit nodes. MixedMode simulation normally uses numerical simulated devices typically only for critical devices. Non-critical devices are modeled using compact models.

13.2 Using MixedMode

Input file specification for MixedMode is different in many respects to the rest of Atlas. But if you're familiar with SPICE and Atlas syntax, you should have little difficulty understanding how these two syntax styles are joined in MixedMode.

Each input file is split in two parts. The first part is SPICE-like and describes the circuit netlist and analysis. The second part is Atlas-like and describes the device simulation model parameters. These two sections of an input file are separated as described in the next section.

The circuit description includes the circuit topology (called the netlist) and the electrical models and parameters of the circuit components. The simulation conditions specify the types of analysis to be performed. These items are described using syntax based on SPICE.

The Atlas device descriptions provide information about device geometry, doping distribution, and meshes. Device descriptions can be prepared using the built-in Atlas syntax, the Athena process simulator, or the DevEdit structure specification and meshing tool. You can also read-in previously calculated device solutions. The device data is read-in from standard structure format files.

When a simulation has finished, the following information is available:

- I-V data (voltages in all circuit nodes and currents in all circuit branches).
- Internal distributions of solution variables (such as electron, hole, and potential distributions) within the numerical devices.

The results of previous runs of MixedMode can be used as initial guesses for future simulations. This is particularly helpful when multiple simulations must be performed from the same starting point.

The accessing and running of examples for Atlas are documented in the DeckBuild User's Manual. We recommend that you run at least one MixedMode example provided on the distribution package before trying their own simulations.

13.2.1 General Syntax Rules

The SPICE-like part of any MixedMode input file starts with the **.BEGIN** statement. The SPICE-like part of the input file ends with **.END**. All parameters related to the device simulation models appear after the **.END** statement. To start MixedMode 3D, specify the 3D parameter in the **.BEGIN** statement.

The first non-comment statement after initializing Atlas (`go atlas`) has to be **.BEGIN**. The order of the following netlist and control statements is arbitrary. The last SPICE-like statement has to be **.END**.

Unlike the rest of Atlas, for SPICE-like statements the `exact` command has to be used, unique abbreviations are not accepted. Statements are not case sensitive.

There has to be at least one numerical Atlas device ("A" device) within the netlist.

Comment characters are # and $, but not *.

All Atlas statements specifying the parameters for the numerical device simulation have to be specified after **.END**.

After all Atlas statements, the simulation has to be explicitly terminated (`quit`, `go <simulator>`).

These rules do not apply to the **SET** statement for parameterization of the input file, since it is interpreted by DeckBuild only.

EXTRACT statements are also an exception similar to **SET**. Since MixedMode input files are parsed completely before execution (see Section 13.2.4 "Recommendations" for more information), extractions can only be done after completion of the simulation. To extract results from a MixedMode simulation, **EXTRACT** should be specified after re-initialization of Atlas (`go atlas`).

13.2.2 Circuit and Analysis Specification

The SPICE-like MixedMode statements can be divided into three categories:

- Element Statements: These statements define the circuit netlist.
- Simulation Control Statements: These statements specify the analysis to be performed.
- Special Statements: These statements typically related to numerics and output (first character being a dot ".").

The specification of the circuit and analysis part has to be bracketed by a **.BEGIN** and an **.END** statements. In other words, all MixedMode statements before **.BEGIN** or after **.END** will be ignored or regarded as an error. The order is between **.BEGIN** and **.END** is arbitrary.

Netlist Statements

Each device in the circuit is described by an element statement. The element statement contains the element name, the circuit nodes to which the element is connected and the values of the element parameters. The first letter of an element name specifies the type of element to be simulated. For example, a resistor name must begin with the letter, R, and can contain one or more characters. This means that `R1`, `RSE`, `ROUT`, and `R3AC2ZY` are all valid resistor names. Some elements, such as diodes and transistors, must always refer to a model. A set of elements can refer to the same model. For some elements, such as resistors and capacitors, model referencing is optional. Each element type has its own set of parameters. For example, a resistor statement can specify a resistance value after the optional model name. The bipolar transistor statement (Q) can specify an area parameter. All parameters have corresponding default values. Independent voltage and current sources have different specifications for transient, DC, and AC phases of simulation. Transient specifications use the keywords: **EXP**, **PULSE**, **GAUSS**, **SFFM**, **SIN**, and **TABLE**. AC parameters start with the keyword, **.AC**.

Elements to be simulated numerically are defined as A devices (Atlas devices). At least one Atlas device in a circuit is mandatory.

MixedMode supports the use of the following circuit elements:

- Numerically simulated Atlas devices (A devices)
- User-defined two-terminal elements (B devices)
- Capacitors (C devices)
- Diodes (D devices)
- Voltage controlled voltage source (E devices)
- Current controlled current source (F devices)
- Voltage controlled current source (G devices)
- Current controlled voltage source (H devices)
- Independent current sources (I devices, may be time dependent)

- JFETs (J devices)
- Coupled (mutual) inductors (K devices)
- Inductors (L devices)
- MOSFETs (M devices)
- Optical sources (O devices)
- Bipolar junction transistors (Q devices)
- Resistors (R devices, may be time dependent)
- Lossless transmission lines (T devices)
- Independent voltage sources (V devices, may be time dependent)
- MESFETs (Z devices)

The physical models for linear elements (resistors, capacitors, sources, and so on) are described Section 13.4.1 “Circuit Element Statements”. The models for diodes: BJTs, JFETs, MESFETs and MOSFETs are described in Section 13.4.2 “Control and Analysis Statements”. You can find more extensive documentation, however, in the SmartSpice/Utmost Modeling Manuals Volumes 1, 2, and 3.

A node is a point in the circuit where two or more elements are connected. A node can be described either in terms of a node name or as a node number. The node names and numbers are arbitrary with the exception of the ground node. The ground node is set either by specifying “0” as the node number or using the name GND. All voltages at the nodes are calculated with respect to the voltage at the ground node.

Example

The netlist for the circuit shown in Figure 13-1 is represented by the following MixedMode input deck fragment.

```
# independent voltage source, 0.1V, connected to node 0 (GND) and
1:
V0    1           0           0.1
# 1kOhm resistor, connected to node 1 and 2
R1    1           2           1K
# ATLAS device, connected to node 2 (anode) and 0 (cathode),
# current scaled by 5e7, mesh from file dio.str
ADIO  2=anode     0=cathode  WIDTH=5e7   INFILE=dio.str
```

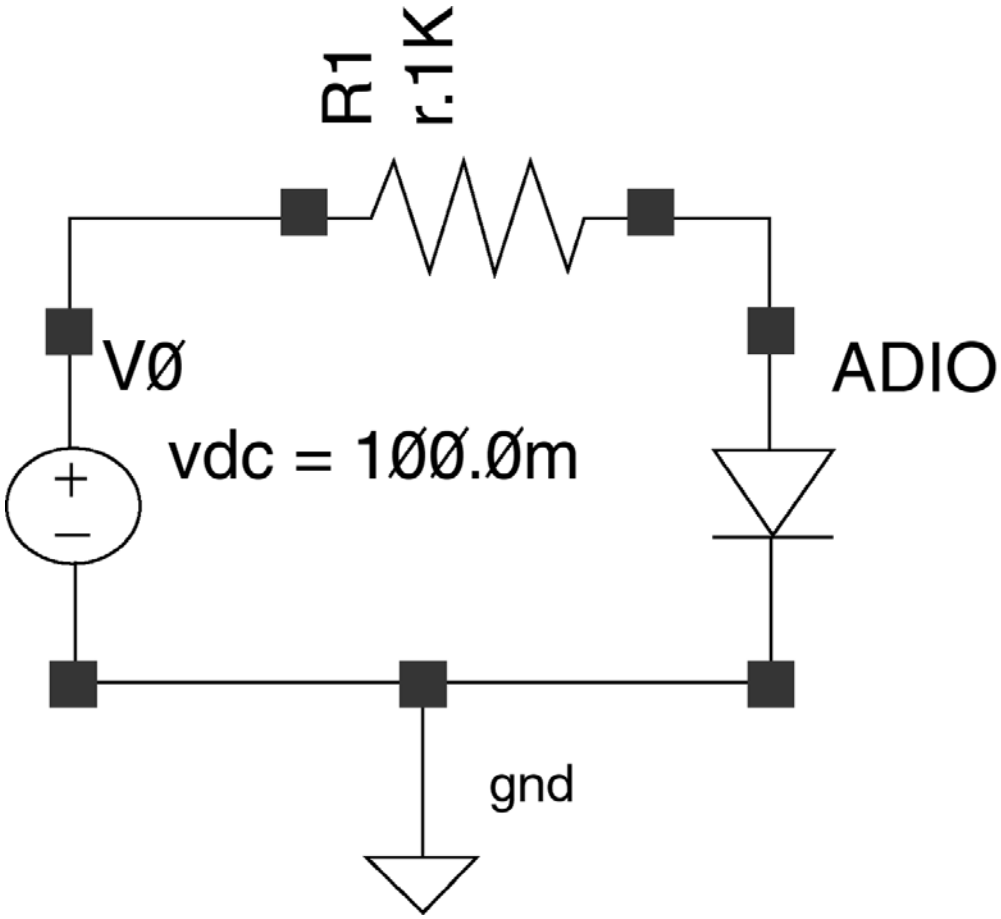


Figure 13-1: Schematic of Primitive Example Circuit

Control Statements

Control statements are used to specify the analysis to be performed in MixedMode. These take the place of the **SOLVE** statements in a regular Atlas input file. At least one of these statements must appear in each MixedMode input file. These statements are:

- **.DC:** steady state analysis including loops
- **.TRAN:** transient analysis
- **.AC:** small signal AC analysis
- **.NET:** small signal parameter extraction (e.g., s-parameters)

If you wish to perform a DC analysis, you should have an associated DC source to relate the DC analysis to. Likewise, for an AC analysis you should have an associated AC source that you relate the AC analysis to. For a transient analysis, you should have an associated source with transient properties that you relate the transient analysis to. Failing to correctly run an analysis on the same sort of source could yield simulation problems, such an running an AC analysis on an independent voltage source where no AC magnitude for the source has been specified.

For more information about these statements, see Section 13.4.2 “Control and Analysis Statements”.

Special statements

Other statements beginning with a dot "." specify special parameters for the circuit simulation. These include numerical options, file input and output, and device parameter output. These statements are listed below.

- compact device models (**.MODEL**)
- the output files (**.LOG**, **.SAVE**)
- initial conditions settings (**.NODESET**, **.IC**)
- initial conditions from a file (**.LOAD**)
- numerics (**.NUMERIC**, **.OPTIONS**)
- device parameter output (**.PRINT**)
- miscellaneous (**.OPTIONS**)

Full descriptions of each statement and associated parameters are found in Section 13.4.2 "Control and Analysis Statements".

13.2.3 Device Simulation Syntax

The second part of a MixedMode command file (after **.END**) is used to define physical models, material parameters, and numerical methods for Atlas devices referenced in the "A"-element statements. The following statements may appear in this part of the command file: **BEAM**, **CONTACT**, **DEFECTS**, **IMPACT**, **INTERFACE**, **INTTRAP**, **MATERIAL**, **MOBILITY**, **METHOD**, **MODELS**, **OUTPUT**, **PROBE**, **TRAP**, and **THERMCONTACT**.

You always need to include an indicator to the circuit element name in each device simulation statement even if there is only one A-device in the circuit. All statements specifying the device properties and models are just supplemented by the DEVICE=name parameter, where name is the circuit element in the netlist, and name will always begin with the letter "A". This makes it possible to define different material properties and model settings for different devices within the circuit.

We recommended that you specify the REGION parameter referring to only one region in **IMPACT**, **MATERIAL**, and **MODELS** statements. If the device consists of more than one region, several statements with the same device parameters and different region parameters are recommended.

For example to specify the bipolar set of models to a device the syntax used might be:

```
MODEL DEVICE=AGTO REGION=2 BIPOLAR PRINT
```

13.2.4 Recommendations

Input Parsing

In regular Atlas (non-MixedMode) simulations, the input is interpreted line by line and each statement is executed immediately. This is very useful and nicely supported by DeckBuild for the interactive development of the input. Circuit simulations, however, require the complete input before any simulation can be performed. Consequently, the following occur:

- The complete input is read and parsed before any simulation is initiated.
- An explicit termination of a simulation is required (`quit`).
- All post processing (extraction and plotting) has to be done after re-initializing Atlas again.

No simulation is started until either a **`QUIT`** statement or a `GO` statement is seen in the input file. Post-processing can be done by restarting Atlas.

Scale and Suffixes

In the MixedMode part of the input, numerical values of parameters are represented in standard floating-point notation. The scale suffix may be followed by a unit suffix (e.g., A for Ampere, V for Volt, and so on). Using a unit suffix can increase the clarity of a command file. The unit suffix is ignored by the program. The scale suffixes are shown in Table 13-1.

Table 13-1 Scale Suffixes

Factor	Name	Suffix
10^{-15}	femto-	F
10^{-12}	pico-	P
10^{-9}	nano-	N
10^{-6}	micro-	U
10^{-3}	milli-	M
10^{3}	kilo-	K
10^{6}	mega-	MG
10^{9}	giga-	G
10^{12}	tera-	T

Numerics

MixedMode solves circuit and device equations simultaneously using fully coupled algorithms. This provides better convergence and requires less CPU time than alternative approaches. The number of circuit variables is often small in comparison with the number of device variables. In this case, the CPU time required for simulation performed using MixedMode does not increase drastically compared to the sum of the simulation times required for the individual numerical physically based devices. MixedMode uses the Newton algorithm for each bias point during steady-state analysis and for each time step during transient analysis. Different variants of the Newton algorithm are used depending on the circumstances [203, 204]. The full Newton method [**.OPTIONS** FULLN] and a modified two-level Newton method [**.OPTIONS** M2LN] are available for steady-state simulation. The full Newton method provides rapid convergence when a good initial guess is available. The modified two-level Newton algorithm is less sensitive to the initial guess. For transient simulation, a good initial guess always exists. The full Newton method therefore works very well. Therefore, it is always used for transient simulation.

When using MixedMode 3D, it is recommended that you specify the DIRECT or GMRES solver in the Atlas part of the MixedMode input deck on the **METHOD** statement. Also, use the NOPROJ parameter in the **.OPTIONS** statement in the MixedMode of the input deck.

Multi-Device Structure Representation

If more than one Atlas device is defined in a MixedMode simulation, the structures are merged together internally. The output solution file is a single file which contains both structures. The first structure referenced will be on top, all other structures will be attached below.

Example

A diode and a bipolar transistor are specified as numerical devices with the following element statements:

```
ABJT 1=BASE 2=EMITTER 4=COLLECTOR WIDTH=1E4 INFILE=bjt.str

ADIO 3=ANODE 4=CATHODE WIDTH=1.5E5 INFILE=dio.str
```

After outputting the solution with:

```
.SAVE MASTER=mas
```

The solution file for the first DC-point, mas_dc_1, contains both structures with the second Atlas device (diode) shifted downwards (see Figure 13-2).

This coordinate shift has to be accounted for eventually when extracting position dependent solution quantities or when defining spatially dependent properties with the C-Interpreter (See Appendix A “C-Interpreter Functions” for more information).

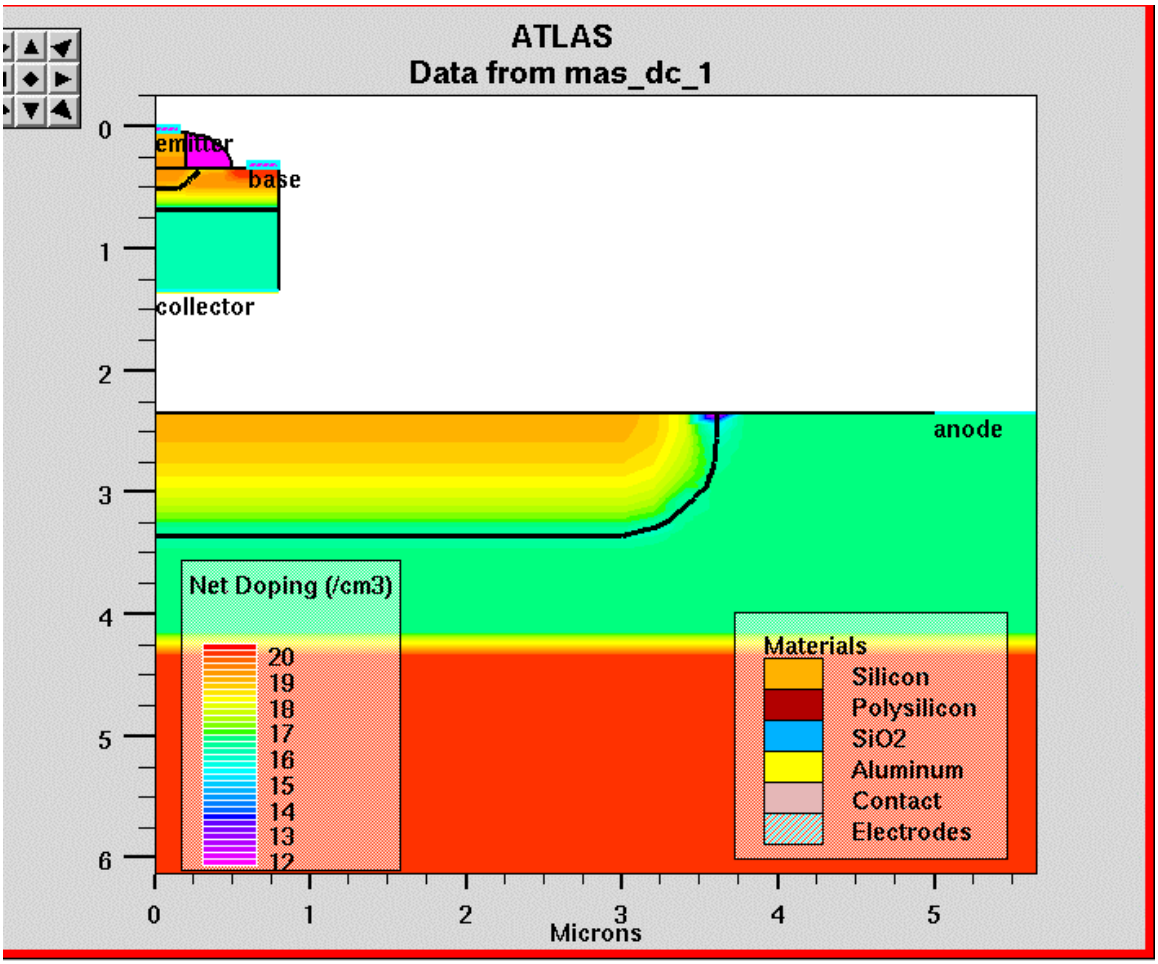


Figure 13-2: Display of a MixedMode solution with two Numerical Devices

Extraction of Results

By default, **EXTRACT** reads its data from the currently opened log-file when executed along with Atlas. Since the extraction of MixedMode log-files require a re-initialization of Atlas (see "Input Parsing" on page 735), **EXTRACT** has to be initialized explicitly with the correct name of the MixedMode log file. To extract voltages at specific nodes, use the syntax `vcct.node."circuit node"`. To extract circuit elements, use `icct.node."circuit element"`.

Example

Specify the log file:

```
.LOG OUTFILE=hallo
```

Subsequent extraction from the transient log-file is done with:

```
go atlas
extract init inf="hallo_tr.log"
extract name="t0" x.val from curve(time,icct.node."Adio_anode") \
        where y.val=0
```

It extracts the time, `t0`, when the transient of the current from the `anode` electrode of the `adio` device in the circuit crosses zero. For more details for the **EXTRACT** syntax, see the DeckBuild User's Manual.

Using MixedMode inside the VWF Automation Tools

Like all other Silvaco products, MixedMode is fully integrated into the VWF framework and can be used for automated experiments. There are, however, some factors to take into account.

One factor is that the **auto-interface** feature doesn't work with MixedMode. All structures have to be explicitly saved in unique files previous to the MixedMode runs and referred to in the `A-` element statements. Another factor is that splits within MixedMode runs are impossible. To overcome this problem, use the **SET** statement to define a variable in a process simulator or in the "dummy" internal run. This variable is used to parameterize the input file.

Example

Capacitance as an independent split variable in a VWF experiment.

```
go internal
# define the independent split variable in a re-entrant simulator:
set cap=5e-9

go atlas

.BEGIN
# use the variable as parameter in MixedMode:
C1    2     3     $cap
```

Another factor is that the automation tools only store files opened by the normal Atlas LOG statement in the VWF database but ignore those defined by **.LOG**. To overcome this, re-initialize Atlas and open the relevant log file of the previous MixedMode run with .log as the append option (so that the file is not reset).

Example

MixedMode log-file definition.

```
.LOG OUTFILE=hallo
```

Re-opening the second DC-log file and the transient log file to get them stored in the VWF database.

```
go atlas

log outfile=hallo_dc_2.log append
log outfile=hallo_tr.log append
```

Initial Settings

Initial convergence is critically dependent on the initial settings of the node voltages (i.e,**.IC** and **.NODESET**). There should not be any problem starting from the zero bias case. Just like starting from a preceding MixedMode solution is simple, since the complete solution of the circuit and the Atlas devices is directly available (i.e, **.LOAD** and **.SAVE**). But when loading solutions for the numerical devices from Atlas, using **.OPTIONS LOADSOLUTIONS**, sometimes precise matching of the initial circuit condition is required. In this case, it is practical to extract the relevant properties in the preceding Atlas run and use them to parameterize the MixedMode input.

In the following example, the voltages and current of an Atlas solution is extracted, and the results are used for the initial definition of the circuit.

The end of the first part is the stand-alone Atlas simulation:

```
# extract the final voltage drop on the anode:
extract name="Von" max(vint."anode")
# extract the gate current:
extract name="I_gate" y.val from curve(vint."anode",i."gate") \
where x.val = $"Von"
extract name="V_gate" y.val from curve(vint."anode",vint."gate") \
where x.val = $"Von"

# now the MixedMode part

go atlas
.BEGIN
# define the gate current source, use extracted value as parameter
I1 0 7 $"I_gate"
#
```

```
# use extracted gate bias and other expressions to calculate
# the node settings:
set Rgl = 10.5
set v7= $V_gate + $I_gate * $Rgl
.NODESET V(1)=2000 V(2)=$"Von" V(3)=$"V_gate" V(4)=$"V_gate" V(5)=-
25 \
V(6)=-15 V(7)=$"v7"
```

13.3 A Sample Command File

A sample MixedMode command file is shown below. This file is used to simulate the reverse recovery of a power diode. Several MixedMode examples are provided with the product which can be accessed using DeckBuild.

```
1. go atlas
2. .BEGIN
3. V1 1 0 1000.
4. R1 1 2 1m
5. L1 2 3 2nH
6. R2 4 0 1MG EXP 1MG 1E-3 0. 20NS 10 200
7. IL 0 4 300
8. ADIODE 3=cathode 4=anode WIDTH=5.E7 INFILE=pd.str
9. .NUMERIC LTE=0.3 TOLTR=1.E-5 VCHANGE=10.
10..OPTIONS PRINT RELPOT WRITE=10
11.$
12..LOAD INFILE=pdsave
13..LOG OUTFILE=pd
14..SAVE MASTER=pd
15.$
16..TRAN 0.1NS 2US
17.$
18..END
19.$
20.MODELS   DEVICE=ADIODE REG=1 CONMOB FLDMOB CONSRH AUGER BGN
21.MATERIAL DEVICE=ADIODE REG=1 TAUN0=5E-6 TAUP=2E-6
22.IMPACT   DEVICE=ADIODE REG=1 SELB
23.$
24.METHOD   CLIM.DD=1.E8 DVMAX=1.E6
25.$
26.go atlas
27.tonyplot pd_tr.log
```

Description

Line 1: All Atlas input files should begin with **GO** `atlas`

Line 2: The **.BEGIN** and **.END** statements indicate the beginning and end of the circuit simulation syntax. These commands are similar to those used in SPICE.

Lines 3-7: Circuit components, topology, and analysis are defined within. Generally, the circuit component definition consists of three parts: the type of component, the lead or terminal mode assignments, and the component value or model name. For example, if the first component definition in this simulation is a DC voltage source, then `V1` defines the component as voltage source number one, `1` and `0` are the two circuit modes for this component, and `1000` indicates that the voltage source value is 1000 volts. The remaining circuit components are resistors (`R1`, `R2`) inductor (`L1`) and independent current source (`IL`).

The reverse recovery of the diode is simulated by dropping the value of output resistor `R2` over a small increment of time. The `R2` statement contains additional syntax to perform this

task. Here, the resistor is treated as a source whose resistance decreases exponentially from 1 mOhm to 1 mOhm over the specified time step. This action essentially shorts out the parallel current source IL, which is also connected to the base of the diode.

Line 8: The ADIODE statement specifies a device to be analyzed by Atlas. The A part of the ADIODE command specifies that this is a device statement. The DIODE portion simply defines the device name. The option INFILE= indicates which device structure file is to be used.

Lines 9-10: These set numerical options for the circuit simulation. WRITE=10 specifies that every tenth timestep will be saved into the solution file specified on the **.SAVE** statement.

Line 12: Specifies a file generated by a previous MixedMode simulation to be used as an initial guess to the voltage.

Line 13-14: Specifies the output log and solution filenames. These names are root names and extensions will be added automatically by the program.

Line 16: Indicates the type of analysis required. In this case, it is a transient simulation lasting 2 microseconds with an initial timestep of 0.1 nanoseconds.

Line 18: Indicates the end of the circuit description. All following statements will be related to the Atlas device.

Lines 20-22: To completely specify the simulation, the physical models used by Atlas must be identified. Note that DEVICE=ADIODE must be specified for each line. The **MODELS** statement is used to turn on the appropriate transport models. This set includes:

- conmob: the concentration dependent mobility mode,
- fldmob: the lateral electric field-dependent mobility model,
- consrh: Shockley-Read-Hall recombination using concentration dependent lifetimes,
- auger: recombination accounting for high level injection effects,
- bgn: band gap narrowing.

The **MATERIAL** statement is used to override default material parameters. In this case, the carrier recombination fixed lifetimes are set. Finally, the Selberherr impact ionization model is enabled using the **IMPACT** statement with the SELB option.

Line 24: The **METHOD** statement specifies numerical options for the device simulation. The METHOD statement must come after all other device simulation statements.

Line 26: The command **GO** Atlas or a **QUIT** statement is needed to initiate simulation. Since a plot of the final log file is desired, the **GO** ATLAS option is used to restart Atlas after the end of the MixedMode simulation.

Line 27: The TonyPlot command is used to plot the resulting log file.

13.4 MixedMode Syntax

This section is split into two parts. The first part describes circuit element statements, which describe the netlist. The second part describe the control and analysis statements.

13.4.1 Circuit Element Statements

A – Atlas device to be simulated using device simulation

Syntax

```
Axxx n1=name1 n2=name2 [n3=name3 ...] infile=filename [width=val]
```

Description

This statement defines a device to be represented by a numerical Atlas model. The device description with all necessary information (geometry, mesh, doping, models, electrode names, and so on) must be available in a standard structure file prior to starting a MixedMode simulation.

Axxx	Name of the element. It must begin with A.
n1	Circuit node to which the Atlas device electrode with the name, name1, is connected. The Atlas device must have at least two electrodes. The maximum number of electrodes allowed in Atlas is 50. This means that up to 25 Atlas devices can be specified. For example, 25 devices with 2 electrodes each, or 10 devices with 5 electrodes each can be specified. The Atlas device models should be used sparingly, because it can be very time consuming. Circuit models should be used for less important circuit components to conserve CPU time. If an Atlas device electrode has been defined as FLOATING on the **CONTACT** statement, then this electrode should be connected to a circuit node with a negative node number. MixedMode treats all circuit nodes with negative node numbers as dummy connection nodes and will not apply a voltage to these nodes.
infile	Name of standard structure format file with device geometry, mesh, doping, electrodes names, and so on. The number of electrodes and their names should match those mentioned in this statement. Optionally, this file can contain a solution, which MixedMode will use as an initial guess (see the **.OPTIONS** statement for more details).
width	Device width. This is an optional parameter (default=1). All currents through Atlas device terminals calculated using the 2D Atlas model will be multiplied by this parameter to account for the third dimension of the device. width can still be used as a multiplier to the Atlas current if a 3D Atlas structure is used in MixedMode 3D.

Example

```
ABJT1 3=EMITTER 4=BASE 6=COLLECTOR INFILE=BJT1.STR WIDTH=10
```

Note: Optional parameters for a statement are shown with square brackets (e.g., [n3=name3]).

B – User-Defined Two-terminal Elements

Syntax

```
Bxxx n+ n- INFILE=file_name FUNCTION=function_name
```

Description

Bxxx	User-defined two terminal element name. It must begin with B.
n+, n-	Positive and negative terminal nodes.
INFILE	Name of the text file (file_name) that contains C source code for a user-defined function that describes element behavior. This file can contain more than one function description.
function_name	Name of the function ("function_name") from the file.

Example

```
B1 2 3 infile=ud.c function=rc
```

MixedMode users who are familiar with C-Interpreter can define their own two-terminal elements using the B statement and a function written in C that defines the behavior of the element.

User-Defined Model

A user-defined model is specified by defining the following:

- The dependencies of the device terminal current on the terminal voltages
- The derivatives of the terminal current with respect to the terminal voltages
- The device current (I) is described by Equation 10-1.

$$I = F1(U,t) + F2(U,t) \cdot \left(\frac{dU}{dt}\right) \qquad 10\text{-}1$$

where:

U is the device voltage, t is time, and $F1$ and $F2$ are functions that determine the behavior of the device. The first term on the right hand side describes the “DC” current and the second term describes “capacitive” current.

The following four functions are user-specified:

- **F1(U,t):** the DC current.
- **F2(U,t):** the capacitance.
- **dF1(U,t)/dU:** the DC differential conductance.
- **Q:** the charge associated with F2(U,t).

To define the element, prepare a text file that contains an appropriate function written in C. A template for this user-defined function is shown below:

```
  int udef(double v, double temp, double ktq, double time, double
*curr,
   double *didv, double *cap, double *charge)
  {
  /* user-supplied code here  */
  return(0);
  }
```

Input Parameters

Four input parameters are supplied to the function and can be used in the user-defined code. The input parameters are:

- **v:** the voltage across the element (V)
- **temp:** the temperature (K)
- **ktq:** the thermal voltage kT/q (V)
- **time:** transient time (sec); a value of 0 is supplied during DC calculations.

Output Parameters

The four output parameters that must be returned by the function are:

- **curr:** the value of F1 (Amps)
- **didv:** the value of dF1(v, time)/dU (A/V)
- **cap:** the value of F2(v, time)
- **charge:** the value of the charge (Q)

Example

Consider an element that consists of a resistor R and a capacitor C connected in parallel. The equation for the total current through this combination is:

$$I(U,t) = \frac{U}{R} + C \cdot \left(\frac{dU}{dt}\right) \quad \text{10-2}$$

The quantities that must be user-defined are:

$$F1(U,t) = \frac{U}{R} \quad \text{10-3}$$

$$F2(U,t) = C \quad \text{10-4}$$

$$\frac{dF(U,t)}{dU} = \frac{1}{R} \quad \text{10-5}$$

$$Q = C \cdot U \quad \text{10-6}$$

When R=2$k\Omega$ and C=100pF, a user-defined function could have the following form:

```
intrc(double v, double temp, double ktq, double time, double *curr,
double *didv, double *cap, double *charge)
          {
       *curr = v/2000.0;
       *didv = 1.0/2000.0
       *cap = 1.0e-10;
       *charge=1.0e-10*v;
       return(0); /* 0 - ok */
  }
```

C – Capacitor

Syntax

```
Cxxx n+ n- value
```

Description

Cxxx	Name of a capacitor element. It must begin with C.
n+, n-	Positive and negative terminal nodes.
value	Capacitance in farads.

Example

```
Cload 3 0 1pF
```

D – Diode

Syntax

```
Dxxx n+ n- mname [area] [L=val] [W=val] [PJ=val] [WP=val] [LP=val]
[WM=val]
[LM=val] [OFF] [IC=val] [M=val] [TEMP=val] [DTEMP=val]
```

Description

Dxxx	Name of the diode element. It must begin with D.
n+, n-	Positive (anode) and negative (cathode) terminal nodes.
mname	Diode model name. It must refer to a diode model.
area	Area factor. The default is 1.0.
L	Length of the diode in meters. Used for LEVEL 3 diode model only.
W	Width of the diode in meters. Used for LEVEL 3 diode model only.
PJ	Periphery of the diode junction. Calculated from W and L if they are specified (in the LEVEL 3 diode model). The ISW and CJSW model parameters are affected by the value of PJ.
WP	Width of the polysilicon capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.
LP	Length of the polysilicon capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.
WM	Width of the metal capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.
LM	Length of the metal capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.
OFF	Sets ON/OFF startup condition for DC analysis. Default is ON.
IC	Initial voltage across the diode.
M	Multiplier used to describe multiple parallel diodes.
TEMP	Device operating temperature (^{o}C).
DTEMP	Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
D1 2 3 dmodel1
Dclmp 3 7 Diol 3.0 IC=0.3
```

Note: See the SmartSpice/Utmost Modeling Manual Volume 2 for a complete description of the diode models.

E – Linear voltage controlled source

Syntax

```
Exxx n+ n- nc+ nc- gain
```

Description

Exxx	Name of the linear voltage controlled voltage source. It must begin with E.
n+, n-	Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.
nc+, nc-	Positive and negative controlling node numbers.
gain	Voltage gain.

The linear voltage-controlled voltage source is characterized by Equation 10-7.

$$v(v+,n) = gain*v(nc+,nc) \quad \text{10-7}$$

Example

```
ER 4 5 6 7 55
```

F – Linear current controlled current source

Syntax

```
Fxxx n+ n- vcontrolname gain
```

Description

Fxxx	Name of the linear current controlled current source. It must begin with F.
n+, n-	Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.
vcontrolname	Name of the voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vcontrolname.
gain	Current gain.

The linear current-controlled current source is characterized by Equation 10-8.

$$v(n+,n-) = gain*i(vcontrolname) \quad \text{10-8}$$

Example

```
F12 4 5 VIN 0.1
```

G – Linear voltage controlled current source

Syntax

```
Gxxx n+ n- nc+ nc- transconductance
```

Description

Gxxx	Name of the linear voltage controlled current source. It must begin with G.
n+, n-	Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.
nc+, nc-	Positive and negative controlling node numbers.
transconductance	Transconductance (in 1/Ohms).

The linear voltage controlled current source is characterized by Equation 10-9.

$$i(n+,n-) = transconductance * v(nc+,nc-) \quad \text{10-9}$$

Example

```
G2 4 5 6 7 5.5
```

H – Linear current controlled voltage source

Syntax

```
Hxxx n+ n- vcontrolname transresistance
```

Description

Hxxx	Name of the linear current controlled voltage source. Must begin from H.
n+, n-	Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.
vcontrolname	Name of voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vcontrolname.
transresistance	Transresistance (in Ohms).

The linear current controlled voltage source is characterized by Equation 10-10.

$$v(n+,n-) = transresistance * i(vcontrolname) \quad \text{10-10}$$

Example

```
H12 4 5 V1 0.1K
```

I – Independent current source

Syntax

```
Ixxx n+ n- value [AC acmag] [transient_parameters]
```

Description

Ixxx	Name of the independent current source. It must begin with I.
n+, n-	Positive and negative terminal nodes.
value	DC value of the source (ampers).
AC	Keyword for the AC source value.
acmag	AC magnitude.
transient_parameters	The transient parameters are described in Section 13.4.3 “Transient Parameters”.

Example

```
I1 2 8 0. PULSE 0 200 0 20ns 20ns 100ns 10 100
I2 1 5 1u AC 2u
```

J – Junction Field-Effect Transistor (JFET)

Syntax

```
Jxxx nd ng ns [nb] mname [area] [M=val] [L=val] [W=val] [OFF]
[IC=vds,vgs]
[TEMP=val] [DTEMP=val]
```

Description

Jxxx	Name of the JFET element. It must begin with J.
nd, ng, ns, nb	Drain, gate, source and bulk terminal nodes. The bulk node doesn’t need to be specified. If the bulk node is not specified, then the bulk is connected to the source node
mname	Model name. It must refer to a JFET model.
area	Area factor. The default is 1.0.
M	Multiplier used to describe multiple parallel JFETs.
L	Length of the gate in meters.
W	Width of the gate in meters.
OFF	Sets ON/OFF startup condition for DC analysis. Default is ON.

IC	Initial condition specification for *vds* and *vgs*.
TEMP	Device operating temperature (^{o}C).
DTEMP	Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
J44 1 4 6 jmodel
```

Note: See SmartSpice/Utmost Modeling Manual, Volume 2 for a complete description of the JFET models.

K – Coupling between two inductors

Syntax

```
Kxxx Lyyy Lzzz kval
```

Description

This is not a real circuit element. This statement defines only the coupling between two inductors.

Kxxx	Name. This parameter is not important and is used only to distinguish the statement. It must begin with K.
Lyyy	First inductor element name. It must begin with an L and match one of the inductor names from the circuit.
Lzzz	Second inductor element name. It must begin with an L and match one of the inductor names from the circuit.
kval	Coefficient of mutual coupling, which must be in the range 0 < kval < 1.

The mutual inductance *M* will be determined from Equation 10-11.

$$M = kval*L1*L2 \quad (10\text{-}11)$$

Example

```
K1 L22 LLOAD 0.99
```

L – Inductor

Syntax

```
Lxxx n+ n- value
```

Description

Lxxx	Name of the inductor. It must begin with L.
n+, n-	Positive and negative terminal nodes.
value	Inductance in henries.

Example

```
L2 2 3 2.5nH
```

M – MOSFET

Syntax

```
Mxxx nd ng ns [nb] mname [L=val] [W=val] [AD=val] [AS=val]
[PD=val] [PS=val] [NRD=val] [NRS=val] [OFF] [IC=vds,vgs,vbs]
[M=val]
[TEMP=val] [DTEMP=val] [GEO=val] [DELVTO=val]
```

Description

Mxxx	MOSFET element name. It must begin with M.
nd, ng, ns, nb	Drain, gate, source, and bulk terminal nodes. The bulk terminal node name is optional. If it is unspecified, ground is used.
mname	Model name. It must refer to a MOSFET model.
L=val	Channel length in meters.
W=val	Channel width in meters.
AD	Drain diffusion junction area (meters^2). This default is 0.
AS	Source diffusion junction area (meters^2). This default is 0.
PD	Drain diffusion junction perimeter (meters). This default is 0.
PD	Source diffusion junction perimeter (meters). This default is 0.
NRD	The Number of squares of drain diffusion for resistance calculations. The default is 0.
NRS	The Number of squares of source diffusion for resistance calculations. The default is 0.
OFF	Sets ON/OFF startup condition for DC analysis. Default is ON.

IC	Initial voltage condition specification for *vds*, *vgs*, and *vbs*.
M	Multiplier used to describe multiple parallel MOSFETs. The default is 1.
TEMP	Device operating temperature (^{o}C).
DTEMP	Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.
DELVTO	Threshold-voltage shift. When specified on the device line, the value overrides the value of the model parameter, DELVTO. If not specified, the value of the model parameter is used.

Example

```
M1 2 4 8 9 mod1
Mout2 19 20 21 0 nmos L=5u W=2u TEMP=50
M22 3 5 7 8 mosmod1 L=10u W=5u AD=150p AS=150p PD=50u
PS=50u NRD=10 NRS=20
```

Note: See SmartSpice/Utmost Modeling Manual, Volume 1 for a complete description of the MOSFET models.

O – Optical source

Syntax

```
Oxxx beam value [transient_parameters]
```

Description

Oxxx	Name of an independent optical source. It must begin with O.
beam	Beam number. The beam with this number should be described in the Atlas section of the command file. See Chapter 11 "Luminous: Optoelectronic Simulator" for a complete description of optoelectronic simulation.
value	DC optical intensity value (W/cm2).
transient_parameters	The transient parameters are described in Section 13.4.3 "Transient Parameters".

Note: The treatment of optical sources is fully similar to the treatment of independent voltage/current sources. In other words you can use, **.DC** statements to simulated DC light responses of the circuit and transient parameters in order to describe the transient behavior of the optical sources.

Example

```
O1 1 0.001 pulse 0.001 0.002 0 2ns 2ns 100ns 10 100
```

Q – Bipolar junction transistor

Syntax

```
Qxxx nc nb ne [ns] mname [area] [OFF] [IC=vbe,vce] [M=val]
[TEMP=val] [DTEMP=val]
```

or

```
Qxxx nc nb ne [ns] mname [area=val] [areab=val] [areac=val] [OFF]
[IC=vbe,vce] [M=val] [TEMP=val] [DTEMP=val]
```

Description

Qxxx	Name of a bipolar junction transistor. It must begin with Q.
nc, nb, ne, ns	Collector, base, emitter, and substrate nodes. The substrate terminal node name is optional. If it is unspecified, ground is used.
mname	Model name. It must refer to a BJT model.
area	Emitter area factor. The default value is 1.0.
areab	Base area factor. The default is area.
areac	Collector area factor. The default is area.
OFF	Sets ON/OFF startup condition for DC analysis. Default is ON.
IC	Initial voltage condition specification for *vbe*, *vce*.
M	Multiplier used to describe multiple parallel BJTs. The default is 1.
TEMP	Device operating temperature (^{o}C).
DTEMP	Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
Q1 2 3 9 npnmod 1.5 IC=0.6,5.0
Q9 10 11 12 20 mod22 OFF TEMP=50
```

Note: See SmartSpice Modeling Manual, Volume 2 for a complete description of the BJT models.

R – Resistor

Syntax

```
Rxxx n+ n- value [transient_parameters]
```

Description

Rxxx	Name of the resistor element. It must begin with "R".
n+, n-	Positive and negative terminal nodes.
value	Resistance in ohms.
transient_parameters	The transient parameters are described in Section 13.4.3 "Transient Parameters".

Note: Unlike the traditional SPICE program, transient parameters are acceptable for resistor elements. This allows simulation of different kinds of time-dependent resistors and switches in a simple way.

Example

```
R12 4 5 100k
```

S-Voltage-Controlled Switch

Syntax

```
Sxxx n+ n- nc+ nc- <mname> <on|or> <roff=val> <ron=val> <vt=val>
<vh=val>
```

Description

Sxxx	Name of the voltage-controlled switch element.
n+, n-	Positive and negative terminal node names.
nc+, nc-	Positive and negative controlling node names.
mname	Optional model name.
on, off	Initial condition of the switch if the switch condition cannot be determined from the controlling voltages.
roff	Off resistance (default is 1/DCGMIN ohm).
ron	On resistance (default is 1 ohm).
vt	Threshold voltage (default is 0V).
vh	Hysteresis voltage (default is 0V).

The switch resistance (R) is calculated from the control voltage (Vc =- V(nc+) - V(nc-)) as:

```
R = roff if Vc < Vt - Vh
R = ron if Vc > Vt - Vh
R = Rold if Vt - Vh <= Vc <= Vt + Vh
```

T – Lossless transmission line

Syntax

```
Txxx n1 n2 n3 n4 Z0= val TD=val
```

Description

Txxx	Name of the transmission line element. It must begin with T.
n1, n2	Nodes at port 1.
n3, n4	Nodes at port 2.
Z0	Characteristic impedance.
TD	Transmission delay.

Example

```
T1 1 0 2 0 Z0=50 TD=10ns
```

V – Independent voltage source

Syntax

```
Vxxx n+ n- value [AC acmag] [transient_parameters]
```

Description

Vxxx	Name of the independent voltage source. It must begin with V.
n+, n-	Positive and negative terminal nodes.
value	DC value of the source in units of volts.
AC	Keyword for the AC source value.
acmag	AC magnitude.
transient_parameters	The transient parameters are described in Section 13.4.3 “Transient Parameters”.

Example

```
VCC 5 0 10.5
VIN 2 4 5.5 AC 1
```

X - Subcircuit Call

Syntax

```
Xxxx n1 <n2 n3 ...> subcktname <<PARAMS:> parname=val ...>
```

Description

Xxxx	Subcircuit names must begin with X.
n1, n2, ...	Node names for external reference.
subcktname	Subcircuit definition.
PARAMS	The optional parameter preceding the list of parameters and their values.
parname	The parameter name whose value is set to val, which will be local to the subcircuit. valis either a numerical value, an expression enclosed in single quotes containing previously defined parameters, or a string, which defines a model's name, that can be transmitted in subcircuit. If the same parameter is assigned a value on more than one statement in the input deck (**.PARAM**, **.SUBCKT**, and **X** statements), then will be one of the following: • a value assigned in a global **.PARAM** statement (outside subcircuits) is used if it exists. • a value assigned in a corresponding **X** statement is used if it exists. • a value assigned in a corresponding **.SUBCKT** statement is used if it exists. • a value assigned in a local **.PARAM** statement (inside the subcircuit) is used if it exists.

This statement creates instance Xxxx of subcircuit subcktname. There must be the same number of nodes n1 ... as in the subcircuit definition (see **.SUBCKT** statement). These nodes replace the formal node names in the **.SUBCKT** statement.

Subcircuits can be nested. The number of nesting levels is not limited but nesting must not be recursive. When the circuit is loaded, all subcircuit device names are expanded to the form devtype.subcktname.devname, where devtype is the first letter of devname. All local node names in the subcircuit are expanded to the form subcktname.nodename.

For nested subcircuits, expanded names have multiple dot-separated qualifiers. For example, if a circuit has a call to subcircuit subcktname1 and the subcktname1 definition contains a call to subcircuit subcircuit2, then expanded names will have the following format:

```
devtype.subcktname1.subcktname2.devname

subcktname1.subcktname2.nodename

subcktname1.subcktname2.modelname
```

YVLG-User-defined Verilog-A Element

Syntax

```
YVLGxxx n1 n2 ... mname <user_params=param-values>
```

Description

YVLGxxx	Name of the voltage-controlled switch element.
n1, n2, ..	Terminal node names.
mname	Verilog-A module name or VLG model name.
user_params	Optional list of user-defined Verilog-A parameters.

The following restrictions apply during a YVLGXXX statement:

- A .verilog statement that specifies the file containing the model referenced by mname must be present in the netlist.
- The number of terminal nodes must match the number of nodes declared in the Verilog-A module.

Z – MESFET

Syntax

```
Zxxx nd ng ns [nb] mname [area] [M=val] [L=val] [W=val] [OFF]
[IC=vds,vgs]
[TEMP=val] [DTEMP=val]
```

Description

Jxxx	Name of the MESFET element. It must begin with Z.
nd, ng, ns, nb	Drain, gate, source and bulk terminal nodes. The bulk node doesn't need to be specified. If the bulk node it not specified, then the bulk is connected to the source node.
mname	Model name. It must refer to a MESFET model.
area	Area factor. The default is 1.0.
M	Multiplier used to describe multiple parallel MESFETs.
L	Length of the gate in meters.
W	Width of the gate in meters.
OFF	Sets ON/OFF startup condition for DC analysis. Default is ON.
IC	Initial condition specification for *vds* and *vgs*.
TEMP	Device operating temperature (^{o}C).
DTEMP	Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
Z44 1 4 6 jmodel
```

Note: See SmartSpice Modeling Manual, Volume 2 for a complete description of the MESFET models.

13.4.2 Control and Analysis Statements

.AC

`.AC` performs an AC linear small-signal analysis on the circuit. MixedMode first creates a linearized small-signal model at the operating point of the circuit and the computes the frequency response over a user-specified range of frequencies.

Syntax

```
.AC DEC|OCT|LIN nump fstart fstop/ sweep source_name start stop
step
```

Description

DEC	Sweep frequency by decades.
OCT	Sweep frequency by octaves.
LIN	Linear frequency sweep. This is default.
nump	Total number of points per decade or per octave, or the total number of points of the linear sweep.
fstart	Starting frequency (Hz).
fstop	Final frequency (Hz).

Several `.AC` statements can be specified in the same command file. In this case, they will be executed sequentially. Before executing the first **.AC** statement, the program will execute all **.DC** statements (if any), regardless of the order of the **.AC** and **.DC** statements in the command file. At least one independent voltage or current source must have a AC specification.

source_name	Name of the independent voltage, current, or optical source to be swept.
start	Starting value of the sweep.
step	Increment value of the sweep.
stop	Final value of the sweep.
sweep	Performs a DC sweep at every frequency point.

Example

```
.AC DEC 3 1.e3 1.e12
.AC LIN 20 1.e5 2.e6
```

.BEGIN

.BEGIN indicates the start of the circuit part of a MixedMode command file. The synonym for this parameter is START.

```
.BEGIN 3D
```

Description

3D	Specifies that MixedMode 3D will be used instead of MixedMode.

.DC

.DC causes a DC transfer curve to be computed for the circuit with all capacitors opened and all inductors shorted.

Syntax

```
.DC DEC|OCT|LIN source_name start stop increment/numbers_steps
source_name2 DEC|OCT|LIN start2 stop2 number_steps2
```

Description

DEC	Sweep DC bias (voltage or current) by decades.
OCT	Sweep DC bias by octaves.
LIN	Linear DC bias sweep. This is the default.
source_name	Name of the independent voltage, current, or optical source to be swept.
start	Starting value of the sweep argument.
stop	Final value of the sweep argument.
number_steps	Number of steps of the inner sweep
source_name2	Name of the secondary sweep source.
start2	start value of the secondary sweep source
stop2	Final value of the secondary sweep source
number_steps2	Number of steps of the secondary sweep.

Several **.DC** statements can be specified in a command file. In this case, they will be executed sequentially. Before executing the first **.DC** statement, the program will simulate the circuit with the independent source values given in the description of those sources. The **.DC** statement is also often used to increment the values of independent voltage and current sources in a circuit to avoid convergence problems.

Examples

```
.DC VIN 0. 5. 0.25
.DC IE 50 500 50
```

.END

.END indicates the end of the circuit part of a MixedMode command file. The synonym for this parameter is FINISH.

.ENDL

.ENDL specifies the end of a library definition.

Syntax

```
.ENDL [library_name]
```

Description

library_name	Name of the library being terminated.

.GLOBAL

.GLOBAL allows globally accessible nodes to be defined for use in subcircuits.

Syntax

```
.GLOBAL node1 <node2 ...>
```

Description

node1	Node name that is to be declared global. Any internal subcircuit modes that have the same name as global node will be assumed to refer to the global node.

Example

```
.GLOBAL V1
```

.IC

.IC sets specified node voltages during the steady-state simulation.

Syntax

```
.IC [V(I)=val_I...]
```

Description

This statement forces the specified node voltages to be set to specified values during the steady-state simulation. These voltages are release when the transient simulation begins.

Example

```
.IC V(1)=10
.IC V(node1)=-0.5
```

.INCLUDE

.INCLUDE includes a file into the input file.

Syntax

```
.INCLUDE filename
```

Description

filename	Name of the file to be inserted into the input deck at the point where the **.INCLUDE** statement appears.

Example

```
.INCLUDE inverter.in
```

.LIB

Syntax

```
.LIB filename [entryname]
```

Description

filename	Name of the library file.
entryname	The entry name of the section of the library name to be included in the input deck. If entryname is not specified, then the entire library file will be included.

Examples

```
.LIB lib1
.LIB lib1 mos1
```

Library File Structure

Each section of the library file should start with a **.LIB** followed by entryname and ends with **.ENDL** followed by an optional entry name:

```
.LIB entryname
....
.ENDL [entryname]
```

.LOAD

.LOAD loads a solution file.

Syntax

```
.LOAD INFILE=filename
```

Description

INFILE	Name of a file (filename) to be loaded as an initial guess for further simulation. This file must have been saved during a previous run of MixedMode using the **.SAVE** statement.

Example

```
.LOAD INFILE=pdsave
```

Note: This statement is not used to load SSF format solution files from Atlas (see **.OPTIONS LOADSOLUTIONS**).

.LOG

.LOG specifies the filename for the circuit voltages and currents that will be saved.

Syntax

```
.LOG OUTFILE=filename CSVFILE=filename ONEFILE TSTART
```

Description

CVSFILE	Name of a file (filename) for the circuit voltages and currents to be saved in comma-separated values format files.
ONEFILE	Enables saving of log files of the same simulation type to a single file. If this is specified, then all **.DC** log files will be written to one file, all **.AC** log files will be written to one file, and all **.NET** log files will be written to one file.
OUTFILE	Name of a file (filename) for the circuit voltages and currents to be saved in standard structure format files.
TSTART	The transient log file writting will not start until TSTART and the time values will be shifted by TSTART.

These files will have the following names:

For steady-state analysis:

```
"filename"_dc_1.log
"filename"_dc_2.log
"filename"_dc_3.log
..
```

(new file will be created for each **.DC** statement).

For AC analysis:

```
"filename"_ac_1.log
..
For network parameter extraction:
"filename"_net_1.log
..
```

For transient analysis:

```
"filename"_tr.log
..
```

To plot results of and entire steady-state analysis simultaneously, load all files related to steady-state analysis into TonyPlot.

CSVFILE	Name of a file for the circuit voltages and currents to be saved in Comma Seperated Value (CSV) format.

Example

```
.LOG OUTFILE=pd
```

.MODEL

.MODEL specifies the circuit element model to be used for diodes, BJTs, or MOSFETs, and the numerical values of parameters associated with the model.

Syntax

```
.MODEL name type <parameters>
```

Description

name	This is the model name. Circuit element definition statements refer to this name to link elements to models.
type	This is the model type. This type must be consistent with the type of the circuit elements that uses the model. The type can be one of the following: • **D** - Diode model • **NMOS** - n-channel MOSFET model. • **PMOS** - p-channel MOSFET model. • **NPN** - npn BJT model • **PNP** - pnp BJT model • **NJF** - n-channel JFET/MESFET model • **PJF** - p-channel JFET/MESFET model
parameters	Model parameters. The parameters are described in the SmartSpice/Utmost Modeling Manuals Volumes 1, 2 and 3.

Example

```
.MODEL MODBJT NPN IS=1.E-17 BF=100 CJE=1F TF=5PS \
CJC=0.3F RB=100 RBM=20
```

.NET

.NET specifies that a network parameter extraction is to be performed.

Syntax

```
.NET INPORT OUTPORT DEC|OCT|LIN nump fstart fstop [Z0] [INDIN]
[RSIN] [INDOUT]] [RSOUT] [CIN] [COUT] [SWEEP] [source_name] [start]
[stop] [step]
```

Description

INPORT	Input port description. It should be in one of the following formats. • **V(n+,n-):** two nodes (positive (n+) and negative (n-)). • **Vxxxx:** where Vxxxx is the name of an existing voltage source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node. • **Ixxxx:** where Ixxxx is the name of an existing current source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node.

Note: If the nodes specified as the input port are the same nodes as an existing current or voltage source, then the name of the source must be specified as inport. Also, remove all AC parameters from voltage or current sources before using the **.NET** statement.

OUTPORT	Output port description. It should be in one of the following formats. • **V(n+,n-):** two nodes (positive (n+) and negative (n-)). • **Vxxxx:** where Vxxxx is the name of an existing voltage source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node. • **Ixxxx:** where Ixxxx is the name of an existing current source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node.

Note: If the nodes specified as the output port are the same nodes as an existing current or voltage source, then the name of the source must be specified as outport.

DEC	Sweep frequency by decades.
OCT	Sweep frequency by octaves.
LIN	Linear frequency sweep. This is default.

nump	Total number of points per decade or per octave, or the total number of points of the linear sweep.
fstart	Starting frequency (Hz).
fstop	Final frequency (Hz).

Additional optional parameters may also be specified on the **.NET** statement.

Z0	Matching Impedance (default = 50 Ohms).
INDIN	Inductance through which the DC voltage source is connected to the input source (only if INPORT is given as Vxxxx).
RSIN	Series resistance of INDIN.
INDOUT	Inductance through which the DC voltage source is connected to the output source (only if OUTPORT is given as Vxxxx).
RSOUT	Series resistance of INDOUT.
CIN	Capacitance through which the S-parameter test circuit is connected to the input port.
COUT	Capacitance through which the S-parameter test circuit is connected to the output port.
SWEEP	Performs a DC sweep at every frequency point.
source_name	Name of the independent voltage, current, or optical source to be swept.
start	Starting value of the sweep.
stop	Final value of the sweep.
step	Increment value of the sweep.

Note: The S-parameters will be automatically saved to the log file. The Z, Y, H, ABCD, and gain small-signal parameters can also be written to the log file. These are selected through the **.OPTIONS** statement. You can also view the default values if PRINT is specified in the **.OPTIONS** statement

Examples

```
.NET V1 V2 DEC 10 1e6 1e10
.NET I1 V2 DEC 10 1e6 1e10 Z0=75 RSOUT=100
.NET V(1,0) V(2,3) DEC 10 1e6 1e10
```

.NODESET

.NODESET sets initial values for circuit node voltages.

Syntax

```
.NODESET [V(I)=VAL_I ...]
```

Description

This statement specifies the initial values for circuit node voltages. If a node voltage is not specified, the program will try to find a solution using zero as an initial guess for this node. This statement can significantly reduce the CPU time needed to calculate the initial condition.

Example

```
.NODESET V(1)=50 V(2)=49.4 V(3)=10 V(5)=-1.5
.NODESET V(in1)=0 V(2)=2 V(out1)=-1
```

.NUMERIC

.NUMERIC specifies special numeric parameters for the circuit analysis.

Syntax

```
.NUMERIC   [parameters]
```

Parameter	Type	Default	Units
AMP.PRECOND	Logical		
DTMAX	Real	$1*10^{100}$	s
DTMIN	Real	$1*10^{-25}$	s
ILK.PRECOND	Logical		
ITLIMIT.SOLVER	Real		500
FILL.LEVEL	Real		2
FILL.RATIO	Real	5.5	
IMAXDC	Integer	25	
IMAXTR	Integer	15	
LTE	Real	0.1	
PAM.BICGST	Logical		False
PAM.GMRES	Logical		False
TCOMP	Real		K
TOLDC	Real	$1*10^{-4}$	
TOLTR	Real	$1*10^{-4}$	
VCHANGE	Real	$5*10^{7}$	V
VMAX	Real	$5*10^{7}$	V
VMIN	Real	$-5*10^{7}$	V

Description

AMP.PRECOND	Specifies that the AMP preconditioner will be used if PAM.BICGST or PAM.GMRES are specified on the .NUMERIC statement.
DTMAX	Specifies the maximum time step.
DTMIN	Minimum time step value for transient analysis.

FILL.LEVEL	Specifies the fill level used when calculating the conductivity matrix using an iterative solver with the ILK preconditioner.
FILL.RATIO	Specifies the fill ratio used when calculating the conductivity matrix using the iterative solver.
ILK.PRECOND	Specifies that the ILK preconditioner will be used if PAM.BICGST or PAM.GMRES are specified on the .NUMERIC statement.
ITLIMIT.SOLVER	Specifies the maximum number of iterations for conductivity matrix iterative solver.
IMAXDC	Maximum number of mixed circuit-device iterations to be performed during steady-state analysis.
IMAXTR	Maximum number of mixed circuit-device iterations to be performed during transient analysis.
LTE	Local truncation error for transient analysis.
PAM.BICGST	Specifies that the PAM.BICGST parallel iterative solver will be used to calculate the conductivity matrix in 3D.
PAM.GMRES	Specifies that the PAM.GMRES parallel iterative solver will be used to calculate the conductivity matrix in 3D.
TCOMP	Temperature compliance value.
TOLDC	Relative accuracy to be achieved during steady-state analysis for the calculation of voltages in circuit nodes.
TOLTR	Relative accuracy to be achieved during transient analysis for the calculation of voltages in circuit nodes.
VCHANGE	Maximum allowable change in circuit node voltages between two mixed circuit-device iterations. This parameter can be useful for reaching steady-state convergence with a bad initial guess.
VMAX	Maximum value for circuit node voltages.
VMIN	Minimum value for circuit node voltages.

Example

```
.NUMERIC LTE=0.05 TOLDC=1.*10-8 DTMIN=1ns
```

.OPTIONS

.OPTIONS specifies various circuit simulation options.

Syntax

```
.OPTIONS [parameters]
```

Parameter	Type	Default	Units
ABCD.PARAM	Logical	False	
ACTRAN.FREQ	Logical	False	
CNODE	Real	$1*10^{-16}$	F
CYLINDR	Logical	False	
DC.WRITE	Real	1	
DT.NOCHECK	Logical	False	
DCGMIN	Real	$1*10^{-12}$	1/ohms
FULLN	Logical	True	
GAINS	Logical	False	
GMIN	Real	$1*10^{-12}$	1/ohms
H.PARAM	Logical	False	
LOADFLOATING	Logical	False	
LOADSOLUTIONS	Logical	False	
M2LN	Logical	False	
M2LN.TR	Logical	False	
NOPROJ	Logical	False	
NOSHIFT	Logical	False	
PRINT	Logical	False	
RELPOT	Logical	False	
RV	Real	$1*10^{-4}$	W
STR.XOFFSET	Real	1	μm
STR.XSHIFT	Logical		
STR.YOFFSET	Real	1	μm
STR.YSHIFT	Logical		

Parameter	Type	Default	Units
STR.ZOFFSET	Real	1	μm
STR.ZSHIFT	Logical		
TEMP	Real	300	K
TNOM	Real	300	K
WRITE	Integer	1	
Y.PARAM	Logical	False	
Z.PARAM	Logical	False	
ZIP.SOLVER	Logical	False	

Description

ABCD.PARAM	ABCD parameters will be written to the log file. This is used in conjunction with the **.NET** statement.
ACTRAN.FREQ	Specifies the frequency used to calculate the device small-signal AC capacitance and conductance at every transient time step.
CNODE	A very small capacitance, which for algorithmic reasons automatically connected from each circuit node to ground. This value can be set to 0.
CYLINDR	Cylindrical coordinate system for all Atlas devices.
DC.WRITE	Specifies how often a structure file (as specified by the MASTER parameter on the **.SAVE** statement) will be written during **.DC** solutions. If this parameter is set to 0, then no **.DC** structure files will be written.
DT.NOCHECK	Specifies that the transient pulse and table definitions will not be taken into account when calculating the time step.
DCGMIN	Specifies the minimum conductance used in SPICE models during DC analysis.
FULLN	Full Newton solution method is used during steady-state simulation.
GAIN	Stability factor (K), unilateral power gain (GU), maximum unilateral transducer power gain (GTUmax) and $\|H21\|^2$ are written to the LOG file. This is used in conjunction with the **.NET** statement.
GMIN	Specifies the minimum conductance used in SPICE models during transient analysis.
H.PARAM	H-parameters will be written to the LOG file. This is used in conjunction with the **.NET** statement.

LOADFLOATING	Specifies that the charge bias value for floating electrodes will be loaded.
LOADSOLUTIONS	Solutions, structures, doping distributions, and meshes, are to be loaded from standard structure files. The solutions are used as the initial guess or initial conditions for subsequent MixedMode simulation. To use this feature, you must: • Calculate a solution for each Atlas device and save each solution in a separate standard structure format file using **.SAVE** or **SOLVE**.... MASTER. • For each Atlas device, use the A statement in the MixedMode command file to specify the associated standard structure format file • Set node voltages to appropriate values with the **.NODESET** statement • Specify **LOADSOLUTIONS** in the **.OPTIONS** statement

Note: If using this feature, specify solutions for all Atlas devices.

The .NODESET statement must always be used when **LOADSOLUTIONS** is used. The **.NODESET** statement is used to make the initial circuit voltages match those device solutions that were obtained. You may also need to specify the **NOSHIFT** parameter of the **.OPTIONS** statement. By default, MixedMode shifts device terminal voltages with respect to the voltage on the first terminal that is specified in the A statement. You must either prepare initial solutions with this terminal grounded, or specify **NOSHIFT** in the **.OPTIONS** statement.

M2LN	Uses the modified two-level Newton solution method during steady-state simulation. The full NEWTON method provides faster solution than the modified two-level Newton method when a good initial guess is available. The modified two-level method is more reliable when the initial guess is far from the solution. The default is the full NEWTON method.
M2LN.TR	Uses the modified two-level Newton solution method during transient simulations.
NOPROJ	Disables the initial guess project method for the Atlas iterations. MixedMode attempts to extrapolate the values of the Atlas device variables (such as potential and carrier concentration) for the each iteration. Specifying NOPROJ disables the extrapolation and the previous values of potential and carrier concentration are used instead.
NOSHIFT	Disables the shift of voltages for Atlas device models. MixedMode normally shifts the voltages on Atlas device terminals to be referenced to the voltage on the first terminal. From the physical point of view, the state of the p-n diode is the same for voltages of 0V and 0.5V on the diode terminals with 1000V and 1000.5V, but the first situation is better for numerical simulation.
PRINT	Enables printing of circuit nodes voltages after the calculation for each bias point (DC analysis) or time step (transient the analysis).

RELPOT	Enables the use of relative convergence criteria for potential for Atlas models. By default, Atlas models use absolute convergence criteria for potential. When bias voltages are large (a common situation for power devices), then absolute convergence criteria are not appropriate and this parameter should be specified.
RV	Defines the ohmic resistance that MixedMode associates with all voltage sources and all inductances. This value should never be set to 0. The default value is small enough to avoid errors due to the influence of the internal resistance. Usually, extremely small values of this parameters can cause convergence problems. It is usually acceptable to decrease this parameter to the range of $1*10^{-6}$-$1*10^{-7}$. This parameter should not be varied unless there is a compelling reason to do so.
STR.XOFFSET	Specifies the X direction shift offset for multiple Atlas devices.
STR.XSHIFT	Specifies that multiple Atlas device will be shifted in the X direction.
STR.YOFFSET	Specifies the Y direction shift offset for multiple Atlas devices.
STR.YSHIFT	Specifies that multiple Atlas device will be shifted in the Y direction.
STR.ZOFFSET	Specifies the Z direction shift offset for multiple Atlas devices.
STR.ZSHIFT	Specifies that multiple Atlas device will be shifted in the Z direction.
TEMP	Device temperature to be use during the simulation.
TNOM	Circuit temperature to be use during the simulation.
WRITE	How often the solution is to be saved in standard structure files during the simulation. For example, `write=3` specifies that the solution will be saved at every third timestep. Specifying this parameter can help avoid disk overflow.
Y.PARAM	Y-parameters should be written to the log file. This is used in conjunction with the `.NET` statement.
Z.PARAM	Z-parameters should be written to the log file. This is used in conjunction with the `.NET` statement.
ZIP.SOLVER	Specifies that the ZIP libary version of BICGST iternative solver will be used to calculate the device conduction matrix instead of the CGS or BICGST solver. If you specify the `DIRECT` parameter in the **METHOD** statement, then the direct solver will be used for both the device conduction matrix calculation and the Atlas solution.

Example

```
.OPTIONS TNOM=293 FULLN
```

.PRINT

.PRINT specifies which device output parameters will be printed to the log files.

Syntax

```
.PRINT [antype]  parmeter(device_name) [parmeter2(device_name)...]
```

Description

antype	Type of analysis for which the outputs are desired. If antype is unspecified, the outputs of all simulation types will be printed. antype must be one of the following keywords: • **AC**: AC analysis outputs • **DC**: DC analysis outputs • **NET**: Network analysis outputs • **TRAN**: Transient analysis outputs • **parameter:** Output variables or expressions to be printed. • **device_name:** The device name.

Example

```
.PRINT ic(q1) ib(q1) is(q1)
.PRINT AC ic(q1) ib(q2)
.PRINT DC ic(q1) ib(q2) i(d1)
.PRINT TRAN cd(m1) cg(m1) cs(m1) cb(m1)
```

.PARAM

.PARAM specifies a parameter definition.

Syntax

```
.PARAM parname1=val1 [parname2=val2 ...]
```

The **.PARAM** statement defines one or more parameter labels by assigning the names parname1, parname2 to constant values val1, val2. Parameter labels must begin with alphabetic character(s) or the underscore character(s).

The **.PARAM** statement can be globally and locally (i.e., inside a subcircuit definition) specified. Once defined, a global parameter label or an expression containing global parameter labels can be used in the input deck when a numerical value is expected. A local parameter label can be used only inside the subcircuit.

.SAVE

.SAVE saves simulation results into files for visualization or for future use as an initial guess.

Syntax

```
.SAVE OUTFILE=name [MASTER=mname][TSAVE=timepts]
```

Description

MASTER	This will cause MixedMode to output standard structure files for visualization in TonyPlot. These files, with the base name, mname, will be written after the calculation of each bias point during DC simulation, and after of each time step during transient simulation. For example, if you wish to save a structure file for each DC bias solution, you would write: `.save master=my_structure.str` MixedMode will then output structure files at each DC bias point. MixedMode will also automatically append a number to the end of the file name indicating which bias point the file is for.

Example

```
.SAVE MASTER=my_filename.str
```

OUTFILE	Specifies that after the simulation is finished the solution is to be written to a file called my_filename.str. In other words, only one structure file will output after the simulation has finished. Even if you have several .dc solutions, MixedMode will only output one structure file if for example: `.save outfile=filename.str` has been used. The Atlas model solutions will be written to the file (name) and the circuit solution will be written to the file, name.cir. These files can be used later for loading solutions to be used as an initial guess (see .LOAD statement).

Example

```
.save outfile=my_filename.str
```

TSAVE	Specifies the transient time points at which a MASTER file will be written. The time points should be specified as a comma seperated list.

The program will automatically add the following suffixes to mname:

- During DC simulation: _dc_number, where number is the number of the DC point.
- During transient simulation: _tr_number, where number is the number of the time step.

Example

```
.SAVE OUTFILE=pdsave MASTER=pd
```

.SUBCKT

.SUBCKT specifies a subcircuit defintion.

Syntax

```
.SUBCKT subcktname <n1 n2 ...>
+ <OPTIONAL: optn1=defval1 <optn2=defval2 ...>>
+ <<PARAMS|PARAM:> parname1=val1 <parname2=val2 ...>>
```

Description

subcktname	The subcircuit name. It is used by an X element statement to reference the subcircuit.
n1 ...	The optional list of external nodes. Ground nodes (zero) are not allowed.
PARAMS|PARAM	The parameter preceding the list of parameters and their values.
parname1, parname2, ...	The name of a parameter with value set to val1, val2, ... that will be local to the subcircuit. val is a numerical value. If the same parameter is assigned, a value on more than one statement in the input deck (**.PARAM**, **.SUBCKT**, and X statements) will be one of the following: • a value assigned in a global **.PARAM** statement (outside subcircuits) is used if it exists. • a value assigned in a corresponding **X** statement is used if it exists. • a value assigned in a corresponding **.SUBCKT** statement is used if it exists. • a value assigned in a local **.PARAM** statement (inside the subcircuit) is used if it exists.

A subcircuit definition is initiated by a **.SUBCKT** statement. The group of element statements, which immediately follow the **.SUBCKT** statement define the subcircuit. The last statement in a subcircuit definition is the **.END** statement. Control statements cannot appear within a subcircuit definition; however, subcircuit definitions may contain anything else, including other subcircuit definitions, device models, and subcircuit calls.

All internal nodes (if not included on the **.SUBCKT** statement) are local with the exception of the global node 0.

.TRAN

.TRAN specifies that a transient analysis is to be performed.

Syntax

```
.TRAN tstep tstop DTCONST DTMAX DTMIN
```

Description

tstep	Time interval in seconds.
tstop	Final time value for which the simulation is to be performed.
DTCONST	Specifies that a constant time step is to be used for the current **.TRAN** statement.
DTMAX	Specifies the maximum time step for the current **.TRAN** statement.
DTMIN	Specifies the minimum time step for the current **.TRAN** statement.

Transient analysis is performed only after the execution of all .DC statements. If no DC statements are used, the transient starts after the calculation of the initial circuit state with the values of the independent sources given in the descriptions of those sources.

Multiple **.TRAN** statements are supported. The TSTOP parameter is not reset between each **.TRAN** statement.

Example

```
.TRAN 1ns 100ns
```

.VERILOG

.VERILOG specifies a Verilog-A file to be included.

Syntax

```
.VERILOG filename <module_name> <compile-options>
```

Description

filename	The name of a VerilogA file.
module_name	Optional Verilog-A module name.
compile_options	Optional Verilog-A Compilation options.

Example

```
.verilog "resistor.va"
```

13.4.3 Transient Parameters

MixedMode allows you to specify transient parameters for voltage sources (Vxxx), current sources (Ixxx), and resistors (Rxxx). These parameters describe the time development behavior of the source.

EXP

EXP is used to define an exponential waveform. The waveform is specified as follows:

```
EXP i1 i2 td1 tau1 td2 tau2
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td1** is the rise delay time.
- **td2** is the fall delay time.
- **tau1** is the rise time constant.
- **tau2** is the fall time constant.

The transient behavior is shown in Table 13-2.

Table 13-2 Transient behavior for EXP

Time	Value
$0 < t < td1$	i1
$td1 \leq t < td2$	i1+(i2-i1).(1-exp[-(t-td1)/tau1])
$td2 \leq t$	i1+(i2-i1).(1-exp[-(t-td1)/tau1]) +(i1-i2).(1-exp[-(t-td2)/tau2])

GAUSS

GAUSS is used to define a Gaussian waveform. The waveform is specified as follows:

```
GAUSS i1 i2 td1 tau1 td2 tau2
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td1** is the rise delay time.
- **td2** is the fall delay time.
- **tau1** is the rise time constant.
- **tau2** is the fall time constant.

The transient behavior is shown in Table 13-3.

Table 13-3 The Transient behavior for GAUSS	
Time	**Value**
0 < t < td1	i1
td1 ≤ t < td2	$i1+(i2-i1).(1-\exp[-((t-td1)/tau1)^2]$
td2 ≤ t	$i1+(i2-i1).(1-\exp[-((t-td1)/tau1)^2]$ $+(i1-i2).(1-\exp[-((t-td2)/tau2)^2]$

PULSE

PULSE is used to define a pulse waveform. The waveform is specified as follows:

```
PULSE i1 i2 td tr tf pw per
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td** is the delay time before the pulse is started.
- **tr** is the rise time of the pulse.
- **tf** is the fall time of the pulse.
- **pw** is the pulse length per period
- **per** is the period.

The transient behavior is described in Table 13-4. Intermediate points are found by linear interpolation.

Table 13-4 Transient behavior for PULSE	
Time	**Value**
0	i1
td	i1
td + tr	i2
td + tr + pw	i2
td + tr + pw + tf	i1
td + per	i1
td + per + tr	i2 (second period)

PWL

PWL is used to define a piece-wise linear waveform. The waveform is specified as a list of time and value point with the time values in ascending order. The waveform is specified as follows:

```
PWL t1, v1 [t2,v2 ....][R=tval]
```

where:

- **t1** is the first time value.
- **v1** is the bias value at time=t1.
- **t2** is the second time value (t2>t1).
- **v2** is the bias value at time=t2.\
- **R** is the repeat time point. This specifies the time point from which the PWL waveform should be repeated. The section of the waveform between tval and the end of the PWL will be repeated until the transient analysis is completed.

PWLFILE

PWLFILE is used to define a file name containing a piece-wise linear waveform. The waveform is specified as a list of time and value points with time values in ascending order. The file is specified as follows:

```
PWLFILE filename [R=tval]
```

where:

- **filename** is the PWL file.
- **R** is the repeat time point. This specifies the time point from which the PWLFILE waveform should be repeated. The section of the waveform between tval and the end of the PWL will be repeated until the transient analysis is completed.

SFFM

SFFM is used to define a modulated sinusoidal waveform. The waveform is specified as follows:

```
SFFM io ia fc mdi fs
```

where:

- **io** is the DC offset.
- **ia** is the amplitude.
- **fc** is the carrier frequency.
- **mdi** is the modulation index.
- **fs** is the signal frequency.

The transient behavior will be:

$$value(t)=io + ia \cdot \sin[\pi \cdot fc \cdot t + mdi \cdot \sin(2\pi \cdot fs \cdot t)]$$

SIN

SIN is used to define a sinusoidal waveform. The waveform is specified as follows:

```
SIN io ia freq td theta
```

where:

- **io** is the offset.
- **ia** is the amplitude.
- **freq** is the frequency.
- **td** is the delay.
- **theta** is the damping factor.

The transient behavior is shown in Table 13-5.

Table 13-5 Transient Behavior for SIN

Time	Value
t < td	value(t) = io
t ≥ td	value(t) = i0+ia.exp[-(t-td)/THETA] .sin[2π.freq.(t-td)]

TABLE

TABLE is used to define a waveform using a table of values. This parameter is used as follows:

```
TABLE infile=<table_file_name>[R=tval]
```

where `table_file_name` is an ASCII text file that contains the tabulated time-dependence of a variable in the following format:

```
t1 v1
t2 v2
t3 v3
...
tN vN
end
```

Each line contains two numbers. The first number is the time in seconds. The second number is the time-dependent variable) voltage in volts, the current in amps, or the resistance in ohms). Up to 1000 lines can be used. Input is terminated by the word end.

If during the simulation the transient time becomes larger than the last value in the table, then the last value will be used for the remainder of the simulation.

R is the repeat time point. This specifies the time point from which the TABLE waveform should be repeated. The section of the waveform between `tval` and the end of the PWL will be repeated until the transient analysis is completed.

13.4.4 Expressions

Expressions can be used at input deck statements to specify (by means of calculations) the value of a parameter.

Functions

MixedMode supports the functions listed in Table 13-6. The table gives the definition for each function, and shows the types of function arguments.

Table 13-6: Functions

Function	Scalar	Vector	Description	Notes
`abs(x)`	+	+	Absolute value x	
`acos(x)`	+	+	Arc cosine x	
`acosh(x)`	+	+	Hyperbolic arc cosine x	
`asin(x)`	+	+	Arc sine x	
`asinh(x)`	+	+	Hyperbolic arc sine x	
`arctan(x)`	+	-	Arc tangent x	
`atan(x)`	+	+	Arc tangent x	
`atanh(x)`	+	+	Hyperbolic arc tangent x	
`cos(x)`	+	+	Cosine of x	
`cosh(x)`	+	+	Hyperbolic cosine of x	
`db(x)`	-	+	Magnitude of the complex vector in decibels: $db(x) = 20 \times \log(mag(x))$	
`erf(x)`	+	+	Error function $erf(x) = \frac{2}{\sqrt{\pi}} \times \int_0^x e^{-t^2} dt$	
`erfc(x)`	+	+	Error function complement *erfc(x) = 1 - erf(x)*	
`exp(x)`	+	+	Exponential: *e* raised to power *x*	
`int(x)` or `trunc(x)`	+	+	Returns the integer part of x	
`log10(x)`	+	+	Logarithm (base 10) of x	
`log(x)` or `ln(x)`	+	+	Natural logarithm (base *e*) of x	

Table 13-6: Functions

Function	Scalar	Vector	Description	Notes
`max(x,y)`	+	+	Maximum of two values, if $x>y$ then $max(x,y)=x$; otherwise $=y$	
`min(x,y)`	+	+	Minimum of two values, if $x<y$ then $min(x,y)=x$; otherwise $=y$	
`pow(x,y)`	+	-	Absolute power: absolute value of x raised to integer part of y: pow(x,y) = abs(x)^int(y)	
`rnd(x)`	+	+	Returns a random integer number (or the vector with each component a random integer number) between 0 and absolute value x (absolute value of the vector x components)	
`sgn(x)`	+	+	Sign of value, if $x<0$ then $sgn(x) = -1$; if $x=0$ then $sgn(x) = 0$; if $x>0$ then $sgn(x) = 1$	
`sin(x)`	+	+	Sine of x	
`sinh(x)`	+	+	Hyperbolic sine of x	
`sqrt(x)`	+	+	Square root of x	
`tan(x)`	+	+	Tangent of x	
`tanh(x)`	+	+	Hyperbolic tangent of x	
`valif(log_cond,x,y)` or if `(log_cond, x,y)`- in `'-pspice'` mode	+	-	Logical expression: if(`log_cond`) then x else y	`log_cond` is the a logical condition, which consist of two expressions compared using operators: ==; !=; <; >; <=; >= or their alphabetical equivalents: eq, ne, lt, gt, le,ge.

Operators

Table 13-7 lists the operations supported by MixedMode. The results of the modulo operation is the remainder when the first number is divided by the second number. Both modulo operation arguments are rounded down to the nearest integer before the operation is performed.

Table 13-7: Operations

Operation Type	Operator	Meaning	Example	Synonym	Notes
Algebraic	+	Plus	a + b		
	+	Unary Plus	+a		
	-	Minus	a - b		
	-	Unary Minus	-a		
	*	Multiply	a*b		
	/	Divide	a/b		
	^	Power	a^b		
	%	Modulo	a%b		
	,	Comma	a,b		
Logical	&	And	a&b	and	
	|	Or	a|b	or	
	&&	And	a&&b	and	
	||	Or	a||b	or	
	!	Unary negation	!a	not	
	[cond] ? x:y	Ternary conditional expression			See Example below
Relational	<	Less than	a<b	lt	
	>	Greater than	a>b	gt	
	<=	Less than or equal	a<=b	le	
	>=	Greater than or equal	a>=b	ge	
	==	Equal	a==b	eq	
	!=	Not equal	a!=b	ne	

Example

```
.param Param1=5 Param2=10
+ Test='(Param>Param2) ? 1 : 2'
+ Test2='5*((Param1<Param2) ? (Param+1) : (Param2+2))'
```

Chapter 14
Quantum: Quantum Effect Simulator

14.1 Quantum Effects Modeling

There are several quantum effects models that work with Atlas to simulate various effects of quantum mechanical confinement. These models are generally independent and should not be used simultaneously in any given simulation. These models are described in the following sections.

In Section 14.2 “Self-Consistent Coupled Schrodinger Poisson Model”, we discuss the self-consistent Schrodinger-Poisson model. This model self-consistently solves Poisson's equation (for potential) and Schrodinger's equation (for bound state energies and carrier wavefunctions). This model should not be used to look at carrier transport problems.

In Section 14.3 “Density Gradient (Quantum Moments Model)”, we describe the density gradient model. This model is based on the moments of the Wigner function equations of motion and calculates a quantum correction to the carrier temperatures in the transport equations. This model can accurately reproduce the carrier concentration predicted by the Schrodinger-Poisson model but can also predict transport properties. This model cannot, however, predict the bound state energies or wave functions given by the Schrodinger-Poisson model.

In Section 14.4 “Bohm Quantum Potential (BQP)”, we describe the Bohm quantum potential model. This model can be used to address a similar set or problems as the density gradient model. The Bohm quantum potential model has two advantages over the density gradient model: 1) better convergence and 2) better calibration to the Schrodinger-Poisson model.

In Section 14.5 “Quantum Correction Models”, we describe two quantum correction models. These models by Van Dort and Hansch are both phenomenological models that give corrections for the effects of quantum confinement in the inversion layer under the gate of MOSFET devices.

In Section 14.6 “Parabolic Quantum Well Model”, we describe a general quantum well model used to predict the gain and spontaneous recombination in quantum well light emitting devices. This model solves Schrodinger equation to calculate the bound state energies and wave functions. The bound state energies are used to predict the gain and spontaneous recombination rates and the wave functions can be used to predict the overlap integral. The capture-escape model is used together with Quantum Well Model to predict position of Fermi levels of quantum wells.

In Section 14.8 “Quantum Transport: Non-Equilibrium Green’s Function Approach”, we describe a fully quantum non-equilibrium Green’s function approach to modeling transport in nanoscale devices with strong transverse confinement. This models predicts eigen energies and eigen functions, quantum electron and curent density, and ballistic current-voltage characteristics.

In Section 14.9 “Drift-Diffusion Mode-Space Method (DD_MS)”, we describe a semiclassical approach to modeling transport in nanoscale devices with strong transverse confinement. This model is coupled to Atlas models for mobility, generation, and recombination. It also predicts eigen energies and eigen functions, quantum electron and curent density, and drift-diffusion current-voltage characteristics.

14.2 Self-Consistent Coupled Schrodinger Poisson Model

The solution of Schrodinger's equation gives a quantized description of the density of states in the presence of quantum mechanical confining potential variations.

The N.SCHRO parameter of the **MODELS** statement enables the self-consistent coupled Schrodinger-Poisson model for electrons. Set the P.SCHRO parameter in the **MODELS** statement to enable the Schrodinger-Poisson model for holes. You can specify both SCHRO and P.SCHRO on the same statement to model two carrier types simultaneously. Atlas2D can solve Schrodinger equation in 1D slices or 2D plane. In the case of cylindrical coordinates, Schrodinger equation is solved in radial direction for different orbital quantum numbers and for all slices perpendicular to the axis. Atlas3D solves a 1D Schrodinger equation along one of the axes at all in-plane nodes or 2D Schrodinger equation in plane slices. To set up dimensionality and direction of the solver, use the SP.GEOM parameter on the **MODELS** statement. Table 14-1 shows the possible values of the SP.GEOM parameter. If the parameter is not specified, Atlas will use a default value, depending on the situation. Note that the 2DXY option in cylindrical coordinates corresponds to a case of a 3D quantum confinement in radial, axial, and azimuthal directions.

Table 14-1 Geometry of Schrodinger solver

SP.GEOM	Geometry	Comments
1DX	1D in X direction	default for cylindrical mesh in Atlas2D.
1DY	1D in Y direction	default for non-cylindrical mesh in Atlas2D.
1DZ	1D in Z direction	Unavailable in Atlas2D.
2DXY	2D in XY plane	Default in Atlas2D if option 2DXY.SCHRO is on Default in Atlas3D for cylindrical or non Atlas mesh or if option 2DXY.SCHRO is on
2DXZ	2D in XZ plane	Unavailable in Atlas2D.
2DYZ	2D in YZ plane	Default for Atlas mesh in Atlas3D. Unavailable in Atlas2D.

When the quantum confinement is in one dimension (along Y axis), the calculation of the quantum electron density relies upon a solution of a 1D Schrodinger equation solved for eigen state energies $E_{i\nu}(x)$ and wavefunctions $\Psi_{i\nu}(x,y)$ at each slice perpendicular to the X axis and for each electron valley (or hole band) ν.

$$-\frac{\hbar^2}{2}\frac{\partial}{\partial y}\left(\frac{1}{m_y^{\nu}(x,y)}\frac{\partial \Psi_{i\nu}}{\partial y}\right) + E_C(x,y)\Psi_{i\nu} = E_{i\nu}\Psi_{i\nu} \qquad 14\text{-}1$$

Here, $m_y^{\nu}(x,y)$ is a spatially dependent effective mass in Y direction for the ν-th valley and $E_C(x,y)$ is a conduction band edge. The equation for holes is obtained by substituting hole effective masses instead of electron ones and valence band edge $-E_V(x,y)$ instead of $E_C(x,y)$. Analogously, in cylindrical coordinates, the equation for the radial part of the wavefunction $R_{im\nu}(r)$ for each orbital quantum number m reads

$$-\frac{\hbar^2}{2}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{1}{m_r^{\nu}(r,z)}r\frac{\partial R_{im\nu}}{\partial r}\right)-\frac{1}{m_r^{\nu}(r,z)}\frac{m^2}{r^2}\right]+E_C(r,z)R_{im\nu}=E_{im\nu}R_{im\nu} \quad (14\text{-}2)$$

Finally, a 2D Schrodinger equation reads:

$$-\frac{\hbar^2}{2}\left[\frac{\partial}{\partial x}\left(\frac{1}{m_x^{\nu}(x,y)}\frac{\partial\Psi_{i\nu}}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{m_y^{\nu}(x,y)}\frac{\partial\Psi_{i\nu}}{\partial y}\right)\right]+E_C(x,y)\Psi_{i\nu}=E_{i\nu}\Psi_{i\nu} \quad (14\text{-}3)$$

In order to take into account exchange-correlation effects, you can specify the SP.EXCORR parameter on the **MODELS** statement. The exchange-correlation, treated within local density approximation, is an additional negative potential energy term in the Hamiltonian, which depends on the local carrier density and is given by

$$V_{exc}=-\frac{q^2}{4\pi^2\varepsilon\varepsilon_0}[3\pi^2 n(x,y,z))]^{1/3} \quad (14\text{-}4)$$

The number of hole bands is set by NUM.BAND parameter on the **MODELS** statement. If NUM.BAND is set to 1, a solution for only one valence band MV (on the **MATERIAL** statement) is obtained. If NUM.BAND is set to 3, a solution for heavy holes, light holes, and split-off band holes (MHH, MLH, and MSO on the **MATERIAL** statement) is obtained. If NUM.BAND is set to 2, only heavy holes and light holes are taken into account. Hole effective mass is assumed to be isotropic.

To choose the number of electron valleys and anisotropy of effective mass, specify the NUM.DIRECT on the **MODELS** statement. If NUM.DIRECT is set to 1, a solution for one Γ-valley with isotropic effective mass MC (on the **MATERIAL** statement) is obtained. If NUM.DIRECT is set to 3, a solution for three doubly-degenerate X-valleys with anisotropic effective mass is obtained. If NUM.DIRECT is set to 4, a solution for four L-valley with anisotropic effective mass is obtained.

In case of anisotropic electron effective mass, when NUM.DIRECT is set to 3 or 4, the value of effective mass in the quantization and the perpendicular directions in each valley depends on the crystallographic orientation, which is set by X.ORIENT, Y.ORIENT, and Z.ORIENT on the **REGION** statement. For example:

```
region num=1  y.min=0 y.max=1 mat=silicon y.orient='<1,1,1>'

region  num=1  y.min=0  y.max=1  mat=germanium  y.orient='<1,1,0>'
x.orient='<1,-1,0>'
```

The default orientation is <1,0,0> along X, <0,1,0> along Y and <0,0,1> along Z axis. If one or two axes are not specified, they are chosen as perpendiculars to the specified axes.

Effective mass tensor $\omega_{i,j}$, whose principal values are ω_{t1}=MT1^{-1},ω_{t2}= MT2^{-1}, and ω_l=ML^{-1}, where MT1, MT2, and ML are set on the **MATERIAL** statement is rotated in accordance with the relative orientation of X, Y, and Z axes of the device and the principal axes of each valley. In case of 1D Schrodinger equation, to get effective masses in the quantization (q) and the two perependicular directions ($p1$,$p2$), the following expressions are used [301]:

$$m_q=1/\omega_{q,q} \quad (14\text{-}5)$$

$$m_{p1}=1/(\omega_{p1,p2}-\omega_{p1,q}^2/\omega_{q,q}) \quad (14\text{-}6)$$

$$m_{p2} = 1/(\omega_{p2,p2} - \omega_{p2,q}^2/\omega_{q,q}) \quad 14\text{-}7$$

In case of 2D Schrodinger equation, to get effective masses in two quantization ($q1$,$q2$) and the one perependicular directions (p), the following expressions are used [30]:

$$m_{q1} = 1/\omega_{q1,q1} \quad 14\text{-}8$$

$$m_{q1} = 1/\omega_{q1,q2} \quad 14\text{-}9$$

$$m_p = 1/(\omega_{q1,q1}\omega_{q2,q2} - \omega_{q1,q2})/\omega_{t1}\omega_{t2}\omega_l \quad 14\text{-}10$$

In case of cylindrical Schrodinger equation, to get effective masses in radial ($q1$,$q2$) and axial directions (r), the following expressions are used:

$$m_{\mathrm{rad}} = 2/(\omega_{q1,q1} + \omega_{q2,q2}) \quad 14\text{-}11$$

$$m_{\mathrm{axis}} = 1/(\omega_{q1,q1}\omega_{q2,q2} - \omega_{q1,q2})/\omega_{t1}\omega_{t2}\omega_l \quad 14\text{-}12$$

The electron valleys are denoted in TonyPlot as Valley #1, Valley #2, Valley #3 and Valley #4.

Schrodinger equation uses a conduction or valence band edge of the bulk material. You have an option to set an additional band off-set manually without getting into details of its origin. To do so, use the following parameters from the **MODELS** or **REGION** or **MATERIAL** statements.

Table 14-2 Band off-set

Valley/Band	Band off-set
Longitudinal	DEC.C1
Transverse1	DEC.C2
Transverse2	DEC.C3
Isotropic Electron Band	DEC.ISO (when NUM.DIRECT=1)
Heavy Holes	DEV.HH
Light Holes	DEV.LH
Split-off Holes	DEV.SO
Isotropic Hole band	DEV.ISO(when NUM.BAND=1)

The expression for the electron concentration is obtained by using Fermi statistics and eigen anergies and wave functions:

$$n(x,y) = 2\frac{k_B T}{\pi h^2}\sum_{\nu}\sqrt{m_x^{\nu}(x,y)m_z^{\nu}(x,y)}\sum_{i=0}^{\infty}\left|\Psi_{i\nu}(x,y)\right|^2 \ln\left[1+\exp\left(-\frac{E_{i\nu}-E_F}{k_B T}\right)\right] \quad 14\text{-}13$$

for 1D confinement and

$$n(x,y) = 2\frac{\sqrt{2k_B T}}{h}\sum_{\nu}\sqrt{m_z^{\nu}(x,y)}\sum_{i=0}^{\infty}\left|\Psi_{i\nu}(x,y)\right|^2 F_{-1/2}\left(-\frac{E_{i\nu}-E_F}{k_B T}\right) \quad 14\text{-}14$$

for a 2D confinement case and cylindrical case, where $F_{-1/2}$ is the Fermi-Dirac integral of order -1/2.

Atlas will search for a minimum number of filled eigen states in the range between $(-\infty; 20kT]$. You can also use EIGEN=X when a certain fixed number X of eigen states is required. In the case of cylindrical mesh in Atlas2D, Schrodinger equation has two quantum numbers: radial number n_r and orbital number m. All states with non zero m are doubly degenerate, which corresponds to $+m$ and $-m$ or to clockwise and counter-clockwise polarization of electron wave function. Therefore, when you set the EIGEN parameter to X, Atlas will store X states for m=0 and X states for each of 2*(X-1) orbital numbers $m = \pm 1, \pm 2, \ldots \pm X$. This will add up to a total of X*(2X-1) states. In TonyPlot, the states are sorted in ascending order with respect to their eigen energies, while the information about values of radial and orbital numbers (n_r,m) is not stored. You can still determine n_r and m of a state, using a general property of wave fuctions $R_{nr,m}(r)$. n_r is always equal to the number of times the wave function changes sign $+1$. For $m=0$, $R_{nr,m}(r)$ is always non zero and has zero first derivative at r=0. For $m=1$, $R_{nr,m}(r)$ is always zero and has zero second derivative at r=0. For $m>1$, $R_{nr,m}(r)$ is always zero and has zero first derivative at r=0.

By default, electron penetration into insulator regions is not allowed. Use the OX.SCHRO parameter on the **MODELS** statement to allow electron penetration and OX.MARGIN= to specify the maximum penetration distance (default is 0.0003 microns). By default, Atlas attempts to find a solution in all semiconducting regions. If a solution is unnecessary in a particular region, switch off SCHRO parameter on the **REGION** statement for that particular region.

When Schrodinger equation is solved in 1D or 2D slices, it is required that all nodes are aligned (1D) or lie in one plane (2D). In case of unstructured or non-Atlas mesh, Atlas can find a solution on an additional rectangular mesh and then interpolate results onto original mesh. To set an additional rectangular mesh, use **SPX.MESH**, **SPY.MESH**, and **SPZ.MESH** statements in a way similar to **X.MESH**, **Y.MESH**, and **Z.MESH**. In Atlas3D, if **Z.MESH** is not specified, Z planes of original mesh will be used.

Note that 2D Schrodinger does not require rectangular mesh within the 2D slice and can handle meshes created outside of Atlas (e.g., DevEdit or Athena). In devices with crossection close to rectangular, Atlas gives an option to speed up a solution by using the SP.FAST parameter on the **MODELS** statement. When this parameter is on, a 2D solution is sought in the form of linear combination of products on 1D solutions in the middle of the solution domain along perpendicular axes. The solution will be exact for rectangular structure and deviate slightly for non rectangular. SP.FAST parameter requires a rectangular mesh. If it is unavailable, use **SPX.MESH**, **SPY.MESH**, and **SPZ.MESH** to set up an additional rectangular mesh.

Since the wavefunctions diminish rapidly from the confining potential barriers in the Schrodinger solutions, the carrier concentrations become small and noisy. You can refine these carrier concentrations by setting a minimum carrier concentration using the QMINCONC parameter on the **MODELS** statement. This parameter sets the minimum carrier concentration passed along to the Poisson solver and the output to the structure files. The transition between the Schrodinger solution and the minimum concentration is refined between 10×QMINCONC and QMINCONC so that it is continuous in the first derivative.

Once the carrier concentration is calculated using Equations 14-13 and 14-14, it is substituted into the charge part of Poisson's Equation. The potential derived from solution of Poisson's equation is substituted back into Schrodinger's equation. This solution process (alternating between Schrodinger's and Poisson's equations) continues until convergence and a self-consistent solution of Schrodinger's and Poisson's equations is reached. A default predictor-corrector scheme is used to avoid instability and oscillations of Poisson convergence. In this scheme, after wave functions are found and fixed, eigen energies are changed locally and plugged into Poisson equation (predictor). After several predictor iterations, a new set of wave functions and eigen energies is computed (corrector). You can control the number of Predictor iterations by a parameter NPRED.NEGF= (default is 7) on the **MODELS** statement.

The convergence criterion for potential for Schrodinger-Poisson as well as for NEGF and DDMS models is given by the QCRIT.NEGF parameter on the **MODELS** statement with a default of 0.001 eV. Maximum number of iterations after which Schrodinger-Poisson proceeds to the next bias is set by SP.NUMITER with a default of 30. Atlas also checks for oscillatory convergence behavior for the last SP.NUMOSC (default 30) iterations starting from iteration number SP.STAOSC (default 3). If logarithm of potential update or residual is smaller than SP.MINDIF (default 0.001), then Atlas will proceed to the next bias point.

For modeling of devices with transverse confinement, it is sometimes more preferable to set von Neumann Boundary Conditions (BC) for potential in a contact. Von Neumann BC will be applied if you set the REFLECT parameter on the **CONTACT** statement and if SCHRO or P.SCHRO parameter is present on the **MODELS** statement.

To set number of eigen states stored in a structure file, use the EIGENS parameter on the **OUTPUT** statement. Use the **SAVE** statement or the OUTFILE parameter on the **SOLVE** statement to write the solutions of the self-consistent system into a structure file.

In obtaining self-consistent solutions for the Schrodinger's Equation, an assumption is made about the location of the electron or hole quasi-fermi level. Normally, Atlas tries to set the quasi-fermi to that of a contact attached to the region where solution is obtained. To ensure that the fermi level is zero, set FIXED.FERMI parameter on the **MODELS** statement.

If you want to solve Schrodinger equation as a postprocessing step under non-equilibrium, specify CARRIERS=2 or CARRIERS=1 HOLES or CARRIERS=1 ELECTRONS on the METHOD statement. In this case, a classical quasi Fermi level will be computed by solving drift-diffusion equations. It is then used to find quantum electron density self-consistently with Poisson equation. Flag FIXED.FERMI will be ignored. In case of bias sweep, Atlas will solve Schrodinger for each bias point. The SP.LAST parameter on the **SOLVE** statement allows to solve the Schrodinger equation with classically computed Fermi levels only for the last bias point.

14.3 Density Gradient (Quantum Moments Model)

The quantum models apply to several different types of problems. These problems are HEMT channel confinement simulation, thin gate oxide MOS capacitors and transistors, and other problems such as small geometry MESFETs and heterojunction diodes.

The effects due to confinement of carriers associated with variations of local potential on the scale of the electron wave functions (i.e., quantum effects) can be modeled in Atlas using a density gradient. This model is based on the moments of the Wigner function equations-of-motion [379, 380, 361, 348], which consists of quantum correction to the carrier temperatures in the carrier current and energy flux equations (see Equations 3-20 to 3-23).

In the density gradient model, the expression for electron current given in Equation 3-11 is replaced by the expression given in Equation 14-15.

$$\vec{J}_n = qD_n\nabla n - qn\mu_n\nabla(\psi - \Lambda) - \mu_n n(kT_L\nabla(\ln n_{ie})) \quad 14\text{-}15$$

Here, Λ is a quantum correction potential.

Equation 14-16 is the density gradient model for holes.

$$\vec{J}_p = -qD_p\nabla p - qp\mu_p\nabla(\psi - \Lambda) + \mu_p p(kT_L\nabla(\ln n_{ie})) \quad 14\text{-}16$$

The quantum potential can be calculated using either of the two expressions given in Equations 14-17 and 14-18.

$$\Lambda = -\frac{\gamma h^2}{12m}\left[\nabla^2 \log n + \frac{1}{2}(\nabla \log n)^2\right] \quad 14\text{-}17$$

$$\Lambda = -\frac{\gamma h^2}{6m}\frac{\nabla^2\sqrt{n}}{\sqrt{n}} \quad 14\text{-}18$$

where γ is a fit factor, m is the carrier effective mass, and n represents electron or hole concentration as appropriate. To choose between these expressions, specify DGLOG for Equation 14-17 or DGROOT for Equation 14-18 in the **MODELS** statement. By default, DGLOG is used. The fit factors for electrons and holes can be specified independently using the DGN.GAMMA or the DGP.GAMMA parameters of the **MODELS** statement respectively.

These expressions and particularly their derivatives can be seen to be sensitive to low carrier concentrations. You can specify a minimum concentration to use in these calculations using the QMINCONC parameter in the **MODELS** statement. This parameter specifies the minimum concentration used in the calculations of Equations 14-17 and 14-18 in cm^{-3}.

You can activate the model for electrons and holes independently. To activate the quantum model with the quantum moments equation for electrons, use the quantum switch, **MODELS** QUANTUM, in the **MODELS** statement. To activate the quantum moments equation for holes, use the quantum switch, **MODELS** P.QUANTUM.

Specify T.QUANTUM in the **OUTPUT** statement to write the quantum temperatures in the standard structure file. Once written you can examine the quantum temperature distribution using TonyPlot.

Since the distributions of carriers given by the density gradient can vary greatly from the distributions predicted by the standard drift-diffusion or energy-balance models, the standard initial guess strategies (e.g., INIT) aren't usually suitable for obtaining solutions for quantum moments. Until more suitable initial guess strategies can be devised, we've included a damping factor to gradually apply to the density gradient.

This damping factor is specified by the QFACTOR parameter in the **SOLVE** statement. The QFACTOR is implemented as a pre-factor to the expression for the quantum correction potential, Λ, in Equations 14-17 and 14-18. As such, a value of QFACTOR of 0.0 implies that the density gradient is either turned off or not applied. A value of QFACTOR of 1.0 implies that the Quantum Moment Model is turned on and applied.

You can vary the value of QFACTOR between 0.0 and 1.0 to overcome the problems of initial guess. During this ramping of QFACTOR, PREVIOUS should be used as an initial guess. Also, while varying the QFACTOR, trapping is available if a given solution doesn't converge. In such cases, the QFACTOR is reduced and solutions proceed until convergence is obtained or the maximum number of trap steps is exceeded.

The following is a typical fragment from an input file that ramps the QFACTOR parameter from 0 to 1:

```
MODEL NUMCARR=1 QUANTUM FLDMOB CONMOB SRH PRINT
OUTPUT CON.BAND VAL.BAND BAND.PARAM T.QUANTUM
SOLVE INIT
SOLVE QFACTOR=0.0
SOLVE QFACTOR=0.0001
SOLVE QFACTOR=0.001
SOLVE QFACTOR=0.01
SOLVE QFACTOR=0.1
SOLVE QFACTOR=1.0
LOG OUTF=test.log
SOLVE VDRAIN=0.01
```

14.4 Bohm Quantum Potential (BQP)

This model was developed for Silvaco by the University of Pisa and has been implemented into Atlas with the collaboration of the University of Pisa. This is an alternative to the Density Gradient method and can be applied to a similar range of problems. There are two advantages to using Bohm Quantum Potential (BQP) over the density gradient method. First, it has better convergence properties in many situations. Second, you can calibrate it against results from the Schrodinger-Poisson equation under conditions of negligible current flow.

The model introduces a position dependent quantum potential, Q, which is added to the Potential Energy of a given carrier type. This quantum potential is derived using the Bohm interpretation of quantum mechanics [135] and takes the following form

$$Q = \frac{-h^2\gamma\underline{\nabla}(M^{-1}\underline{\nabla}(n^{\alpha}))}{n^{\alpha}} \qquad \text{14-19}$$

where γ and α are two adjustable parameters, M^{-1} is the inverse effective mass tensor and n is the electron (or hole) density. This result is similar to the expression for the quantum potential in the density gradient model with $\alpha = 0.5$, but there are some differences about how they are implemented. Q is added to the continuity equations the same way Λ is for the density gradient method as shown in Equations 14-15 and 14-16.

The Bohm Quantum Potential (BQP) method can also be used for the Energy balance and hydrodynamic models, where the semi-classical potential is modified by the quantum potential the same way as for the continuity equations.

The default iterative scheme used to solve the non-linear BQP equation along with a set of semi-classical equations is as follows. After an initial semi-classical solution has been obtained, the BQP equation is solved on its own Gummel iteration to give Q at every node in the device. The semi-classical potential is modified by the value of Q at every node and the set of semi-classical equations is then solved to convergence as usual (using a Newton or block iterative scheme). Then, the BQP equation is solved to convergence again and the process is repeated until self-consistency is achieved between the solution of the BQP equation and the set of semi-classical equations. The set of semi-classical equations solved can be any of the combinations usually permitted by Atlas.

You can enable an alternative iteration scheme by specifying the `BQP.NEWTON` flag on the **`METHOD`** statement. This causes the BQP equation to be included in the fully coupled Newton iteration. This can only be used with either `BQP.N` or `BQP.P` models seperately and not with both enabled. Additionally, this model also modifies the intrinsic carrier density, n_i, by the Bohm quantum potential. This ensures that under equilibrium conditions $np - n_i^2 = 0.0$. On subsequent `SOLVE` statements, the `FREEZEBQPNI` flag should be enabled to ensure that the value of n_i used is the equilibrium one

To use the BQP model for electrons (or holes), specify `BQP.N` (`BQP.P`) in the **`MODELS`** statement. You can also set the parameter values (α and γ) and the direction of the quantization (confinement). Tables 14-3 and 14-4 show the parameters to use for the **`MODELS`** statement.

Table 14-3 MODEL Statement Parameters for Electrons			
Parameter	**Type**	**Default**	**Units**
BQP.N	Logical	false	
BQP.NGAMMA	Real	1.2	--
BQP.NALPHA	Real	0.5	--
BQP.QDIR	Integer	2	--

Table 14-4 MODEL Statement Parameters for Holes			
Parameter	**Type**	**Default**	**Units**
BQP.P	Logical	false	
BQP.PGAMMA	Real	1.0	--
BQP.PALPHA	Real	0.5	--
BQP.QDIR	Integer	2	--

BQP.N switches on the model for electrons. BQP.P switches it on for holes. The BQP.NGAMMA and BQP.PGAMMA allow you to set the γ parameter for electrons and holes respectively. BQP.NALPHA and BQP.PALPHA allow you to set the α parameter for electrons and holes respectively.

You can specify BQP.N and BQP.P separately. But if quantum effects for electrons and holes occur in the same device, then you can specify them together in the same **MODELS** statement.

The BQP.QDIR parameter specifies the principal quantization direction.

BQP.QDIR = 1 means the X direction.

BQP.QDIR = 2 means the Y direction.

BQP.QDIR = 3 means the Z direction.

BQP.QDIR = 4 means the radial direction (for 3D cylindrical structures only).

For example, if a MOSFET channel is in the XY plane, the direction of quantization (quantum confinement) is the Z direction you would set BQP.QDIR=3. For semiconductors with spherical bands, BQP.QDIR will have no effect.

The special case of BQP.QDIR = 4 applies to 3D cylindrical structures only. It specifies quantization in the radial direction. Assuming cubic symmetry of the semiconductor band structure, the quantization direction will be aligned with the principal valley axes 4 times through a 360° rotational span of the structure.

In other words, the effective mass tensor has four-fold rotational symmetry about a vertical axis. For convenience, the principal directions are taken to be the (±1, 0, 0) and the (±0, 1, 0) directions in (x,y,z) space. Therefore to use this model correctly, the angular mesh must cover at least 90°.

14.4.1 Calibration against Schrodinger-Poisson Model

You can obtain close agreement between BQP and the results of Schrodinger-Poisson (S-P) calculations for any given class of device. To obtain comparisons with S-P results, we recommend to use either the new quasistatic capacitance-voltage profile feature or compare charge-voltage curves. This will ensure similar charge control properties between the two models.

The first part of the calibration is to choose a suitable biasing for the device. There should be negligible current flow and quantum confinement effects that manifest at the chosen biases. The second part of the calibration is to set the appropriate BQP parameters in the **MATERIAL** or **MODELS** statements, and to set CARRIERS=0 in the **METHOD** statement.

This will cause the BQP equation to be coupled with Poisson's equation using the charge density terms.

$$n = N_c \exp\left(-\frac{(E_c + qQ)}{kT_L}\right) \quad \text{14-20}$$

$$p = N_v \exp\left(-\frac{(qQ - E_v)}{kT_L}\right) \quad \text{14-21}$$

This gives the same results as solving the current continuity equations with the constraint of zero current density.

The third part of calibration is to choose the quantity to compare with S-P results.

For example, for a MOSFET holding the drain and source voltages at the same bias and ramping the gate bias will give us a bias dependent capacitance with negligible current flow. So for an NMOS, you may have the statement

```
SOLVE VGATE=0.0 NAME=GATE VSTEP=0.01 VFINAL=2.0 QSCV
```

to give us the quasi-static C-V curve as it is biased into inversion. It is best to use a fine voltage step with QSCV to give good resolution. The process can be repeated by setting the S-P model in the **MODELS** statement instead of BQP to obtain the same set of curves for the S-P model.

The BQP model is then rerun with different sets of parameters until an acceptable agreement with the curves produced by the S-P model is achieved.

14.4.2 Post Calibration Runs

After obtaining the parameters for BQP, either the Drift-Diffusion or energy balance (hydrodynamic) equations can be solved as usual. For cases where Lattice Heating is important then LAT.TEMP can be enabled at the same time.

The iteration scheme uses a modified version of BLOCK. Set BLOCK in the **METHOD** statement (although NEWTON and GUMMEL are ignored, BLOCK is always used if the BQP model is set). If an energy balance model is chosen (HCTE.EL or HCTE.HO on the **MODELS** statement), then an alternative iteration scheme will become available by specifying BQP.ALTEB in the **METHOD** statement. This method is slower and is only available in Atlas2D but may have better convergence properties.

By using BQP.NOFERMI, the BQP equation will only use its Boltzmann statistics form. Without this parameter, the statistics used are those specified for the other equations. With fermi statistics, the convergence can be poor for very high carrier densities, and this parameter can circumvent some convergence properties.

Re-calibrate the BQP parameters if you set BQP.NOFERMI.

Table 14-5 METHOD Statement Parameter

Parameter	Type	Default	Units
BQP.ALTEB	Logical	false	
BQP.NEWTON	Logical	false	
BQP.NOFERMI	Logical	false	
BQPX.TOL	Real	2.5×10^{-7}	BQPX.TOL
BQPR.TOL	Real	1.0×10^{-26} (2D) 1.0×10^{-18} (3D)	BQPR.TOL
BQP.NMIX	Real	1	BQP.NMIX
BQP.PMIX	Real	1	BQP.PMIX
NBLOCKIT	Integer	15	NBLOCKIT
ITLIMIT	Integer	25	ITLIMIT
GUMITS	Integer	100	GUMITS

To speed up convergence, specify the NOCURRENT parameter on the first **SOLVE** statement after **SOLVE** INIT. It should prevent the need to use the QFACTOR parameter as was necessary for the density gradient method. QSCV enables the quasistatic capacitance calculation and output.

Table 14-6 SOLVE Statement Parameters			
Parameter	**Type**	**Default**	**Units**
FREEZEBQPNI	Logical	false	
NOCURRENT	Logical	false	
QSCV	Logical	false	
QFACTOR	Real	1.0	

Use the parameters in Table 14-5 to control solution convergence behavior.

The solution comprises of OUTER iterations (which end when self-consistency occurs) and within these are solutions of groups of equations or single equations. The BQP equation is implemented so that both LHS and RHS convergence are required. You cannot override this default behavior. Tests carried out have shown it gives the best performance. ITLIMIT controls how many iterations can occur for the BQP equation in **SOLVE** INIT. This is 10 * ITLIMIT. For CARRIERS=0 solutions, the maximum number of iterations for the individual equation solvers is GUMITS. The maximum number of OUTER iterations is ITLIMIT. NBLOCKIT controls the number of OUTER cycles in the drift-diffusion and energy-balance cases. the maximum number of drift-diffusion solution and carrier-energy solution iterations is ITLIMIT, and the maximum number of BQP solutions is GUMITS. The convergence may become unsteady when the carrier density is very high and Fermi statistics is used. In this case, you may decrease BQP.NMIX or BQP.PMIX parameters or both to a value between 0 and 1. This will cause relaxation of Fermi factors between two consecutive iterations and result in steadier convergence.

Note: The Bohm Quantum potential is stored in Atlas as an energy and so the convergence criteria are actually of order KT less than those for scaled electrostatic potential. In Atlas 3D, the BQP equation had an extra scaling factor applied and so the RHS norms are larger than in 2D. LHS norms, however, are the same.

The Bohm Quantum Potential is automatically calculated in insulators. If you specify the SEMICONDUCTOR parameter in the **MATERIAL** statement for the insulator, the drift-diffusion equations will be solved in the insulator and will include the BQP corrections. It is also recommended to set the densities of states of the insulator to be the same as the semiconductor to which they interface. For example for Silicon/SiO2 system, set

```
MATERIAL REGION=N NC300=2.8E19 NV300=1.04E19
```

where N is region number of SiO2.

You can also set the BQP parameters on a material by material basis, which may be necessary for heterojunction based devices. The BQP model is applied globally and cannot be set on a region by region basis.

Table 14-7 shows the material parameters that have a direct impact on the BQP equation.

Table 14-7 MATERIAL Statement

Parameter	Type	Default
SEMICONDUCTOR	Logical	false
EG300	Real	
AFFINITY	Real	
NC300	Real	
NV300	Real	
BQP.NALPHA	Real	0.5
BQP.NGAMMA	Real	1.2
BQP.PALPHA	Real	0.5
BQP.PGAMMA	Real	1.0

There is one more pertinent parameter, BQP.NEUMANN, which can be set on the **MODELS** statement.

BQP.NEUMANN sets the normal gradient of classical+quantum potential explicitly to zero on non-contact boundaries. If cleared using **MODELS** ^BQP.NEUMANN, the Neumann conditions will then be applied implicitly. Clearing this flag may give a slight speed increase and slight differences in solution.

Table 14-8 OUTPUT Statement Parameters

Parameter	Type	Default
P.QUANTUM	Logical	False

Setting the P.Quantum parameter in the **OUTPUT** statement will cause the Bohm Quantum Potential to output to a structure file. The Electron Quantum Potential and Hole Quantum Potential will both output.

14.5 Quantum Correction Models

In deep submicron MOS devices, quantum effects in the channel can have significant effects on the device characteristics. These effects are directly due to the increased doping levels and thinner gate oxide. In the channel of such devices, the potential well formed during inversion where the effects of quantum confinement must be considered. The direct effect of such quantum confinement is that the peak of the carrier concentration is shifted away from the interface and the thinner gate oxide can cause a marked difference in gate capacitance. Consideration of the effects may be essential for accurate prediction of the device turn-on voltage.

To fully treat quantum effects, solving the Schrodinger's equation, isn't always desirable or necessary.

14.5.1 Hansch's Model

The quantum mechanical correction given by Hansch [114] is suitable for accurate simulation of the effects of quantum mechanical confinement near the gate oxide interface in MOSFETs. To enable the model, specify the HANSCHQM parameter of the **MODELS** statement. This correction model is a modification of the density of states as a function of depth below the Si/SiO_2 interface, which is given by Equation 14-22.

$$N_C^* = N_C\left[1 - \exp-\left(\frac{z}{\text{LAMBDA}}\right)^2\right] \qquad 14\text{-}22$$

Here, N_C is the standard density of states, z is the depth below the interface. LAMBDA is a user-definable parameter in the **MODELS** statement.

Table 14-9 User-definable parameter for Equation 14-22

Statement	Parameter	Default	Units
MODELS	LAMBDA	1.0×10^{-3}	µm

14.5.2 Van Dort's Model

In the Van Dort's model [323], the effects of the quantum confinement are modeled by broadening the bandgap near the surface, which makes a function of a perpendicular electric field and distance from the surface. In Van Dort's model, the change in bandgap is given by the expression in Equation 14-23.

$$\Delta Eg = (\text{B.DORT})\ \beta\left(\frac{\varepsilon_{si}}{4qk_BT_L}\right)^{1/3}(E_\perp - E_c)^{\alpha}g(y) \qquad 14\text{-}23$$

Here, B.DORT is a user-definable parameter in the MODEL statement, $E_\perp$ is the perpendicular electric field, and g(y) is a function to restrict the application of the model to the channel region, given by Equation 14-24.

$$g(y) = \frac{2e^{-(Y/(\text{D.DORT}))^2}}{1 + e^{-2(Y/(\text{D.DORT}))^2}} \qquad 14\text{-}24$$

The default parameter values for β, E_c, and α are dependent on technology as described in Table 14-10.

Table 14-10 Technology Dependent Parameter Values of General Van Dort Model

Technology	E_c	α	β
electron inversion	NINV.EC	NINV.EXP	NINV.BETA
hole inversion	PINV.EC	PINV.EXP	PINV.BETA
electron accumulation	NACC.EC	NACC.EXP	NACC.BETA/(1+Nd/NACC.N)
hole accumulation	PACC.EC	PACC.EXP	PACC.BETA/(1+Na/PACC.N)

The default values for the various parameter of the **MODELS** statement are given in Table 14-11.

Table 14-11 Default Parameter Values for the Van Dort Model

Parameter	Type	Default	Units	Reference
NINV.EC	Real	0	V/cm	[323]
NINV.BETA	Real	5.92E-8	(1)	
NINV.EXP	Real	0.666666	(1)	
PINV.EC	Real	100000	V/cm	[116]
PINV.BETA	Real	6.10E-8	(2)	
PINV.EXP	Real	0.666666	(2)	
NACC.EC	Real	0	V/cm	[117]
NACC.BETA	Real	8.45E-8	(3)	
NACC.EXP	Real	0.666666	(3)	
NACC.N	Real	2.0E19	cm^{-3}	[117]
PACC.EC	Real	0	V/cm	[117]
PACC.BETA	Real	4.97E-8	(3)	
PACC.EXP	Real	0.666666	(3)	
PACC.N	Real	3.6E19	cm^{-3}	[117]

The new models are enabled for the various condicitons are enabled by the flags NINV.DORT, NACC.DORT, PINV.DORT, and PACC.DORT. The flags N.DORT is aliased to NINV.DORT and P.DORT is aliased to PINV.DORT.

14.6 Parabolic Quantum Well Model

The parabolic quantum well model is used to predict bound state energies in a region for subsequent use in modelling optoelectronic gain, radiative recombination and absorption. Figure 14-1 shows the simulation flow for such modeling.

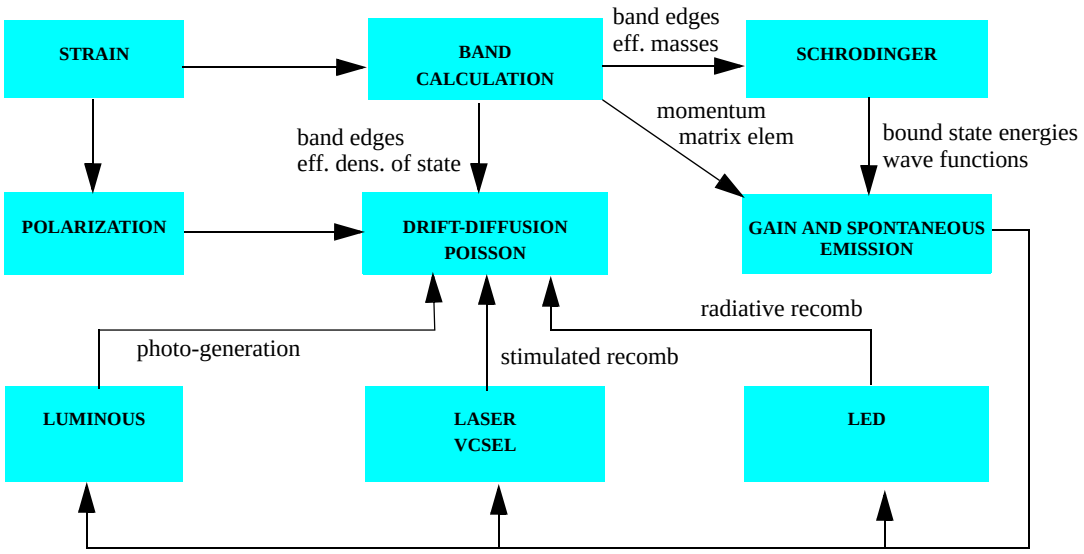


Figure 14-1: Simulation Flow For Physically Based Optoelectronic Models

The solution for the bound state energies is done by solving Schrodinger equation (see Equation 14-1) along discrete slices in the quantization direction. Effective masses and band-edge parameters are taken from multi-band k.p-based models set by ZB.ONE, ZB.TWO, ZB.THREE, and WZ.THREE parameters on the **MODELS** statement, which stand for zincblende or wurtzite type of material and the number of valence bands used in the calculation.

This model treats the electronic states within the effective mass approximation. Away from the zone center, the valence bands calculated within this model can quickly deviate from the exact band structure depending on the strength of inter-mixing of the hole states. The optical response calculated within this model also becomes less reliable for large spatial changes in effective mass parameters between the well and the barrier region. Section 14.7 “Multiband kp Models” describes the multiband **k.p** models that take into account this inter-mixing of states to provide a much more accurate band structure and optical response over a wider region of the the Brillouin zone. However since the parabolic model implements the summation over states in momentum space analytically, it is much faster than the more general **k.p** model. When determining the applicability of the parabolic model, you should carefully consider the range of energies over which the optical response is required, the strength of valence and conduction band coupling, and the differences in the effective mass between the well and the barrier regions.

You can enable the quantum well model by specifying QWELL in the **REGION** or **MODELS** statement. The orientation and dimensionality of Schrodinger solver is set by SP.GEOMETRY parameter on the **MODELS** statement with a default of 1DY. Alternatively, you may solve 1DX and 1DZ cases. Regions with QWELL parameter are treated as independent quantum wells, unless they are adjacent to each other and have the same orientation, in which case they are

merged into one quantum well region. This means that a single quantum well region can include different materials. Merging of adjacent regions into a single quantum well can be overridden by assigning different numbers to different wells, using QWNUM parameter on the **REGION** or the **MODELS** statement (e.g., QWNUM=1).

Whenever a quantum well region is specified, the solution domain for Schrodinger equation will actually be expanded into neighboring regions to account for wavefunction penetration into the barriers. The maximum penetration length is set by WELL.MARGIN parameter on the **MODELS** statement with a default value of 0.01 micron. It is perfectly admissible to have wavefunctions of two different wells to overlap.

By default, Atlas mesh is used for discretization. If the Atlas mesh too coarse, you can specify WELL.NX or WELL.NY or WELL.NZ or all three in the **REGION** or the **MODELS** statement to use an auxiliary quantum well mesh. If an auxiliary mesh is used, the sampling is uniform and bound state energy values are interpolated back onto the device mesh for the optical modelling. Generally, you should select WELL.NX, WELL.NY, and WELL.NZ to adequately resolve the geometry but there is a trade-off between computational accuracy and speed.

When Schrodinger equation is discretized, the conduction and valence bands are modified in the barriers to avoid bound states at the edges. The schematics below shows the original conduction band $E_c(x)$ and the modified band $E'_c(x)$ as seen by confined carriers.

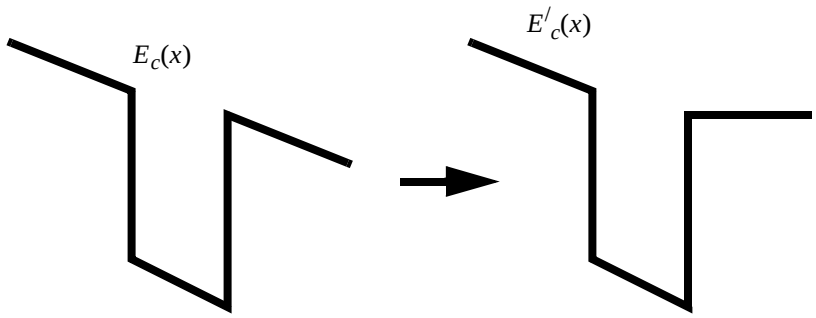


Figure 14-2: Modification of conduction band $E_c(x)$, as seen by confined carriers

The WELL.CNBS and WELL.VNBS parameters of the **REGION** or the **MODELS** statement specify the maximum number of bound state energies to resolve for each conduction and valence band with default values of 1. An output structure file of the QWELL model contains bound state energies and wavefunctions for each well, band, and sub-band. The actual number of bound states found may be less than or equal to the value specified by WELL.CNBS and WELL.VNBS. Additionally, the output contains band edge energies for each band and bound carrier densities.

You can also specify regions for quantum well simulation in the SUPERLATTICE (**DBR**) statement. To enable the model, specify the QWELL1 and QWELL2 parameters. These enable quantum modeling for the first and second superlattice cycles respectively. The sampling of the quantum wells are controlled by the WELL1.NX, WELL2.NX, WELL1.NY and WELL2.NY parameters. The numbers of bound states are controlled by the WELL1.CNBS, WELL2.CNBS, WELL1.VNBS and WELL2.VNBS parameters.

14.6.1 Superlattice Model

In some applications, you may want to model a long set of independent quantum wells. In other cases, you may be interseted in collective bound and traveling states of a set of strongly coupled quantum wells.

In the first case, you can specify regions for quantum well simulation in the SUPERLATTICE (**DBR**) statement. To enable the model, specify the QWELL1 and QWELL2 parameters. These enable quantum modeling for the first and second superlattice cycles respectively. The sampling of the quantum wells are controlled by the WELL1.NX, WELL2.NX, WELL1.NY, and WELL2.NY parameters. The numbers of bound states are controlled by the WELL1.CNBS, WELL2.CNBS, WELL1.VNBS, and WELL2.VNBS parameters. The rest of the model is identical to QWELL model.

In the second case, when a set of quantum wells is strongly coupled, you may set SLATT parameter on the **REGION** statement. All adjacent regions with the SLATT attribute will be lumped into the same superlattice domain. To assign adjacent regions to different superlattice domains, use the SL.NUMBER parameter. You also need to set direction of Schrodinger solver using SL.GEOM = 1DX or 1DY (default). The SLATT model uses the same band structure as the QWELL model. It will also reuse WELL.NX and WELL.NY parameters, if specified. Calculation of collective states of a superlattice domain is done using Quantum Transmitting Boundary Method (QTBM), which is a non-linear Schrodinger equation with open-boundary contacts. Calculated eigen states are sorted into bound and traveling based on the location of the wavefunction and eigen energy with respect to band edge. SLATT model can coexist with QWELL model or used independently.

14.6.2 Radiative Recombination in Quantum Wells

Knowledge of bound states allows Atlas to calculate spontaneous recombination and gain due to interband transitions. To calculate spontaneous recombination and include it into the continuity transport equations, use SPONTANEOUS parameter on the **MODELS** statement. If the QWELL region is also assigned LED parameter, then luminous power from this region will be computed and stored in the IV log file.

Spontaneous emission rate per unit energy and gain are given as functions of photon energy $h\omega$ and polarization υ=*TE, TM* by the following expressions:

$$r_{spon}^{\upsilon}(h\omega) = \left(\frac{n_r e^2 \omega}{\pi h c^3 \varepsilon_0 m_0^2}\right) \sum_{n,m} \int \rho_r^{2D}(E) \frac{\Gamma/(2\pi)}{(E-h\omega)^2 + (\Gamma/2)^2} f_c^n \left(1 - f_v^m\right) \left|I_{m,n}\right|^2 \left|M_b^{\upsilon}\right|^2 dE \qquad 14\text{-}25$$

$$g^{\upsilon}(h\omega) = \left(\frac{\pi e^2}{n_r c \omega m_0^2 \varepsilon_0}\right) \sum_{n,m} \int \rho_r^{2D} \frac{\Gamma/(2\pi)}{(E-h\omega)^2 + (\Gamma/2)^2} \left(f_c^n - f_v^m\right) \left|I_{m,n}\right|^2 \left|M_b^{\upsilon}\right|^2 dE \qquad 14\text{-}26$$

where

- $\rho_r^{2D} = m_r/\pi h^2 L_z$ is the two-dimensional density of states.
- m_r is a reduced electron-hole effective mass.
- n_r is a material refractive index.
- $I_{m,n} = \langle \phi_m | \phi_n \rangle$ is an overlap integral of electron and hole wavefunctions.

- M_b^{υ} is a polarization- dependent bulk momentum matrix element.
- Γ is the line-width due to Lorentzian broadening.
- The sum is taken over all transitions between electron sub-bands with energies E_n and hole sub-bands with energies E_m of all valence bands.

Fermi-Dirac distribution functions have the following form ($E_{mn} \equiv E_n - E_m$):

$$f_c^n(E) = \frac{1}{1 + exp\{[E_n + (m_r / m_e^*)(E - E_{mn}) - E_{Fc}]/kT\}} \quad 14\text{-}27$$

$$f_v^m(E) = \frac{1}{1 + exp\{[E_m - (m_r / m_h^*)(E - E_{mn}) - E_{Fv}]/kT\}} \quad 14\text{-}28$$

Here, we assume that bound carriers are in the detailed balance with bulk carriers. Therefore, we use local Fermi levels of bulk carriers to compute the recombination rate at each node. See Section 14.6.4 “Capture-Escape Model” for more realistic treatment of bound carrier dynamics.

In order to compute total radiative recombination rate, we perform averaging over polarizations and integration over the whole emission spectrum:

$$R_{sp}(x, y, z) = \int_0^{\infty} \frac{\left(2 r_{spon}^{TE}(h\omega) + r_{spon}^{TM}(h\omega)\right)}{3} d(h\omega) \quad 14\text{-}29$$

The integration over the spectrum is done using Gauss-Laguere quadrature rule. The number of energy points is set by WELL.NERSP (default is 32) parameter on the **METHOD** statement. If the parameter is set to zero (not recommended), a trapezoidal integration will be used with a spacing set by WELL.DENERGY parameter on the **MODELS** or **MATERIAL** statement.

Spectrum of spontaneous recombination and gain can be stored in a separate file using SPECTRUM='filename' parameter on the **SAVE** statement. The spectra are averaged over all slices of the quantum well.

14.6.3 Intersubband Radiative Transitions

In order to compute intersubband gain and spontaneous emission, specify INTERSUB.SPONT on the **MODELS** statement. In order to save spectrum of intersubband transitions, set SPEC.INTERSUB='filename.log' parameter on the **SAVE** statement.

Intersubband spontaneous emission rate per unit energy and gain are given as functions of photon energy $h\omega$ by the following expressions (TE light does not couple to intesubband transitions):

$$r_{spon}^{TM}(h\omega) = \left(\frac{n_r e^2 \omega^3}{\pi h c^3 \varepsilon_0}\right) \sum_{n,m} |d_{m,n}|^2 \rho_r^{2D} \frac{\Gamma/(2\pi)}{(E_{mn} - h\omega)^2 + (\Gamma/2)^2} \int f^n (1 - f^m) dE_t \quad 14\text{-}30$$

$$g^{TM}(h\omega) = \left(\frac{\omega \pi e^2}{n_r c \varepsilon_0}\right) \sum_{n,m} |d_{m,n}|^2 \rho_r^{2D} \frac{\Gamma/(2\pi)}{(E_{mn} - h\omega)^2 + (\Gamma/2)^2} \int (f^n - f^m) dE_t \quad 14\text{-}31$$

where

- $d_{m,n} = \langle \phi_m | x | \phi_n \rangle$ is a dipole moment between initial and final states.
- Γ is the line-width due to Lorentzian broadening.
- The sum is taken over all transitions between sub-bands with energies E_n and E_m of the same carrier type.
- Integral is taken over transverse energy E_t

Fermi-Dirac distribution functions have the following form:

$$f^n(E) = \frac{1}{1 + exp\{[E_n + E_t - E_F]/kT\}} \quad \text{14-32}$$

14.6.4 Capture-Escape Model

The model is launched by adding WELL.CAPT parameter to the **MODELS** statement. Capture-escape model treats the dynamics of bound carriers by breaking carriers into bulk and bound subsystems, coupled to each other by a capture-escape rate. The capture-escape rate is added to the continuity equation to remove bulk carriers. The same rate serves as a generation mechanism for bound carriers and the following additional equations are solved for each slice of each quantum well to find quantum well Fermi energies:

$$\frac{\partial \tilde{n}^{2D}}{\partial t} = \tilde{R}_{capt,n} - \tilde{R}_{recomb} - \frac{1}{e}\nabla \tilde{J}_n^{2D} \quad \text{14-33}$$

$$\frac{\partial \tilde{p}^{2D}}{\partial t} = \tilde{R}_{capt,p} - \tilde{R}_{recomb} - \frac{1}{e}\nabla \tilde{J}_p^{2D} \quad \text{14-34}$$

where ~ means that the integration of the variable is performed over the box, belonging to the slice of the quantum well. The Fermi energies thus found are used in the expressions for radiative transitions and all other important phenomena, such as in-plane transport, inter-well tunneling and inter-well spontaneous emission.

The capture-escape rate, which is added to the continuity equation for bulk carriers is given by

$$R_{capt} = \frac{n^{3D}}{\tau}\sum_{\nu}\left(1 - f\left(E_{F,\nu}^{2D}, E_\nu\right)\right)\left(1 - exp\left(\frac{E_{F,\nu}^{2D} - E_F^{3D}}{kT}\right)\right) \quad \text{14-35}$$

where n^{3D} is a bulk carrier density, τ is a capture time, $E_{F,\nu}^{2D}$ is a fermi level of sub-band ν, E_F^{3D} is a local bulk fermi level, and E_ν is the sub-band energy. The summation is performed over all sub-bands of the same carrier type. In the first bracket under the summation, f is a Fermi-Dirac function, which reduces capture rate as the sub-band is getting filled. Carrier capture is a multi-phonon process, whose detailed physics is not the objective of this model. Rather, we use capture time τ as a parameter set by WELL.TAUN (for electrons) and WELL.TAUP (for holes) parameters on the **MATERIAL** statement.

As seen from the expression above, capture rate may become negative when Fermi level for bound carriers exeeds that for bulk carriers. In this case, it will act as an escape mechanism of 2D carriers from the well into the bulk.

An alternative equation for the Capture-Escape rate is available. It is

$$R_{\text{capt}} = \begin{cases} \frac{n^{3d}}{\tau}\Sigma_\nu(1-f(E^{2d}_{F,\nu},E_\nu))\left(F_{1/2}(\frac{E^{3d}_F-E^{2d}_{F,\nu}}{KT})-F_{1/2}(\frac{E^{2d}_{F,\nu}-E^{3d}_F}{KT})\right) & \text{if } E^{3d}_F \geq E^{2d}_{F,\nu} \\ \frac{n^{3d}}{\tau}\Sigma_\nu f(E^{2d}_{F,\nu},E_\nu)\left(F_{1/2}(\frac{E^{3d}_F-E^{2d}_{F,\nu}}{KT})-F_{1/2}(\frac{E^{2d}_{F,\nu}-E^{3d}_F}{KT})\right) & \text{if } E^{3d}_F < E^{2d}_{F,\nu} \end{cases} \quad \text{14-36}$$

where the top term corresponds to carrier capture by the well, and the bottom term to carrier escape from the well. The function $F_{1/2}$ is the Fermi-Dirac function of order one half. The energy is electron energy for electrons and hole energy for holes. You select this alternative R_{capt} model by specifying CAPT.ALT on the METHOD statement.

For LED and laser applications, it is usually sufficient to have one Fermi level for all sub-bands in the same slice of the same carrier type. However, if an individual equation for each sub-band of each slice is required, specify WELL.SEPARATE on the **MODELS** statement.

In both equations for R_{capt} there is no nett capture rate if the quasi-Fermi level for the bound carrier system coincides with the quasi-Fermi level for the bulk carriers.

If SPONTANEOUS parameter is specified on the **MODELS** statement, a second term will be added, which will depopulate 2D sub-bands. When capture-escape model is active, spontaneous recombination due to quantum wells is no longer a part of bulk continuity equations.

In order to take into account spontaneous emission between bound states in the neighboring wells, specify the WELL.WWSPONT parameter on the **MODELS** statement. This mechanism is a must for type-II quantum wells and may be important when the wave function overlap of neighboring wells is large.

In addition to the radiative recombination rate produced by the SPONTANEOUS flag, there are three non-radiative mechanisms for recombination-generation of the quantum confined carrier densities. The first method is the Shockley-Read-Hall model which you enable by specifying the CAPT.SRH flag on the MODELS statement. The rate is given by

$$R_{srh} = \frac{N_{qc}P_{qc} - N^{eq}_{qc}P^{eq}_{qc}}{\tau_p(N_{qc}+NI)+\tau_n(P_{qc}+NI)} \quad \text{14-37}$$

where N_{qc} and Pqc are the quantum confined carrier densities (See Eqn 14-13) and N^{eq}_{qc} and P^{eq}_{qc} are the values of these densities in thermodynamic equilibrium. The quantity $NI = \sqrt{N^{eq}_{qc}P^{eq}_{qc}}$. The recombination lifetime τ_n is specified by the CAPT.SRH.TAUN parameter and the lifetime τ_p by the CAPT.SRH.TAUP parameter. At the interface between a quantum well and a barrier, the recombination rate can be enhanced by specifying a surface recombination velocity. You do this with the parameters S.WELL.N and S.WELL.P on the INTERFACE statement. These modify the recombination lifetimes τ_n and τ_p as follows

$$\frac{1}{\tau_n} = \frac{1}{\tau_n} + \frac{S.W\ ELL.N\ D_i}{A_i} \quad \text{14-38}$$

$$\frac{1}{\tau_p} = \frac{1}{\tau_p} + \frac{S.W\ ELL.P\ D_i}{A_i} \quad \text{14-39}$$

where D_i is the interface length associated with a mesh node and A_i is the semiconductor area associated with the same node. For 3D devices or 2D devices with a CYLINDRICAL mesh, then D_i is an interface area and A_i is an associated semiconductor volume.

The second non-radiative recombination rate is Auger recombination and you enable it by specifying the CAPT.AUGER flag on the MODELS statement. The rate is given by

$$R_{auger} = (CAPT.AUGERN\ N + CAPT.AUGERP\ P)(N_{qc}P_{qc} - N_{qc}^{eq}P_{qc}^{eq}) \quad \text{14-40}$$

The recombination rates as a function of position can be output to structure file using the SAVE statement and can be viewed in Tonyplot as ‘Quantum Well SRH recombination rate’ and ‘Quantum well Auger recombination rate’ respectively. Specifying CAPT.SRH and CAPT.AUGER on the PROBE statement causes the recombination rates to be saved to log file. Information about the parameters for these models is given in Table 14-12.

The third non-radiative recombination rate is recombination via charged traps at interfaces between the quantum well and insulator materials. The recombination rates are calculated from the equations 3-91 and 3-92, but with the quantum confined carrier densities being used instead of the bulk values. To enable this model you specify CAPT.TRAP on the MODELS statement, and also QWELL on any INTTRAP statements that you wish to include into equations 14-33 and 14-34. Any INTTRAP statements without the QWELL flag specified will be included in the bulk drift-diffusion equations.

The third term, switched on by `WELL.INPLANE` parameter on the **`MODELS`** statement, models the in-plane transport of 2D carriers in the well, which becomes significant when there is a voltage drop along the well. Electron and hole current for 2D carriers are given by $J_n^{2D} = -e\sum_\nu \mu_n n_\nu^{2D} \nabla E_{Fn,\nu}^{2D}$ and $J_p^{2D} = -e\sum_\nu \mu_p p_\nu^{2D} \nabla E_{Fp,\nu}^{2D}$.

The default values for the mobilities μ_n and μ_p, are those of the drift-diffusion equations for the bulk carriers. Alternatively you can set a constant mobility for the in-plane carrier transport using the CAPT.MUN0 parameter for μ_n and CAPT.MUP0 parameter for μ_p. Additionally, the C-Interpreter parameters F.CAPT.MUN and F.CAPT.MUP allow you to specify the name of a file which contains your function to give an in-plane mobility which depends on position, material composition, doping level, lattice temperature and field. You specify any of these parameters on the MATERIAL statement.

Table 14-12 User Specified Parameters for Capture-Escape Non-adiative Recombination Models for Quantum Confined Carriers

Statement	Parameter	Type	Default	Units
INTERFACE	S.WELL.N	Real	0.0	*cm/s*
INTERFACE	S.WELL.P	Real	0.0	*cm/s*
INTTRAP	QWELL	Logical	False	
MATERIAL	CAPT.SRH.TAUN	Real	10^{-3}	*s*
MATERIAL	CAPT.SRH.TAUP	Real	10^{-3}	*s*
MATERIAL	CAPT.AUGERN	Real	10^{-30}	cm^6s^{-1}
MATERIAL	CAPT.AUGERP	Real	10^{-30}	cm^6s^{-1}
MATERIAL	CAPT.MUN0	Real	0	cm^2/Vs
MATERIAL	CAPT.MUP0	Real	0	cm^2/Vs
MATERIAL	F.CAPT.MUN	Char		
MATERIAL	F.CAPT.MUP	Char		
MATERIAL	WELL.TAUN	Real	10^{-12}	
MATERIAL	WELL.TAUP	Real	10^{-12}	
METHOD	CAPT.ALT	Logical	False	
MODELS	CAPT.AUGER	Logical	False	
MODELS	CAPT.SRH	Logical	False	
MODELS	CAPT.TRAP	Logical	False	
MODELS	WELL.SEPARATE	Logical	False	
MODELS	WELL.WWSPONT	Logical	False	
MODELS	WELL.INPLANE	Logical	False	
MODELS	SPONTANEOUS	Logical	False	
PROBE	CAPT.SRH	Logical	False	
PROBE	CAPT.AUGER	Logical	False	
PROBE	CAPT.N.CONC	Logical	False	
PROBE	CAPT.P.CONC	Logical	False	
PROBE	SPONT.EMISS	Logical	False	

Parameters on the **METHOD** statement regulate the convergence of the model. WELL.ERROR (default 1e-5 V) is the largerst relative error of bound carrier density. WELL.ITERMAX (default is 20) sets the maximum number of Newton iterations, after which the 2D carrier densities are assumed to be converged. WELL.PROJ (default is true) sets the projection method as initial guess. At each Newton iteration, the solution of the coupled system of Poisson, bulk continuity and quantum well rate equations is done using the Schur Complement method by default. A direct method, although less effective, can still be called by switching off the SCHUR parameter on the **METHOD** statement.

The output structure file will contain Fermi levels for each sub-band, capture rates and in-plane current densities.

14.6.5 Charge Self-Consistency

In order to take into account confined carriers in the Poisson equation, set WELL.SELFCON parameter on the **MODELS** statement. This parameter is automatically enabled when the Capture-Escape model is enabled. The parameter will trigger the following changes to the normal computation. At first, the conduction and valence bands in the well are modified before plugging into continuity equation, so that the well is effectively removed for bulk carriers. The schematics below shows the original conduction band $E_c(x)$ and the modified band $E'_c(x)$, as seen by bulk carriers. The value of the conduction band in the well is given by the energy of the lowest of the left and right barriers. This reduces bulk carrier density in the well by not letting the bulk carriers to fill it. Then, the 2D carrier density is computed with Equation 14-13 added to the bulk carrier density and used in Poisson equation $n_{tot} = n^{3D} + n^{2D}$. The modified conduction band and valence band energies are automatically output to any Standard Structure file if WELL.SELFCON is enabled. These can be viewed in Tonyplot as ‘Modified Conduction Band Energy’ and ‘Modified Valence Band Energy’ .

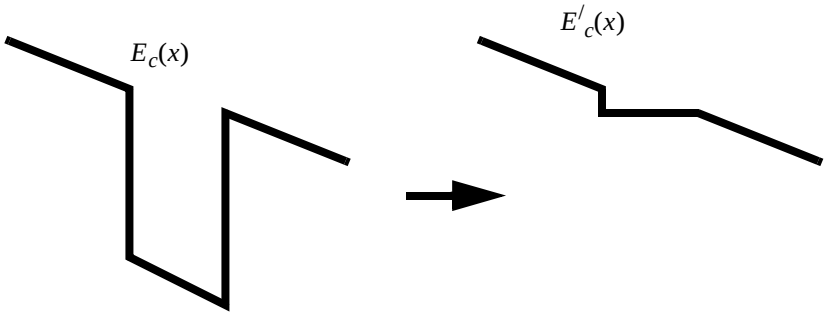


Figure 14-3: Modification of conduction band $E_c(x)$, as seen by bulk carriers

14.7 Multiband kp Models

Optical response of semiconductors often involves states far from the fundamental gap, where a band structure calculation becomes necessary to obtain the correct energy dispersion and electronic states. The $\boldsymbol{k}\cdot\boldsymbol{p}$ method is a very economical way to systematically construct the band structure near a particular point in the Brillouin zone. The standard $\boldsymbol{k}\cdot\boldsymbol{p}$ models, which are implemented in Atlas, expand the band structure around the center of the Brillouin zone, $\boldsymbol{k}$=0, which is called the Γ point.

As a consequence of the periodicity of the crystal potential, the general form of the single electron eigenstates of the crystal is gurranteed by the Bloch theorem to be

$$\psi_{n\boldsymbol{k}}(\boldsymbol{r}) = e^{i\boldsymbol{k}\cdot\boldsymbol{r}}u_{n\boldsymbol{k}}(\boldsymbol{r}) \qquad 14\text{-}41$$

where the Bloch envelope functions $u_{n\boldsymbol{k}}(\boldsymbol{r})$ have the same periodicity as the crystal, i.e., $u_{n\boldsymbol{k}}(\boldsymbol{r}) = u_{n\boldsymbol{k}}(\boldsymbol{r}+\boldsymbol{R})$, where $\boldsymbol{R}$ is a lattice vector. Since the complete set of functions $u_{n\boldsymbol{k}}(\boldsymbol{r})$ at each $\boldsymbol{k}$ span the entire Hilbert space of electronic states, the functions at the Γ point form a perfectly valid basis set to expand $u_{n\boldsymbol{k}}(\boldsymbol{r})$ at any other $\boldsymbol{k}$ in Equation 14-41. Thus, another form for Equation 14-41 for bulk materials is

$$\psi_{j\boldsymbol{k}}(\boldsymbol{r}) = e^{i\boldsymbol{k}\cdot\boldsymbol{r}}\sum_{n'}u_{n'}(\boldsymbol{r})c_{n'j}(\boldsymbol{k}) \qquad 14\text{-}42$$

where $u_{n'}(\boldsymbol{r})$ are the Bloch envelope functions at the Γ point, and $c_{n'j}(\boldsymbol{k})$ are the coefficients of expansion for a state in band j at point $\boldsymbol{k}$. Since the $\boldsymbol{k}\cdot\boldsymbol{p}$ theory does not make explicit use of the Bloch envelope functions, each eigenstate $\psi_{j\boldsymbol{k}}$ can be specified by the vectors $c_j(\boldsymbol{k})$ with components $c_{n'j}(\boldsymbol{k})$.

The $\boldsymbol{k}\cdot\boldsymbol{p}$ theory assumes that the states $u_{n'}(\boldsymbol{r})$ are known and yields the coefficients $c_{n'j}(\boldsymbol{k})$ by substituting Equation 14-42 into the single particle time-independent Schrödinger equation. The substitution yields a matrix equation,

$$\sum_{n'}H_{nn'}(\boldsymbol{k})c_{n'j}(\boldsymbol{k})(\boldsymbol{r}) = E_{n\boldsymbol{k}}c_{nj}(\boldsymbol{k}) \qquad 14\text{-}43$$

$$H_{nn'}(\boldsymbol{k}) = \left\{E_n + \frac{\hbar^2k^2}{2m_0}\right\}\delta_{nn'} + \frac{\hbar^2k^2}{2m_0} + \frac{\hbar}{m_0}\boldsymbol{k}\cdot\boldsymbol{p}_{nn'} \qquad 14\text{-}44$$

in which E_n are the band edge energies at the Γ point and $\boldsymbol{p}_{nn'}$ is the momentum operator represented as a matrix in the Γ point basis. For brevity, we have lumped both the linear momentum and the $\boldsymbol{k}$ dependent spin orbit coupling contributions into this matrix. The second equation defines the matrix representation of the single particle Hamiltonian in the Bloch basis. The Hamiltonian is diagonal at the Γ point, while the off-diagonal contributions arise from the $\boldsymbol{k}\cdot\boldsymbol{p}$ term.

Equation 14-43 is exact, but includes infinite number of states. At this stage, an approximation is introduced using the Löwdin perturbation method in which a small number of bands closest to the band gap (set A) are decoupled from the remaining bands (set B) by constructing a unitary transformation of the Hamiltonian matrix. Since this cannot be carried

out exactly analytically, the method becomes essentially perturbative by expanding this transformation in orders of the wavevector $\boldsymbol{k}$. This expansion restricts the area of the Brillouin zone that the $\boldsymbol{k}\cdot\boldsymbol{p}$ theory can cover in practice. This area increases as the number of bands in set A is increased.

When using the Γ (or other high symmetry points of the Brillouin zone) as the point of expansion, the choice of basis states and the functional form of the terms in perturbation expansion can be based on the point group symmetry of the crystal. The coefficients of the perturbation expansion form a set of parameters characterizing a particular $\boldsymbol{k}\cdot\boldsymbol{p}$ model. A particularly useful aspect of the $\boldsymbol{k}\cdot\boldsymbol{p}$ theory is that many of these parameters have a direct relationship to the resulting band structure and experimentally measurable quantities. Semiconductor theory provides a standard prescription of $\boldsymbol{k}\cdot\boldsymbol{p}$ Hamiltonian matrices with each term represented by a polynomial in terms of the components of the wavevector k [335, 353]. The polynomial coefficients are tabulated as material-dependent parameters for most of the widely used semiconductor materials. The resulting models are the most well-developed for the diamond, zinc-blende and wurtzite crystal classes.

The models implemented in Atlas include terms up to the second order in $\boldsymbol{k}$, which is the most common scenario in calculations of optical response of direct gap semiconductors. Following the above discussion, the general form of the Hamiltonian implemented in Atlas is

$$H_{nm}(\boldsymbol{k}) = H_{nm}^{(0)} + \sum_{\beta = x,\,y,\,z} k_\beta \overline{Q}_{nm}^{\beta} + Q^{\beta} k_\beta + \sum_{\alpha\beta = x,\,y,\,z} k_\alpha D_{nm}^{\alpha\beta} k_\beta \qquad \text{14-45}$$

where subscripts, n and m, denote the basis states, each matrix element is specified by the parameters of the model, and k_α are the cartesian components of the wave vector. In bulk, both the k_α and the matrix elements are numbers so that the order they appear does not matter. The ordering of k_α and the matrix they multiply are an important part of applying the $\boldsymbol{k}\cdot\boldsymbol{p}$ theory to nanostructures.

Bulk $\boldsymbol{k}\cdot\boldsymbol{p}$ model is applied to planar heterostructures structures via the so-called envelope function approximation (EFA) [59]. In this approximation, each layer is treated via the bulk $\boldsymbol{k}\cdot\boldsymbol{p}$ model of its constituent material thus making all the matrices in [38] spatially varying. The electronic wavefunctions are modulated at a length scale much longer than the unit cell of the underlying crystal, by making the vectors $c_j(\boldsymbol{k})$ in [42] also depend on the coordinate along the growth direction. These spatially dependent vectors now define vector value envelope functions, which we will denote as $\psi_j(\boldsymbol{k})$ below.

A crucial assumption in the extension of the bulk model to nanostructures is that the spatially dependent envelope functions $\psi_j(\boldsymbol{k}, x)$ must be slowly varying on the scale of the unit cell of the underlying crystal. The scale at which the basis states $u_n(\boldsymbol{r})$ vary. These assumptions are generally satisfied for layers thicker than about 5-7 monolayers [356]. In this case, the equations for the $\psi_j(\boldsymbol{k}, x)$ follow from Equations 14-43 and 14-44 by making component of the $\boldsymbol{k}$ in the growth direction a differential operator. Thus with [001] as the growth direction, aligned with the X-axis, the envelope functions obey the equation

$$\left[\frac{d}{dx}D^{xx}(x)\frac{d}{dx} + \frac{d}{dx}\overline{Q}^{x}(\boldsymbol{k}_{\|}, x) + Q^{x}(\boldsymbol{k}_{\|}, x)\frac{d}{dx} + H^{(0)}(\boldsymbol{k}_{\|}, x) + V(x)\right]\psi_j(\boldsymbol{k}_{\|}, x) = E_j(\boldsymbol{k}_{\|})\psi_j(\boldsymbol{k}_{\|}, x) \qquad \text{14-46}$$

where j identifies the different solutions to this equation, each of which corresponds to a subband state at the two-dimensional wavector $\boldsymbol{k}_{\|}$ in the plane perpendicular to the growth axis. $V(x)$ is the confinement potential.

$$Q(\boldsymbol{k}_{\|}, x) = Q^{x}(x) + \sum_{\alpha = y, z} k_{\alpha} D^{\alpha x}(x) \quad \text{14-47}$$

$$\overline{Q}(\boldsymbol{k}_{\|}, x) = \overline{Q}^{x}(x) + \sum_{\alpha = y, z} D^{x\alpha}(x) k_{\alpha} \quad \text{14-48}$$

$$H^{(0)}(\boldsymbol{k}_{\|}, x) = \sum_{\alpha = y, z} [Q^{\alpha}(x) + \overline{Q}^{\alpha}(x)] k_{\alpha} + \sum_{\alpha, \beta = y, z} D^{\alpha\beta}(x) k_{\alpha} k_{\beta} \quad \text{14-49}$$

Since the band parameters now also depend on the coordinate along this direction, the EFA requires a careful treatment of the ordering of band parameters and the components of $\boldsymbol{k}_{\|}$ in the growth direction. The quantum well modeling in Atlas is based on the theoretical approach due to Burt [42, 43], and developed further by Foreman [90, 91], who derived the correct ordering implied by the Schrödinger equation itself. The Burt theory dictates how the $\boldsymbol{k} \cdot \boldsymbol{p}$ parameters are introduced into the matrices $H^{(j)}_{\alpha\beta}$ [91].

Modeling of quantum wells within the above general framework has two aspects:

- the bulk $\boldsymbol{k} \cdot \boldsymbol{p}$ models for each of the constituent materials, and
- the modeling and solving the Schrödinger with position dependent material parameters.

Atlas currently provides multiband spatially dependent $\boldsymbol{k} \cdot \boldsymbol{p}$ model only for layered nanostructures grown along the [001] crystallographic direction. The finite element method applied to the EFA [82] is used to solve the coupled multiband Schrödinger equation [90] numerically. Spurious solutions, with highly oscillatory or sharply decaying envelope functions, are known to arise in many band structure calculations that use EFA within the \kp framework. A major advantage of Burt's formulation described in the general discussion above, and its numerical implementation in Atlas is that spurious solutions do not arise when solving the eigenvalue equation [90].

14.7.1 Bulk Model

The 8-band Kane Model suffices for most optoelectronic applications involving processes near the Γ point. At the zone center, the top valence electron states of zinc-blende materials are *p*-like, and the lowest conduction state are *s*-like state. The valence states can thus be represented by six states: $|X\sigma\rangle$, $|Y\sigma\rangle$, $|Z\sigma\rangle$, where *X*,*Y*, and *Z* denote the spatial part of the state, which transform like *x*,*y*, and *z* coordinates, and σ is the spin. This defines a set of six states to which the two *S*-like conduction states with spin up and down, $|S\sigma\rangle$, are added to define an 8-band model.

In zinc-blende crystals, the eigenstates at the Γ point are the eigenstates of the total angular momentum *J*=*L*+*S*, where *L* is the spatial angular momentum and *S* the spin angular momentum. The six valence states consists of two heavy-hole HH states ($J = 3/2$ and $J_z = \pm 3/2$) and two light-hole LH states ($J = 3/2$, $J_z = \pm 1/2$) and two split-off (SO) states ($J = 1/2$, $J_z = \pm 1/2$) an energy Δ (`DELTA0` parameter in the `MATERIALS` statement)

below the valence edge. The six valence states are decoupled from the conduction state at the Γ point, while this coupling grows linearly in k and is proportional to the momentum matrix element P between the valence and conduction states (see Figure 14-4).

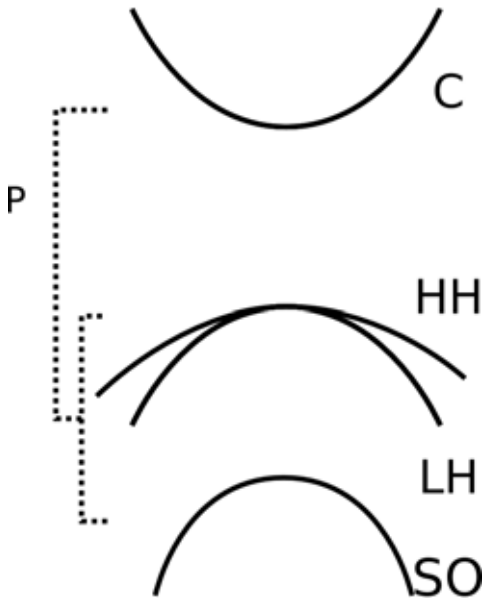


Figure 14-4: Schematic diagram of bands around the Γ point in the 8-Band Kane Model

The angular momentum basis diagonalizes the zinc-blende Hamiltonian at the Γ point. These states then become coupled at all other $\boldsymbol{k}$. In the absence of any further symmetry breaking, the states are doubly degenerate at each $\boldsymbol{k}$. Therefore, it is often convenient to deal with only one set of these states, reducing the problem by half. This is achieved by rotating the angular momentum basis to block diagonalize the Hamiltonian within the six valence states. The new basis inter-mix the doubly degenerate states (the two HH, LH, and SO between themselves) as follows:

$$|1\rangle = \alpha|\frac{3}{2},\frac{3}{2}\rangle - \alpha^*|\frac{3}{2},-\frac{3}{2}\rangle \quad \text{14-50}$$

$$|2\rangle = -\beta^*|\frac{3}{2},\frac{1}{2}\rangle + \beta|\frac{3}{2},-\frac{1}{2}\rangle$$

$$|3\rangle = \beta^*|\frac{1}{2},\frac{1}{2}\rangle + \beta|\frac{1}{2},-\frac{1}{2}\rangle$$

$$|4\rangle = \alpha|\frac{3}{2},\frac{3}{2}\rangle + \alpha^*|\frac{3}{2},-\frac{3}{2}\rangle$$

$$|5\rangle = \beta^*|\frac{3}{2},\frac{1}{2}\rangle + \beta|\frac{3}{2},-\frac{1}{2}\rangle$$

$$|6\rangle = \beta^*|\frac{1}{2},\frac{1}{2}\rangle - \beta|\frac{1}{2},-\frac{1}{2}\rangle$$

Here, the complex parameters α and β depend on $\boldsymbol{k}$ and are related to the phases of the HH and LH couplings in the angular momentum basis [51]. When instructed to use the block diagonalized basis, Atlas first transforms the Hamiltonian to the angular momentum basis to obtain α and β at each $\boldsymbol{k}$, and then performs the block diagonalization.

When using the block diagonlized basis (see Table 14-13), Atlas uses the “upper block” states (1, 2, and 3), while treating “lower block” states 4, 5, and 6 as degenerate with 1, 2, and 3 respectively. The degeneracy of these states (or equivalently, the angular momentum doublets) is lifted in asymmetric quantum well potentials. This results in the so-called “structure induced asymmetry” (SIA) and requires both members of the doublets to be included in the calculation. SIA introduces small spin splitting of bands and can often be ignored in relation to other approximations in the simulation. SIA is automatically ignored when using the block diagonalized bases, while models that use the full set of valence states include it automatically.

As mentioned above for the general $\boldsymbol{k}\cdot\boldsymbol{p}$ theory, the Kane model also depends on a set of band parameters, which specify the energy and effective mass of each state at the Γ point, and the $\boldsymbol{k}$-dependent couplings among these states. Models involving a subset of the eight states in the Kane model are obtained by using the degenerate perturbation theory, which amounts to modifications of band parameters the analytical formulas for which are widely available in literature [353, 187, 59].

Atlas implements the 8-band Kane model and all its standard lower-dimensional forms using these band parameter modification formulas. The flags for activating these models and the Γ point states included in each model are summarized in Table 14-13. In models that include the conduction band, the momentum matrix element, *P*, is set to zero when solving for the band structure via Equation 14-43. This produces a parabolic conduction band, which is a good approximation for wide gap materials. For narrow gap materials, the non-parabolicity of the conduction band due to $P \neq 0$ can be significant. The mixing of conduction and valence states away from Γ affects the valence states as well. In this case, the bands can be coupled by setting `KP.CVCOUPLED` on the **`MODELS`** statement. Setting *P*=0 affects only the band structure calculation, and not the optical response calculations that follow from it. In practice, leaving `KP.CVCOUPLED` unspecified provides a speedup in the quantum well eigenvalue calculations because it leads to matrices that are positive definite (see Section 14.7.2 “Modeling Quantum Wells”).

At least one flag listed in Table 14-13 must be set in order to activate the multiband $\boldsymbol{k}\cdot\boldsymbol{p}$ model. The flags apply to both the zinc-blende and wurtzite crystal classes. Unlike the zinc-blende band structure where the split-off band can often be ignored due to large split-off energy, all six valence states play an important role in the wurtzite band structure, even at the Γ point. Atlas will exit with an error if `KP.CV2` or `KP.V2` are specified for a material with wurtzite crystal class. The flags in Table 14-13 set the bulk model from which the Schrödinger equations for the quantum wells are constructed and solved as explained Section 14.7.2 “Modeling Quantum Wells”.

Finally, the $\boldsymbol{k}\cdot\boldsymbol{p}$ model is built in the HH and LH basis only if the split-off band is ignored. In this case, the doubly degenerate HH and LH states (for zero magnetic fields) can be decoupled analytically using a transformation due to Broido *et. al.* [39, 59], which reduces the Hamiltonian from a 4×4 to a 2×2 matrix. This simplification is also applied in the case of `KP.CV2` when `KP.CVCOUPLED` is false or unset. Thus when using `KP.CV2` model, the matrix sizes are halved when `KP.CVCOUPLED` is set to zero, which can result in a significant reduction in computation time when modeling quantum wells (see Section 14.7.2 “Modeling Quantum Wells”).

You can also set the KP.PARABOLIC flag, which solves the selected $\boldsymbol{k}\cdot\boldsymbol{p}$ model in the effective mass approximation. In this approximation, the solution is constructed at $\boldsymbol{k}$=**0**. The eigenfunctions at all other $\boldsymbol{k}$ are set equal to the eigenfunctions at $\boldsymbol{k}$=0. The eigenvalues are obtained by adding the diagonal of the average of the term of zeroth order in the wavevector component along the growth direction,

$$\frac{1}{L}\int_0^L H^{(0)}(\boldsymbol{k}_{\|}, x)dx \qquad 14\text{-}51$$

Specifying KP.PARABOLIC allows you to gauge the effects of interband mixing for their particular structure and device application, and allows you to determine whether they play an important role.

14.7.2 Modeling Quantum Wells

Setting up the geometry

The spatially dependent $\boldsymbol{k}\cdot\boldsymbol{p}$ model is automatically enabled for each region associated with a **MODELS** statement, where one of the flags in Table 14-13 is specified. We name such a region a "quantum region" for sake of clarity below. By default, each quantum region is treated independently from the rest with the Dirichlet boundary condition of vanishing wavefunctions at its boundary. Coupled models are generated by default by specifying QWELL or SLATT. When using the $\boldsymbol{k}\cdot\boldsymbol{p}$ models, SLATT and QWELL are equivalent and below we only refer to QWELL.

Table 14-13 Flags for using various *k*·*p* models and the zone center states they contain

		Cartesian Basis (Zinc-Blende and Wurtzite)							
Flag	**Cond-Val Coupling**	$\|S\uparrow\rangle$	$\|S\downarrow\rangle$	$\|X\uparrow\rangle$	$\|Y\uparrow\rangle$	$\|Z\uparrow\rangle$	$\|X\downarrow\rangle$	$\|Y\downarrow\rangle$	$\|Z\downarrow\rangle$
KP.CV6	**User-Defined**	X	X	X	X	X	X	X	X
KP.V6	0			X	X	X	X	X	X
		Angular Momentum Basis (Zinc-Blende only)							
	User-Defined	$\|S\uparrow\rangle$	$\|S\downarrow\rangle$	$\|\frac{3}{2},\frac{3}{2}\rangle$	$\|\frac{3}{2},\frac{1}{2}\rangle$	$\|\frac{3}{2},\frac{-1}{2}\rangle$	$\|\frac{3}{2},\frac{-3}{2}\rangle$	$\|\frac{1}{2},\frac{1}{2}\rangle$	$\|\frac{1}{2},\frac{-1}{2}\rangle$
KP.CV4	0	X	X	X	X	X	X		
KP.V4				X	X	X	X		
		Block Diagonalized Basis (Zinc-Blende only)							
	User-Defined	$\|S\uparrow\rangle$	$\|S\downarrow\rangle$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$			
KP.CV3	0	X	X	X	X	X			

Table 14-13 Flags for using various *k·p* models and the zone center states they contain									
KP.CV2		X	X	X	X				
KP.V3			X	X	X	X			
KP.V2				X	X				

If the QWELL parameter is true for any quantum region, then that region is combined with the quantum regions that are its neighbors in real space. If the $\boldsymbol{k \cdot p}$ models for the two regions are different sub-models of the 8-band model, then a larger model containing the basis states of both is defined for the combined region.

The above process continues until each quantum region with QWELL is surrounded by regions for which none of the parameters is in Table 14-13 is true. Therefore, the presence of QWELL flag results in treating all neighboring quantum regions as coupled by default, which yields the coupled multiple quantum well model. The other important aspect of this flag is to perform the calculation of linear optical response for the full quantum region automatically after solving for the band structure. However, Atlas keeps track of the sub-regions for which you have specified the QWELL parameter on the **REGION** or **MODELS** statements. It localizes the computed optical response to those sub-regions, scaling the gain coefficient by the ratio *L/R*, where *L* is the full length of the quantum region, and *R* is the sum of the lengths of the sub-regions for which QWELL is explicitly specified. This allows multiple quantum wells to be treated as coupled. Only one expensive calculation is then performed per corrector cycle. At the same time, the optical gain is localized to the well region for use in the drift-diffusion and photon rate equations.

In systems confined in one spatial direction, the optical gain (absorption) coefficients are dimensionless. They represent the fraction of photons generated (absorbed) from a flux incident on the quantum region. However, it is common practice to maintain the uniformity of definitions and divide the optical gain in 2D systems by the length of the confinement direction. Furthermore, along the confinement direction, we can only determine the wavefunction and associated probabilities of locating a carrier. This prevents specification of gain and radiative recombination localized to a point in space, as required in drift diffusion and photon rate equations. Using QWELL parameter as described above allows you to localize the gain to a small but finite volume. This type of locality is necessary to interface the model with the classically motivated rate equations. In certain cases, especially if the amalmagated region becomes too large, this classical locality with the quantum mechanical results can be made more consistent by preventing one or more regions to be joined with others. This allows you full control over the extent to which the quantum region expands automatically.

You can exclude a particular region completely from the procedure joining multiple regions. In this case, specify a different value for the QWNUM parameter at the **REGION** statement. Regions are combined as above only if they have the same QWNUM value. They must also have the same confinement direction specified by the SP.GEOM parameter. An alternative route to achieving the same effect is to define disjoint quantum regions, so that two regions marked for quantum calculation via the parameters in Table 14-13 are separated by a region for which none of the parameters are true.

A separate calculation is performed for each independent quantum region. The quantum regions (after expansion) are then allowed to overlap. This results in multiple values of physical quantities at mesh nodes lying within the overlap of quantum regions. In this case, the average of the multiple values will be computed at those nodes.

Once a set of quantum regions is identified as described above, a 1D spatial mesh is applied across each region along the orientation defined by the SP.GEOM parameter. The orientation is currently assumed to be the [001] crystallographic direction. The grid spacing is set to the average grid spacing of the Atlas mesh inside the region. This can be overridden for one or more regions by specifying the WELL.NY (for 1DY) or WELL.NX (for 1DX) on the **MODELS** statement for the corresponding regions. The mesh is duplicated to create a uniform distribution of slices along the direction orthogonal to the orientation. The number of slices is set by WELL.NX (for 1DY) or WELL.NY (for 1DX) if specified. Otherwise, this number is also determined by setting the spacing between the slices to be the average grid spacing of Atlas mesh in the corresponding direction.

Atlas solves the Schrodinger equations of a ***k.p*** model using the method of finite elements. By default, second order polynomials are used in each cell of the spatial mesh. The cell basis are converted to nodal basis via the wavefunction continuity imposed at each cell boundary. The default corresponds to the KP.FEM.LAGRANGE parameter on the **MODELS** statement, which uses second order polynomials. The order can be changed to third order Hermite polynomials by specifying KP.FEM.HERMITE. With Hermite polynmials, the extra degree of freedom can be used to match the derivatives of the wavefunctions by integrating the matrix ***k.p*** equation across each cell boundary. This results in a scattering matrix relating the coefficients in one cell to its neighbors, and yields the same number (but different type) of nodal basis as in the second order case. To activate matching, specify KP.FEM.INTERFACE on the **MODELS** statement. A different scattering matrix must be constructed at each ***k*** when derivative matching is on. This results in different nodal bases (and a different overlap matrix) at each ***k***, which is more time consuming than the second order case. You can quantify corrections arising from explicit derivative matching for a particular problem by comparing results with KP.FEM.INTERFACE set to True with those when KP.FEM.INTERFACE is set to False.

Parameters Controlling The Execution Of A Calculation

A set of eigenvalues $E_n(\boldsymbol{k})$ and the corresponding eigenstates $\psi_n(\boldsymbol{k}_{||}, x)$ are calculated at each $\boldsymbol{k}_{||}$ point.The eigenvalues and eigenstates of this Hamiltonian are calculated at energies closest to the fundamental gap using the restarted Arnoldi process implemented in the Arpack package. The number of valence sub-bands to calculate is specified by WELL.VNBS and the number of conduction sub-bands by WELL.CNBS. The Arpack package is faster when searching for a cluster of eigenvalues separated from the rest of the spectrum. You can do this by setting WELL.VNBS to a multiple of the number of valence bands in the underlying bulk model. The tolerance for the Arpack solver is set by KP.SOLVERTOL.

The axial approximation is used in $\boldsymbol{k}_{||}$ space, which yields an isotropic band structure as a function of the magnitude $k = |\boldsymbol{k}_{||}|$ of the wave vector. A 1D mesh is initially created with KP.NUMKPTS many points uniformly distributed between k=0 and KP.KMAX nm^{-1}. The density of states (DOS) for each band is calculated by summing the DOS for each band by performing the integral

$$\rho(E) = \sum_j 2\pi \int dk k \delta(E - E_{jk}) \quad \text{14-52}$$

analytically within each mesh element by using a second order polynomial fit interpolating through the calculated E_{jk} at the endpoints and the midpoint of the element. If the KP.ADAPTIVE parameter is set, then each mesh element in $\boldsymbol{k}_{\|}$ space is split into two sub-elements and the integral is repeated for each sub-element. An error is estimated as the difference between the integral over the original element, and the sum of its sub-elements. If this error is larger than the tolerance specified by KP.DOSTOL, the sub-division is continued for each sub-element and so forth until the tolerance is met in each element of the $\boldsymbol{k}_{\|}$ space cell. This generates an accurate DOS over the range of $\boldsymbol{k}_{\|}$ points specified. The DOS is generated exactly (within numerical precision) for a parabolic band. The range of energies for DOS is set automatically by the maximum and minimum band energies over the $\boldsymbol{k}_{\|}$ points in the mesh. The resolution of DOS is set in eV by KP.DOSRES.

Once a DOS is computed, Atlas computes the Fermi level of electrons and holes from the expectation value of the areal density over the quantum region. The expected areal density is computed by integrating the volume density along the growth axis over the entire quantum region. The areal densities for electrons and holes, ρ_e and ρ_h respectively, are related to the Fermi levels, μ_e and μ_h by

$$\rho_e = \sum_n \int_{E_{max}}^{E_0} dE \frac{\rho_n(E)}{1 + \exp(E - \mu_e)} / kT \quad \text{14-53}$$

$$\rho_h = \sum_n \int_{E_{min}}^{E_0} dE \frac{\rho_n(E)}{1 + \exp(\mu_h - E)} / kT$$

where E_{min} are the maximum and minimum energies of all the bands. E_0 is placed at middle of the gap. No contribution to the integral comes from the energies between the band edge and E_0. Therefore, the choice of this parameter is arbitrary as long as it lies inside the gap. Atlas solves the above equations by first using the bisection method to find the energy interval where the Fermi level lies. It then uses the Newton method to converge Fermi levels to within a user-specified tolerance in density. It is important to have sufficiently high KP.KMAX so that the generated band structure allows the carrier densities to drop off to zero from the band edges towards the maximum and minimum energies for conduction and valence states respectively.

Since a continuous set of non-parabolic bands is generated, discrete energies and the associated eigenfunctions at the sub-band edges no longer provide sufficient information about the local behavior of electronic states. Instead, as the calculation proceeds a local density of states (LDOS) is created, which is defined by

$$\rho(x, E) = 2\pi \sum_{jn'} \int dk k |\psi_{n'j}(\boldsymbol{k}, x)|^2 \delta(E - E_{jk}) \quad \text{14-54}$$

The LDOS gives the probability density per unit of energy of finding an electron at energy E and point x. It is akin to the probability density associated with the wavefunction and produces the DOS when integrated over all space

$$\rho(E) = \int dx \rho(x, E) \quad \text{14-55}$$

The LDOS is generated for the same energies as the DOS. The KP.CNLDOS parameter specifies the number of samples of LDOS between the minimum and maximum conduction sub-band energy. The KP.VNLDOS parameter specifies the samples within the valence sub-bands. The band structure must be calculated over sufficiently large KP.KMAX so that the occupation of the conduction states and the hole states drops off to zero as the highest and lowest energies within the calculated band structure are reached.

Calculation of Optical Response

Following the above calculations, the linear optical response is calculated for each quantum region for which QWELL is true. The linear optical response can be given by the imaginary part of the optical susceptibility alone, where the real part can be constructed using Kramers-Kronig relations. The response is fundamentally related to the Hamiltonian for the interaction of electrons and a classical electromagnetic field

$$H_{int}(\boldsymbol{r}, t) = e\boldsymbol{A}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{v}} \quad \text{14-56}$$

where $\hat{\boldsymbol{v}} = \hat{p}/m_0$ is the velocity operator. The linear optical response at energy $E = \hbar\omega$ follows by setting $\boldsymbol{A}(\boldsymbol{r}, t) = \boldsymbol{A}e^{-i\omega t}$, where the amplitude $\boldsymbol{A}$ is treated as uniform over the entire quantum region. The uniformity approximation is generally valid since the optical wavelengths are often much larger than the quantum region. Once the sub-band states $\boldsymbol{\psi}_n(k, x)$ are calculated as vectors over the bulk basis of the model, the matrix elements of the velocity operator are generated from the Hamiltonian by exploiting the relation $\hbar\boldsymbol{v}(k) = \partial H(k)/\partial k$, which holds for the full Hamiltonian. Thus, the matrix elements of $\boldsymbol{v}$ are diagonal in $\boldsymbol{k}_{\|}$. The matrix elements of the in-plane components of $\boldsymbol{v}$, between the sub-band states j and l are given by

$$\nu_{jl}^{\alpha}(k) = \frac{1}{\hbar}\sum_{nm}\int\Bigg([Q_{nm}^{\alpha}(x) + \overline{Q}_{nm}^{\alpha}(x) + (D_{nm}^{\alpha\beta} + D_{nm}^{\beta\alpha})k_{\beta}]\psi_{nj}^{*}(k, x)\psi_{mj}(k, x) + D_{nm}^{\alpha x}(x)\psi_{nj}^{*}(k, x)\frac{d\psi_{ml}(k, x)}{dx} + D_{nm}^{x\alpha}(x)\frac{d\psi_{nj}^{*}(k, x)}{dx}\psi_{ml}(k, x)\Bigg)dx \quad \text{14-57}$$

where n and m enumerate the bulk basis states. The components along the growth direction are given by

$$\nu_{jl}^{x}(k) = \frac{1}{\hbar}\sum_{nm}\int\Big([D_{nm}^{\alpha x} + D_{nm}^{x\alpha}]k_{\alpha}\psi_{nj}^{*}(k, x)\psi_{mj}(k, x) + [Q_{nm}^{x}(x) + D_{nm}^{xx}(x) + D_{nm}^{\alpha x}(x)k_{\alpha}]\psi_{nj}^{*}(k, x)\frac{d}{dx}\psi_{mj}(k, x) + [\overline{Q}_{nm}^{x}(x) + D_{nm}^{xx}(x) + D_{nm}^{x\alpha}(x)k_{\alpha}]\psi_{mj}(k, x)\frac{d}{dx}\psi_{nj}^{*}(k, x)\Big)dx \quad \text{14-58}$$

The velocity matrix elements can have components along the diagonal (i.e., intra-band components for j=l). The diagonal components of the in-plane velocity matrix represent the

energy band gradients. For example, $v_{jj}^{\alpha}(k)$ is the group velocity component in direction α of a carrier wavepacket in band *j* narrowly located at quasi-momentum *k*.

These matrix elements enter the Fermi Golden rule calculation of the transition rates among the electronic states, which yields the imaginary part of the linear optical susceptibility. The net optical gain is proportional this susceptibility and is a more useful quantity to define for device calculations. The latter consists of $g(E) = u(E) - w(E)$ where the emission contribution from downward transitions are contained in $u(E)$, and the upward transitions contributing the absorption are contained in $w(E)$. These two functions switch roles when we change the sign of *E*, giving the fundamental requirement of zero gain for static perturbations *E*=0. Thus, these functions are completely defined for *E*>0 and are given by

$$w(E) = \frac{\pi e^2 \hbar}{\varepsilon_0 c n_R E} \sum_{nm} \int dk k (\hat{e} \cdot \boldsymbol{v}_{nm}(k))(\boldsymbol{v}_{nm}(k) \cdot \hat{e}^*) \delta(E_m(k) - E_n(k) - E) f_n(1 - f_m) \quad \text{14-59}$$

$$u(E) = \frac{\pi e^2 \hbar}{\varepsilon_0 c n_R E} \sum_{nm} \int dk k (\hat{e} \cdot \boldsymbol{v}_{nm}(k))(\boldsymbol{v}_{nm}(k) \cdot \hat{e}^*) \delta(E_m(k) - E_n(k) - E) f_m(1 - f_n) \quad \text{14-60}$$

In the above equations, $\hat{e}$ is the polarization direction of the electric field. Atlas computes *w*(*E*) and *u*(*E*) for two orthogonal directions: TE (for $\hat{e}$ parallel to the well plane) and TM $\hat{e}$ perpendicular to the well plane. The resulting $w^{TE}(E)$ and $u^{TE}(E)$ are written separately in the output and they are applied separately within the optical absorption models to allow for polarization dependence of the optical response. However, when using these models as input to Luminous, the latter sums the TE and TM modes and loses polarization dependence as a result.

Atlas calculates *u*(*E*), *w*(*E*), and *g*(*E*) at energies ranging over all possible transitions over the calculated band structure. It automatically selects the energy grid to cover this range. The resolution of this grid can be specified in eV by the KP.CHIRES parameter. From this calculation, Atlas then computes the spontaneous emission rate per eV per cm^3 by multiplying *u*(*E*) with the photon density of states. The total radiative recombination rate is calculated as the integral of the spontaneous emission over all energies.

Depending on the spatial mesh, the bulk model, and the tolerances specified above, the $\boldsymbol{k} \cdot \boldsymbol{p}$ calculations can be very time consuming. However, it often suffices to calculate the optical response at a given bias point, without requiring self-consistency between the quantum states and the Poisson potential. In this case, KP.OFF can be specified on a SOLVE statement to bypass the $\boldsymbol{k} \cdot \boldsymbol{p}$ calculation entirely. Once a bias point is reached where the optical response is desired, issuing a **SOLVE** PREV without KP.OFF) will perform the $\boldsymbol{k} \cdot \boldsymbol{p}$ calculations with the potential at that point, and for the areal carrier densities obtained by integrating the densities within the quantum region. If SPONT is set on the **MODELS** statement, this repeat calculation will be performed with the radiative recombination rates in the quantum region calculated from the optical response calculated above. To perform a calculation using the results of the $\boldsymbol{k} \cdot \boldsymbol{p}$ calculation at the previous **SOLVE** statement, specify KP.PREV instead of KP.OFF.

14.8 Quantum Transport: Non-Equilibrium Green's Function Approach

This section describes a Non Equilibrium Green's Function approach (NEGF). This fully quantum method treats such effects as source-to-drain tunneling, ballistic transport, quantum confinement on equal footing. In the first part, we discuss a mode space approach, where NEGF formalism in transport direction is coupled with Schrodinger equation in transverse plane. This situation is common to double gate transistors, FinFETs, nanowire FETS, and so on. In the second part, we discuss a planar NEGF approach, where the transverse plane is assumed to be infinite. This approach is more applicable to multiple barrier structures, such a resonant tunneling diode (RTD).

14.8.1 Mode Space NEGF Approach

By specifying the NEGF_MS and SCHRODINGER options on the **MODELS** statement, you can launch an NEGF solver to model ballistic quantum transport in such devices as double gate or surround gate MOSFET in Atlas2D or Atlas3D. All the parameters described in Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model" have the same meaning in the context of NEGF.

An effective-mass Hamiltonian H_o of a two-dimensional device, given by

$$H_o = -\frac{h^2}{2}\left[\frac{\partial}{\partial x}\left(\frac{1}{m_x^v(x,y)}\frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{m_y^v(x,y)}\frac{\partial}{\partial y}\right)\right] \quad 14\text{-}61$$

is discretized in real space using a finite volume method. A corresponding expression in cylindrical coordinates is

$$H_o = -\frac{h^2}{2}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{1}{m_r^v(r,z)}r\frac{\partial}{\partial r}\right)-\frac{1}{m_r^v(r,z)}\frac{m^2}{r^2}+\frac{\partial}{\partial z}\left(\frac{1}{m_z^v(r,z)}\frac{\partial}{\partial z}\right)\right] \quad 14\text{-}62$$

Instead of solving a 2D or 3D problem, which may take too much computational time, a Mode Space (MS) approach is used. A Schrodinger equation is first solved in each slice of the device to find eigen energies and eigen functions. Then, a transport equation of electrons moving in the sub-bands is solved. Because only a few lowest eigen sub-bands are occupied and the upper sub-bands can be safely neglected, the size of the problem is reduced. Moreover, in the devices where the cross-section does not change (e.g., a nanowire transistor), the sub-bands are not quantum-mechanically coupled to each other and the transport equations become essentially 1D for each sub-band. Therefore, you can further divide the method into Coupled (CMS) or Uncoupled Mode Space (UMS) approaches. Atlas will automatically decide on the minimum number of sub-bands required and the method to be used. It is possible, however, to set the number of sub-bands by using the EIGEN parameter on the **MODELS** statement. To enforce either CMS or UMS approaches, you can use NEGF_CMS or NEGF_UMS instead of NEGF_MS on the **MODELS** statement. The transformation of a real space Hamiltonian H_o to a mode space is done by taking a matrix element between m-th an n-th wave functions of k-th and l-th slices:

$$H^{MS}{}_{mnkl} = \langle \Psi^k{}_m(y)|H_o|\Psi^l{}_n(y)\rangle \quad 14\text{-}63$$

Note, that in cylindrical geometry, different orbital quantum numbers will result in zero matrix element and are therefore uncoupled. The quantum transport equations in the mode space read:

$$\left(E - H^{MS}{}_{mnkl} - \Sigma_S^R - \Sigma_D^R\right) G^R = I \qquad 14\text{-}64$$

$$\left(E - H^{MS}{}_{mnkl} - \Sigma_S^R - \Sigma_D^R\right) G^< = \left(\Sigma_S^< + \Sigma_D^<\right) G^A \qquad 14\text{-}65$$

Here, E is an electron (hole) energy. $\Sigma_{S,D}{}^R$ are retarded self-energies, carrying information about the density of states of source and drain semi-infinite contacts. $\Sigma_{S,D}{}^<$ are less-than self-energies, carrying information about the Fermi distribution function in contacts.

The solution of these equations are a retarded Green's function $G^R(E)$, whose diagonal elements have a meaning of the local density of states as a function of energy and a less-than Green's function $G^<(E)$, whose diagonal elements have a meaning of carrier density as a function of energy. $G^A(E) = (G^R(E))^\dagger$ is an advanced Green's function. A non-uniform energy grid will be generated internally, while you can control the number of energy grid points using a parameter ESIZE.NEGF= on the **MODELS** statement. After solving the above equations at each energy, following quantities of interest are computed by Atlas and stored in a structure file (expressions are given for rectangular grid):

- carrier density

$$n(x_i, y_j) = -\frac{i}{L_z}\sum_{k_z \sigma}\sum_{mn}\int G^<{}_{mnii}(E)\Psi^i{}_m(y_j)\Psi^{*i}{}_n(y_j)\frac{dE}{2\pi} \qquad 14\text{-}66$$

- x-component of current density

$$J_x(x_i, y_j) = -\frac{2e}{hL_z\Delta_x}\sum_{k_z \sigma}\sum_{mn}\int \Re e(t_{i+1ij}G^<{}_{mnii+1}(E))\Psi^i{}_m(y_j)\Psi^{*i+1}{}_n(y_j)\frac{dE}{2\pi} \qquad 14\text{-}67$$

- y-component of current density

$$J_y(x_i, y_j) = -\frac{2e}{hL_z\Delta_y}\sum_{k_z \sigma}\sum_{mn}\int \Re e(t_{iijj+1}G^<{}_{mnii}(E))\Psi^i{}_m(y_j)\Psi^{*i}{}_n(y_{j+1})\frac{dE}{2\pi} \qquad 14\text{-}68$$

- total current density

$$J = (J_x^2 + J_y^2)^{1/2} \qquad 14\text{-}69$$

Here, t_{ijkl} is an off-diagonal element of real space Hamiltonian H_o, which couples nodes (x_i,y_k) and (x_j,y_l). The summation over k_z is absent in the cylindrical geometry. Instead, a summation over orbital quantum numbers is performed.

Additionally, you can store energy dependent quantities as a spectrum at a certain coordinate along transport direction or as a contour plot of energy and coordinate. There are two ways to save spectrum or contour file :

1. Specify FILENAME= parameter on **PROBE** statement and set NEGF.LOG on the **SAVE** or **SOLVE** statement. An incremental number will be added to the file name, each time a new spectrum is saved. To save contour, specify FILENAME parameter on **PROBE** statement and set NEGF.CONT on the SAVE or SOLVE statement.
2. To save spectrum log file, specify NEGF.LOGFILE= parameter on the **SAVE** or **SOLVE** statement. To save contour, specify NEGF.CONTFILE= parameter on the **SAVE** or **SOLVE** statement

The following quantities can be stored into spectrum or contour file:

- Transmission coefficient for m-th sub-band (only for spectrum file)

$$T_m(E) = tr\left\{G^R\Gamma_S G^A\Gamma_D\right\}, \quad \Gamma_{S,D} = 2\Im m\left\{\Sigma^R_{S,D}\right\} \qquad 14\text{-}70$$

You must set a TRANSMISSION parameter on the **PROBE** statement. It is followed by BAND= and STATE= parameters to specify the effective mass band (valley) and the number of sub-band respectively. If BAND= or STATE= is missing, a summation over this index is performed.

- Density of states in the m-th sub-band

$$DOS_m(E, x_i) = 2\Im m\left\{G^R_{mmii}(E)\right\} \qquad 14\text{-}71$$

To save DOS into spectrum file, you must set a DOSVSE parameter on the **PROBE** statement. To save DOS into a contour file, set DOS.CONT on the **PROBE** statement. It is followed by BAND= , STATE= , and (only for spectrum file) X= parameters to specify the effective mass band (valley), the number of sub-band, and the location along transport direction respectively. If BAND= or STATE= is missing, a summation over this index is performed.

- Electron density per energy in m-th sub-band

$$N_m(E, x_i) = \Im m\left\{G^<_{mmii}(E)\right\} \qquad 14\text{-}72$$

To save density into spectrum file, you must set a parameter DENSVSE on the PROBE statement. To save density into a contour file, set DENS.CONT on the **PROBE** statement. It is followed by BAND=, STATE=, and (only for spectrum file) X= parameters to specify the effective mass band (valley), the number of sub-band, and the location along transport direction respectively. If BAND= or STATE= is missing, a summation over this index is performed.

- Current density per energy in the m-th sub-band

$$J_m(E, x_i) = \frac{2e}{h}\Re e\left\{h_{mmi+1i}G^<_{mmii+1}(E)\right\} \qquad 14\text{-}73$$

To save current density into spectrum file, you must set a parameter CURDVSE on the **PROBE** statement. To save density into a contour file, set CURD.CONT on **PROBE** statement. It is followed by BAND=, STATE=, and (only for spectrum file) X= parameters to specify the effective mass band (valley), the number of sub-band, and the location along transport direction respectively.

The energy dependent quantities are stored on a uniform energy grid. You can specify the size of the grid by setting a parameter ESIZEOUT.NEGF= on **OUTPUT** statement. If it is not specified, the size will be the same as ESIZE.NEGF on the **MODELS** statement

A solution of NEGF equations is performed self-consistently with Poisson equation using a predictor-corrector scheme. You can specify the number of predictor iterations using a NPRED.NEGF= parameter on the **MODELS** statement. The criterion for convergence of Poisson iterations is set by a QCRIT.NEGF= parameter on the **MODELS** statement.

By default, Dirichlet boundary conditions (BC) for potential and open-boundary condition for electrons are used in the contacts. You can also use von Neumann boundary conditions for potential by specifying REFLECT parameter on the **CONTACT** statement. This type of BC is useful because ballistic carriers do not create voltage drop near the source and drain contacts. Moreover, the potential in the source and drain must be allowed to float because the number of electrons reflected backward and thus total carrier concentration in contacts can change with bias. The third type of BC is applicable to Schottky contacts, where potential is fixed by Dirichlet BC, but carriers are injected at all energies, mimicking a near-constant density of states of metal contact near the Fermi level. To set Schottky BC, set NEGF.TUNNEL on the **CONTACT** statement and also specify metal density of states by DOS.NEGF with the default of 0.1 1/eV. In the case of Schottky contacts, contact self-energy is given by

$$\Sigma_c^R = i \cdot t^2 \cdot \mathrm{DOS.NEGF} \qquad 14\text{-}74$$

Finally, a few comments has to be made about NEGF model:

- NEGF is a fully quantum approach to electron transport, which takes into account the wave nature of electron in both transport and transverse directions. This is the main difference of this method and drift-diffusion or Boltzmann Transport equation, which treat electron classically.
- In the present version, the transport is ballistic (i.e., no scattering occurs). Currently, this model gives an upper limit of current. In future versions, electron-phonon scattering may be added.
- The model is robust to predict subthreshold current, threshold voltage. The Poisson convergence, however, may be unstable when a device in completely open because the assumption of equilibrium contacts breaks down. In the long run, adding scattering will mitigate this problem.
- A discretized effective-mass Hamiltonian has a finite bandwidth, which depends on the mesh spacing in the transport direction and is given by $h^2/(2m\Delta^2_x)$. A spacing in the transport direction should be fine enough, so that a half-bandwidth of discretized effective mass Hamiltonian is larger than applied bias. Atlas will give a warning if the grid is not fine enough.

14.8.2 Planar NEGF approach

Most of the equations and syntax of the mode-space approach are also applicable to planar devices. Here, we discuss only a few new parameters specific to planar approach. Planar NEGF solver is based on 1D NEGF with effective mass Hamiltonian discretized in real space. Current and carrier density are obtained by analytically integrating Green's function over 2D transverse k-space. The solution is done for each electron or hole band independently of all other bands. Ballistic transport is assumed. In real devices, such as resonant tunneling diodes (RTD), injection into the double barrier structure occurs from emitter quasi-bound states. To model this effect without including inelastic phonon scattering, we treat emitter and collector as quasi-equilibrium regions, while the central region with double barrier structure is treated as non-equilibrium.

To launch the model, specify `N.NEGF_PL1D` or `P.NEGF_PL1D` or both on the **`MODELS`** statement for electrons and holes respectively. This will tell Atlas to solve 1D NEGF equation in the first slice in transport direction and copy carrier and current density to all other slices. If the slices are not equivalent, use `N.NEGF_PL` or `P.NEGF_PL` or both instead to solve 1D NEGF equation for each slice separately.

To specify a region as quasi-equilibrium, use `EQUIL.NEGF` on the **`MODELS`** or the **`REGION`** statements. It is a good idea to specify all regions in front of the first barrier and behind the last barrier as quasi-equilibrium, so that all possible quasi-bound states are filled. Quasi-equilibrium regions are characterized by a broadening, which can be set by `ETA.NEGF` parameter on the **`MODELS`** or the `REGION` statements, with a default of 0.0066 eV.

The output quantities of interest are the IV characteristics, spectrum of transmission, DOS, density and current, and also spatial profiles of carrier density and conduction or valence band edge.

The solution of 1D NEGF equations starts with resonance finding in the whole device using Quantum Transmitting Boundary Method (QTBM) and a construction of the energy grid. It is the resolution of these resonances that is critical for correct simulation. The resonances can be seen as extremely narrow peaks in the transmission vs. energy plot. Use log scale in TonyPlot to get a better understanding of the position of the transmission and DOS resonances.

In case of poor convergence, energy grid size (`ESIZE.NEGF` on the **`MODELS`** statement) and magnitude of the broadening in quasi-equilibrium regions have to be increased.

14.9 Drift-Diffusion Mode-Space Method (DD_MS)

This model is a semiclassical approach to transport in devices with strong transverse confinement and is a simpler alternative to mode-space NEGF approach (NEGF_MS) as described earlier. Similar to NEGF_MS, the solution is decoupled into Schrodinger equation in transverse direction and 1D transport equations in each sub-band. In this model, however, a classical drift-diffusion equation is solved instead of a quantum transport equation. Thus, the model captures quantum effects in transverse direction and yet inherits all familiar Atlas models for mobility, recombination, impact ionization, and band-to band tunneling. Unlike NEGF_MS, the model can handle only devices with uniform cross-section because quantum mechanical coupling between electron sub-bands is neglected. To activate the model, use DD_MS together with SCHRO or P.SCHRO parameters or both on the **MODELS** statement. Note that CARRIERS parameter on the **METHOD** statement should be set to 0 because multi-dimensional Atlas continuity equation solvers are not used in this method.

One-dimensional drift-diffusion transport in each sub-band, characterized by sub-band index ν and effective mass index b is described by

$$\frac{1}{q}\frac{\partial J_{\nu b}}{\partial x} = R_{\nu b} - G_{\nu b} \qquad 14\text{-}75$$

$$J_{\nu b} = q\mu_{\nu b} n_{\nu b}\frac{\partial E_{\nu b}}{\partial x} - qD_{\nu b}\frac{\partial n_{\nu b}}{\partial x} \qquad 14\text{-}76$$

where $J_{\nu b}(x)$ is sub-band current, $n_{\nu b}(x)$ is sub-band carrier density to be found, and $E_{\nu b}(x)$ is sub-band eigen energy as found by Schrodinger solver in transverse direction. The 1D transport equation is discretized using Scharfetter-Gummel scheme. Mobility $\mu_{\nu b}(x)$ is defined by Atlas mobility models and is in general dependent on material, doping, local longitudinal electric field, and temperature. Diffusion coeffiecient $D_{\nu b}(x)$ is related to mobility by

$$D_{\nu b} = \mu_{\nu b}\left(\frac{\partial n_{\nu b}}{\partial E_{\nu b}}\right)^{-1} \qquad 14\text{-}77$$

As Fermi-Dirac statistics is always assumed, the relation between sub-band carrier densities (cm^{-3}), quasi-fermi levels and eigen energies is given by the following relations:

$$n_{\nu b} = 2\frac{k_B T}{A\pi h^2}\sum_{\nu}\sqrt{m_x^{\nu b} m_z^{\nu b}}\ln\left[1 + \exp\left(-\frac{E_{\nu b} - E_{F,\nu b}}{k_B T}\right)\right] \qquad 14\text{-}78$$

in case of 1D confinement in y direction and

$$n_{\nu b} = 2\frac{\sqrt{2k_B T}}{Ah}\sum_{\nu}\sqrt{m_x^{\nu b}}F_{-1/2}\left(-\frac{E_{\nu b} - E_{F,\nu b}}{k_B T}\right) \qquad 14\text{-}79$$

in case of 2D confinement in y-z plane, where A is a normalizing area.

Recombination rates $R_{\nu b}(x)$ are computed by taking into account recombination of a particular electron (hole) sub-band with all hole (electron) sub-bands:

$$R_{\nu b} = \sum_{\lambda b'} R_{\nu b \lambda b'} \equiv \sum_{\lambda b'} \frac{n_{\nu b} p_{\lambda b'} - n_i^2}{\tau_{\nu b \lambda b'}} \quad 14\text{-}80$$

Here, $R_{\nu b \lambda b'}(x)$ are computed using Atlas bulk recombination models with a substitution of sub-band carrier densities for bulk densities, sub-band eigen energies for conduction and valence bands, and sub-band fermi levels for bulk fermi levels. Unlike the rest of Atlas, recombination rates are set to 0 when only one carrier type is solved for.

Total impact ionization generation rates are computed by taking a sum over all electron and hole sub-bands:

$$G_{total} = \sum_{\nu b \lambda b'} (\alpha_n^{\nu b} |J_{\nu b}| + \alpha_p^{\lambda b'} |J_{\lambda b'}|) \quad 14\text{-}81$$

Impact ionization coefficients α_n and α_p are computed using Atlas bulk impact ionization models. Then, the generation rate in each sub-band is given by G_{total}, weighted by the carrier population in the sub-band:

$$G_{\nu b} = G_{total} \cdot \frac{n_{\nu b}}{n_{total}} \quad 14\text{-}82$$

$$G_{\lambda b'} = G_{total} \cdot \frac{p_{\lambda b'}}{p_{total}} \quad 14\text{-}83$$

Unlike the rest of Atlas, it is allowed to use impact ionization model with only one carrier type. While one carrier type calculation may give incorrect answer, it is sometimes very useful to understand the effect of electrons and holes separately.

When generation-recombination mechanisms are present, you can iterate between carrier density and G-R rates before solving the Poisson equation. The self-consistency between carrier density and G-R rates, achieved in this inner iteration procedure, will result in a more stable Poisson convergence. To control the number of inner iterations, use the `RGITER.DDMS=N` and the `RGCONV.DDMS=X` parameters on the **`METHOD`** statement. Here, `N` is the maximum number of inner iterations (default is 0) and `X` is minimum error between 0 and 1 (default is 1e-5).

After 1D transport equations are solved in each sub-band and quasi-fermi levels are extracted, bulk (3D) carrier densities and currents are computed by summing over all sub-bands and weighting with corresponding wave function squared as done in Equation 14-13 or Equation 14-14. The same procedure is used to compute total generation and recombination rates, which are stored in output structure file.

A usual predictor-corrector scheme is used to solve Poisson equation self-consistently. Similar to `NEGF_MS`, Dirichlet boundary conditions for potential in contacts are default. Von Neumann boundary conditions can be used by specifying `REFLECT` on the **`CONTACT`** statement. Schottky contacts boundary conditions can be set by `DDMS.TUNNEL` parameter on the **`CONTACT`** statement. In the case of Schottky contacts, potential is fixed but new generation

terms will be added to the drift-diffusion equations. The local tunneling generation rate is given by

$$G_{TUN}(x) = \frac{A^{*}T}{k}\vec{F}(x)\Gamma_{\nu}(x)\ln\left(\frac{1+\exp(E_{\nu}(x)-E_{F,\nu}(x))}{1+\exp(E_{\nu}(x)-E_{F,contact})}\right) \quad , \qquad 14\text{-}84$$

where A^* is Richardson constant, $F(x)$ is the local electric field, and $\Gamma(x)$ is the tunneling probability, computed within WKB approximation:

$$\Gamma_{\nu}(x) = \exp\left(-\frac{4\pi}{h}\int_{0}^{x}\sqrt{2m(E_{\nu}(0)-E_{\nu}(x'))}dx'\right) \qquad 14\text{-}85$$

An output structure file contains quantities summed over all sub-bands. If you want detailed information about 1D sub-band-resolved eigen energies, carrier densities, currents, quasi fermi levels or generation-recombination rates, you can save an additional log file along with master structure file by specifying the `DDMS.LOG` option on the **SAVE** or **SOLVE** statements. If you specify `DDMS.LOG` on the **SOLVE** statement, then you must also set `MASTER` and `OUTF`. The name of the extra file is the same as the corresponding structure file but ending with `"_ddms.log"`.

Chapter 15
TFT: Thin-Film Transistor Simulator

15.1 Polycrystalline and Amorphous Semiconductor Models

TFT is an Atlas module that simulates disordered material systems. TFT doesn't contain material models so you need to combine either S-Pisces or Blaze with TFT to simulate these material systems.

TFT enables you to define an energy distribution of defect states in the bandgap of semiconductor materials. This is necessary for the accurate treatment of the electrical properties of such materials as polysilicon and amorphous silicon.

The syntax used by TFT is a part of the Atlas syntax so there's no need for you to learn how to use a new simulator to run TFT simulations.

Before continuing with this chapter, you should be familiar with Atlas physics. If not, read Chapters 2 "Getting Started with Atlas" and 3 "Physics" before proceeding further. For more information on S-Pisces and Blaze, see Chapters 5 "S-Pisces: Silicon Based 2D Simulator" and 6 "Blaze: Compound Material 2D Simulator".

15.2 Simulating TFT Devices

This section is intended to illustrate the basic building blocks for a thin-film transistor simulation.

To use TFT, specify the addition of defect states into the bandgap of a previously defined crystalline material. Throughout this section the example of polysilicon is used. Amorphous silicon and other disordered materials are handled in a similar manner.

15.2.1 Defining The Materials

Use the Atlas command syntax to define a simple polysilicon thin-film transistor structure. The **MESH**, **X.MESH**, and **Y.MESH** statements can be used to construct a mesh as described in Section 2.6 "Defining A Structure". For more information on **MESH** statements, Sections 22.34 "MESH" and 22.2 "A.MESH, R.MESH, X.MESH, Y.MESH, Z.MESH".

When defining the material regions, use the following syntax.

```
REGION Y.MIN=-0.05 Y.MAX=0   OXIDE
REGION Y.MIN=0     Y.MAX=0.2 SILICON
REGION Y.MIN=0.2   Y.MAX=2   OXIDE
```

Note that the region is defined as silicon. You can also define the material as polysilicon. But note that it is the defect distribution rather than this initial material definition that will determine the electrical characteristics.

15.2.2 Defining The Defect States

Disordered materials contain a large number of defect states within the band gap of the material. To accurately model devices made of polycrystalline or amorphous materials, use a continuous density of states. The **DEFECTS** statement is used to specify the density of defect states (DOS) as a combination of exponentially decaying band tail states and Gaussian distributions of mid-gap states [151, 111]. In addition, you may need to model the grain-grain boundary interface as a thermionic field emission boundary [155].

Note: The INTDEFECTS, INTERFACE S.S, and TRAP.COULOMBIC models are avaiable for grain-grain boundary interface modeling. These models have been developed in collaboration with Epson and Cambridge University. For more information, see Sections 22.23 "INTDEFECTS", 22.24 "INTERFACE", and 22.37 "MODELS".

15.2.3 Density of States Model

It is assumed that the total density of states (DOS) and $g(E)$, is composed of four bands: two tail bands (a donor-like valence band and an acceptor-like conduction band) and two deep level bands (one acceptor-like and the other donor-like) which are modeled using a Gaussian distribution.

$$g(E) = g_{TA}(E) + g_{TD}(E) + g_{GA}(E) + g_{GD}(E) \quad \text{15-1}$$

Here, E is the trap energy, E_C is the conduction band energy, E_V is the valence band energy and the subscripts (*T, G,A, D*) stand for tail, Gaussian (deep level), acceptor and donor states respectively.

$$g_{TA}(E) = \mathrm{NTA}\ \exp\left[\frac{E - E_c}{\mathrm{WTA}}\right] \quad 15\text{-}2$$

$$g_{TD}(E) = \mathrm{NTD}\ \exp\left[\frac{E_v - E}{\mathrm{WTD}}\right] \quad 15\text{-}3$$

$$g_{GA}(E) = \mathrm{NGA}\ \exp\left[-\left[\frac{\mathrm{EGA} - E}{\mathrm{WGA}}\right]^2\right] \quad 15\text{-}4$$

$$g_{GD}(E) = \mathrm{NGD}\ \exp\left[-\left[\frac{E - \mathrm{EGD}}{\mathrm{WGD}}\right]^2\right] \quad 15\text{-}5$$

For an exponential tail distribution, the DOS is described by its conduction and valence band edge intercept densities (NTA and NTD), and by its characteristic decay energy (WTA and WTD).

For Gaussian distributions, the DOS is described by its total density of states (NGA and NGD), its characteristic decay energy (WGA and WGD), and its peak energy distribution (EGA and EGD). Table 15-1 shows the user-specifiable parameters for the density of defect states.

Table 15-1 User-Specifiable Parameters for Equations 15-2 to 15-5

Statement	Parameter	Default	Units
DEFECTS	NTA	1.12×10^{21}	cm^{-3}/eV
DEFECTS	NTD	4.0×10^{20}	cm^{-3}/eV
DEFECTS	NGA	5.0×10^{17}	cm^{-3}/eV
DEFECTS	NGD	1.5×10^{18}	cm^{-3}/eV
DEFECTS	EGA	0.4	eV
DEFECTS	EGD	0.4	eV
DEFECTS	WTA	0.025	eV
DEFECTS	WTD	0.05	eV
DEFECTS	WGA	0.1	eV
DEFECTS	WGD	0.1	eV

15.2.4 Trapped Carrier Density

The ionized densities of acceptor and donor like states (n_T and p_T respectively) are given by:

$$p_T = p_{TA} + p_{GA} \quad \text{15-6}$$

$$n_T = n_{TD} + n_{GD} \quad \text{15-7}$$

where n_{TA}, n_{GA}, p_{TD}, and p_{GD} are given below.

$$p_{TA} = \int_{E_V}^{E_C} g_{TA}(E) \cdot f_{t_{TA}}(E, n, p)\, dE \quad \text{15-8}$$

$$p_{GA} = \int_{E_V}^{E_C} g_{GA}(E) \cdot f_{t_{GA}}(E, n, p)\, dE \quad \text{15-9}$$

$$n_{TD} = \int_{E_V}^{E_C} g_{TD}(E) \cdot f_{t_{TD}}(E, n, p)\, dE \quad \text{15-10}$$

$$n_{GD} = \int_{E_V}^{E_C} g_{GD}(E) \cdot f_{t_{GD}}(E, n, p)\, dE \quad \text{15-11}$$

$f_{t_{TA}}(E, n, p)$ and $f_{t_{GA}}(E, n, p)$ are the ionization probabilities for the tail and Gaussian acceptor DOS, while $f_{t_{TD}}(E, n, p)$ and $f_{t_{GD}}(E, n, p)$ are the ionization probabilities for the donors.

In the steady-state case, the probability of occupation of a trap level at energy E for the tail and Gaussian acceptor and donor states are given by Equations 15-12 through 15-15.

$$f_{t_{TA}}(E, n, p) = \frac{v_n\, \texttt{SIGTAE}\, n + v_p\, \texttt{SIGTAH}\, n_i \exp\left[\frac{E_i - E}{kT}\right]}{v_n\, \texttt{SIGTAE}\left(n + n_i \exp\left[\frac{E - E_i}{kT}\right]\right) + v_p\, \texttt{SIGTAH}\left(p + n_i \exp\left[\frac{E_i - E}{kT}\right]\right)} \quad \text{15-12}$$

$$f_{t_{GA}}(E, n, p) = \frac{v_n\, \texttt{SIGGAE}\, n + v_p\, \texttt{SIGGAH}\, n_i \exp\left[\frac{E_i - E}{kT}\right]}{v_n\, \texttt{SIGGAE}\left(n + n_i \exp\left[\frac{E - E_i}{kT}\right]\right) + v_p\, \texttt{SIGGAH}\left(p + n_i \exp\left[\frac{E_i - E}{kT}\right]\right)} \quad \text{15-13}$$

$$f_{t_{TD}}(E,n,p) = \frac{v_p\ \texttt{SIGTDH}\ p + v_n\ \texttt{SIGTDE}\ n_i\ \exp\left[\frac{E-E_\zeta}{kT}\right]}{v_n\ \texttt{SIGTDE}\left(n + n_i\ \exp\left[\frac{E-E_i}{kT}\right]\right) + v_p\ \texttt{SIGTDH}\left(p + n_i\ \exp\left[\frac{E_i-E}{kT}\right]\right)} \quad \text{15-14}$$

$$f_{t_{GD}}(E,n,p) = \frac{v_p\ \texttt{SIGGDH}\ p + v_n\ \texttt{SIGGDE}\ n_i\ \exp\left[\frac{E-E_i}{kT}\right]}{v_n\ \texttt{SIGGDE}\left(n + n_i\ \exp\left[\frac{E-E_i}{kT}\right]\right) + v_p\ SIGGDH\left(p + n_i\ \exp\left[\frac{E_i-E}{kT}\right]\right)} \quad \text{15-15}$$

where v_n is the electron thermal velocity and v_p is the hole thermal velocity, n_i is the intrinsic carrier concentration. SIGTAE and SIGGAE are the electron capture cross-section for the acceptor tail and Gaussian states respectively. SIGTAH and SIGGAH are the hole capture cross-sections for the acceptor tail and Gaussian states respectively and SIGTDE, SIGGDE, SIGGDH, and SIGGDH are the equivalents for donors states.

Table 15-2 User-Specifiable Parameters for Equations 15-12 to 15-15

Statement	Parameter	Default	Units
DEFECTS	SIGTAE	1.0×10^{-16}	cm^2
DEFECTS	SIGTDE	1.0×10^{-14}	cm^2
DEFECTS	SIGGAE	1.0×10^{-16}	cm^2
DEFECTS	SIGGDE	1.0×10^{-14}	cm^2
DEFECTS	SIGTAH	1.0×10^{-14}	cm^2
DEFECTS	SIGTDH	1.0×10^{-16}	cm^2
DEFECTS	SIGGAH	1.0×10^{-14}	cm^2
DEFECTS	SIGGDH	1.0×10^{-16}	cm^2

15.2.5 Steady-State Trap Recombination

For steady-state conditions, the net recombination/generation rate is identical for electrons (R_n) and holes (R_p) (i.e., instantaneous equilibrium). Using Equations 15-12 through 15-15 to give the values of f_t and following the derivation by Shockley and Read [294] and Hall [113], the Shockley-Read-Hall recombination/generation rate due to the defect states is given by:

$$R_{n,p} = \int_{E_V}^{E_C} \left(\frac{v_n\, v_p\, \mathtt{SIGTAE}\, \mathtt{SIGTAH}\, (n\, p - n_i^2)\, g_{TA}(E)}{v_n\, \mathtt{SIGTAE} \left(n + n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) + v_p\, \mathtt{SIGTAH} \left(p + n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right)} \right. \quad \text{15-16}$$

$$+ \frac{v_n\, v_p\, \mathtt{SIGTGAE}\, \mathtt{SIGGAH}\, (n\, p - n_i^2)\, g_{GA}(E)}{v_n\, \mathtt{SIGGAE} \left(n + n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) + v_p\, \mathtt{SIGGAH} \left(p + n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right)}$$

$$+ \frac{v_n\, v_p\, \mathtt{SIGTDE}\, \mathtt{SIGTDH}\, (n\, p - n_i^2)\, g_{TD}(E)}{v_n\, \mathtt{SIGTDE} \left(n + n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) + v_p\, \mathtt{SIGTDH} \left(p + n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right)}$$

$$\left. + \frac{v_n\, v_p\, \mathtt{SIGGDE}\, \mathtt{SIGGDH}\, (n\, p - n_i^2)\, g_{GD}(E)}{v_n\, \mathtt{SIGGDE} \left(n + n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) + v_p\, \mathtt{SIGGDH} \left(p + n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right)} \right) dE$$

15.2.6 Transient Traps

For the transient case, time is required for carriers to be emitted or captured and therefore instantaneous equilibrium cannot be assumed. This means that Equation 15-16 is no longer valid for transient simulations. Instead, the total recombination/generation rate for electrons (which is equal to electron recombination rate minus the generation rate for electrons) is calculated using the transient probabilities of occupation for acceptors (f_{tTA} and f_{tGA}). These are calculated by solving additional rate equations (Equations 15-17 and 15-18).

$$\frac{d}{dt}(p_{TA}) = \int_{E_V}^{E_C} g_{TA}(E) \left[v_n\ SIGTAE \left(n\, (1 - f_{t_{TA}}(E)) - f_{t_{TA}}(E)\, n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) \right. \quad \text{15-17}$$

$$\left. - v_p\ SIGTAH \left(p\, f_{t_{TA}}(E) - (1 - f_{t_{TA}}(E))\, n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right) \right] dE$$

$$\frac{d}{dt}(p_{GA}) = \int_{E_V}^{E_C} g_{GA}(E) \left[v_n\ \mathtt{SIGGAE} \left(n\, (1 - f_{t_{GA}}(E)) - f_{t_{GA}}(E)\, n_i \exp\left[\frac{E - E_i}{k\, T}\right]\right) \right. \quad \text{15-18}$$

$$\left. - v_p\ \mathtt{SIGGAH} \left(p\, f_{t_{GA}}(E) - (1 - f_{t_{GA}}(E))\, n_i \exp\left[\frac{E_i - E}{k\, T}\right]\right) \right] dE$$

The total hole recombination/generation rate can also be determined from the transient values of f_{tTD} and f_{tGD} (see Equations 15-19 and 15-20).

$$\frac{d}{dt}(n_{TD}) = \int_{E_V}^{E_C} g_{TD}(E)\left[v_p \ \texttt{SIGTDH}\left(p\ (1-f_{t_{TD}}(E)) - f_{t_{TD}}(E)\ n_i\ \exp\left[\frac{E_i-E}{k\ T}\right]\right)\right. \quad \text{15-19}$$

$$\left. - v_n \ \texttt{SIGTDE}\left(n\ f_{t_{TD}}(E) - (1-f_{t_{TD}}(E))\ n_i\ \exp\left[\frac{E-E_i}{k\ T}\right]\right)\right] dE$$

$$\frac{d}{dt}(n_{GD}) = \int_{E_V}^{E_C} g_{GD}(E)\left[v_p \ \texttt{SIGGDH}\left(p\ (1-f_{t_{GD}}(E)) - f_{t_{GD}}(E)\ n_i\ \exp\left[\frac{E_i-E}{k\ T}\right]\right)\right. \quad \text{15-20}$$

$$\left. - v_n \ \texttt{SIGGDE}\left(n\ f_{t_{GD}}(E) - (1-f_{t_{GD}}(E))\ n_i\ \exp\left[\frac{E-E_i}{k\ T}\right]\right)\right] dE$$

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. Specifying FAST on the **DEFECTS** statement will neglect the trap rate equation from the simulation.

Trap-Assisted Tunneling

Trap-Assisted Tunneling models the trap-to-band phonon-assisted tunneling effects for Dirac wells. At high electric fields, tunneling of electrons from the valence band to the conduction band through trap or defect states can have an important effect on the current.

Trap-assisted tunneling is modeled by including field-effect enhancement terms [133] (Γ_n^{DIRAC} and Γ_p^{DIRAC}) in the trap lifetimes in capture cross-sections. This model is enabled by specifying TRAP.TUNNEL in the **MODELS** statement.

The electron capture cross-section (SIGTAE, SIGGAE, and SIGTDE, and SIGGDE) are modified by including the electron field-effect term (Γ_n^{DIRAC}). For example, the electron capture cross-section for acceptor tail states (SIGTAE) becomes:

The field-effect enhancement term for electrons is given by:

$$\texttt{SIGTAE} \times (1 + \Gamma_n^{DIRAC}) \quad \text{15-21}$$

SIGGAE, SIGTDE, and SIGGDE are also modified this way.

$$\Gamma_n^{DIRAC} = \frac{\Delta En}{kT_L}\int_0^1 \exp\left(\frac{\Delta En}{KT_L}u - k_n u^{3/2}\right) du \quad \text{15-22}$$

While the field-effect enhancement term for hole is:

$$\Gamma_p^{DIRAC} = \frac{\Delta Ep}{kT_L}\int_0^1 \exp\left(\frac{\Delta En}{KT_L}u - k_p u^{3/2}\right)du \quad 15\text{-}23$$

The hole capture cross-sections (SIGTAH, SIGGAH, SIGTDH, and SIGGDH) are modified by including the hole field effect term (Γ_n^{DIRAC}). For example, SIGTAH now becomes:

$$\texttt{SIGTAH} \times 1 + \Gamma_n^{DIRAC} \quad 15\text{-}24$$

SIGGAH, SIGTDH, and SIGGDH are also modified this way.

where u is the integration variable, ΔE_n is the energy range where tunneling can occur for electrons, ΔE_p is the tunneling energy range for holes, and k_n and k_p are given by:

$$k_n = \frac{4}{3}\frac{\sqrt{2m_0\texttt{MASS.TUNNEL}\Delta En^3}}{3qh|E|} \quad 15\text{-}25$$

$$k_p = \frac{4}{3}\frac{\sqrt{2m_0\texttt{MASS.TUNNEL}\Delta Ep^3}}{3qh|E|} \quad 15\text{-}26$$

$\hbar$ is the reduced Planck's constant, m_0 is the rest mass of an electron, and MASS.TUNNEL is the relative effective mass (You can specify MASS.TUNNEL by setting the MASS.TUNNEL parameter in the **MODELS** statement).

Poole-Frenkel Barrier Lowering

The Poole Frenkel barrier lowering effect enhances the emission rate for trap-to-band phonon-assisted tunneling and pure thermal emissions at low electric fields. The Poole Frenkel effect occurs when the Coulombic potential barrier is lowered due to the electric field and only occurs in traps with a Coulombic potential (such as traps that are neutral when filled).

The Poole-Frenkel effect is modeled by including field-effect enhancement terms for Coulombic wells (Γ_n^{COUL} and Γ_p^{COUL}) and thermal emission (χ_F) [189] in the capture cross-sections. The model also includes the trap-assisted tunneling effects in the Dirac well. To enable this model, specify TRAP.COULOMBIC in the **MODELS** statement.

In the electron recombination/generation term (R_N), the Coulombic barrier lowering term (cF), and the electron Coulombic field-effect enhancement term (Γ_n^{COUL}) are applied to the electron capture cross-sections.

For example, SIGTAE now becomes:

$$\texttt{SIGTAE} \times (\chi_F + \Gamma_n^{COUL}) \quad 15\text{-}27$$

SIGGAE, SIGTDE, and SIGGDE are modified in the same manner.

The hole capture cross-section are modified by including the Dirac field-effect enhancement term for holes (Γ_n^{DIRAC}).

For example, SIGTAH now becomes:

$$\text{SIGTAE} \times (\chi_F + \Gamma_n^{COUL}) \quad \text{15-28}$$

SIGGAH, SIGTDH, and SIGGDH are modified in the same manner.

In the hole recombination/generation term (R_p), the Coulombic terms are applied to the hole capture cross-sections and the Dirac terms applied to the electron capture cross-sections.

The Poole-Frenkel thermal emission enhancement factor, χ_F, is defined as:

$$\chi_F = \text{A.TRAPCOULOMBIC} exp\left(\frac{\Delta E_{fp}}{kT_L}\right) \quad \text{15-29}$$

ΔEf_p is the barrier lowering term for a Coulombic well (see Equation 15-30).

$$\Delta E_{fp} = \sqrt{\frac{\text{B.TRAPCOULOMBIC} q^3 |E|}{\pi \varepsilon}} \quad \text{15-30}$$

The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_p^{COUL}, are defined as:

$$\Gamma_n^{COUL} = \frac{\Delta En}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} \exp\left(\frac{\Delta En}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u \Delta E_n}\right)^{5/3}\right]\right) du \quad \text{15-31}$$

$$\Gamma_p^{COUL} = \frac{\Delta E_p}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} \exp\left(\frac{\Delta E_p}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u \Delta E_p}\right)^{5/3}\right]\right) du \quad \text{15-32}$$

Table 15-3 User-Specificable Parameters for Equations 15-29 and 15-30

Statement	Parameter	Default	Units
MATERIAL	A.TRAPCOULOMBIC	1.0	
MATERIAL	B.TRAPCOULOMBIC	1.0	

15.2.7 Continuous Defects

If CONTINUOUS is specified in the **DEFECTS** statement, then the integral equations for the charge and recombination are evaluated using a numerical integral scheme [288]. In this case, the NUMA and NUMD (**DEFECTS** statement) parameters correspond to the number of acceptor and donor energy level intervals used in the integral.

15.2.8 Discrete Defects

If CONTINUOUS is not specified on the **DEFECTS** statement, the equation is modeled with discrete energy levels. The integrals terms in Equations 15-8 to 15-11 are replaced by summations, which run over the number of discrete energy levels (NUMA and NUMD). The acceptor and donor density of states terms are integrated separately. For example, the equation for the electron trap concentration (Equation 15-6) is replaced by:

$$n_T = \sum_{i=0}^{\mathrm{NUMA}} \left(f_{t_{TA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{TA}(E)\, dE + f_{t_{GA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{GA}(E)\, dE \right) \quad \text{15-33}$$

Table 15-4 Additional Parameters for the DEFECTS Statement

Statement	Parameter	Default	Units
DEFECTS	FAST	FALSE	
DEFECTS	CONTINUOUS	FALSE	
DEFECTS	NUMA	12	
DEFECTS	NUMD	12	

Syntax for a typical defect states definition is given below.

```
DEFECTS NTA=1.12E21 NTD=4.E20 WTA=0.025 WTD=0.05 \
     NGA=5.E17 NGD=1.5E18 EGA=0.4 EGD=0.4 \
     GA=0.1 WGD=0.1 SIGTAE=1.E-16 \
     SIGTAH=1.E-14 SIGTDE=1.E-14 \
     SIGTDH=1.E-16 SIGGAE=1.E-16 SIGGAH=1.E-14 \
     SIGGDE=1.E-14 SIGGDH=1.E-16
```

Figure 15-1 shows how the syntax is used to define the peak density of states and distribution widths for the two tail states and two Gaussian distributions.

15.2.9 Amphoteric Defects

In hydrogenated amorphous silicon (a-Si:H), the effect of dangling-bonds states on recombination can be significant. The dangling-bond states are amphoteric and located around the middle of the bandgap. If you specify the AMPHOTERIC parameter on the **DEFECTS** or **INTDEFECTS** statements, the dangling-bond density of states for a-Si:H will be calculated using the defect-pool model [250, 251] as will the amphoteric defect density and recombination [320].

The amphoteric defects are given by:

$$n^{+}_{AMP} = D(E) f^{+}_{AMP}(E) \quad \text{15-34}$$

$$n^{0}_{AMP} = D(E) f^{0}_{AMP}(E) \quad \text{15-35}$$

$$n^{-}_{AMP} = D(E) f^{-}_{AMP}(E) \quad \text{15-36}$$

where:

- $n^{+}_{AMP}(E)$ is the occupied density of the positive amphoteric traps.
- $n^{0}_{AMP}(E)$ is the occupied density of the neutral amphoteric traps.
- $n^{-}_{AMP}(E)$ is the occupied density of the negative amphoteric traps.
- $f^{+}_{AMP}(E)$ is the probability of occupation of the positive amphoteric traps,
- $f^{0}_{AMP}(E)$ is the probability of occupation of the neutral amphoteric traps,
- $f^{-}_{AMP}(E)$ is the probability of occupation of the negative amphoteric traps.
- $D(E)$ is the dangling-bond density of states given by Equation 15-38.

The total charge caused by the presence of amphoteric traps is subtracted from the right hand side of Poisson's equation. The total charge value is defined by:

$$Q_{TAMP} = q \int_{E_V}^{E_C} (n^{+}_{AMP}(E) - n^{-}_{AMP}(E)) dE \quad \text{15-37}$$

$$D(E) = \mathrm{NV0.AMPHOTERIC}\left(\frac{\mathrm{HCONC.AMPHOTERIC}}{\mathrm{NSISI.AMPHOTERIC}}\right)^{\frac{k\mathrm{T0.AMPHOTERIC}}{4\mathrm{EV0.AMPHOTERIC}}} \quad 15\text{-}38$$

$$\cdot\left(\frac{2\mathrm{EV0.AMPHOTERIC}^2}{2\mathrm{EV0.AMPHOTERIC} - k\mathrm{T0.AMPHOTERIC}}\right)$$

$$exp\left[-\frac{1}{2\mathrm{EV0.AMPHOTERIC}}\left(\mathrm{EP.AMPHOTERIC} - E_V - \frac{\mathrm{SIGMA.AMPHOTERIC}^2}{4\mathrm{EV0.AMPHOTERIC}}\right)\right]$$

$$\cdot\left(\frac{2}{f^0_{AMP_0}(E)}\right)^{\frac{k\mathrm{T0.AMPHOTERIC}}{2\mathrm{EV0.AMPHOTERIC}}} \cdot \frac{1}{\sqrt{2\pi\mathrm{SIGMA.AMPHOTERIC}^2}}$$

$$\cdot exp\left[-\left(\frac{E + \frac{\mathrm{SIGMA.AMPHOTERIC}^2}{2\mathrm{EV0.AMPHOTERIC}} - \mathrm{EP.AMPHOTERIC}}{2\mathrm{SIGMA.AMPHOTERIC}^2}\right)^2\right]$$

where:

- EP.AMPHOTERIC is the most probable potential defect energy.
- E_V is the valence band edge.
- EV0.AMPHOTERIC is the characteristic energy.
- HCONC.AMPHOTERIC is the density of hydrogen.
- NSISI.AMPHOTERIC is the density of Si-Si bonds.
- NV0.AMPHOTERIC is the density of states of the valance-band tail exponential region extrapolated to the valence-band edge.
- SIGMA.AMPHOTERIC is the defect pool width.
- T0.AMPHOTERIC is the freeze-in temperature.
- $f^0_{AMP_0}(E)$ is the equilibrium probability of occupation of the neutral amphoteric traps.

The probabilities of occupation can be calculated from the equilibrium probabilities of occupation ($f^+_{AMP_0}(E)$, $f^0_{AMP_0}(E)$, and $f^-_{AMP_0}(E)$) and the capture and emission rates:

$$f^+_{AMP}(E) = \frac{1}{1 + \left[\frac{e_p^+ + n\nu_n\mathrm{SIGNP.AMPHOTERIC}}{e_n^0 + p\nu_p\mathrm{SIGP0.AMPHOTERIC}}\right]\left[1 + \frac{e_p^0 + n\nu_n\mathrm{SIGN0.AMPHOTERIC}}{e_n^- + p\nu_p\mathrm{SIGPN.AMPHOTERIC}}\right]} \quad 15\text{-}39$$

$$f^0_{AMP}(E) = \frac{1}{1 + \left[\frac{e^0_n + p\nu_p \texttt{SIGP0.AMPHOTERIC}}{e^+_p + n\nu_n \texttt{SIGNP.AMPHOTERIC}}\right] + \left[\frac{e^0_p + n\nu_n \texttt{SIGN0.AMPHOTERIC}}{e^-_n + p\nu_p \texttt{SIGPN.AMPHOTERIC}}\right]} \quad (15\text{-}40)$$

$$f^-_{AMP}(E) = 1 - f^0_{AMP}(E) - f^+_{AMP}(E) \quad (15\text{-}41)$$

where:

- SIGN0.AMPHOTERIC is the electron capture cross-section for neutral defects.
- SIGNP.AMPHOTERIC is the electron capture cross-section for positive defects.
- SIGP0.AMPHOTERIC is the hole capture cross-section for neutral defects.
- SIGPN.AMPHOTERIC is the hole capture cross-section for negative defects.
- e^0_n is the electron emission coefficient for neutral defects.
- e^-_n is the electron emission coefficient for negative defects.
- e^+_p is the hole emission coefficient for positive defects.
- e^0_p is the hole emission coefficient for neutral defects.

$$e^0_n = \frac{n_0 f^+_{AMP_0}(E)\nu_n \texttt{SIGNP.AMPHOTERIC}}{f^0_{AMP_0}(E)} \quad (15\text{-}42)$$

$$e^-_n = \frac{n_0 f^0_{AMP_0}(E)\nu_n \texttt{SIGN0.AMPHOTERIC}}{f^-_{AMP_0}(E)} \quad (15\text{-}43)$$

$$e^0_p = \frac{p_0 f^-_{AMP_0}(E)\nu_p \texttt{SIGPN.AMPHOTERIC}}{f^0_{AMP_0}(E)} \quad (15\text{-}44)$$

$$e^+_p = \frac{p_0 f^0_{AMP_0}(E)\nu_p \texttt{SIGP0.AMPHOTERIC}}{f^+_{AMP_0}(E)} \quad (15\text{-}45)$$

$$f^+_{AMP_0}(E) = \frac{1}{1 + 2\exp\left[\frac{E_F - E_T}{kT}\right] + \exp\left[\frac{2E_F - 2E_T - \texttt{EU.AMPHOTERIC}}{kT}\right]} \quad (15\text{-}46)$$

$$f^0_{AMP_0}(E) = \frac{2\exp\left[\frac{E_F - E_T}{kT}\right]}{1 + 2\exp\left[\frac{E_F - E_T}{kT}\right] + \exp\left[\frac{2E_F - 2E_T - \texttt{EU.AMPHOTERIC}}{kT}\right]} \quad (15\text{-}47)$$

$$f^{-}_{AMP_0}(E) = 1 - f^{+}_{AMP_0}(E) - f^{0}_{AMP_0}(E) \quad \text{15-48}$$

where:

- n_0 is the equilibrium electron concentration.
- p_0 is the equilibrium hole0 concentration.
- E_F is the Fermi energy.
- E_T is the trap energy.
- EU.AMPHOTERIC is the defect electron correlation energy.

The recombination terms are given by:

$$R^{+}_{AMP} = D(E)[n\nu_n \texttt{SIGNP.AMPHOTERIC} f^{+}_{AMP}(E) - e^{0}_{n} f^{0}_{AMP}(E)] \quad \text{15-49}$$

$$R^{-}_{AMP} = D(E)[n\nu_n \texttt{SIGN0.AMPHOTERIC} f^{0}_{AMP}(E) - e^{-}_{n} f^{-}_{AMP}(E)] \quad \text{15-50}$$

Table 15-5 User-Specifiable Parameters for Equations 15-34-15-50

Statement	Parameter	Default	Units
DEFECTS (INTDEFECTS)	EP.AMPHOTERIC	1.27	eV
DEFECTS (INTDEFECTS)	EU.AMPHOTERIC	0.2	eV
DEFECTS (INTDEFECTS)	EV0.AMPHOTERIC	0.056	eV
DEFECTS (INTDEFECTS)	HCONC.AMPHOTERIC	5×10^{21} (5×10^{17})	cm^{-3} (cm^{-2})
DEFECTS (INTDEFECTS)	NSISI.AMPHOTERIC	2×10^{23} (2×10^{19})	cm^{-3} (cm^{-2})
DEFECTS (INTDEFECTS)	NV0.AMPHOTERIC	NV300	cm^{-3} (cm^{-2})
DEFECTS (INTDEFECTS)	SIGMA.AMPHOTERIC	0.19	eV
DEFECTS (INTDEFECTS)	SIGN0.AMPHOTERIC	1e-16	cm^2
DEFECTS (INTDEFECTS)	SIGNP.AMPHOTERIC	1e-16	cm^2
DEFECTS (INTDEFECTS)	SIGP0.AMPHOTERIC	1e-16	cm^2
DEFECTS (INTDEFECTS)	SIGPN.AMPHOTERIC	1e-16	cm^2
DEFECTS (INTDEFECTS)	T0.AMPHOTERIC	300	K

15.2.10 Amphoteric Defect Generation

This model calculates the dangling bond density of amphoteric defects as a function of bias stress time. If the parameter TIME.EP on the **MODELS** statement is specified, then the value of EP.AMPHOTERIC in the amphoteric defect model will be replaced by *Ep*:

$$Ep = \mathrm{EP.AMPHOTERIC}\ \exp\left(-\left(\frac{\mathrm{TIME.EP}}{\mathrm{EPT0.AMPHOTERIC}}\right)^{\mathrm{EPBETA.AMPHOTERIC}}\right) \quad 15\text{-}51$$

where *Ep* is the stress time dependent most probable potential defect energy.

Table 15-6 User-Specifiable Parameters for Equation 15-51

Statement	Parameter	Type	Default	Units
MODELS	TIME.EP	Real	0	s
DEFECTS	EPT0.AMPHOTERIC	Real	1.3e6	s
DEFECTS	EPBETA.AMPHOTERIC	Real	0.5	

15.2.11 Light Induced Defect Generation

This model calculates the dangling bond density of amphoteric defects as a function of light [211]. The model is enabled by specifying TRAP.LID on the **MODELS** statement. The total dangling bond density of amphoteric defects (N_{AMP}) is given by:

$$N_{AMP} = \left[3G^2\mathrm{TIME.LID}\left(\frac{\mathrm{KD.LID} - 2\mathrm{KH.LID}}{\mathrm{SIG.LID}^2}\right) + N_{AMP_0}\right] \quad 15\text{-}52$$

where

- N_{AMP_0} is the initial total dangling bond density.
- G is the photogeneration rate.
- KD.LID is the light induced SiHD creation constant.
- KH.LID is the dissociation constant of a hydrogen atom from a SiHD defect.
- SIG.LID is the capture coefficient of free carriers by dangling bonds.
- TIME.LID is the time values used in the light induced defect generation model.

By default, the photogeneration rate (G) is calculated from all active **BEAM** statements. If you specify the BEAM.LID parameter on the **MODELS** statement, then only active **BEAM** statements that were defined with the parameter BEAM.LID will be used in the photogeneration rate calculation. If you specify the parameter GENRATE.LID parameter on the **MODELS** statement, then this value will be used as the photogeneration rate in Equation 15-52.

Table 15-7 User-Specifiable Parameters for Equation 15-52				
Statement	**Parameter**	**Type**	**Default**	**Units**
MODELS	TIME.LID	Real	0	s
MATERIAL	KD.LID	Real	5.0e-15	$cm^3\ s^{-1}$
MATERIAL	KH.LID	Real	1.0e-18	$cm^3\ s^{-1}$
MATERIAL	SIG.LID	Real	1.0e-8	$cm^3\ s^{-1}$

15.2.12 Plotting The Density Of States Versus Energy

The DFILE and AFILE parameters were used to allow you to specify output file names for capture of defect densities as a function of energy for donor states and acceptor states, respectively. For example, if you want to look at the donor and acceptor defect distributions, specify the following line:

```
DEFECTS DFILE=donors AFILE=acceptors
```

The files, donors and acceptors, could then be loaded into TonyPlot to look at the distributions of donor and acceptor defects as a function of energy.

15.2.13 Using the C-Interpreter to define DEFECTS

The C-Interpreter can be used to define the defect states in the bandgap. The F.TFTDON and F.TFTACC parameters of the **DEFECTS** statement indicate the filenames containing the C functions. For more information on using the C-Interpreter, See Appendix A "C-Interpreter Functions".

```
DEFECTS F.TFTDON=mydefects.c F.TFTACC=mydefects.c
```

The file, mydefects.c, will contain C functions for donor and acceptor defect densities as a function of energy. These user defined defects are added to the existing defect distribution. If you want to use only your own function, set the gaussian and tail functions to zero. The following example defect states are defined in the file, tft.lib. These are added to a zero background set using the tail and gaussian state syntax. The resultant distribution of defects versus energy can be plotted in the files, don.dat and acc.dat.

```
DEFECTS F.TFTDON=tft.lib F.TFTACC=tft.lib DFILE=don.dat
AFILE=acc.dat \
NTA=0 NTD=0 WTA=1.0 WTD=1.0 \
NGA=0 NGD=0 EGA=0.6 EGD=0.6 WGA=1 WGD=1 \
SIGTAE=1.E-16 SIGTAH=1.E-14 SIGTDE=1.E-14 SIGTDH=1.E-16
SIGGAE=1.E-16 SIGGAH=1.E-14 SIGGDE=1.E-14 SIGGDH=1.E-16
```

15.2.14 Setting Mobility and Other Models

TFT uses adaptations of the standard models of S-Pisces or Blaze. The example below shows how to select the models and material parameters for polysilicon.

```
MATERIAL MUN=300 MUP=30

MODELS SRH
```

Typical mobility values for amorphous silicon can be set by:

```
MATERIAL MUN=20 MUP=1.5
```

Other models are also available in TFT. These include impact ionization and tunneling. Set them by using:

```
MODELS BBT.STD IMPACT
```

Note: Don't use concentration dependent mobility models (CONMOB, ANALYTIC, ARORA, KLA) because they overwrite the low field mobilities set in the **MATERIAL** statement.

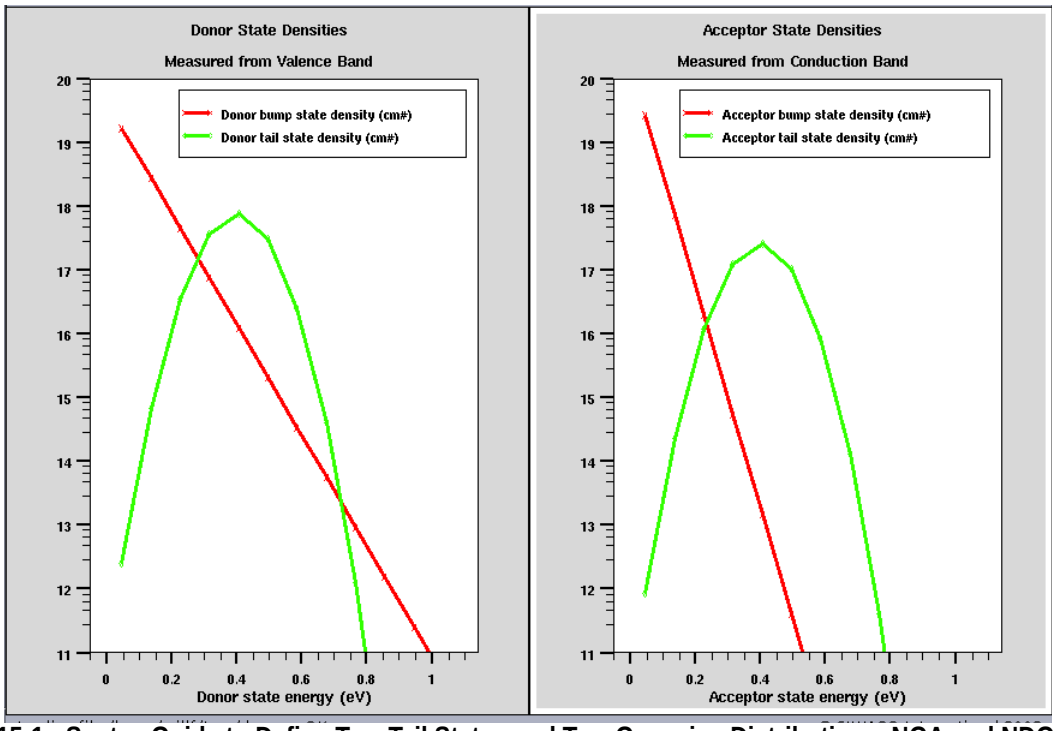


Figure 15-1: Syntax Guide to Define Two Tail States and Two Gaussian Distributions. NGA and NDG are the integrated values of the Gaussian distributions. Gaussians are entered on energies EGA and EGD respectively.

Chapter 16
Organic Display and Organic Solar: Organic Simulators

16.1 Organic Device Simulation

Organic Device simulation in the Atlas framework is supported by two products: Organic Display and Organic Solar. As implied by the names, these simulators are targeted at specific applications requiring specific physical features. In the case of Organic Display, the simulator is set up to perform analysis of organic based active displays. This entails simulation of principally organic FET devices, OFETs, and organic light emitting devices, OLEDs. The Organic Solar simulator is specific to light sensitive devices, particularly solar cells. Organic based imaging sensors and the like may also be simulated using Organic Solar.

The following sections describe each of the simulators in more detail. This is followed by details of the physical models used in each case.

16.1.1 Organic Display

Organic Display allows simulation of complex organic light emitting devices and circuits. Much of this capability is discussed elsewhere in this manual. Before reading this section, you should already be familiar with the general concepts discussed in Chapters 2 “Getting Started with Atlas” and 3 “Physics”. Chapter 2: “Getting Started with Atlas” gives an overview of the general concepts of building a simulation deck and using the simulator. Chapter 3: “Physics” gives a fairly detailed discussion of the basic physical concepts and models that define device modeling.

If you are interested in simulation of organic thin film transistors, OTFTs, then also read Chapter 15: “TFT: Thin-Film Transistor Simulator”. The drift diffusion concepts discussed in that chapter are completely anologous to those involved in simulation of OTFTs. Also if you are interested in embedding physical devices in lumped element (SPICE) circuits, then read Chapter 13: “MixedMode: Mixed Circuit and Device Simulator”. If you are interested in simulating organic light emitting devices, OLEDs, then also read Chapter 12: “LED: Light Emitting Diode Simulator”. This chapter discusses the physical concepts for simulation of light emitting devices in general including OLEDs. Finally, those interested may also want to look over Chapter 21 “Numerical Techniques” where they will be given a description of the workings of the numerics of the simulator including information on convergence.

In this section, we will lead you through the important differences that you will need to know about when simulating organic display devices. We will do this in a sequential manner similar to the description of deck construction presented in Chapter 2: “Getting Started with Atlas”.

First, with respect to definition of the device structure, there is little difference with what is described in Chapter 2: “Getting Started with Atlas”. The main thing you should be aware of is that there are several default material names that are recognized by Organic Display. These materials are `Organic`, `Pentacene`, `Alq3`, `TPD`, `PPV`, `Tetracene`, and `NPB`. For insulating dielectrics and conductive materials, see Appendix B “Material Systems”.

If you want to simulate a material not included in the default set recognized by Organic Display, you should follow the steps described in Section B.2.5 “Specifying Unknown or User-Defined Materials in Atlas”.

After you have specified the device structure, you may decide how to handle interfaces between homogenous layers. These interfaces are analogous to heterojunctions in the drift diffusion formulation where the band edges are replaced by the HOMO and LUMO.

You may choose to make the interfaces obey the physics of thermionic emission and tunneling. To do this, specify the **INTERFACE** statement with the parameters THERMIONIC and TUNNEL set. There are various ways in which characteristics may be assigned to a particular interface geometrically using principally the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. These types of interfaces are discussed in more detail in Section 6.2.2 "The Thermionic Emission and Field Emission Transport Model".

Next, you may want to fix the workfunction of electrodes to a certain value relative to the LUMO and HOMO. To do this, use the **CONTACT** statement where the workfunction is specified with the WORKF parameter. A particular **CONTACT** statement is associated with an electrode using the NAME or NUMBER parameters. Also, you can assign various Schottky characteristics to the electrode using, for example, the SURF.REC parameter to enable surface recombination or BARRIER to enable barrier lowering. For more details, see Section 22.6 "CONTACT".

Key parameters in the simulation of organic devices are the energies of the LUMO and HOMO. In Organic Display, these are set (indirectly) on the **MATERIAL** statement using the EG300 and AFFINITY parameters. The LUMO energy coresponds to the negative of the value specified by the AFFINITY parameter. The HOMO energy coresponds to the negative of the sum of the values of the AFFINITY and EG300 parameters.

Defect states can be specified in organic materials using the **ODEFECTS** and **OINTDEFECTS** statements. These statements are described in Section 16.2.2 "Organic Defects Model". The organic defects are important to correctly model defect recombination (Equation 16-40), hopping mobility (Equations 16-43 and 16-44), and dopant light emission rates (see Section 16.3.1 "Dopants").

You should also enable Poole-Frenkel mobility modeling in organic devices by specifying PFMOB on the MODEL statement. This should be done in conjunction with the Langevin recombination model enabled by the LANGEVIN parameter of the **MODELS** statement. Langevin recombination is needed to enable exchange between charged carriers and singlet and triplet excitons as described in Equations 16-63 and 16-106. To enable the singlet and triplet exciton rate equations, specify SINGLET and TRIPLET on the **MODELS** statement.

Radiative emission can come from two sources: the host material singlets and the dopants. The dopant rate equation is given in Equation 16-108 and is enabled by specifying DOPANT on the ODEFECT statement. Organic Display cannot predict the output spectra so you must supply it. These are in the form of tabular descriptions of intensity vs. wavelength as described in Section 12.8.1 "User-Definable Emission Spectra". To connect materials and their emission spectra, assign the USER.SPECT and DOPE.SPECT parameters of the **MATERIAL** statement to the file names containing the appropriate tables. The spectra need not be normalized as this is done automatically. To output the spectra to TonyPlot compatible log files, specify OUT.USPEC and OUT.DSPEC.

The normalized emission spectra are scaled by the radiative emission rates given in Equations 16-63 and 16-108. The PHEFF.EXCITON and DPHEFF.EXCITON parameters are quantum efficiencies for host and dopant emission and act to scale the emission rates. The emission spectra are critical in calculation through reverse ray tracing (see Section 12.6 "Reverse Ray-Tracing") of output coupling and output emisson spectrum. Also for reverse ray tracing, you should accurately specify the complex indices of refraction for the various materials. This can be done as described in Section 11.8 "Defining Optical Properties of Materials".

Finally, a variety of useful information can be output after each solution using the LED statement (see Section 12.6 “Reverse Ray-Tracing” and Section 22.27 “LED”). These information include the angular distribution of emission spectra and output coupling.

16.1.2 Organic Solar

Organic Solar allows simulation of complex organic light sensitive devices such as solar cells and imaging sensors. Much of this capability is discussed elsewhere in this manual. Before reading this section, you should already be familiar with the general concepts discussed in Chapters 2 “Getting Started with Atlas” and 3 “Physics”. Chapter 2: “Getting Started with Atlas” gives an overview of the general concepts of building a simulation deck and using the simulator. Chapter 3: “Physics” gives a fairly detailed discussion of the base physical concepts and models that define device modeling.

To understand the general concepts of photdetection, read Chapter 11: “Luminous: Optoelectronic Simulator”. If you are interested in embedding physical devices in lumped element (SPICE) circuits, then read Chapter 13: “MixedMode: Mixed Circuit and Device Simulator”. Those interested may also want to look over Chapter 21 “Numerical Techniques” where they will be given a description of the workings of the numerics of the simulator including information on convergence.

In this section, we will lead you through the important differences you will need to know about when simulating organic display devices. We will do this in a sequential manner similar to the description of deck construction presented in Chapter 2: “Getting Started with Atlas”.

First with respect to definition of the device structure, there is little difference with what is described in Chapter 2: “Getting Started with Atlas”. You should be aware of is that there are several default material names that are recognized by Organic Display. These materials are Organic, Pentacene, Alq3, TPD, PPV, Tetracene, and NPB. For insulating dielectrics and conductive materials, see Appendix B “Material Systems”.

If you want to simulate a material not included in the default set recognized by Organic Solar, you should follow the steps described in Section B.2.5 “Specifying Unknown or User-Defined Materials in Atlas”.

After you have specified the device structure, you may decide how to handle interfaces between homogenous layers. These interfaces are analogous to heterojunctions in the drift diffusion formulation where the band edges are replaced by HOMO and LUMO.

As such, you may choose to make the interfaces obey the physics of thermionic emission and tunneling. To do this, specify the **INTERFACE** statement with the parameters THERMIONIC and TUNNEL set. There are various ways in which interfaces may be specified to a particular interface using principally the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. These types of interfaces are discussed in more detail in Section 6.2.2 “The Thermionic Emission and Field Emission Transport Model”.

Next, you may want to fix the workfunction of electrodes to a certain value relative to the LUMO and HOMO. To do this, use the **CONTACT** statement where the workfunction is specified with the WORKF parameter. A particular **CONTACT** statement is associated with an electrode using the NAME or NUMBER parameters. Also, you can assign various Schottky characteristics to the electrode using, for example, the SURF.REC parameter to enable surface recombination or BARRIER to enable barrier lowering. For more details, see Section 22.6 “CONTACT”.

Key parameters in the simulation of organic devices are the energies of the LUMO and HOMO. In Organic Solar, these are set (indirectly) on the **MATERIAL** statement using the EG300 and AFFINITY parameters. The LUMO energy coresponds to the negative of the value specified by the AFFINITY parameter. The HOMO energy coresponds to the negative of the sum of the values of the AFFINITY and EG300 parameters.

Defect states can be specified in organic materials using the **ODEFECTS** and **OINTDEFECTS** statements. These statements are described in Section 16.2.2 “Organic Defects Model”. The organic defects are important to correctly model defect recombination (Equation 16-40), hopping mobility (Equations 16-43 and 16-44), and dopant light emission rates (see Section 16.3.1 “Dopants”).

You should also enable Poole-Frenkel mobility modeling in organic devices by specifying PFMOB on the **MODELS** stateement. This should be done in conjunction with the Langevin recombination model enabled by the LANGEVIN parameter of the **MODELS** statement. Langevin recombination is needed to enable exchange between charged carriers and singlet and triplet excitons as described in Equations 16-63 and 16-106. To enable the singlet and triplet exciton rate equations, specify SINGLET and TRIPLET on the **MODELS** statement.

Photodetection in organic photo detectors is a two part process. First, absorbed photons generate singlet excitons. The QE.EXCITON parameter of the **MATERIAL** statement (see Section 16.3.2 “Light Generation of Excitons”) specifies the efficiency of this conversion. Then, excitons are generated and they diffuse to interfaces in the "blend" material where they experience dissociation as described in Section 16.3.3 “Exciton Dissociation”. Once electrons and holes dissociate from the singlets, they experience built-in electric fields. They are then separated and detected at the electrodes. To enble dissociation, specify S.DISSOC on the **MODELS** statement.

16.2 Organic Materials Physical Models

This section describes the model definition required for the simulation of organic devices.

16.2.1 Gaussian Band Structure

In organic semiconductors the density of states in the conduction (LUMO) and valence (HOMO) levels are often described by a Gaussian function [326].

$$g_C(E-E_C) = \frac{\mathtt{NTC.GAUSS}}{\sqrt{2\pi}\mathtt{SIGC.GAUSS}}\exp\left(-\frac{(E-E_C)^2}{2\mathtt{SIGC.GAUSS}^2}\right) + \frac{\mathtt{GNTC.GAUSS}}{\sqrt{2\pi}\mathtt{GSIGC.GAUSS}}\exp\left(-\frac{((E-E_C)+\mathtt{GDEC.GAUSS})^2}{2\mathtt{GSIGC.GAUSS}^2}\right) + \frac{\mathtt{G2NTC.GAUSS}}{\sqrt{2\pi}\mathtt{G2SIGC.GAUSS}}\exp\left(-\frac{((E-E_C)+\mathtt{G2DEC.GAUSS})^2}{2\mathtt{G2SIGC.GAUSS}^2}\right) \quad (16\text{-}1)$$

$$g_V(E_V-E) = \frac{\mathtt{NTV.GAUSS}}{\sqrt{2\pi}\mathtt{SIGV.GAUSS}}\exp\left(-\frac{(E_V-E)^2}{2\mathtt{SIGV.GAUSS}^2}\right) + \frac{\mathtt{GNTV.GAUSS}}{\sqrt{2\pi}\mathtt{GSIGV.GAUSS}}\exp\left(-\frac{((E_V-E)+\mathtt{GDEV.GAUSS})^2}{2\mathtt{GSIGV.GAUSS}^2}\right) + \frac{\mathtt{G2NTV.GAUSS}}{\sqrt{2\pi}\mathtt{G2SIGV.GAUSS}}\exp\left(-\frac{((E_V-E)+\mathtt{G2DEV.GAUSS})^2}{2\mathtt{G2SIGV.GAUSS}^2}\right) \quad (16\text{-}2)$$

E_C and E_V are, respectively, the centers of the conduction and valence Gaussian DOS. Using the Fermi-Dirac occupation probability, Equation 3-25, the electron density in the conduction band and the hole density in the valence band are given by

$$n(E_F) = \int_{-\infty}^{\infty} g_C(E-E_C)f(E-E_F)dE \quad (16\text{-}3)$$

$$p(E_F) = \int_{-\infty}^{\infty} g_V(E_V-E)(1-f(E-E_F))dE \quad (16\text{-}4)$$

These equations are written in the same form as Equations 3-27 and 3-28.

$$n(E_F) = N_C G_C\left(\frac{E_F-E_C}{k_BT}\right) \quad (16\text{-}5)$$

$$p(E_F) = N_V G_V\left(\frac{E_V-E_F}{k_BT}\right) \quad (16\text{-}6)$$

Here, N_C and N_V are effective band-edge density of states and G_C and G_V are Gaussian equivalents of the Fermi-Dirac integral. G_C and G_V have to be evaluated numerically.

An equilibrium intrinsic carrier density and intrinsic Fermi level can be found for the Gaussian band structure where

$$n(E_{Fi}) = p(E_{Fi}) = n_i \quad (16\text{-}7)$$

The effective band-edge density of states are chosen so that Equations 3-34 and 3-35 are still valid. The parameters to define a Gaussian DOS are in Table 16-1.

Table 16-1 Guassian DOS Parameters

Statement	Parameter	Default	Units
MATERIAL	NTC.GAUSS	NTV.GAUSS or 1.0e21	cm^{-3}
MATERIAL	SIGC.GAUSS	SIGV.GAUSS or 0.1	eV
MATERIAL	GNTC.GAUSS	0.1×NTC.GAUSS	cm^{-3}
MATERIAL	GSIGC.GAUSS	SIGC.GAUSS	eV
MATERIAL	GDEC.GAUSS	3×(SIGC.GAUSS+GSIGC.GAUSS)	eV
MATERIAL	G2NTC.GAUSS	0.1×NTC.GAUSS	cm^{-3}
MATERIAL	G2SIGC.GAUSS	SIGC.GAUSS	eV
MATERIAL	G2DEC.GAUSS	3×(SIGC.GAUSS+G2SIGC.GAUSS)	eV
MATERIAL	G2NTV.GAUSS	0.1×NTV.GAUSS	cm^{-3}
MATERIAL	G2SIGV.GAUSS	SIGV.GAUSS	eV
MATERIAL	G2DEV.GAUSS	3×(SIGV.GAUSS+G2SIGV.GAUSS)	eV
MATERIAL	NTV.GAUSS	NTC.GAUSS or 1.0e21	cm^{-3}
MATERIAL	SIGV.GAUSS	SIGC.GAUSS or 0.1	eV
MATERIAL	GNTV.GAUSS	0.1×NTV.GAUSS	cm^{-3}
MATERIAL	GSIGV.GAUSS	SIGV.GAUSS	eV
MATERIAL	GDEV.GAUSS	3×(SIGV.GAUSS+GSIGV.GAUSS)	eV
MATERIAL	CRELMIN.N	1e-10	
MATERIAL	CRELMIN.P	1e-10	
MATERIAL	CRELMAX.N	0.5	
MATERIAL	CRELMAX.P	0.5	
MATERIAL	G3MAX.N	100.0	
MATERIAL	G3MAX.P	100.0	
MATERIAL	PROFILE.GAUSS		

To use the Gaussian band structure model, either specify NTC.GAUSS, SIGC.GAUSS, NTV.GAUSS, or SIGC.GAUSS on a MATERIAL statement.

If a Gaussian DOS is used for one band, then it will automatically be used for the other band. If a parameter is defined for one band, then that value becomes the default, if the

corresponding parameter is not explicitly set, for the other band. For example, setting the conduction band to

```
MATERIAL NTC.GAUSS = 2.0e22 SIGC.GAUSS = 0.15
```

will have Nt = 2.0e22 and σ = 0.15 for a Gaussian valence band as well.

Only a single parameter is required to activate a Gaussian band structure. For example

```
MATERIAL NTC.GAUSS = 2.0e22
```

The numerical defaults are used if a parameter is not explicitly given for either band.

Two guest Gaussians for each band are optional and can only be present if a host Gaussian has been defined. Each guest Gaussian has three parameters: the concentration NT, the width SIG, and the energy between the center of the host and the guest DE. At least one of these parameters must be defined for the guest DOS to be added. Adding a guest to one band does not automatically add it to the other band. The guest Gaussian is not automatically ionized. If the guest is to act as a doping, this concentration must be supplied as well. For example

```
SET gntc_gauss=1e20
REGION NUMBER=1 MATERIAL=organic DONOR=$gntc_gauss
MATERIAL NUMBER=1 NTC.GAUSS=1e21 SIGC.GAUSS=0.1 \
GNTC.GAUSS=$gntc_gauss GSIGC.GAUSS=0.1 GDEC.GAUSS=0.5
```

The guest DOS is defined on the **MATERIAL** statement, but the resultant doping is defined with "DONOR=$gntc_gauss" on the **REGION** statement.

The PROFILE.GAUSS parameter defines a filename to save a Gaussian band structure profile to. This shows DOS as a function of energy and carrier density as a function of Fermi energy. This file is saved when the band structure is initialized during a "**SOLVE** INIT".

In the Drift-Diffusion model, the diffusion coefficient is

$$D = \frac{\mu}{q} \frac{n}{dn/dE_F} \tag{16-8}$$

With Parabolic band structure and Boltzmann statistics

$$\frac{dn}{dE_F} = \frac{n}{k_B T} \tag{16-9}$$

and we get the Einstein diffusion coefficient

$$D_{Einstein} = \frac{k_B T \mu}{q} \tag{16-10}$$

With Gaussian bands, we write the diffusion coefficent as

$$D = g_3 D_{Einstein} \tag{16-11}$$

with

$$g_3, n = \frac{1}{k_B T} \frac{n}{dn/dE_F} \tag{16-12}$$

and

$$g_3, p = \frac{1}{k_B T} \frac{p}{dp/dE_F} \quad (16\text{-}13)$$

These equations hold for

$$\texttt{CRELMIN.N} \le \frac{n}{\texttt{NTC.GAUSS}} \le \texttt{CRELMAX.N} \quad (16\text{-}14)$$

and

$$\texttt{CRELMIN.P} \le \frac{p}{\texttt{NTV.GAUSS}} \le \texttt{CRELMAX.P} \quad (16\text{-}15)$$

If the normalized carrier density is outside these limits, then g_3 is evaluated at the appropriate limit. `G3MAX.[NP]` is the maximum value that g_3 is allowed to have.

Analytical Approximation for the Gauss-Fermi Integral

By default the Gauss-Fermi integral

$$n(E_F) = \int_{-\infty}^{\infty} g_C(E - E_C) f(E - E_F) dE \quad (16\text{-}16)$$

is evaluated by Gauss-Hermite integration. Paasch and Scheinert, in [230], give an analytic expression for the evaluation of the integral. Their expression is

$$n(E_F) = N_0 G(\zeta; \hat{s}) \quad (16\text{-}17)$$

where ζ is the normalized energy

$$\zeta = \frac{E_F - E_C}{kT} \quad (16\text{-}18)$$

and $\hat{s}$ is the normalized width of the Gaussian DOS

$$\hat{s} = \frac{\sigma}{kT} \quad (16\text{-}19)$$

There are two regions $G(\zeta; \hat{s})$ for $\zeta \le -\hat{s}^2$

$$G(\zeta; \hat{s}) = \exp\left(\frac{\hat{s}^2}{2} + \zeta\right) \frac{1}{\exp[K(\hat{s}^2)(\zeta + \hat{s}^2)] + 1} \quad (16\text{-}20)$$

and for $\zeta \ge -\hat{s}^2$

$$G(\zeta; \hat{s}) = \frac{1}{2} efrc\left(-\frac{\zeta}{\hat{s}\sqrt{2}} H(\hat{s})\right) \quad (16\text{-}21)$$

The factor $H(\hat{s})$ is given by

$$H(\hat{s}) = \frac{\sqrt{2}}{\hat{s}} efrc^{-1}\left[\exp\left(-\frac{\hat{s}^2}{2}\right)\right] \quad (16\text{-}22)$$

and over the range $2 \le \hat{s} \le 8$ is approximated by the polynomial

$$H(\hat{s}) = 1.02964 - \frac{0.50216}{\hat{s}} - \frac{0.12889}{\hat{s}^2} \quad 16\text{-}23$$

The factor $K(\hat{s})$ is given by

$$K(\hat{s}) = 2\left\{1 - \frac{H(\hat{s})}{\hat{s}}\sqrt{\frac{2}{\pi}}\exp\left[\frac{1}{2}\hat{s}(1 - H(\hat{s})^2)\right]\right\} \quad 16\text{-}24$$

and over the range $2 \le \hat{s} \le 8$ is approximated by the polynomial

$$K(\hat{s}) = -0.08566 + \frac{1.65879}{\hat{s}} - \frac{0.74604}{\hat{s}^2} \quad 16\text{-}25$$

To use this model specify PAASCH.GFI=1 or PAASCH.GFI=2 on the **MATERIAL** statement. PAASCH.GFI=1 uses the analytic expressions for $H(\hat{s})$ and $K(\hat{s})$, whereas PAASCH.GFI=2 uses the polynomial fit.

16.2.2 Organic Defects Model

Density of States

The **ODEFECTS** statement is used to specify the energy distribution of defect states. The total density of states (DOS) is composed of an acceptor-like conduction band DOS ($g_A(E)$) and a donor-like valence band DOS ($g_D(E)$). There are two defect distribution models available. The first model defines the defects based on a characteristic temperature and density [289].

$$g_A(E) = \frac{\mathrm{HA}}{k\mathrm{TCA}} exp\left(\frac{E - E_c}{(k\mathrm{TCA})}\right) \quad 16\text{-}26$$

$$g_D(E) = \frac{\mathrm{HD}}{k\mathrm{TCD}} exp\left(\frac{E_v - E}{(k\mathrm{TCD})}\right) \quad 16\text{-}27$$

HA is the acceptor traps density of states, HD is the donor traps density of states, TCA is the acceptor traps characteristic temperature, TCD is the donor traps characteristic temperature, and E is the energy.

Statement	Parameter	Default	Units
ODEFECTS	HA	0	cm^{-3}
ODEFECTS	HD	0	cm^{-3}
ODEFECTS	TCA	0	K
ODEFECTS	TCD	0	K

The trapped state concentrations for electrons (n_A) and holes (p_D) are given by:

$$n_A = \int_{Ev}^{Ec} g_A(E) \cdot f_{tA}(E, n, p)dE \quad 16\text{-}28$$

$$p_D = \int_{Ev}^{Ec} g_D(E) \cdot f_{tD}(E, n, p)dE \quad 16\text{-}29$$

where $f_{tA}(E,n,p)$ is the acceptor probability of occupation and $f_{tD}(E,n,p)$ is the donor probability of occupation.

The second model is the effective transport hopping energy model and defines the defects using a double peak Gaussian distribution [14],[15].

$$g_A(E) = \frac{\mathtt{NIA}}{\sqrt{2\pi}\ \mathtt{SIGMAIA}} exp\left(-\frac{(E-E_c)^2}{2\mathtt{SIGMAIA}^2}\right) + \frac{\mathtt{NA}}{\sqrt{2\pi}\ \mathtt{SIGMAA}} exp\left(-\frac{(E-E_c+EA)^2}{2\mathtt{SIGMAA}^2}\right) \quad 16\text{-}30$$

$$g_D(E) = \frac{\mathtt{NID}}{\sqrt{2\pi}\ \mathtt{SIGMAID}} exp\left(-\frac{(E_v-E)^2}{2\mathtt{SIGMAID}^2}\right) + \frac{\mathtt{ND}}{\sqrt{2\pi}\ \mathtt{SIGMAD}} exp\left(-\frac{(E_v-E+\mathtt{ED})^2}{2\mathtt{SIGMAD}^2}\right) \quad 16\text{-}31$$

NIA is the total intrinsic density for acceptor-like traps. NID is the total intrinsic density for donor-like traps. SIGMAIA is the Gaussian width for the intrinsic acceptor-like traps. SIGMAID is the Gaussian width for the intrinsic donor-like traps. NA is the total doping density for acceptor-like traps. ND is the total doping density for donor-like traps. SIGMAA is the Gaussian width for the doping acceptor-like traps. SIGMAD specifies the Gaussian width for the doping donor-like traps. EA is the energy shift between the intrinsic and doping states for acceptor-like traps. ED is the energy shift between the intrinsic and doping states for donor-like traps.

You can use the DOPING.PERCENT parameter to specify the doping percentage. If DOPING.PERCENT is specified, then the doping density for acceptor-like and donor-like traps is

$$\mathtt{NA} = \mathtt{NA}\frac{\mathtt{DOPING.PERCENT}}{100} \quad 16\text{-}32$$

$$\mathtt{ND} = \mathtt{ND}\frac{\mathtt{DOPING.PERCENT}}{100} \quad 16\text{-}33$$

The trapped state concentrations are given by:

$$n_A = \int_{E_v}^{E_{tr_n}} gA(E) f_{tA}(E, n, p)dE \quad 16\text{-}34$$

$$p_D = \int_{E_{tr_p}}^{E_c} gD(E) f_{tD}(E, n, p)dE \quad 16\text{-}35$$

where E_{tr_n} is the effective transport energy for electrons and E_{tr_p} is the effective transport energy for holes. You must specify the HOPPING parameter on the ODEFECTS statement to use the effective transport hopping energy model. The effective transport energy for electrons and holes is calculate by solving Equations 16-30 and 16-31 along with Equations 16-34 and 16-35.

$$\int_{-\infty}^{E_{tr_n}} gA(E)(E_{tr_n}-E)^3 dE = \frac{6}{\pi}(\mathrm{HOPN.GAMMA}kT)^3 \quad \text{16-36}$$

$$\int_{-\infty}^{E_{tr_p}} gD(E)(E_{tr_p}-E)^3 dE = \frac{6}{\pi}(\mathrm{HOPP.GAMMA}kT)^3 \quad \text{16-37}$$

where HOPN.BETA is the electron percolation constant, HOPP.BETA is the hole percolation constant, HOPN.GAMMA is 1/carrier localization radius for electrons and HOPP.GAMMA is 1/carrier localization radius for holes.

Statement	Parameter	Value	Units
ODEFECTS	DOPING.PERCENT		%
ODEFECTS	EA	0.0	eV
ODEFECTS	ED	0.0	eV
MATERIAL	HOPN.BETA	1.5	
MATERIAL	HOPN.GAMMA	5.0×10^{7}	cm^{-1}
MATERIAL	HOPP.BETA	1.5	
MATERIAL	HOPP.GAMMA	5.0×10^{7}	cm^{-1}
ODEFECTS	HOPPING	False	
ODEFECTS	NA	0.0	cm^{-3}
ODEFECTS	ND	0.0	cm^{-3}
ODEFECTS	NIA	0.0	cm^{-3}
ODEFECTS	NID	0.0	cm^{-3}
ODEFECTS	SIGMAA	0.0	eV
ODEFECTS	SIGMAD	0.0	eV
ODEFECTS	SIGMAIA	0.0	eV
ODEFECTS	SIGMAID	0.0	eV

In steady-state, the probabilities of occupation are given by:

$$f_{tA} = \frac{v_n \mathtt{SIGAE} n + v_p \mathtt{SIGAH} n_i exp\left[\frac{E_i - E}{kT_L}\right]}{v_n \mathtt{SIGAE}\left(n + n_i exp\left[\frac{E - E_i}{kT_L}\right]\right) + v_p \mathtt{SIGAH}\left(p + n_i exp\left[\frac{E_i - E}{kT_L}\right]\right)} \quad 16\text{-}38$$

$$f_{tD} = \frac{v_p \mathtt{SIGDH} p + v_n \mathtt{SIGDE} n_i exp\left[\frac{E - E_i}{kT_L}\right]}{v_n \mathtt{SIGDE}\left(n + n_i exp\left[\frac{E - E_i}{kT_L}\right]\right) + v_p \mathtt{SIGDH}\left(p + n_i exp\left[\frac{E_i - E}{kT_L}\right]\right)} \quad 16\text{-}39$$

where v_n is the electron thermal velocity, v_p is the hole thermal velocity, n_i is the intrinsic carrier concentration, SIGAE and SIGAH are the acceptor electron and hole capture cross-sections respectively and SIGDE and SIGDH are the donor electron and hole capture cross-sections respectively.

Statement	Parameter	Default	Units
ODEFECTS	SIGAE	10^{-20}	cm^2
ODEFECTS	SIGAH	10^{-20}	cm^2
ODEFECTS	SIGDE	10^{-20}	cm^2
ODEFECTS	SIGDH	10^{-20}	cm^2

Steady-State and Transient Recombination

The steady-state net recombination/generation rate is identical for electrons (R_n) and holes (R_p) as equilibrum is assumed to be instantaneous. The Schockley-Read-Hall recombination/generation rate due to the defect states is given by:

$$R_{n,p} = \int_{Ev}^{Ec} \left[\frac{v_n v_p \mathtt{SIGAE}\ \mathtt{SIGAH}(np - n_i^2) g_A(E)}{v_n \mathtt{SIGAE}\left(n + n_i exp\left[\frac{E - E_i}{kT_L}\right]\right) + v_p \mathtt{SIGAH}\left(p + n_i exp\left[\frac{E_i - E}{kT_L}\right]\right)} \right. \quad 16\text{-}40$$

$$\left. + \frac{v_n v_p \mathtt{SIGDE}\ \mathtt{SIGDH}(np - n_i^2) g_D(E)}{v_n \mathtt{SIGDE}\left(n + n_i exp\left[\frac{E - E_i}{kT}\right]\right) + v_p \mathtt{SIGDH}\left(p + n_i exp\left[\frac{E_i - E}{kT}\right]\right)} \right] dE$$

In transient simulations, the instantaneous equilibrium can no longer be assumed. Instead, the recombination/generation rate is calculated using the transient acceptor and donor probabilities of occupation. These are calculated by solving additional rate equations:

$$\frac{d}{dt}(n_A) = \int_{Ev}^{Ec} g_A(E)\left[v_n \texttt{SIGAE}\left(n(1-f_{tA}(E)) - f_{tA}(E) n_i \exp\left[\frac{E-E_i}{kT_L}\right]\right)\right.$$
$$\left. - v_p \texttt{SIGAH}\left(p f_{tA}(E) - (1-f_{tA}(E)) n_i exp\left[\frac{E_i - E}{kT_L}\right]\right)\right] dE \quad \text{16-41}$$

$$\frac{d}{dt}(p_D) = \int_{Ev}^{Ec} g_D(E)\left[v_p \texttt{SIGDH}\left(p(1-f_{tD}(E)) - f_{tD}(E) n_i exp\left[\frac{E_i - E}{kT_L}\right]\right)\right.$$
$$\left. - v_n \texttt{SIGDE}\left(n f_{tD}(E) - (1-f_{tD}(E)) n_i \exp\left[\frac{E-E_i}{kT_L}\right]\right)\right] dE \quad \text{16-42}$$

Additional Parameters

If the parameter FAST is specified in the **ODEFECTS** statement, then the traps are assumed to reach equilibrum instantaneously. Therefore, Equations 16-41 and 16-42 will not be solved. In this case, the steady-state probabilities of a occupation will be used in transient simulations.

You can use the C-Interpreter to define the defect states in the bandgap. The F.OTFTDON and F.OTFTACC parameters of the **ODEFECTS** statement indicate the filenames containing the C functions. For more information on using the C-Interpreter, see Appendix A "C-Interpreter Functions".

```
ODEFECTS F.OTFTDON=mydefects.c F.OTFTACC=mydefects.c
```

The file, mydefects.c, will contain C functions for donor and acceptor defect densities as a function of energy. The following example defect states are defined in the file, otft.lib. The resultant distribution of defects versus energy can be plotted in the files, odon.dat and oacc.dat.

```
ODEFECTS   F.OTFTDON=otft.lib   F.OTFTACC=otft.lib   DFILE=odon.dat
AFILE=oacc.dat \

SIGAE=1.E-16 SIGAH=1.E-14 SIGDE=1.E-14 SIGDH=1.E-16
```

If you specify the CONTINUOUS parameter in the **ODEFECTS** statement, then a numerical integral will be used to calculate the charge and recombination integrals. In this case, you can use the parameters NUMA and NUMD to specify the number of acceptor and donor energy levels intervals used in the integral. If CONTINUOUS is not specified, then the integrals will be modelled with discrete energy levels. The integral terms are replaced by summations, which run over the number of discrete energy levels (NUMA for acceptors and NUMD for donors).

Statement	Parameter	Default	Units
ODEFECTS	AFILE		
ODEFECTS	CONTINUOUS	False	

ODEFECTS	DFILE		
ODEFECTS	F.OTFTACC		
ODEFECTS	F.OTFTDON		
ODEFECTS	FAST	False	
ODEFECTS	NUMA	12	
ODEFECTS	NUMD	12	

16.2.3 Hopping Mobility Model

If the effective transport hopping energy model is specified (HOPPING on the **ODEFECTS** statement), then the field-independent hopping mobility can be used (HOPMOB on the **MODELS** statement). This is given by [15]:

$$\mu_{n0} = \frac{q\text{HOPN.V0}}{kT}\left[\int_{-\infty}^{E_{tr_n}} g_A(E)dE\right]^{-2/3} exp\left[-2\left(\frac{3\text{HOPN.BETA}}{4\pi}\right)^{1/3}\text{HOPN.GAMMA}\left[\int_{-\infty}^{E_{tr_n}} g_A(E)dE\right]^{-1/3}\right] \quad 16\text{-}43$$

$$\mu_{p0} = \frac{q\text{HOPP.V0}}{kT}\left[\int_{-\infty}^{E_{tr_p}} g_D(E)dE\right]^{-2/3} exp\left[-2\left(\frac{3\text{HOPP.BETA}}{4\pi}\right)^{1/3}\text{HOPP.GAMMA}\left[\int_{-\infty}^{E_{tr_p}} g_D(E)dE\right]^{-1/3}\right] \quad 16\text{-}44$$

where HOPN.V0 is the attempt-to-jump frequency for electrons and HOPP.V0 is the attempt-to-jump frequency for holes.

If the effective transport hopping energy model is specified on a region with a Gaussian band structure the full equations from reference [15] are used. The transport energy is found by solving

$$\int_{-\infty}^{E_{tr,n}} \frac{g_C(E_n)(E_{tr,n}-E_n)^3}{1+\exp(-(E_n-E_{Fn})/kT)}dE_n = \frac{6}{\pi}(\text{HOPN.GAMMA}kT)^3 \quad 16\text{-}45$$

$$\int_{-\infty}^{E_{tr,p}} \frac{g_V(E_p)(E_{tr,p}-E_p)^3}{1+\exp(-(E_p-E_{Fp})/kT)}dE_p = \frac{6}{\pi}(\text{HOPP.GAMMA}kT)^3 \quad 16\text{-}46$$

And the mobilities are calculated from

$$\mu_{n0} = \frac{q\text{HOPN.V0}}{kTn}\left[\int_{-\infty}^{E_{tr,n}} g_C(E_n)dE_n\right]^{-2/3}\int_{-\infty}^{E_{tr,n}} \frac{g_C(E_n)}{1+\exp(-(E_n-E_{Fn})/kT)}\exp\left(\frac{E_n-E_{tr,n}}{kT}\right)dE_n \quad 16\text{-}47$$

$$\mu_{p0} = \frac{q\text{HOPP.V0}}{kTp}\left[\int_{-\infty}^{E_{tr,p}} g_V(E_p)dE_p\right]^{-2/3}\int_{-\infty}^{E_{tr,p}} \frac{g_C(E_p)}{1+\exp(-(E_p-E_{Fp})/kT)}\exp\left(\frac{E_p-E_{tr,p}}{kT}\right)dE_p$$

Statement	Parameter	Value	Units
MODELS	HOPMOB	False	
MOBILITY	HOPMOB.N	False	
MOBILITY	HOPMOB.P	False	
MATERIAL	HOPN.BETA	1.5	
MATERIAL	HOPN.GAMMA	5.0e7	cm^{-1}
MATERIAL	HOPN.V0	1.0e11	Hz
MATERIAL	HOPP.BETA	1.5	
MATERIAL	HOPP.GAMMA	5.0e7	cm^{-1}
MATERIAL	HOPP.V0	1.0e11	Hz

16.2.4 Poole-Frenkel Mobility Model

The Poole-Frenkel mobility model is given by [99][129]:

$$\mu_{n_{PF}}(E) = \mu_{n0}exp\left(-\frac{\text{DELTAEN.PFMOB}}{kT_{neff}} + \left(\frac{\text{BETAN.PFMOB}}{kT_{neff}} - \text{GAMMAN.PFMOB}\right)\sqrt{|E|}\right) \quad 16\text{-}48$$

$$\mu_{p_{PF}}(E) = \mu_{p0}exp\left(-\frac{\text{DELTAEP.PFMOB}}{kT_{peff}} + \left(\frac{\text{BETAP.PFMOB}}{kT_{peff}} - \text{GAMMAP.PFMOB}\right)\sqrt{|E|}\right) \quad 16\text{-}49$$

where $\mu_{n_{PF}}(E)$ and $\mu_{p_{PF}}(E)$ are the Poole-Frenkel mobilities for electrons and holes respectively, μ_{n0} and μ_{n0} are the zero field mobilities for electrons and holes respectively, and E is the electric field. DELTAEN.PFMOB and DELTAEP.PFMOB are the activation energy at zero electric field for electrons and holes respectively. BETAN.PFMOB is the electron Poole-Frenkel factor, and BETAP.PFMOB is the hole Poole-Frenkel factor.

T_{neff} is the effective temperature for electrons and is defined as:

$$T_{neff} = \frac{\text{T0N}\cdot\text{PFMOB}\ T_L}{\text{T0N}\cdot\text{PFMOB}-T_L} \quad 16\text{-}50$$

T_{peff} is the effective temperature for holes and is defined as:

$$T_{peff} = \frac{\text{T0P}\cdot\text{PFMOB}\ T_L}{\text{T0P}\cdot\text{PFMOB}-T_L} \quad 16\text{-}51$$

If you specify the BETANAUTO.PFMOB parameter on the **MOBILITY** statement, then BETAN.PFMOB in Equation 16-48 will be calculated from the permittivity:

$$\mathrm{BETAN.PFMOB} = q\sqrt{\frac{q}{\pi\varepsilon\varepsilon 0}} \qquad 16\text{-}52$$

Similarly, if you specify BETAPAUTO.PFMOB, then BETAP.PFMOB in Equation 16-49 will be calculated from the permittivity.

If BETAN.PFMOB, BETAP.PFMOB, BETANAUTO.PFMOB, or BETAPAUTO.PFMOB are not specified, you will then use a simplified form of the Poole-Frenkel mobility [129][272]:

$$\mu_{n_{PF}}(E) = \mu_{n0}\exp\left(-\frac{\mathrm{DELTAEN.PFMOB}}{kT_{neff}} + \sqrt{\frac{|E|}{\mathrm{E0N.PFMOB}}} - \mathrm{GAMMAN.PFMOB}\sqrt{|E|}\right) \qquad 16\text{-}53$$

$$\mu_{p_{PF}}(E) = \mu_{p0}\exp\left(-\frac{\mathrm{DELTAEP.PFMOB}}{kT_{peff}} + \sqrt{\frac{|E|}{\mathrm{E0P.PFMOB}}} - \mathrm{GAMMAP.PFMOB}\sqrt{|E|}\right) \qquad 16\text{-}54$$

where $E0$ is the characteristic field. You can use the parameters E0NAUTO.PFMOB and E0PAUTO.PFMOB in the **MOBILITY** statement to calculate the electron and hole characteristic field values from the permittivity:

$$E0 = \left(\frac{kT_L}{q}\right)^2\left(\frac{\pi\varepsilon\varepsilon 0}{q}\right) \qquad 16\text{-}55$$

If you specify the PFMOB parameter on the **MODELS** statement, the Poole-Frenkel mobility model will be used for both electrons and holes. You can specify this model individually for electrons and holes by using the PFMOB.N and PFMOB.P parameters in the **MOBILITY** statement.

Statement	Parameter	Default	Units
MOBILITY	BETAN.PFMOB	0	$eV(cm/V)^{1/2}$
MOBILITY	BETAP.PFMOB	0	$eV(cm/V)^{1/2}$
MOBILITY	BETANAUTO.PFMOB	False	
MOBILITY	BETAPAUTO.PFMOB	False	
MOBILITY	DELTAEN.PFMOB	0	eV
MOBILITY	DELTAEP.PFMOB	0	eV
MOBILITY	E0N.PFMOB	1956	V/cm
MOBILITY	E0P.PFMOB	1956	V/cm
MOBILITY	E0NAUTO.PFMOB	False	

MOBILITY	E0PAUTO.PFMOB	False	
MOBILITY	GAMMAN.PFMOB	0	$(cm/V)^{1/2}$
MOBILITY	GAMMAP.PFMOB	0	$(cm/V)^{1/2}$
MOBILITY	PFMOB	False	
MOBILITY	PFMOB.N	False	
MOBILITY	PFMOB.P	False	
MOBILITY	TON.PFMOB	0	K
MOBILITY	TOP.PFMOB	0	K
MOBILITY	VTHN.PFMOB		cm/s
MOBILITY	VTHP.PFMOB		cm/s

Due to the strong dependence on the electric field, the Poole-Frenkel mobility model can cause convergence problems. To increase the stability of the Poole-Frenkel mobility model, it is combined with a limiting mobility ($\mu_{n_{lim}}(E)$ and $\mu_{p_{lim}}(E)$) calculated from the thermal velocities (ν_n and ν_p).

$$\mu_n(E) = \frac{1}{\frac{1}{\mu_{n_{PF}}(E)} + \frac{1}{\mu_{n_{lim}}(E)}} \quad 16\text{-}56$$

$$\mu_p(E) = \frac{1}{\frac{1}{\mu_{p_{PF}}(E)} + \frac{1}{\mu_{p_{lim}}(E)}} \quad 16\text{-}57$$

where:

$$\mu_{n_{lim}}(E) = \frac{\nu_n}{E} \quad 16\text{-}58$$

$$\mu_{p_{lim}}(E) = \frac{\nu_p}{E} \quad 16\text{-}59$$

If you specify VTHN.PFMOB on the **MOBILITY** statement, then Equation 16-59 becomes:

$$\mu_{n_{lim}}(E) = \frac{\text{VTHN.PFMOB}}{E} \quad 16\text{-}60$$

Similarly, if you specify VTHP.PFMOB on the **MOBILITY** statement, then Equation 16-60 becomes:

$$\mu_{p_{lim}}(E) = \frac{\texttt{VTHP.PFMOB}}{E} \quad (16\text{-}61)$$

16.2.5 Bimolecular Langevin Recombination Model

To enable this model, specify the parameter LANGEVIN in the **MODELS** statement. The Langevin recombination rate coefficient is given by [32][284]:

$$r_L(x,y,t) = \texttt{A.LANGEVIN}\frac{q[\mu_n(E)+\mu_p(E)]}{\varepsilon_r\varepsilon_0} \quad (16\text{-}62)$$

If you specify the logical parameter KOSTER on the **MODELS** statement, the Langevin rate is given by:

$$r_L(x,y,t) = \texttt{A.LANGEVIN}\frac{qmin(\mu_n,\mu_p)}{\varepsilon_r\varepsilon_o}$$

Statement	Parameter	Default	Units
MATERIAL	A.LANGEVIN	1	

The Langevin recombination rate is given by:

$$R_L n,p = r_L(x,y,t)(np - ni^2) \quad (16\text{-}63)$$

When enabled, the Langevin recombination rate is included in recombination terms in the carrier continuity equations (Equations 3-3 and 3-4). This rate is also included in the exciton singled and triplet continuity equations described in Section 16.3 “Singlet and Triplet Excitons”.

16.2.6 Juska Two-Dimensional Langevin Recombination Model

This is a 2D Langevin recombination model that only considers the drift current. To enable this model, specify the DR.2D.LANGEVIN parameter on the **MODELS** statement. The Langevin recombination rate is given by [145]:

$$R_{L,n,p} = \texttt{A·2D.LANGEVIN}\sqrt{\pi}\frac{q\texttt{L.2D.LANGEVIN}^{3/2}c^{1/2}}{\varepsilon_r\varepsilon_0}(\mu_n(E)+\mu_p(E))np \quad (16\text{-}64)$$

with

$$c = \sqrt{np}$$

L.2D.LANGEVIN is the thickness of the 2D lamellas in the organic structure. In [145], the value of the prefactor A.2D.LANGEVIN is 3/4. Another paper [220] has a similar expression with a prefactor of 15/8.

Statement	Parameter	Default	Units
MATERIAL	A.2D.LANGEVIN	0.75	
MATERIAL	L.2D.LANGEVIN	1.6	nm

When enabled, the Langevin recombination rate is included in recombination terms in the carrier continuity equations (Equations 3-3 and 3-4). This rate is also included in the exciton singlet and triplet continuity equations described in Section 16.3 "Singlet and Triplet Excitons".

16.2.7 Nenashev Two-Dimensional Langevin Recombination Model

This is a 2D Langevin recombination model that considers both the drift and the diffusion current. To enable this model, specify the DRDI.2D.LANGEVIN parameter on the **MODELS** statement. The Langevin recombination rate is given by [220]:

$$R_{L,n,p} = \frac{4\pi kT\,\mathrm{L.2D.LANGEVIN}}{q\ln\left(\frac{\mathrm{ALPHA.2D.LANGEVIN}}{a^2\,\mathrm{L.2D.LANGEVIN}c}\right)}((\mu_n(E)+\mu_p(E))np) \qquad 16\text{-}65$$

with

$$c = \sqrt{np}$$

and the Onsager radius

$$a = \frac{q^2}{4\pi\varepsilon_r\varepsilon_0 kT}$$

L.2D.LANGEVIN is the thickness of the 2D lamellas in the organic structure. The ALPHA.2D.LANGEVIN parameter can either be specified directly or by providing G.2D.LANGEVIN for the equation

$$\mathrm{ALPHA.2D.LANGEVIN} = \frac{1}{\pi\exp(2\gamma + \mathrm{2D.LANGEVIN})} \qquad 16\text{-}66$$

where γ is Euler's constant. If ALPHA.2D.LANGEVIN is provided then any value given for G.2D.LANGEVIN is ignored.

Statement	Parameter	Default	Units
MATERIAL	L.2D.LANGEVIN	1.6	nm
MATERIAL	ALPHA.2D.LANGEVIN	0.037	
MATERIAL	G.2D.LANGEVIN	1	

When enabled, the Langevin recombination rate is included in recombination terms in the carrier continuity equations (Equations 3-3 and 3-4). This rate is also included in the exciton singlet and triplet continuity equations described in Section 16.3 "Singlet and Triplet Excitons".

16.2.8 Generalized Gaussian Disorder Transport Model

There are two similar Gaussian disorder hopping mobility models. The Extended Gaussian Disorder Model (EGDM) is described in [239, 326] and the Extended Correlated Disorder Model (ECDM) is described in [35]. The EGDM is activated for electrons and holes respectively with the PASVEER.N and PASVEER.P flags on the MOBILITY statement. The ECDM is activated for electrons and holes respectively with the ECDM.N and ECDM.P flags on the MOBILITY statement.

These models assume a Gaussian density of states

$$N_C(E) = \frac{N_{t,n}}{\sqrt{2\pi \mathrm{SIG.PASV.N}^2}} \cdot \exp\left(-\frac{E_n^2}{2\mathrm{SIG.PASV.N}^2}\right) \quad 16\text{-}67$$

$$N_V(E) = \frac{N_{t,p}}{\sqrt{2\pi \mathrm{SIG.PASV.P}^2}} \cdot \exp\left(-\frac{E_p^2}{2\mathrm{SIG.PASV.P}^2}\right) \quad 16\text{-}68$$

$N_C(E)$ and $N_V(E)$ are the distributions for modeling electrons and holes respectively. E_n and E_p are the energies relative to the center of the appropriate Gaussian distribution.

The densities of sites, N_t, are described by the average distance, a, between sites

$$N_{t,n} = \frac{1}{\mathrm{A.PASV.N}^3} \quad 16\text{-}69$$

$$N_{t,p} = \frac{1}{\mathrm{A.PASV.P}^3} \quad 16\text{-}70$$

The organic semiconductor is disordered so the carrier wave-functions are localized with an inverse length of ?. The inverse of the electron localization length is ALP.PASV.N and the inverse of the hole localization length is ALP.PASV.P.

The expressions in [326] are for the case where $?^{-1}=0.1\times a$. If ALP.PASV.N or ALP.PASV.P are defined, then the associated A.PASV.N or A.PASV.P are calculated as

$$\mathrm{A.PASV.N} = \frac{10}{\mathrm{ALP.PASV.N}} \quad (16\text{-}71)$$

$$\mathrm{A.PASV.P} = \frac{10}{\mathrm{ALP.PASV.P}} \quad (16\text{-}72)$$

If both A.PASV.N and ALP.PASV.N are defined, then the explicit value of A.PASV.N is used. Similarly, if both A.PASV.P and ALP.PASV.P are defined, then the explicit value of A.PASV.P is used.

If a Gaussian band structure (Section 16.2.1 "Gaussian Band Structure") is being used then the default parameter values for the Pasveer model are extracted from this band structure.

$$\mathrm{A.PASV.N} = \mathrm{NTC.GAUSS}^{-1/3} \quad (16\text{-}73)$$

$$\mathrm{SIG.PASV.N} = \mathrm{SIGC.GAUSS} \quad (16\text{-}74)$$

$$\mathrm{A.PASV.P} = \mathrm{NTV.GAUSS}^{-1/3} \quad (16\text{-}75)$$

$$\mathrm{SIG.PASV.P} = \mathrm{SIGV.GAUSS} \quad (16\text{-}76)$$

Any parameters explicitly given on a **MOBILITY** statement overload these defaults (but do not change the equivalent parameters in the Gaussian band structure).

These models are best expressed in terms of normalized parameters.

$$\hat{n} = n \cdot \mathrm{A.PASV.N}^3 \quad (16\text{-}77)$$

$$\hat{p} = p \cdot \mathrm{A.PASV.P}^3 \quad (16\text{-}78)$$

$$\hat{\sigma}_n = \frac{\mathrm{SIG.PASV.N}}{k_B T} \quad (16\text{-}79)$$

$$\hat{\sigma}_p = \frac{\mathrm{SIG.PASV.P}}{k_B T} \quad (16\text{-}80)$$

$$\hat{E}_n = \frac{q \cdot \mathrm{A.PASV.N} \cdot E}{\mathrm{SIG.PASV.N}} \quad (16\text{-}81)$$

$$\hat{E}_p = \frac{q \cdot \mathrm{A.PASV.P} \cdot E}{\mathrm{SIG.PASV.P}} \quad (16\text{-}82)$$

The EGDM model of the enhancement to the mobility is

$$\mu(T, n, E) = \mu_{300} \cdot g_0(T) \cdot g_1(n, T) \cdot g_2(E, T) \quad (16\text{-}83)$$

The temperature dependent terms, $g_0(T)$, are

$$g_{0,n} = \exp\left(\frac{\mathrm{TC2.PASV.N} \times \mathrm{SIG.PASV.N}^2}{k^2}\left\{\frac{1}{300^2} - \frac{1}{T^2}\right\}\right) \quad (16\text{-}84)$$

$$g_{0,p} = \exp\left(\frac{\texttt{TC2.PASV}\cdot\texttt{P}\times\texttt{SIG.PASV}\cdot\texttt{P}^2}{k^2}\left\{\frac{1}{300^2}-\frac{1}{T^2}\right\}\right) \quad (16\text{-}85)$$

The carrier density dependent enhancement to the mobilities, g_1, are

$$g_{1,c}(c,T) = exp\left[\frac{1}{2}\cdot(\hat{\sigma}_c^2-\hat{\sigma}_c)\cdot(2\hat{c})^{\delta_c}\right] \quad (16\text{-}86)$$

where

$$\delta_c = \frac{2}{\hat{\sigma}_c^2}\cdot\{\ln[\hat{\sigma}_c^2-\hat{\sigma}_c]-\ln(\ln 4)\} \quad (16\text{-}87)$$

These equations hold for

$$\texttt{CMIN.PASV.N} \le \hat{n} \le \texttt{CCUTOFF}\cdot\texttt{N} \quad (16\text{-}88)$$

$$\texttt{CMIN.PASV.P} \le \hat{p} \le \texttt{CCUTOFF}\cdot\texttt{P} \quad (16\text{-}89)$$

If the normalized carrier concentration is outside of this range, the enhancement g_1 is calculated at the appropriate limit.

The electric field dependent enhancement to the mobilities, g_2, is

$$g_{2,n}(E,T) = exp\left\{0.44\cdot[\hat{\sigma}_c^{3/2}-2.2]\cdot\sqrt{1+0.8\hat{E}_c^2}-1\right\} \quad (16\text{-}90)$$

These equations hold for

$$\hat{E}_n \le \texttt{FCUTOFF}\cdot\texttt{N} \quad (16\text{-}91)$$

$$\hat{E}_p \le \texttt{FCUTOFF}\cdot\texttt{P} \quad (16\text{-}92)$$

If the normalized electric field is higher than the cutoff, the enhancement g_2 is calculated at the cutoff electric field.

The ECDM model of the mobility is

$$\mu(T,c,E) = (\mu_{low}(T,c,E)^{q(\hat{\sigma})}+\mu_{high}(c,E)^{q(\hat{\sigma})})^{1/q(\hat{\sigma})} \quad (16\text{-}93)$$

The exponent is given by

$$q_c(\hat{\sigma}_c) = \frac{2.4}{1-\hat{\sigma}_c} \quad (16\text{-}94)$$

The low-field mobility is similar to the EGDM model

$$\mu_{low}(T, c, E) = \mu_{300} \cdot g_0(T) \cdot f_1(c, T) \cdot f_2(c, E, T) \quad \text{16-95}$$

The $g_0(T)$ term is identical to EGDM model (see Equations 16-84 and 16-85).

The carrier density dependent enhancement to the mobilitites, f_1, are similar to the EGDM expressions

$$f_{1,c}(c, T) = \exp[(0.25\hat{\sigma}_c^2 + 0.7\hat{\sigma}_c) \cdot (2\hat{c})^{\delta_c}] \quad \text{16-96}$$

where

$$\delta_c = \frac{2.3}{\hat{\sigma}_c^2} \cdot \{\ln(0.5\hat{\sigma}_c^2 + 1.4\hat{\sigma}_c) - \ln(\ln 4)\} \quad \text{16-97}$$

The field dependent term has a similar form to the EGDM g_2 expression, but is also dependent upon the carrier density.

$$f_{2,c}(c, E, T) = \exp\left[h(\hat{E})\left(1.05 - 1.2\hat{c}^{r(\hat{\sigma}_c)}\right)(\hat{\sigma}_c^{3/2} - 1)(\sqrt{1 + 2\hat{E}} - 1)\right] \quad \text{16-98}$$

The exponent is

$$r(\hat{\sigma}_c) = 0.7\hat{\sigma}_c^{-0.7}$$

The *h(E)* function is chosen to ensure the differential of f_2 goes to zero at $E=0$ (E_{crit}=0.16).

$$\begin{aligned} h(\hat{E}) &= \frac{4}{3}\frac{\hat{E}}{\hat{E}_{crit}} && for && \hat{E} \le \frac{\hat{E}_{crit}}{2} \\ h(\hat{E}) &= \left[1 - \frac{4}{3}\left(\frac{\hat{E}}{\hat{E}_{crit}} - 1\right)^2\right] && for && \frac{\hat{E}_{crit}}{2} < \hat{E} < E \le \hat{E}_{crit} \\ h(\hat{E}) &= 1 && for && \hat{E} > \hat{E}_{crit} \end{aligned} \quad \text{16-99}$$

The high field mobility is

$$\mu_{high}(c, E) = \frac{\exp(-20)}{\hat{E}_c}\mu_0(1 - \hat{c}) = \mu_{300}\frac{K_{\mu high}}{\hat{E}_c}(1 - \hat{c}) \quad \text{16-100}$$

Statement	Parameter	Value	Units
MOBILITY	PASVEER.N	False	
MOBILITY	PASVEER.P	False	
MOBILITY	SIG.PASV.N	0.1	eV
MOBILITY	SIG.PASV.P	0.1	eV

MOBILITY	A.PASV.N	0.1	nm
MOBILITY	A.PASV.P	0.1	nm
MOBILITY	ALP.PASV.N		nm^{-1}
MOBILITY	ALP.PASV.P		nm^{-1}
MOBILITY	CMIN.PASV.N	$1x10^{-10}$	
MOBILITY	CMIN.PASV.P	$1x10^{-10}$	
MOBILITY	CCUTOFF.N	0.1	
MOBILITY	CCUTOFF.P	0.1	
MOBILITY	FCUTOFF.N	2	
MOBILITY	FCUTOFF.P	2	
MOBILITY	TC2.PASV.N	0.42	
MOBILITY	TC2.PASV.P	0.42	
MOBILITY	ECDM.N	False	
MOBILITY	ECDM.P	False	

16.2.9 Thermionic Emission over an Organic Barrier

In [300], it gives the effective barrier height between two organic interfaces as

$$\Delta E_{eff} = \Delta E - qF_{av}a_{av} \quad \text{16-101}$$

Here, ΔE is the difference between the appropriate band edges, F_{av} is the average electric field across the barrier and a_{av} is the average organic molecule size between the two sides of the barrier.

Staudigel [300] give the probability for a carrier to cross the organic/organic barrier as

$$G(\Delta E_{eff}) = c \cdot \int_{-\infty}^{\infty} \Gamma(E_A)dE_A \int_{-\infty}^{\infty} \Gamma(E_B)dE_B \begin{cases} \exp\left(-\dfrac{E_B + \Delta E_{eff} - E_A}{k_B T}\right) & E_B + \Delta E_{eff} > E_A \\ 1 & E_B + \Delta E_{eff} \le E_A \end{cases} \quad \text{16-102}$$

The normalization constant, c, is chosen so that

$$G(0) = 1 \quad \text{16-103}$$

This probability replaces the exp term in Equations 6-48 and 6-49, so we get the electron and hole current at an organic-organic interface.

$$J_n = qv_n(n^+ - n^- G(\text{Q.THERMIONIC}\Delta E_{eff})) \; J_p = qv_p(p^+ - p^- G(\text{Q.THERMIONIC}\Delta E_{eff})) \quad \text{16-104}$$

The probability as a function of ΔE_{eff} for a range of Gaussian widths is shown in Figure 16-1.

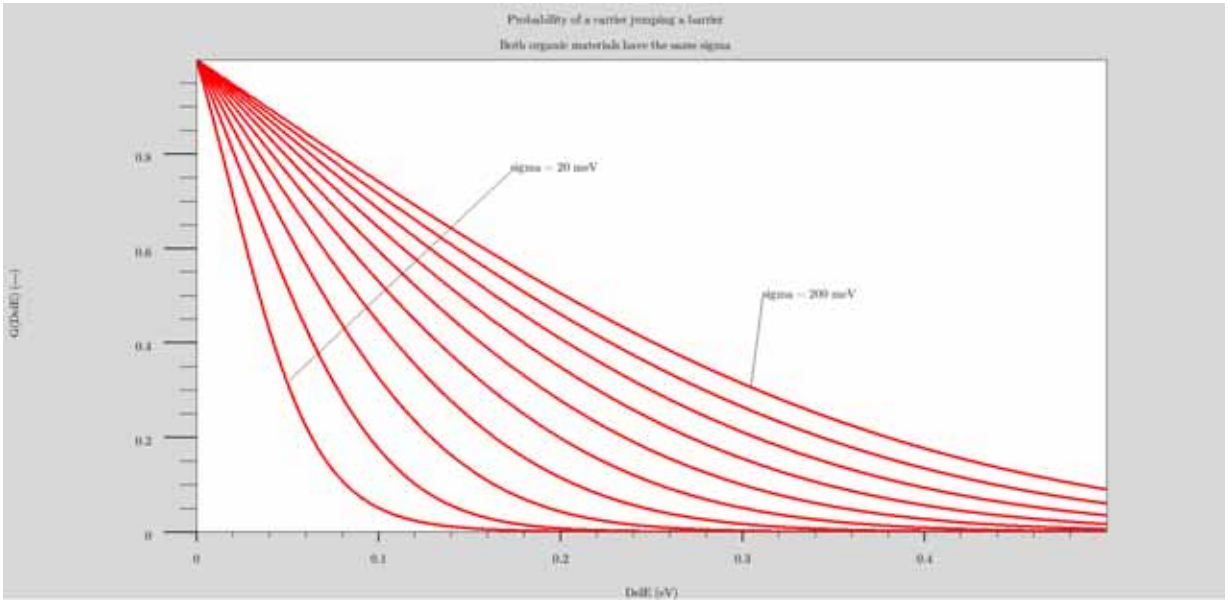


Figure 16-1: The probability of a charge carrier jumping over the boundary in an organic-organic interface as a function of barrier height. Calculated at different widths of the Gaussian density of states.

By default the probability is calculated over the range $0 \le G(\Delta E_{eff}) \le 5.12\sigma$ where σ is the width of the host Gaussian (SIGC.GAUSS for electrons and SIGV.GAUSS for holes). If this is insufficient then the upper limit of the table can be extended with the PGG.NSIGMA parameter on the INTERFACE command. If this is specified the probability will be calculated over the range $0 \le G(\Delta E_{eff}) \le$ PGG.NSIGMAσ.

The SAVE.PJUMP parameter on the **INTERFACE** statement tells Atlas to save the $G(\Delta E_{eff})$ calculations during a simulation. The value of this parameter is the root of a filename. For example

```
INTERFACE S.S THERMIONIC SAVE.PJUMP=gde
```

Atlas will generate a file of G(Q.THERMIONICΔE_{eff}) vs. F_{av} for each valid interface and each carrier type.

```
<filename root>.<regA>.<regB>.<carrier type>.log
```

For example, gde.1.2.n.log or gde.2.4.p.log. The data stored in the files are

- the electric field from region A into region B,
- the effective barrier height,
- the probability of a carrier jumping from region A into region B,
- and the probability of a carrier jumping from region B into region A.

One of the two probabilities is expected to be 1 (if there is a barrier hindering a carrier from region A to region B, then there can be no barrier in the other direction). But both probabilities are saved to ensure the direction is known.

If you only want to save the electron data, then add ^SAVE.HOLE to the **INTERFACE** statement. If you only want to save the hole data, then ^SAVE.ELEC should be added to the **INTERFACE** statement.

The SPJ.DE parameter gives a limit on the density of points saved to the files. The data saved is calculated during a simulation so you have little control over the actual F_{av} saved to the file. But SPJ.DE gives the minimum separation of data points in the file. For example, if SPJ.DE=100, then the F_{av} points in the file will be at least 100 V/cm apart. So if $F_{av} = 2500$ has been added to the file, then no additional data in the range $2400 \le F_{av} \le 2600$ will be added.

16.3 Singlet and Triplet Excitons

The distribution of singlet or triplet excitons can be used to infer the radiative rates for luminescense or phosphorescence in organic material LEDs. In Atlas, you can self-consistently solve the singlet and triplet exciton continuity equations along with the electron and hole drift diffusion equations. The singlet exciton continuity equation is given by [272], [300], and [332]:

$$\frac{dS(x,y,t)}{dt} = \mathtt{RST.EXCITON} R_{Ln,p} + \mathtt{F.GENSINGLET}(x,y,z,t,n,p) \quad \text{16-105}$$

$$+ \frac{\mathtt{RST.TT.EXCITON}}{1+\mathtt{RST.TT.EXCITON}} \mathtt{KTT\_EXCITON}\ T^2(x,y,t)$$

$$-\mathtt{KISC.EXCITON}\ S(x,y,t) - \mathtt{KST\cdot EXCITON}\ S(x,y,t)\ T(x,y,t) - K_{dd}S(x,y,t)$$

$$-\mathtt{KSP\cdot EXCITON}\ S(x,y,t)(n(x,y,t)+p(x,y,t)) - \frac{\mathtt{KSS\cdot EXCITON}}{2}\ S^2(x,y,t)$$

$$-\mathtt{KNRS.EXCITON}\ S(x,y,t) - \frac{\mathtt{PHEFF\cdot EXCITON}}{\mathtt{TAUS\cdot EXCITON}} S(x,y,t) + \nabla(D_S \nabla S(x,y,t)) - Q(x,y)S(x,y,z)$$

$$-R_{Dn,p}S(x,y,t) + G_{ph}\mathtt{QE.EXCITON} - \mathtt{SIGSN.EXCITON}(v_{diff}+v_n)S(x,y,t)n(x,y,t)$$

$$-\mathtt{SIGSP.EXCITON}(v_{diff}+v_p)S(x,y,t)p(x,y,t) + \mathtt{KRISC.EXCITON}\ T(x,y,t)$$

while the triplet exciton continuity equation is given by:

$$\frac{dT(x,y,t)}{dt} = (1-\mathtt{RST\_EXCITON})R_{Ln,p} + \mathtt{F.GENTRIPLET}(x,y,z,t,n,p) \quad \text{16-106}$$

$$+\mathtt{KISC.EXCITON}\ S(x,y,t) - \mathtt{KTP\_EXCITON}\ T(x,y,t)(n(x,y,t)+p(x,y,t))$$

$$-\mathtt{KTT\_EXCITON}\ T^2(x,y,t) - \mathtt{KNRT\cdot EXCITON}\ T(x,y,t) - \mathtt{KRISC.EXCITON} T(x,y,t)$$

$$-\frac{Tx,y,t}{\mathtt{TAUT\cdot EXCITON}} - \mathtt{KEE.EXCITON} T(x,y,t) + \nabla(D_T \nabla T(x,y,t))$$

where:

- $S(x,y,t)$ is the spatial and temporal singlet exciton density,
- $T(x,y,t)$ is the spatial and temporal triplet exciton density,
- RST.EXCITON is the fraction of singlets formed during Langevin recombination,
- RST.TT.EXCITON is the fraction of singlets formed during triplet-triplet annihilation,
- $R_{Ln,p}$ is the Langevin recombination rate,
- $n(x,y,t)$ is the electron concentration,
- $p(x,y,t)$ is the hole concentration,
- TAUS.EXCITON is the singlet radiative decay lifetime,
- TAUT.EXCITON is the triplet radiative decay lifetime,
- KISC.EXCITON is the intersystem crossing constant,

- KRISC.EXCITON is the triplet intersystem crossing constant,
- KST.EXCITON is the singlet-triplet constant,
- KSP.EXCITON is the singlet-polaron constant,
- KTP.EXCITON is the triplet-polaron constant,
- KSS.EXCITON is the singlet-singlet constant,
- KTT.EXCITON is the triplet-triplet constant,
- KNRS.EXCITON is the singlet non-radiative decay constant
- KNRT.EXCITON is the triplet non-radiative decay constant.
- k_{dd} is the dipole-dipole exciton transfer rate (Forster).
- KEE.EXCITON is the electron exchange exciton transfer rate.
- $Q(x,y)$ is the metal quenching rate.
- PHEFF.EXCITON is the host photoluminescence quantum efficiency.
- $R_{Dn,p}$ is the exciton dissociation rate described in Section 16.3.3 “Exciton Dissociation”.
- G_{ph} is the photoabsorption rate.
- QE.EXCITON is the proportion of absorbed photons that is responsible for generating singlets.
- F.GENSINGLET gives an independent generation of singlet excitons as a function of position and time via a c-interpreter function.
- F.GENTRIPLET gives an independent generation of triplet excitons as a function of position and time via a c-interpreter function.
- V_{diff} is the exciton diffusion velocity.
- V_n is the electron velocity.
- V_p is the hole velocity.
- SIGSN.EXCITON is the singlet/electron capture cross section.
- SIGSP.EXCITON is the singlet/hole capture cross section.

The singlet diffusion constant, D_S, can be expressed as:

$$D_S = \frac{\mathrm{LDS.EXCITON}^2}{\mathrm{TAUS.EXCITON}} \qquad 16\text{-}107$$

where LDS.EXCITON is the singlet diffusion length.

The triplet diffusion constant, D_T, can be expressed as:

$$D_T = \frac{\mathrm{LDT.EXCITON}^2}{\mathrm{TAUT.EXCITON}} \qquad 16\text{-}108$$

where LDT.EXCITON is the triplet diffusion length.

You can specify the dipole-dipole exciton transfer rate using KDD.EXCITON on the **MATERIAL** statement. If KDD.EXCITON is not specified, K_{dd} can be expressed as

$$K_{dd} = \frac{\mathrm{R0.EXCITON}^6 \ \mathrm{DOPING.PERCENT} \ 0.01}{8 \ \mathrm{TAUS.EXCITON} \ \mathrm{ARADIUS.EXCITON}^6} \qquad 16\text{-}109$$

16-106			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	KDD.EXCITON	0	s^{-1}
MATERIAL	KEE.EXCITON	0	s^{-1}
MATERIAL	KISC.EXCITON	0	s^{-1}
MATERIAL	KNRS.EXCITON	0	s^{-1}
MATERIAL	KNRT.EXCITON	0	s^{-1}
MATERIAL	KSP.EXCITON	0	$cm^3 s^{-1}$
MATERIAL	KST.EXCITON	0	$cm^3 s^{-1}$
MATERIAL	KTP.EXCITON	0	$cm^3 s^{-1}$
MATERIAL	KSS.EXCITON	0	$cm^3 s^{-1}$
MATERIAL	KTT.EXCITON	0	$cm^3 s^{-1}$
MATERIAL	LDS.EXCITON	0.01	microns
MATERIAL	LDT.EXCITON	0.0632	microns
MATERIAL	RST.EXCITON	0.25	
MATERIAL	RST.TT.EXCITON	RST.EXCITON	
MATERIAL	TAUS.EXCITON	1×10^{-7}	s
MATERIAL	TAUT.EXCITON	1×10^{-4}	s
MATERIAL	PHEFF.EXCITON	1.0	
MATERIAL	SIGSN.EXCITON	0	cm^{-2}
MATERIAL	SIGSP.EXCITON	0	cm^{-2}
MATERIAL	KRISC.EXCITON	0	s^{-1}

Statement	**Parameter**	**Default**	**Units**
MATERIAL	KISC.EXCITON	0	s^{-1}
MATERIAL	KNRS.EXCITON	0	s^{-1}

MATERIAL	KNRT.EXCITON	0	s^{-1}
MATERIAL	KSP.EXCITON	4×10^{-8}	$cm^3\ s^{-1}$
MATERIAL	KST.EXCITON	2×10^{-7}	$cm^3\ s^{-1}$
MATERIAL	KTP.EXCITON	2×10^{-9}	$cm^3\ s^{-1}$
MATERIAL	KSS.EXCITON	5×10^{-8}	$cm^3\ s^{-1}$
MATERIAL	KTT.EXCITON	1×10^{-9}	$cm^3\ s^{-1}$
MATERIAL	LDS.EXCITON	0.01	microns
MATERIAL	LDT.EXCITON	0.0632	microns
MATERIAL	RST.EXCITON	0.25	
MATERIAL	TAUS.EXCITON	1×10^{-7}	s
MATERIAL	TAUT.EXCITON	1×10^{-4}	s

To enable the self-constant simulation of both singlet and triplet excitons, specify EXCITONS in the **MODELS** statement. Specifying SINGLET instead of EXCITONS will enable the solution of the singlet exciton equations, while specifying TRIPLET will enable the triplet exciton equation.

Statement	Parameter	Default
MODELS	EXCITONS	False
MODELS	SINGLET	False
MODELS	TRIPLET	False
MODELS	F.GENSINGLET	
MODELS	F.GENTRIPLET	

When solving for excitons, you can examine the density distribution of the singlet and triplet excitons in TonyPlot in files saved in the standard structure file format. You can also specify EXCITON in the **PROBE** statement to save the singlet exciton density to log files. For further analysis of LED devices, see Chapter 12: “LED: Light Emitting Diode Simulator”.

16.3.1 Dopants

The singlet exciton density for dopants [179] can also be calculated. If you specify the parameter DOPANT on the **ODEFECTS** statement, then the trap densities defined using the NA, SIGMAA, EA, ND, SIGMAD, and ED paramaters will be used in dopant singlet exciton calculation.

Note: Only one dopant is allowed per region.

The equation for dopant singlet excitons [179] is given by:

$$\frac{dS_d(x,y,t)}{dt} = \nabla(D_{sd}\nabla S_d(x,y,t)) + \texttt{DRST.EXCITON}\, R_{Ld} + K_{dd}\, S(x,y,t) - \frac{1}{2}\texttt{DKSS.EXCITON}\, S_d^2(x,y,t)$$
$$-S_d(x,y,t)\left(\texttt{DKNRS.EXCITON} + \frac{\texttt{DPHEFF.EXCITON}}{\texttt{DTAUS.EXCITON}} + \texttt{DKCQ.EXCITON}\,\frac{\texttt{DOPING.PERCENT}}{100} + Q(x,y)\right)$$
$$-\texttt{F.DSANNIHILATION}(x,y,t,S_d,n,p)$$

16-110

while the triplet exciton continuity equation for dopants is given by:

$$\frac{dT_d(x,y,t)}{dt} = (1-\texttt{DRST.EXCITON})R_{Ld} + \texttt{KEE.EXCITON}\,T(x,y,t)$$
$$-T_d(x,y,t)\left(\frac{1}{\texttt{DTAUT.EXCITON}} + \texttt{DKTP.EXCITON}(N_{dd}+N_{da})\right)$$

16-111

where

- $S_d(x,y,t)$ is the spatial and temporal singlet exciton density for dopants.
- $T_d(x,y,t)$ is the spatial and temporal triplet exciton density for dopants.
- DRST.EXCITON is the fraction of singlets formed.
- DKNRS.EXCITON is the dopant nonradiative decay rate.
- DTAUS.EXCITON is the singlet dopant radiative decay lifetime.
- DTAUT.EXCITON is the triplet dopant radiative decay lifetime.
- DKCQ.EXCITON is the dopant concentration quenching rate.
- DOPING.PERCENT is the doping percent specified on the **ODEFECTS** statement.
- DKSS.EXCITON is the bimolecular annililation constant.
- DKTP.EXCITON is the dopant triplet-polaron constant.
- $Q(x,y)$ is the metal quenching rate.
- DPHEFF.EXCITON is the dopant photoluminescent quantum efficiency.
- F.DSANNIHILATION specifies the dopant singlet exciton annihiliation rate as a function of position, time, dopant singlet density, electron concentration, and hole concentration via a C-Interpreter function.
- The dopant singlet diffusion constant, D_{Sd} can be expressed as

$$D_{Sd} = \frac{\texttt{DLDS.EXCITON}^2}{\texttt{DTAUS.EXCITON}}$$

16-112

where DLDS.EXCITON is the dopant singlet diffusion length.

- R_{Ld} is the dopant Langevin recombination rate given by:

$$\boldsymbol{R}_{Ld} = \frac{q}{\varepsilon\varepsilon_o}(DAD.LANGEVINn\, N_{dd}\mu_n(E) + DAA.LANGEVINp\, N_{da}\mu_p(E))$$

16-113

where

- N_{dd} is the density of occupied dopant donor traps.
- N_{da} is the density of occupied dopant acceptor traps.
- DAD.LANGEVIN and DAA.LANGEVIN are prefactors to tune the efficiency of the Langevin recombination.

An additional Langevin recombination term [179] is also added to the electron (R_{dn}) and hole (R_{dp}) current continuity equations:

$$R_{dn} = \frac{DAD.LANGEVIN n N_{dd} \mu_n(E)}{\varepsilon \varepsilon_o} \quad \text{16-114}$$

$$R_{dp} = \frac{DAA.LANGEVIN p N_{da} \mu_p(E)}{\varepsilon \varepsilon_o} \quad \text{16-115}$$

Statement	Parameter	Default	Units
MATERIAL	DAD.LANGEVIN	1	
MATERIAL	DAA.LANGEVIN	1	
MATERIAL	DLDS.EXCITON	0	microns
MATERIAL	DKCQ.EXCITON	1.1×10^{-8}	s^{-1}
MATERIAL	DKNRS.EXCITON	0.0	s^{-1}
MATERIAL	DKSS.EXCITON	3.0×10^{-8}	$cm^3\ s^{-1}$
MATERIAL	DKTP.EXCITON	0	s^{-1}
MATERIAL	DPHEFF.EXCITON	1.0	
MATERIAL	DRST.EXCITON	0.25	
MATERIAL	DTAUS.EXCITON	5.0×10^{-9}	s
MATERIAL	DTAUT.EXCITON	0	s
MATERIAL	F.DSANNIHILATION	0.25	
MATERIAL	KDD.EXCITON	0.0	s^{-1}

16.3.2 Light Generation of Excitons

Optical generation of of excitons is available in Organic Solar. All of the capabilities available in Luminous are also available in Organic Solar for the description of the propogation and absorption of light. If you are interested in photogeneration in organic materials, then read Chapter 11: “Luminous: Optoelectronic Simulator”.

The additional parameter QE.EXCITON of the **MATERIAL** statement specifies the fraction of the absorbed photons that generate singlet excitons.

The fraction 1-QE.EXCITON of the absorbed photons leads to the generation of electron hole pairs. By default, the value of QE.EXCITON is 0. For Organic Solar analysis, a value of one is more recommended. The actual detection mechanism requires exciton dissociation as described in the next section.

16.3.3 Exciton Dissociation

Another source of carrier generation in the carrier continuity equations (Equations 3-3 and 3-4) is due to exciton dissociation. Exciton dissociation is the dissociation of a neutral singlet exciton into a free electron hole pair. By default, the dissociation rate is given by:

$$R_D, n, p = \frac{3r_L(x,y,t)}{4\pi \mathrm{A.SINGLET}^3} exp\left(-\frac{\mathrm{S.BINDING}}{kT}\right)\frac{J_1(2\sqrt{-2b})}{\sqrt{-2b}}S(x,y,t) \quad \text{16-116}$$

Here, r_L is the Langevin recombination rate constant (Equation 16-62). A.SINGLET and S.BINDING are user-specified parameters as described in Table 16-16 representing the electron and hole separation distance and the singlet exciton binding energy. J_1 is the first order Bessel function and $S(x,y,t)$ is the local density of singlet excitons.

The parameter b is given by:

$$b = \frac{q^3|E|}{8\pi\varepsilon_r\varepsilon_0 k^2 T^2} \quad \text{16-117}$$

where E is the local electric field and ε_r is the relative permittivity.

Generally, the geminate pair distance can be described by a distribution function specified by SDISS.TYPE on the **MODELS** statement as follows [102]:

$$\begin{aligned} &\mathrm{SDISS.TYPE} = 1 \quad && f(r) = \delta(r-a) \\ &\mathrm{SDISS.TYPE} = 2 \quad && f(r) = r^2 e^{-r^2/a^2} \\ &\mathrm{SDISS.TYPE} = 3 \quad && f(r) = e^{-r^2/a} \\ &\mathrm{SDISS.TYPE} = 4 \quad && f(r) = r^2 e^{-r/a} \end{aligned} \quad \text{16-118}$$

The dissociation rate is also included in the singlet continuity equation described in Section 16.3 “Singlet and Triplet Excitons”. To include the singlet dissociation rate, you must specify S.DISSOC on the **MODELS** statement and also enable Langevin recombination.

Statement	Parameter	Default	Units

MATERIAL	A.SINGLET	1.8	nm
MATERIAL	QR.EXCITON	0	cm^3/s
MATERIAL	S.BINDING	0.1	eV

In the geminate pair dissociation model, the geminate pair separation distance A.SINGLET and the geminate pair binding energy S.BINDING are independent fitting parameters. This can be slightly non-physical. One could suggest that we apply Coulomb's law to link the two parameters as if they have the same dependence as for ion pairs.

This correction can be expressed using a new parameter K.SINGLET in Equation 16-119.

$$S.BINDING = K.SINGLET * \frac{q^2}{A.SINGLET * 4\pi\varepsilon_r\varepsilon_o} \quad 16\text{-}119$$

Here, K.SINGLET is a user-specifiable parameter expressing the relationship between S.BINDING and A.SINGLET. Ideally, K.SINGLET should have a value of one. You can independently express any two of the parameters and the third will be automatically calculated. The values of S.BINDING and A.SINGLET are eventually used in Equation 16-117.

16.3.4 Metal Quenching

The metal quenching rate is given by:

$$Q(x,y) = QR.EXCITON \cdot M(x,y) \quad 16\text{-}120$$

where $M(x,y)$ is the metal atomic density. QR.EXCITON is the quenching constant specifiable on the **MATERIAL** statement. To specify the metal atomic density, use the METAL parameter on the **DOPING** statement. The defaults for QR.EXCITON are given in Table 16-16.

16.3.5 Thermionic Emission

If thermionic emission is activated across an interface (see Section 6.2.2 “The Thermionic Emission and Field Emission Transport Model”), then the singlet and triplet currents across this interface are

$$J_s = v_{ex}^{+} s^{+} G^{+}(Q.THERMIONIC\Delta E_g) - v_{ex}^{-} s^{-} G^{-}(Q.THERMIONIC\Delta E_g) \quad 16\text{-}121$$

$$J_t = v_{ex}^{+} t^{+} G^{+}(Q.THERMIONIC\Delta E_g) - v_{ex}^{-} t^{-} G^{-}(Q.THERMIONIC\Delta E_g) \quad 16\text{-}122$$

s^+ and s^- are the singlet densities on either side of the interface. t^+ and t^- are the triplet densities on either side of the interface. The exciton thermal velocity, v_{ex}, is the average of the electron and hole thermal velocity.

$$v_{ex} = (v_n + v_p)/2$$

G^+ is the probability that an exciton will cross from the "+" region to the "-" region. G^- is the probability than an exciton will cross from the "-" region to the "+" region. This probability is the product of the probability of an electron crossing the boundary and the probability of a hole crossing the boundary.

$$G(Q.THERMIONIC\Delta E_g) = G_n(Q.THERMIONIC\Delta E_c)G_p(Q.THERMIONIC\Delta E_v) \quad \text{16-123}$$

QEX.BARRIER can be used to set a quality for just exciton thermionic emission across a barrier. For example if Q.THERMIONIC=1 but QEX.BARRIER=0 then there would be barriers to electron and hole transport across an interface, but no barrier to exciton transport.

16.3.6 C-Interpreter Defined Exciton Coefficients

C-Interpreter functions can be used to define position dependent coefficients for the exciton continuity equations. In the following (x,y,z) is the position (in μm), where we want to calculate the coefficients. The functions are

```
int kexciton(double x, double y, double z,
        double *rst, double *rsttt, double *kisc, double *ktt);
```

These are the coefficients that effect both the singlet and triplet continuity equations. The returned values correspond to RST.EXCITON, RST.TT.EXCITON, KISC.EXCITON, and KTT.EXCITON.

```
int ksinglet(double x, double y, double z,
         double *taus, double *pheff, double *lds,
         double *knrs, double *ksp, double *kss, double *kst);
```

These are the coefficients that effect only the singlet continuity equation. The returned values correspond to TAUS.EXCITON, PHEFF.EXCITON, LDS.EXCITON, KNRS.EXCITON, KSP.EXCITON, KSS.EXCITON, and KST.EXCITON.

```
int ktriplet(double x, double y, double z,
             double *taut, double *ldt, double *knrt, double *ktp);
```

These are the coefficients that effect only the triplet continuity equation. The returned values correspond to TAUT.EXCITON, LDT.EXCITON, KNRT.EXCITON, and KTP.EXCITON.

To use these C-Interpreter functions, the syntax is

```
MODELS F.KEXCITON=<file1> F.KSINGLET=<file2> F.KTRIPLET=<file3>
```

See Appendix A "C-Interpreter Functions" for more information regarding the C-Interpreter and its functions.

16.4 The Holstein Model

Organic light-emitting diodes (OLEDs) have gained particular interest in the recent years, thanks to their fabrication flexibility and design versatility. This section describes the Holstein model for a molecular chain with Frenkel excitons, which is a powerful tool for simulating OLED devices. The underlying Hamiltonian for the system is as follows [212]:

$$H_{\mathrm{Hol}} = \mathrm{J}\sum_{n}(a^{\dagger}{}_{n}a_{n+1} + a^{\dagger}{}_{n+1}a_{n}) + E_{ex}\sum_{n}a^{\dagger}{}_{n}a_{n} + \sum_{n}b^{\dagger}{}_{n}b_{n} + \sum_{n}a^{\dagger}{}_{n}a_{n}(-g(b^{\dagger}{}_{n} + b_{n}) + g^{2})$$

where the $a^{\dagger}{}_{n}$, a_{n}, $b^{\dagger}{}_{n}$, b_{n} are creation and annihilation operators for a Frenkel exciton at site n and creation and annihilation operators for a phonon localized at site n, respectively. The first two terms describe exciton hopping to the nearest neighbor, where J (EX.HOPPING) is the excitation transfer integral normalized to phonon energy $\hbar\omega$ (EX.EPHONON). The third term counts the number of excitons with E_{ex} (EXCITON.GAP) being the energy gap between an exciton-less and an exciton-full molecule. The fourth term, $b^{\dagger}{}_{n}b_{n}$ counts the number of phonons at site n, which equals total phonon energy for that molecule due to normalization by $\hbar\omega$. The remaining terms describe the linear exciton-phonon interaction, where g (EX.COUPLING) is the exciton-phonon coupling. As this model allows for at most 1 exciton per molecular, the $a^{\dagger}{}_{n}a_{n}$ term evaluates to either 0 or 1.

The electronic ground state of the system is denoted by $|0^{el}\rangle$ and one has

$$|n\rangle \equiv a^{\dagger}{}_{n}|0^{el}\rangle \qquad \text{16-124}$$

The basis states take the form:

$$|n\underline{v}\rangle \equiv |n\rangle|...v_{-1}\tilde{v}_{0}v_{1}...\rangle \equiv a^{\dagger}{}_{n}|0^{el}\rangle \times \frac{1}{\sqrt{v_{0}!}}(\tilde{b}^{\dagger}{}_{n})^{v_{0}}|\tilde{0}_{n}\rangle \times \prod_{m\neq n}\frac{1}{\sqrt{v_{m}!}}(b^{\dagger}{}_{m})^{v_{m}}|0_{m}\rangle \qquad \text{16-125}$$

where $\underline{v}$ represents a cloud of phonons in occupation-number form around the exciton at site n denoted by relative indices, so that there are v_{1} phonons at site n+1, v_{-1} phonons at site n - 1 and so on. The maximum extent of $\underline{v}$ is represented by GS.MAXCLOUDSIZE in the case of exciton-less ground states and EX.MAXCLOUDSIZE in the case of excited states, which indicate the number of sites from the center of the cloud to either edge of the cloud. Likewise, the maximum total number of phonons in the cloud $\left(\sum_{m}v_{m}\right)$ is denoted by

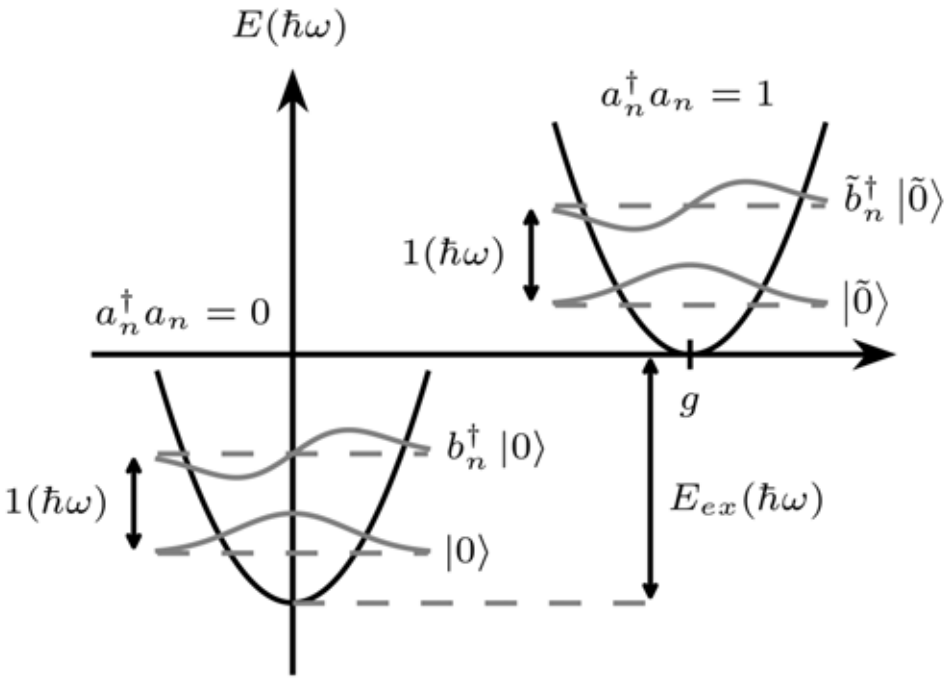


Figure 16-2: The displaced harmonic oscillator description of the Holstein model

GS.MAXPHONONS and EX.MAXPHONONS for exciton-less and exciton-full states, respectively. $|0_m\rangle$ is the absolute ground state for the undisplaced quantum oscillator, denoted by $a^\dagger{}_n a_n = 0$ in Figure 16-2, representing 0 phonons in an exciton-less molecule and $|\tilde{0}_n\rangle = e^{-|g|^2/2} D(g)|0_n\rangle$ is the absolute ground state for the displaced Lang-Firsov quantum oscillator, denoted by $a^\dagger{}_n a_n = 1$ in Figure 16-2, representing 0 phonons in a molecule with one exciton. The corresponding displaced creation and annihilation operators and the displacement operator are defined as

$$\tilde{b}^\dagger{}_n \equiv b^\dagger{}_n - g \qquad \text{16-126}$$

$$\tilde{b}_n \equiv b_n - g \qquad \text{16-127}$$

$$D(\alpha) = e^{\alpha a^\dagger - \alpha^* a} \qquad \text{16-128}$$

It is interesting to note that the interaction terms of the Hamiltonian will take the appropriate number-operator form depending on whether the Hamiltonian is acting on a displaced or undisplaced oscillator. If there is no exciton at site n, $a^\dagger{}_n a_n$ equals zero and one is left with $b^\dagger{}_n b_n$, which is the phonon-number operator. Conversely, if $a^\dagger{}_n a_n = 1$, then

$$b^{\dagger}_n b_n + a^{\dagger}_n a_n(-g(b^{\dagger}_n + b_n) + g^2) = b^{\dagger}_n b_n - g(b^{\dagger}_n + b_n) + g^2 = \tilde{b}^{\dagger}_n \tilde{b}_n \quad \text{16-129}$$

which again takes the form of the number operator but this time for displaced phonons (see Figure 16-2).

Equation 16-126 can be Fourier transformed to arrive at the momentum-space representation of states:

$$|\underline{\nu};k\rangle \equiv \frac{1}{\sqrt{N}}\sum_n e^{ikn}|n\underline{\nu}\rangle \quad \text{16-130}$$

which uses the translational symmetry of the molecular chain to render the quasi-momentum k to a fixed parameter, thanks to the nearest-neighbor assumption in the Hamiltonian. The parameter k is therefore omitted from our notation in the following. The final eigenstates are written as a linear combination of these basis states. For example, an eigenstate with energy E_j will have the form:

$$|\varphi_j\rangle = \sum_{\underline{\nu}} u_{\underline{\nu}j}|\underline{\nu}\rangle \quad \text{16-131}$$

Atlas uses the GS/EX.MAXCLOUDSIZE and GS/EX.MAXPHONONS parameters to construct the set of phonon-cloud basis states. It then calculates the absorption scattering rate from any 0-exciton (ground) state to any 1-exciton (excited) state using [282]:

$$R_{\mathrm{gs}\to\mathrm{ex}}(\hbar\omega) = \frac{1}{\pi\hbar}\int dE \mathrm{Tr}[G^r_{gs}(E;\eta)\Sigma^{<}_{ex}(E+\hbar\omega) + G^{<}_{gs}(E)\Sigma^{a}_{ex}(E+\hbar\omega;\eta)] \quad \text{16-132}$$

where $G^{r/<}_{gs}$ is the retarded / less-than Green function for the 0-exciton manifold, $\Sigma^{</a}_{ex}$ is the less-than / advanced self-energy for the excitonically-excited manifold and η (EX.HOMOBROADENING) is the energy broadening.

The ADDPOLARON flag in the **MATERIAL** statement is used to add an exciton-polaron that can optionally be associated with a unique identifier specified by the MIX.NAME parameter in the same **MATERIAL** statement. This unique identifier is referenced when adding extra states via the ADDSTATE parameter as discussed below. The type of the particle can be specified to be either SINGLET or TRIPLET in the same statement as ADDPOLARON. As an example:

```
material region=2 addpolaron mix.name=DCJTB singlet \
exciton.gap=2.25 ex.hopping=0.1 ex.coupling=1.8 \
ex.maxphonons=6 ex.maxcloud=0 ex.ephonon=0.065 \
ex.homobroadening=0.01 gs.maxphonons=12 gs.maxcloud=0
```

The Holstein model is activated via the HOLSTEIN flag in the **MODELS** statement. It can optionally take parameters such as QSPEC.RES to specify the energy resolution in eV for the optical spectrum and QSPEC.EMIN and QSPEC.EMAX to specify the lower and upper limits of the spectral energy axis.

The following parameters in the **MATERIAL** statement further allow adding extra energy levels, modeled as Voigt lineshapes. These levels are often needed to account for the effects of higher energy states (not Frenkel excitons), on the absorption spectrum. The vibrational modes of electronic ground state other than the single phonon branch in the model can also be added, which account for the low energy featureless tail of the emission spectrum. The added

state is associated with the mixture member specified using MIX.NAME, which must have been defined in an earlier ADDPOLARON statement.

- ADDSTATE indicates that a state is to be added to the material
- EX.ECEN adds a level in the excited state manifold
- GS.ECEN adds a level in the electronic ground state manifold
- EX.AMPLITUDE specifies the amplitude of the lineshape for the level at EX.CEN relative to the exciton-polaron lineshape
- GS.AMPLITUDE specifies the amplitude of the lineshape for the level at
- GS.CEN relative to the exciton-polaron lineshape

Chapter 17
NOISE: Electronic Noise Simulator

17.1 Introduction

NOISE is an Atlas based product that simulates the small-signal noise generated by devices. Electronic noise results in an unavoidable degradation of a circuit's performance. It is important to understand the properties of noise to minimize its effect.

You should already be familiar with Atlas before using NOISE. If not, see Chapter 2 "Getting Started with Atlas".

17.2 Simulating Noise in Atlas

Atlas models the noise of a device by calculating the statistical behavior of equivalent random voltage sources at its ports. You can use the results of this calculation in a circuit simulator (e.g., SmartSpice) to minimize the disruption caused by the (inevitable) presence of noise. The following shows the minimum needed to simulate noise in Atlas.

1. Add the `noise` parameter to a **LOG** command. For example:

```
log outfile=noise.log inport=gate outport=drain noise
```

2. Use the `noise` parameter on the subsequent **SOLVE** command. For example:

```
solve noise frequency=100.0
```

NOISE is an extension of the AC analysis of a device (see Section 2.9.3 “Small-Signal AC Solutions” for more information on AC analysis). You can perform a noise simulation on any device (one-port or two-port) where small-signal AC analysis can be performed.

A direct AC analysis is automatically performed on the device before the noise simulation (i.e., you don’t need to perform the analysis).

17.3 Circuit Level Description of Noise

Noise is the name given to the random fluctuations in the currents and voltages of real devices.

The simple Drude model of an n-type resistor gives

$$\langle J \rangle = \langle \sigma \rangle \cdot E = q \cdot \langle n \rangle \cdot \langle v \rangle \quad \text{17-1}$$

where $\langle J \rangle$ is the mean current density, q is the electronic charge, $\langle n \rangle$ is the mean electron density, and $\langle v \rangle$ is the mean electron drift velocity. Time dependent fluctuations in the instantaneous electron density and drift velocity will cause fluctuations in the resultant current density.

The current at a contact of the resistor would be

$$i(t) = \langle i \rangle + \delta i(t) \quad \text{17-2}$$

where the $\delta i(t)$ term is the noise generated by the resistor.

A noisy device can be represented by external current sources added to the terminals of an ideal (noiseless) device (see Figure 17-1).

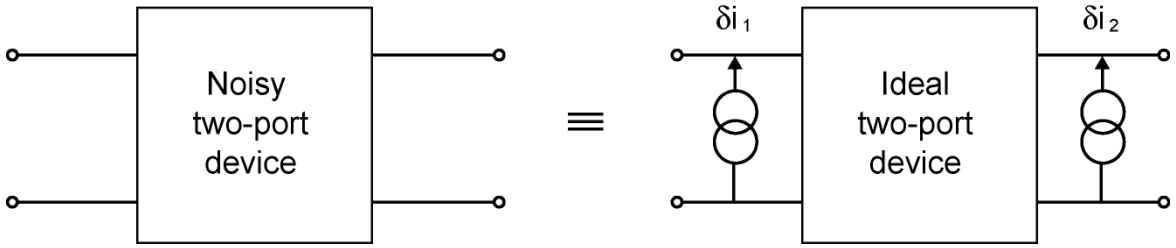


Figure 17-1: A real (noisy) device is modelled as an ideal (noiseless) device with random current sources attached to the ports.

These current sources are small and random. The description of noise is the statistical properties of these current sources. Normally, you describe the behavior of the frequency domain representation of the current sources.

The means of the current sources are zero. For example:

$$\langle \delta i_1(\omega) \rangle = 0 \quad and \quad \langle \delta i_2(\omega) \rangle = 0 \quad \text{17-3}$$

The noise is described by the auto-correlation:

$$\langle \delta i_1^2(\omega) \rangle \quad and \quad \langle \delta i_2^2(\omega) \rangle \quad \text{17-4}$$

and the cross-correlation:

$$\langle \delta i_1(\omega) \cdot \delta i_2(\omega)^* \rangle \quad and \quad \langle \delta i_2(\omega) \cdot \delta i_1(\omega)^* \rangle \quad \text{17-5}$$

It may be more convenient to describe the noise in terms of the behavior of voltage sources instead of current sources. The noise current can be easily translated into a noise voltage. For example:

$$\begin{pmatrix} \delta v_1(\omega) \\ \delta v_2(\omega) \end{pmatrix} = \begin{pmatrix} Z_{11}(\omega) & Z_{12}(\omega) \\ Z_{21}(\omega) & Z_{22}(\omega) \end{pmatrix} \cdot \begin{pmatrix} \delta i_1(\omega) \\ \delta i_2(\omega) \end{pmatrix} \quad \text{17-6}$$

Figure 17-2 shows that both representations are the same.

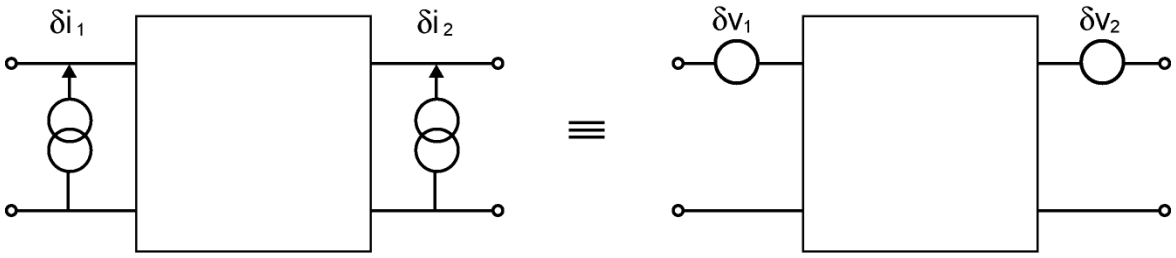


Figure 17-2: The noise can be modelled with random current sources or random voltage sources.

17.3.1 Noise "Figures of Merit" for a Two-Port Device

Figure 17-3 shows a simple two-port circuit.

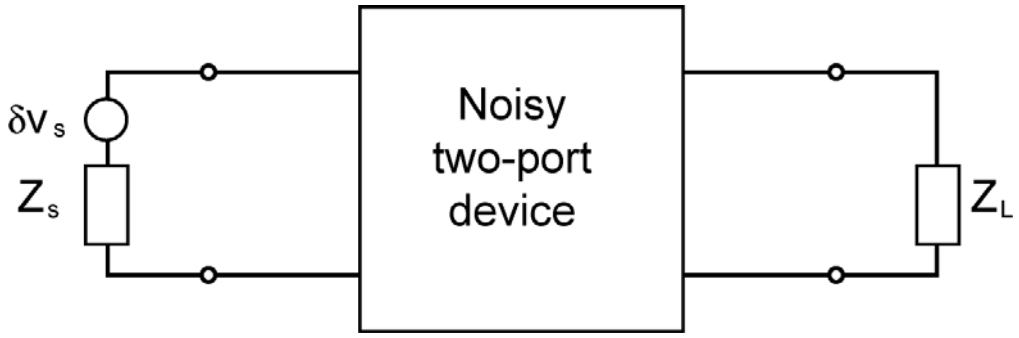


Figure 17-3: A block diagram of a simple circuit. We are interested in how much noise reaches the load.

The noise figure, *F*, is a measure of the increased amount of noise power delivered to the load because of the device's noise.

$$F = \frac{P_{S,D}}{P_S} \quad \text{17-7}$$

Here:

- P_S is the noise power that would be delivered to the load if the device was noiseless (i.e., only the noise from the source is transferred to the load).
- $P_{S,D}$ is the noise power that is delivered to the load from both the source and the device.

The standard definition of noise figure assumes the source is generating thermal noise at the temperature of the device:

$$\langle v_s^2 \rangle = 4kTR_s \quad \text{17-8}$$

Here, we have for the two-port circuit:

$$F = F_{min} + \frac{g_n}{R_s}|Z_S - Z_0|^2 \quad \text{17-9}$$

where the traditional two-port figures of merit for noise are as follows:

- the minimum noise figure, F_{min} ;
- the best source impedance, Z_0 ;
- the noise conductance, g_n.

The best source impedance gives the minimum noise figure (i.e., $F=F_{min}$ when $Z_S=Z_0$). The noise conductance is a measure of how much the noise figure increases as the source impedance moves away from Z_0.

17.4 Noise Calculation

Atlas does the following to calculate the noise of a device.

1. Takes a small volume of a device and calculates the random current flucluations in that volume.
2. Uses the impedence field to calculate the resultant voltage on a contact.
3. Repeats the above steps until the noise from each part of the device has been calculated, which is then the total noise of a device.

17.4.1 The Impedance Field

The impedance field is a transfer function relating current (injected at some point in the device) to the resultant contact voltage.

Suppose a current is injected into a point ***r*** of a device (see Figure 17-4).

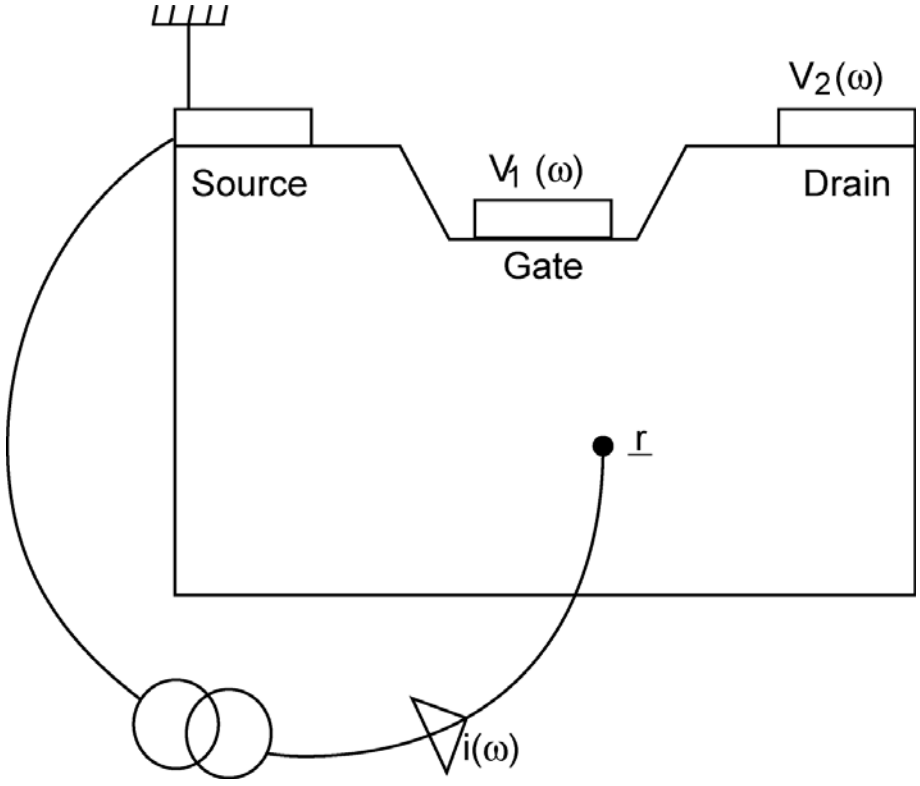


Figure 17-4: We calculate the impedance field at point *r* by injecting a unit current at the point getting the resultant voltages V1 and V2. (This is impossible to do in practice but is easy for Atlas to simulate.)

This will generate a voltage on the open circuit contacts. The device is assumed to be linear so that the voltage will be at the same frequency as the input current. Here, we get

$$v_1(\omega) = Z_1(\boldsymbol{r};\omega) \cdot i(\omega) \qquad 17\text{-}10$$

and

$$v_2(\omega) = Z_2(\boldsymbol{r};\omega) \cdot i(\omega) \quad 17\text{-}11$$

The $Zi(r;\omega)$ parameters, when taken over the whole device, are called the impedance fields.

There are separate impedance fields for when the injected current is electrons or holes. In other words, a two-port device has four associated impedance fields: $Z_1^n(\boldsymbol{r};\omega)$, $Z_2^n(\boldsymbol{r};\omega)$, $Z_1^p(\boldsymbol{r};\omega)$, and $Z_2^p(\boldsymbol{r};\omega)$.

17.4.2 Microscopic Noise Source

We assume the statistical behavior of the noise from one point in the device is totally uncorrelated with the statistical behavior of the noise at all other points. So we describe the correlation of a parameter, x, as:

$$\langle \delta x \cdot \delta x' \rangle(\boldsymbol{r}, \boldsymbol{r}';\omega) = \boldsymbol{k}_{\delta x,\, \delta x}(\boldsymbol{r};\omega) \cdot \delta(\boldsymbol{r} - \boldsymbol{r}') \quad 17\text{-}12$$

where the term $k_{\delta x,\, \delta x}(\boldsymbol{r};\omega)$ is called the microscopic noise source.

The theory of Generation-Recombination noise gives the auto-correlation of the current:

$$k_{\delta i,\, \delta i}(\boldsymbol{r};\omega) \quad 17\text{-}13$$

The theory of Diffusion noise gives the auto-correlation of the current density:

$$k_{\delta \boldsymbol{J},\, \delta \boldsymbol{J}}(\boldsymbol{r};\omega) \quad 17\text{-}14$$

17.4.3 Local Noise Source

The local noise source is the effect that a microscopic noise source has on the noise behavior of the device.

For $k_{\delta i,\, \delta i}(\boldsymbol{r};\omega)$ we have

$$\langle \delta v_\alpha, \delta v_\beta^{\,*} \rangle(\omega) = \int_\Omega Z_\alpha(\boldsymbol{r};\omega) k_{\delta i,\, \delta i}(\boldsymbol{r};\omega) \cdot Z_\beta^{\,*}(\boldsymbol{r};\omega) \cdot d\boldsymbol{r} \quad 17\text{-}15$$

where the integral is over the device volume. The integrand function is called the local noise source.

For $k_{\delta J,\, \delta J}(r;\omega)$ we have

$$\langle \delta v_\alpha, \delta v_\beta^{\,*} \rangle(\omega) = \int_\Omega \boldsymbol{Z}_\alpha(\boldsymbol{r};\omega) k_{\delta \boldsymbol{J},\, \delta \boldsymbol{J}}(\boldsymbol{r};\omega) \cdot \boldsymbol{Z}_\beta^{\,*}(\boldsymbol{r};\omega) \cdot d\boldsymbol{r} \quad 17\text{-}16$$

Again, the integral is over the device volume and the integrand function is called the local noise source.

The vector impedance field is defined as

$$\boldsymbol{Z}(r;\omega) = \nabla Z(r;\omega) \quad 17\text{-}17$$

17.5 Atlas Models

Atlas has models for three types of microscopic noise source: Diffusion Noise, Generation-Recombination Noise, and Flicker Noise.

If noise is being calculated then diffusion noise is on by default. Generation-recombination noise is on if any Generation-Recombination (GR) or Impact Ionization (II) models are defined. Flicker noise is off by default.

The default noise models for diffusion and GR/II have no user-definable parameters. They get all the needed information from the simulation.

17.5.1 Diffusion Noise

Diffusion noise is caused by variations in the velocity of the carriers.

Einstein Diffusion Equation

There are no independent user-definable parameters for this model. You can set the mobility model in a **MATERIAL** statement but that will affect the whole simulation not just the noise calculation. It is on by default and can only be turned off by choosing a different model for diffusion noise.

The equation for the microscopic noise source is

$$k_{\delta \boldsymbol{J}, \delta \boldsymbol{J}}(\boldsymbol{r};\omega) = 4q^2 n(\boldsymbol{r}) D(\omega) \qquad 17\text{-}18$$

where:

- q is the fundamental electronic charge.
- $n(\boldsymbol{r})$ is the carrier concentration.
- $D(\omega)$ is the diffusion coefficient.

The diffusion coefficient is given by Einstein's relationship, which is:

$$D(\omega) = \frac{kT}{q}\mu \qquad 17\text{-}19$$

where:

- k is Boltzmann's constant.
- T is the temperature.
- μ is the carrier mobility.

C-Interpreter

You can define a C-Interpreter function to calculate the microscopic noise source for diffusion noise. If a C-Interpreter function is defined, then the default model is turned off.

For a C-Interpreter function to calculate the electron microscopic noise source, use the **MATERIAL** statement:

```
material F.MNSNDIFF=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for electron diffusion noise.
 * Statement: MATERIAL
 * Parameter: F.MNSNDIFF
 * Arguments:
 * xcomp      composition fraction x
 * ycomp      composition fraction y
 * temp       lattice temperature (K)
 * n          electron concentration (cm^-3)
 * mun        electron mobility (cm^2/V s)
 * e          electric field (V/cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnsndiff(double xcomp, double ycomp, double temp, double n,
double mun, double e, double *mns);
```

For a C-Interpreter function to calculate the hole microscopic noise source, use the **MATERIAL** statement:

```
material F.MNSPDIFF=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for hole diffusion noise.
 * Statement: MATERIAL
 * Parameter: F.MNSPDIFF
 * Arguments:
 * xcomp      composition fraction x
 * ycomp      composition fraction y
 * temp       lattice temperature (K)
 * p          hole concentration (cm^-3)
 * mup        hole mobility (cm^2 / V s)
```

```
 * e          electric field (V / cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnspdiff(double xcomp, double ycomp, double temp, double p,
double mup, double e, double *mns);
```

17.5.2 Generation-Recombination Noise

Generation-Recombination noise is caused by variations in the number of the carriers.

There are no user-definable parameters for GR noise. You can, however, set the parameters for the various GR models in the **MATERIAL** statement and the various II models in the **IMPACT** statement. Setting these parameters will affect the whole simulation, not just the noise calculation.

There are two families of GR in Atlas: direct and trap assisted. Direct GR is when the electron travels directly from the conduction band to the valence band (or vice versa). Trap assisted GR is when the electron travels from a band to a trap level. If more than one trap level is defined, then they are treated as independent trap assisted GR sites (i.e., electrons are not assumed to move between different trap levels).

Generation-Recombination is defined by four parameters:

- G_n
- G_p
- R_n
- R_p

G is the generation rate, R is the recombination rate, n is for electrons, and p is for holes. The generation rate of electrons is the number of electrons that appears (per unit time, per unit volume) in the conduction band. The recombination rate is the number of electrons that disappears from the conduction band.

Direct GR

Since there are no trap levels, all electrons leaving the conduction band end up in the valence band. All electrons leaving the valence band end up in the conduction band.

Therefore:

$$G_n = G_p \quad and \quad R_n = R_p \tag{17-20}$$

The GR parameters are calculated with

$$G = C \cdot n_i^2$$
$$R = C \cdot np \tag{17-21}$$

For optical GR (OPTR in the **MODELS** statement), the C parameter is COPT in the **MATERIAL** statement. For Auger GR (AUGER in the **MODELS** statement), the C parameter is a function of the carrier density.

The microscopic noise sources are

$$k_{\delta n,\,\delta n} = k_{\delta p,\,\delta p} = k_{\delta n,\,\delta p} = 2(G + R) \qquad 17\text{-}22$$

Trap Assisted GR

There are assumed to be no direct transitions in trap assisted GR. Therefore, an electron leaving the conduction band ends up at the trap level. An electron entering the conduction band comes from the trap level. Unlike direct GR, the following four parameters are independent.

$$G_n = \frac{n_1}{\tau_n}\frac{n_t}{N_t}$$

$$G_p = \frac{p_1}{\tau_p}\left(1 - \frac{n_t}{N_t}\right)$$

$$R_n = \frac{n}{\tau_n}\left(1 - \frac{n_t}{N_t}\right) \qquad 17\text{-}23$$

$$R_p = \frac{p}{\tau_p}\frac{n_t}{N_t}$$

The ratio n_t/N_t is the fraction of ionized traps. The concentration n_1 is the electron concentration if the Fermi level were at the trap level. p_1 is the hole concentration if the Fermi level were at the trap level.

You can define the lifetimes τ_n and τ_p using the `TAUN0` and `TAUP0` parameters from the **`MATERIAL`** statement.

The microscopic noise sources are

$$k_{\delta n,\,\delta n} = 2(G_n + R_n) \qquad 17\text{-}24$$

There are no direct transitions between the conduction and valence band. Therefore:

$$k_{\delta p,\,\delta p} = 2(G_p + R_p) \qquad 17\text{-}25$$

Impact Ionization

Impact ionization is a pure generation term. For each electron created in the conduction band, a corresponding hole is also created in the valence band. This is similar to direct GR without a recombination term.

The microscopic noise sources are

$$k_{\delta n,\,\delta n} = k_{\delta p,\,\delta p} = k_{\delta p} = 2G \qquad 17\text{-}26$$

17.5.3 Flicker Noise

Flicker noise is observed experimentally, however, a complete theory about what causes flicker noise isn't known [34].

Hooge

The Hooge model, which is shown below, is a phenomenological microscopic noise source.

$$K_{\delta Jn,\,\delta Jn}(\boldsymbol{r};\omega) = \frac{\alpha_{Hn}}{f}\frac{\left|J_n(\boldsymbol{r})\right|^2}{n(\boldsymbol{r})} \quad and \quad K_{\delta Jp\ ,\delta Jp}(\boldsymbol{r};\omega) = \frac{\alpha_{Hp}}{f}\frac{\left|J_p(\boldsymbol{r})\right|^2}{p(\boldsymbol{r})} \qquad 17\text{-}27$$

In this model:

- α_{Hn} is the electron Hooge constant.
- α_{Hp} is the hole Hooge constant.
- f is the frequency.
- $J_n(\boldsymbol{r})$ is the electron current density.
- $J_p(\boldsymbol{r})$ is the hole current density.
- $n(\boldsymbol{r})$ is the electron concentration.
- p(r) is the hole concentration.

The Hooge constants are defined in the **MATERIAL** statement:

```
material HOOGEN=<value> HOOGEP=<value>
```

The default values for these parameters is zero (which means the flicker noise is turned off).

C-Interpreter

You can define a C-Interpreter function to calculate the microscopic noise source for flicker noise. If a C-Interpreter function is defined, then the default model is turned off.

For a C-Interpreter function to calculate the electron microscopic noise source, use the MATERIAL statement:

```
material F.MNSNFLICKER=<filename>
```

The header for this C-Interpreter function is:

```
/*
 * Microscopic Noise Source for electron flicker (1/f) noise.
 * Statement: MATERIAL
 * Parameter: F.MNSNFLICKER
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * temp       lattice temperature (K)
 * n          electron concentration (cm^-3)
 * jn         electron current density (A / cm^2)
 * f          frequency (Hz)
```

```
 * e          electric field (V / cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnsnflicker(double x, double y, double temp, double n,
double jn, double f, double e, double *mns);
```

For a C-Interpreter function to calculate the hole microscopic noise source, use the **MATERIAL** statement:

```
material F.MNSPFLICKER=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for hole flicker (1/f) noise.
 * Statement: MATERIAL
 * Parameter: F.MNSPFLICKER
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * temp       lattice temperature (K)
 * p          hole concentration (cm^-3)
 * jp         hole current density (A / cm^2)
 * f          frequency (Hz)
 * e          electric field (V/cm)
 * *mns       microscopic noise source (s A^2/cm)
 */
int mnspflicker(double x, double y, double temp, double p,
double jp, double f, double e, double *mns);
```

17.6 Output

The noise simulation results can output to the log file. The intermediate values in the calculation can output to the structure file.

17.6.1 Log Files

There are five groups of data that can output to a log file. They are as follows:

- The two-port figures of merit (FOM): F_{min}, Z_{opt}, g_n.
- The correlation of the total noise voltage sources.
- The correlation of the total noise current sources.
- The correlation of the individual noise voltage sources (GR, II, electron and hole diffusion; electron and hole flicker).
- The correlation of the individual noise current sources (GR, II, electron and hole diffusion; electron and hole flicker).

Table 17-1 shows the logical parameters in the **LOG** command that cause these groups to output. The columns in this table correspond to the groups above.

Table 17-1 Logical Parameters in the LOG command

	two-port FOM (a)	Total V correlation	Total I correlation	Individual V correlation	Individual I correlation
NOISE	X	_(b)	-	-	-
NOISE.V	X	X	-	-	-
NOISE.I	X	-	X	-	-
NOISE.IV	X	X	X	-	-
NOISE.VI	X	X	X	-	-
NOISE.V.ALL	X	X	-	X	-
NOISE.I.ALL	X	-	X	-	X
NOISE.ALL	X	X	X	X	X

Note a: If the device is a one-port, then this data will not be output: the two-port figures of merit have no meaning for a one-port device.

Note b: If the device is a one-port and NOISE is the only noise output in the **LOG** command, then the total noise voltage sources will output.

17.6.2 Structure Files

There are five groups of data that can output to a structure file. They are as follows:

- Total Local Noise Source (LNS)
- Impedance fields
- Short-circuit Green's function
- Individual Microscopic Noise Sources (MNS)
- Individual local noise sources

Table 17-2 shows the logical parameters from the **OUTPUT** command that cause these groups to output. The columns in this table correspond to the groups above.

Table 17-2 Logical Parameters in the OUTPUT command

	Total LNS	Impedance Fields	Short-circuit Green's function	Individual MNS	Individual LNS
NOISE	X	-	-	-	-
NOISE.IMP	-	X	-	-	-
NOISE.ALL	X	X	X	X	X

Chapter 18
Thermal 3D: Thermal Packaging Simulator

18.1 Overview

Thermal3D solves the heat transport equation in planar and non-planar three-dimensional structures. It can be used to compute steady-state temperature distributions, or in the case of time varying heating, the transient temperature distributions. You specify the heat sinks and heat sources on a region by region basis. Peltier heating or cooling may also be specified at relevant interfaces between regions. The thermal conductivity model and parameters can be chosen in each region. For the solution of the transient equation, the heat capacity model and parameters can be chosen in each region.

If you're unfamiliar with Thermal3D, see also Chapter 7 "3D Device Simulator". If you're unfamiliar with Atlas, see Chapter 2 "Getting Started with Atlas".

18.1.1 3D Structure Generation

Thermal3D supports structure defined on 3D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension. There are two methods for creating a 3D structure that can be used with Thermal3D. One method is through the command syntax of Atlas. The other method is through an interface to DevEdit. See Section 7.2 "3D Structure Generation" for more information on both methods.

Defining Steady-State Heat Sources

Heat sources are identified with regions in the 3D structure. Regions are defined in the manner documented in "Specifying Regions And Materials" on page 42 and "Atlas Syntax For 3D Structure Generation" on page 539. A region has a unique number, which is used to identify the region on the **MATERIAL** statement. The POWER parameter of the **MATERIAL** statement is used to set the power of the heat source in watts.

```
MATERIAL REGION=2 POWER=0.35
```

Set the POWER parameter in the **SOLVE** statement if you want to step the power through a range of values. See Section 18.4 "Obtaining Solutions In THERMAL3D" for the proper syntax.

Defining Time-Dependent Heat Sources

There are two options for the time dependence of heat sources. The first option is a linear ramp from one power to another power. These two powers are set on the **MATERIAL** statement and applied to a region. For example, the statement

```
MATERIAL REGION=2 POWERLO=1.0 POWERHI=2.0
```

sets the two values for region 2 in units of Watts. The POWERLO parameter is equivalent to the POWER parameter in the steady-state case. You choose the time of the linear ramp by one of two methods. The first way is to use the TP.RAMPTIME parameter on the **MATERIAL** statement. In this case, the statement

```
MATERIAL REGION=2 POWERLO=1.0 POWERHI=2.0 TP.RAMPTIME=1.0e-5
```

initializes the heat power in region 2 to be ramped from 1 Watt to 2 Watts over 10 microseconds.

If you omit TP.RAMPTIME from the **MATERIAL** statement, the ramp time will be obtained from the RAMPTIME parameter on the **SOLVE** statement. Any region that does not have the TP.RAMPTIME parameter set will use ramp time specified by the RAMPTIME parameter on the **SOLVE** statement.

The second option for specifying the time dependence of the heating powers is to use the SQPULSE flag on the **SOLVE** statement. The heat power of each region then follows a square wave pattern. You specify the low and high values for each region using the POWERLO and POWERHI parameters as described above. You specify the rise time from POWERLO to POWERHI with the TRISE parameter on the **SOLVE** statement. You specify the fall time from POWERHI to POWERLO by the TFALL parameter on the SOLVE statement. You can then either specify the overall pulse duration by using the FREQUENCY parameter on the **SOLVE** statement or the PULSE.WIDTH parameter on the SOLVE statement. PULSE.WIDTH is the overall pulse duration excluding the rise and fall times. The reciprocal of the FREQUENCY parameter is the overall pulse duration including the rise and fall times. For example, the statement

```
SOLVE TSTEP=1.0e-6 TFINAL=1.0 SQPULSE TRISE=1.0e-5 TFALL=1.0e-5
PULSE.WIDTH=2.0e-3
```

will define a sequence of repeating square pulses with individual duration of 2.02 milliseconds with rise and fall times of 10 microseconds each, and 1 millisecond at POWERLO and 1 millisecond at POWERHI. The initial timestep is 1 microsecond. The simulation will proceed until 1 second of pulses have been simulated.

See section 17.4.2 :'Obtaining transient solutions in THERMAL3D'

Note: The parameter TDELAY on the **SOLVE** statement is not active in Thermal3D.

Defining Peltier Heating and Cooling Terms

When electric current passes through interfaces between materials having different Peltier coefficients, heat will be generated or absorbed. You can model this in Thermal3D by using the PELTIER parameter on the **INTERFACE** statement. It does not apply to interfaces with electrical insulators.

For semiconductor-semiconductor interfaces, you must specify the S.S flag along with localization parameters. Set the PELTIER parameter to the rate of Peltier heating in W/cm^2, and use a negative value for Peltier cooling.

Defining Heat Sinks

Heat sinks are identified as electrodes in the 3D structure. Heat sink areas should be defined as electrodes in the manner documented in "Specifying Electrodes" on page 43 and "Atlas Syntax For 3D Structure Generation" on page 539.

Each electrode (heat sink) has a unique number, which is used to set the temperature on the heat sink during simulation. For more information on setting the temperature of the heat sink, see Section 18.4 "Obtaining Solutions In THERMAL3D".

18.2 Model and Material Parameter Selection

18.2.1 Thermal Simulation Model

Thermal3D is capable of solving the heat transport equation of the form

$$C(\vec{r}, T)\frac{\partial T}{\partial t} = \nabla \cdot (k(\vec{r}, T)\nabla T) + H(\vec{r}, t) \quad \text{18-1}$$

where C is the position and temperature-dependent heat capacity, T is the temperature, and k is the position and temperature-dependent thermal conductivity. The quantity H is the supplied heating power density that is spatially uniform within each region and may also have a time-evolution.

The time and position dependent temperature distribution is obtained by solving this equation with some boundary conditions. The specific boundary conditions are that the temperature is fixed on some electrodes. For boundaries where the temperature is not fixed, the heat flow across the boundary is assumed to be zero.

In the case where the heat generation term, H, is independent of time, then the equation simplifies to the Poisson equation

$$\nabla \cdot (k(\vec{r}, T)\nabla T) = -H(\vec{r}) \quad \text{18-2}$$

whose solution gives the steady-state temperature distribution. The units for H are Watts/cm^3, which is calculated for every region of the device by dividing the specified power for the region by its volume. Consequently, the units for heat capacity, C, are $J/K/cm^3$ and for thermal conductivity, k, are $W/cm/K$. See Section 18.4 “Obtaining Solutions In THERMAL3D” for details on how to solve Equations 18-1 and 18-2.

18.2.2 Setting Thermal Conductivity

To set the thermal conductivities, see Section 8.2.2 “Specifying Thermal Conductivity”. The default model for applicable materials is the `TCON.COMP` model.

18.2.3 Specifying Heat Capacity

To set the heat capacities, see Section 8.2.3 “Specifying Heat Capacity”. The default model for applicable materials is the `HC.COMP` model.

18.3 Numerical Methods

Solutions are obtained iteratively. You change the maximum number of iterations permitted using the ITLIMIT parameter on the **METHOD** statement.

An iterative solver is used by default to solve the linear equation system. You use a direct solver instead by specifying DIRECT on the **METHOD** statement. You also set some parameters for the transient solve using the **METHOD** statement as explained in Section 18.4.2 “Obtaining Transient Solutions In THERMAL3D”.

18.4 Obtaining Solutions In THERMAL3D

You may obtain steady-state and transient temperature solutions. In both cases, you specify the THERMAL parameter of the **MODELS** statement. How to obtain steady-state solutions is explained in Section 18.4.1 "Obtaining Steady-State Solutions In THERMAL3D". How to obtain transient solutions is explained in Section 18.4.2 "Obtaining Transient Solutions In THERMAL3D".

18.4.1 Obtaining Steady-State Solutions In THERMAL3D

The **SOLVE** statement in used in Thermal3D for heat-flow solutions much the same as it is used in other Atlas simulations involving electrical biases. The temperature in kelvin on each heat sink is used to prescribe the boundary condition temperatures. For example:

```
SOLVE T1=300 T2=500
```

sets the temperature at 300 K and 500 K on heat sink #1 and #2 respectively.

Multiple **SOLVE** statements are allowed in Thermal3D. This is useful for obtaining solutions for several combinations of heat sinks and thermal power sources. A range of solutions can also be obtained by stepping the value of a heat sink or power source. For example:

```
SOLVE T1=300 POWER2=0.35 POWER3=0.4 NSTEPS=5 STEPREGION=3 \
POWERFINAL=0.8 OUTFILE=thermal_out0
```

increments the thermal power source in region #3 from 0.4 watts to 0.8 watts in 5 steps and:

```
SOLVE T1=300 POWER2=0.35 NSTEPS=3 ELECTRODE=1 \
TEMPFINAL=600 OUTFILE=thermal_out0
```

increments the temperature on electrode #1 from 300 K to 600 K in 3 steps. Thermal power and temperature can be simultaneously sweep.

If more than one region power is specified during a sweep, the region to be stepped must be specified by STEPREGION=# as shown in the first example above. If the STEPREGION parameter is not specified, the smallest numerical value of POWER# is stepped.

During temperature and power sweeps, the output filename is modified according to the following rule. The rightmost character is incremented using the sequence: 0-9, A-Z, a-z. When a character is incremented beyond z, the character is set to 0> and the character to the left (smaller than 'z') is incremented.

Complete syntax information for the **SOLVE** statement can be found in Section 22.58 "SOLVE".

The **SOLVE** statement is used in Thermal3D for heat flow solutions much as it is used in other Atlas simulations for electrical biases. The temperature in Kelvin on each heat sink in the device must be set in the **SOLVE** statement. The Tn parameter, where n is the number of the heat sink, is used to set the temperature. For example:

```
SOLVE T1=300 T2=500
```

This sets 300K on heat sink #1 and 500K on heat sink #2.

18.4.2 Obtaining Transient Solutions In THERMAL3D

For a transient solution, you use a transient **SOLVE** statement and specify whether you want the heating power to be linearly ramped, or to follow a square wave pattern. This is outlined in "Defining Time-Dependent Heat Sources" on page 906 A single steady-state solve should always precede the transient solve in order to ensure that the correct steady-state temperature distribution corresponding to the values of the regional POWERLO parameters is present. The values of the temperatures of the heat sinks are set in an identical way to the steady state case. For transient solutions, you may set the values of heat capacity for each region as described in Section 8.2.3 "Specifying Heat Capacity".

The transient simulation uses the TR-BDF2 discretization and starts with the initial timestep you specify using the TIMESTEP (or TSTEP) parameter on the **SOLVE** statement. The algorithm estimates the local trancation error (LTE) at each step, and changes the step size automatically to attempt to keep the LTE below the value set using the TOL.TIME parameter on the **METHOD** statement. It is recommended to also set values of DT.MIN and DT.MAX on the **METHOD** statement, particularly for square pulse simulation. The simulation continues until it reaches the simulation given by the TFINAL parameter on the SOLVE statement.

For linearly ramped powers, the ramp time you specify with the RAMPTIME parameter is used by default, unless a region has a value of TP.RAMPTIME set.

For square wave powers, the same time parameters apply to all regions.

18.4.3 Boundary Conditions for Thermal simulations

In addition to the ability to make an electrode a heat sink, you can also use the **THERMCONTACT** statement in the same way as for Giga (see Section 8.2.7 "Thermal Boundary Conditions"). The conditions are always applied to the exterior of the relevant part of the device. In other words, the BOUNDARY flag on the **THERMCONTACT** statement is always set to true. For example, the syntax

```
THERMCONTACT NUMBER=1 ELEC.NUM=2 TEMPER=400
```

will create a thermal contact physically the same as electrode number 2 and apply a temperature of 400 Kelvin on the external surface of this electrode. This is different to the syntax

```
SOLVE T2=400
```

described above because in the latter case the entire electrode is set to 400 K. In the former case, heat conduction inside the electrode is modelled, and only the exterior of the electrode is set to 400 K.

Using the **THERMCONTACT** statement, both thermal conduction and thermal radiation boundary conditions can be modeled(see Section 8.2.7 "Thermal Boundary Conditions") . For example

```
THERMCONTACT NUMBER=1 TEMPER=300 ALPHA=0.0 BLACKBODY X.MIN=0.0
X.MAX=0.0 Y.MIN=0

Y.MAX=1000.0 Z.MIN=0.0 Z.MAX=500
```

applies ideal blackbody radiation conditions on the plane x=0, assuming that this is an external surface.

18.5 Interpreting The Results From THERMAL3D

The output of thermal simulation consists of the minimum and maximum calculated temperature of each region and its location. The three-dimensional temperature distribution can be saved in an output structure file by setting the OUTFILE parameter in the **SOLVE** statement and visualized using TonyPlot3D.

For transient simulations, a structure file can be output after every timestep by using the OUTFILE parameter on the **SOLVE** statement. The ONEFILEONLY flag causes only one structure file to be saved per transient **SOLVE** statement.

The runtime output consists of the device maximum and minimum temperatures at each timestep, as well as the volume averaged temperature. The Local Truncation Error estimate is also given, as well as current simulation time and next timestep.

The **LOG** statement can also be used for transient Thermal3D solutions. By default, a volume averaged temperature is output at every timestep. To obtain further temperature information, the THERMAL flag on the **PROBE** statement is used. All the usual **PROBE** functionality is available, and so the minimum and maximum temperatures in the structure can easily be written to the log file.

18.6 More Information

Many examples using Thermal3D have been provided on the distribution package. More information about the use of Thermal3D can be found by reading the text associated with each example.

You can also find some information at www.silvaco.com.

Chapter 19
Mercury: Fast FET Simulator

19.1 Introduction

Mercury is an Atlas module for the fast simulation of FET devices. The Mercury command syntax is similar to the Atlas command syntax. See Chapter 2 “Getting Started with Atlas” if you are unfamilar with the Atlas command syntax.

19.1.1 Accessing Mercury

To access the Mercury module, use the MERCURY command or the MERCURY parameter on the **MESH** command. This command must be specified immediately after the **GO** ATLAS command, so you can activate the Mercury module either with

```
GO ATLAS

MERCURY

MESH WIDTH=100
```

or with

```
GO ATLAS

MESH MERCURY WIDTH=100
```

19.2 External Circuit

The Mercury module can use a Harmonic Balance simulation to calculate the large signal behavior of a FET. As the large signal behavior of a device depends upon the circuit around the device, the Mercury module always embeds the FET under simulation in an external circuit. The system that Mercury simulates is shown in Figure 19-1.

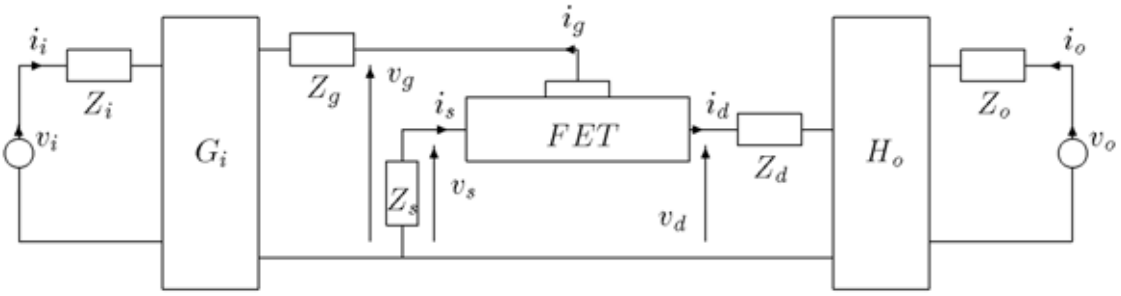


Figure 19-1: FET Embedded In An External Circuit

The input matching network is described by a set of G two-port parameters

$$\begin{pmatrix} I_i \\ V_g \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \cdot \begin{pmatrix} V_i \\ I_g \end{pmatrix} \qquad 19\text{-}1$$

The output matching network is described by a set of H two-port parameters

$$\begin{pmatrix} V_d \\ I_o \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \cdot \begin{pmatrix} I_d \\ V_o \end{pmatrix} \qquad 19\text{-}2$$

The contact elements Z_i, Z_o, Z_s, Z_g, and Z_d are simple impedances. The input and output matching circuits, G_i and H_o, are defined as two-port networks. At DC, all elements are assumed to be real. At RF, all elements can be complex.

The **NETWORK** command is used to define the characteristics of the external circuit.

For a DC or small-signal AC simulation, the external circuit is usually not wanted so the CLEAR parameter will remove the external circuit. This will set all the contact impedances to 0. The input matching network becomes

$$\begin{pmatrix} I_i \\ V_g \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} V_i \\ I_g \end{pmatrix} \qquad 19\text{-}3$$

which just applies the input voltage to the gate and passes the gate current to the output. The output matching network becomes

$$\begin{pmatrix} V_d \\ I_o \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} I_d \\ V_o \end{pmatrix} \qquad 19\text{-}4$$

which just applies the output voltage to the drain and passes the drain current directly to the output.

For a harmonic balance simulation, you must define the external circuit. The circuit should be defined at DC, the fundamental frequency, and at all the harmonics that are being modeled.

At DC, the contact impedances are set as resistances. You can define the input matching network as a set of G-parameters, S-parameters, or Z-parameters. You can define the output matching network as a set of H-parameters, S-parameters, or Z-parameters.

The RF circuits must be defined at a range of frequencies. There are several ways to do this. The first way is to define the impedance as an RLC triplet. This calculates the impedance as

$$Z = R + j.\omega.L - j/(\omega.C) \qquad 19\text{-}5$$

The default capacitance is infinite, so the $1/(\omega.C)$ term goes to zero if the capacitance is not set. The second way is to set an explicit value that is the same for all frequencies. The third way is to use the HARMONIC=<n> parameter and set a value for a single frequency (where HARMONIC=1 is the fundamental, f, HARMONIC=2 is 2.f, HARMONIC=3 is 3.f, and so on).

The RF contact impedances can be defined as RLC triplets, as admittances, or as impedances.

The RF input matching network can be defined as a G-parameter network, an S-parameter network, or as a Z-parameter network. If it is defined as a Z-parameter network, then you can set the z-parameters or you can define them as an RLC triplet.

The RF output matching network can be defined as a H-parameter network, an S-parameter network, or a Z-parameter network. If it is defined as a Z-parameter network, then you can set the Z-parameters or you can define them as an RLC triplet.

Example

```
NETWORK CLEAR
```

This removes the network and provides a clean slate to work with

```
NETWORK ZI.RE.RF=50.0 ZI.IM.RF=0.0 ZO.RE.RF=50.0 ZO.IM.RF=0.0
```

This sets the input and output impedance to 50 Ohms at all frequencies.

```
NETWORK RG.RF=0.01 LG.RF=1e-13
```

This sets a resistance and inductance on the gate contact, the impedance will be calculated at each frequency.

```
NETWORK GI.11.RE.RF=10.0 GI.11.IM.RF=5.0 GI.12.RE.RF=7.0 ...
NETWORK HARMONIC=1 ZI.11.RE.RF=...
```

This sets an input network at all harmonics and then changes the network at the fundamental.

19.3 Device

The FET in Figure 19-1 consists of an intrinsic FET and some parasitics as shown in Figure 19-2.

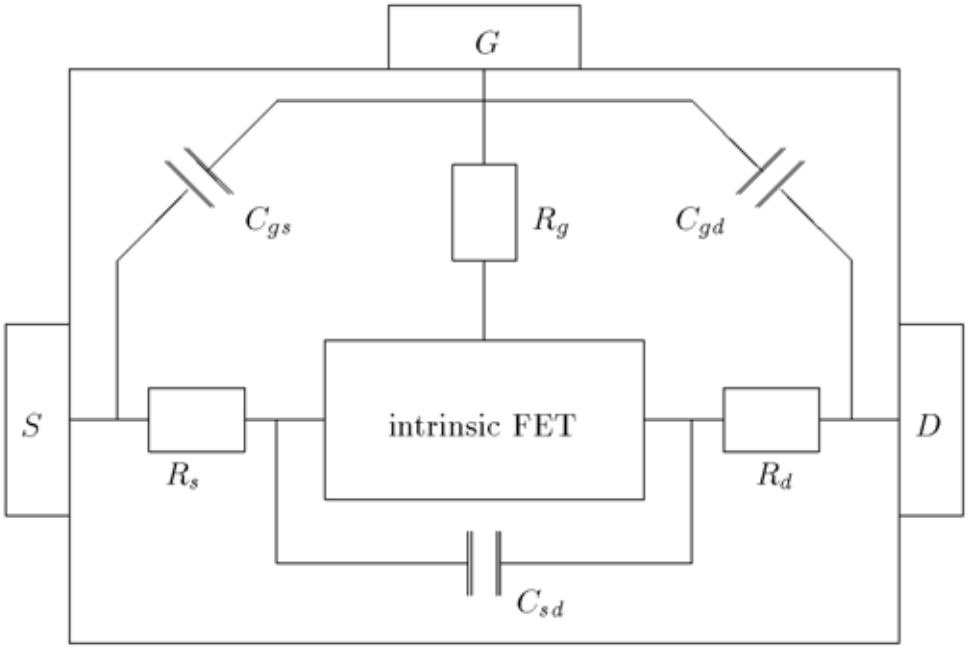


Figure 19-2: Intrinsic FET With Parasitics

The Mercury module imposes some restrictions on device structures and requires some assumptions of device geometry to use the quasi-2D method to speed up the simulation (see Figure 19-3). All material layers and doping profiles are planar: they have no variation in the x-directon. The only exceptions to this rule are the gate and a possible recess in the surface. The source and drain contacts are not considered part of the intrinsic device and are modeled as perfectly ohmic contacts to the channel on the left and right of the device.

The method for defining a device in the Mercury module is similar to the method used in Atlas. First, the **MESH** statements are used to set the z-depth of the device and to define the physical extent of the device in the xy-pane

```
MESH WIDTH=100

X.MESH LOCATION=0.0 SPACING=0.1

X.MESH LOCATION=1.1 SPACING=0.2

X.MESH LOCATION=2.5 SPACING=0.1

Y.MESH LOCATION=0.0 SPACING=0.01

Y.MESH LOCATION=3.0 SPACING=0.01
```

But the Mercury module only uses the extreme values of x and y to define the extent of the device, any locations between these two extremes (such as the "X.MESH LOCATION=1.1" command) are ignored.

Then **REGION** commands are used to define the epilayers. Only the y-extent of the regions are recognized by Mercury: each layer must cover the whole x-range of the device

```
REGION MATERIAL=GaAs THICKNESS=0.2
REGION MATERIAL=AlAs THICKNESS=0.3
REGION MATERIAL=GaAs THICKNESS=2.0
```

Then, **DOPING** commands are used to define the doping. Again, these commands can only have a y-variation. Any x-variation in the doping will be ignored.

```
DOPING UNIFORM N.TYPE CONCENTRATION=1e17 Y.MIN=0.0 Y.MAX=0.2
```

The only non-uniformity in the X direction is a recess in the surface of the device

```
SURFACE X=0.50 Y=0.00
SURFACE X=0.65 Y=0.15
SURFACE X=1.35 Y=0.15
SURFACE Y=1.50 Y=0.00
```

and the position of the gate

```
ELECTRODE NAME=gate X.MIN=0.7 X.MAX=1.3
```

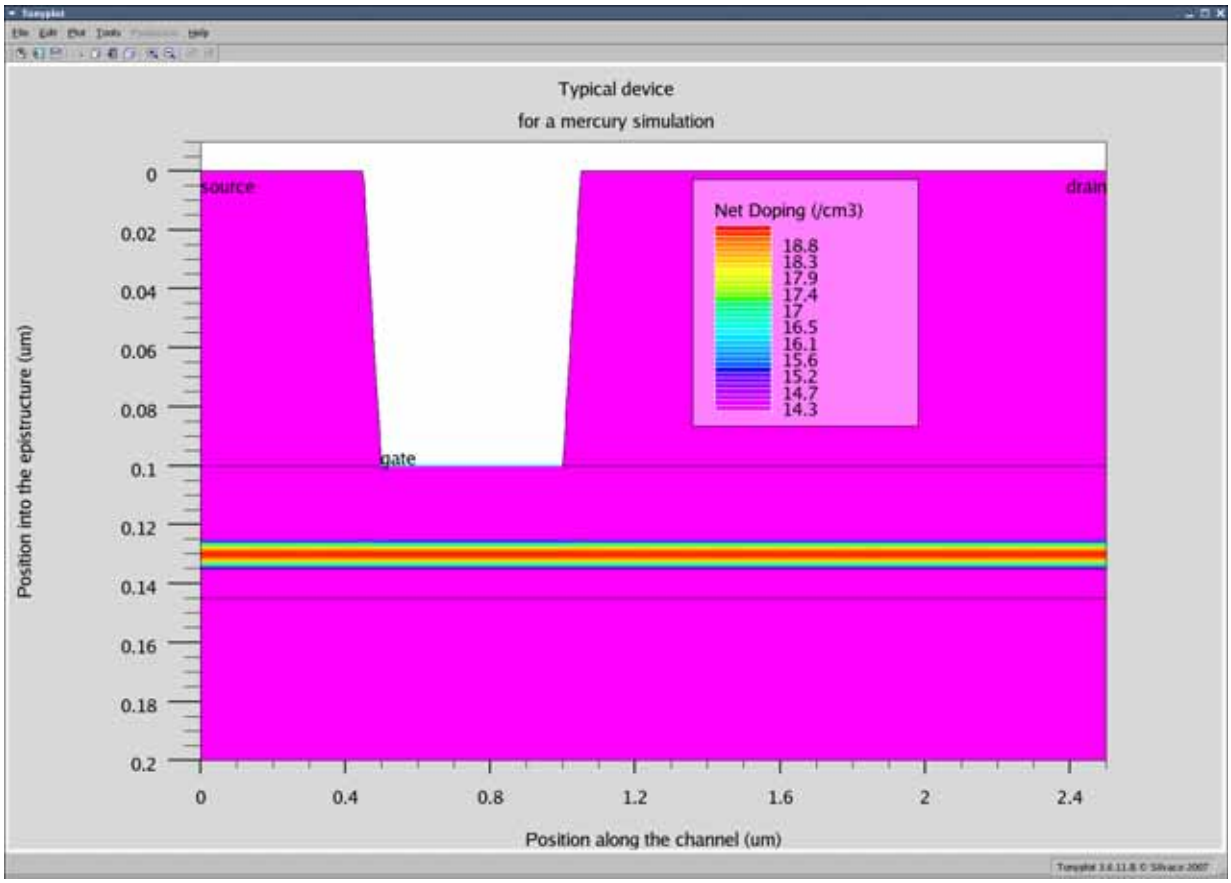


Figure 19-3: Typical Planar Device for a Mercury Simulation

19.4 Quasi-2D Simulation

The quasi-2D method has two parts. The first calculates Poisson's equation in the Y direction. The second calculates the transport equations in the X direction.

19.4.1 Poisson Equation

The first part of the simulation calculates Poisson's equation in the Y direction of the device over a range of surface conditions. This calculates the carrier density as a function of depth into the device (see Figure 19-4). The program then converts this profile into a "channel" with a width and a constant 3D density (see Figure 19-5).

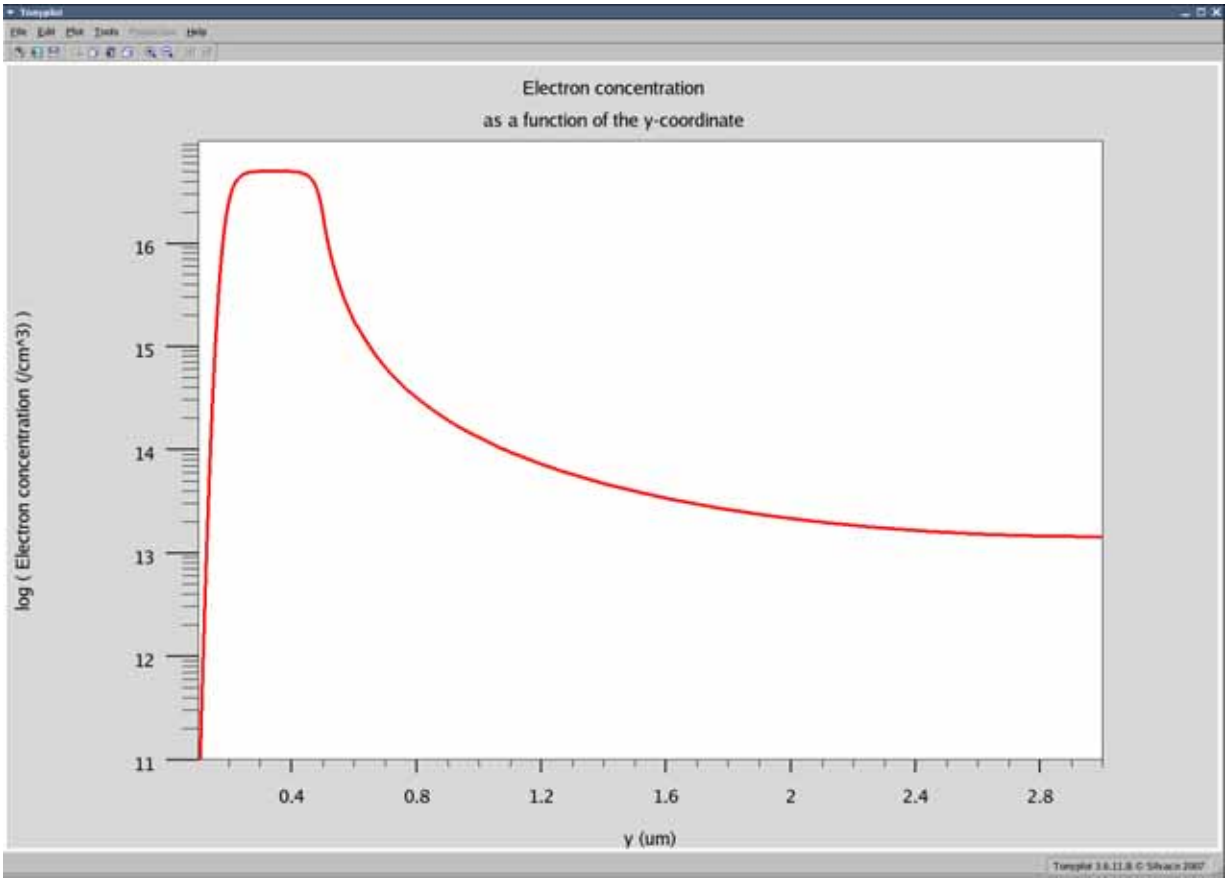


Figure 19-4: Carrier Density as a Function of Depth

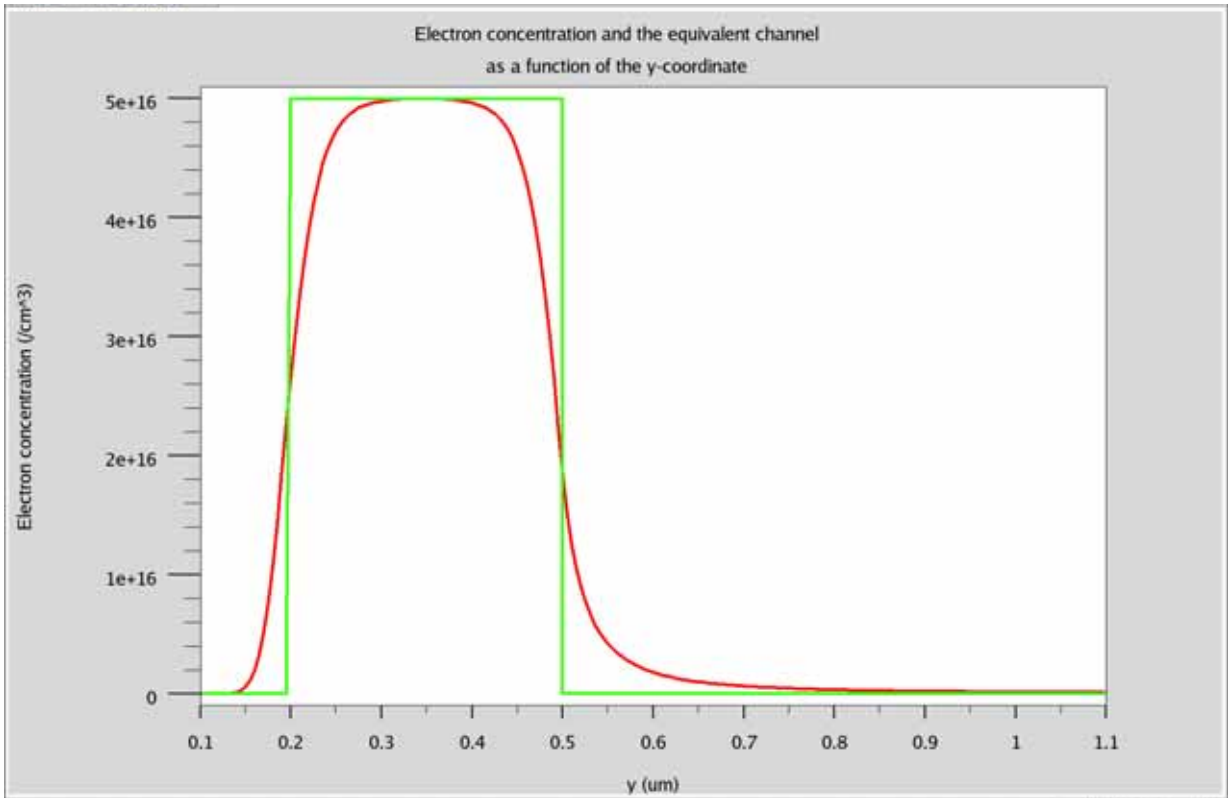


Figure 19-5: Channel with a Width and a Constant 3D Density

Two look-up tables are generated. One from the results of Poisson's equation for an ungated slice over the range of recess depths of the device (if the device is planar this is a single point). The other from the results of Poisson's equation for a gated slice over the range of contact voltages expected in the simulation, see Figure 19-6.

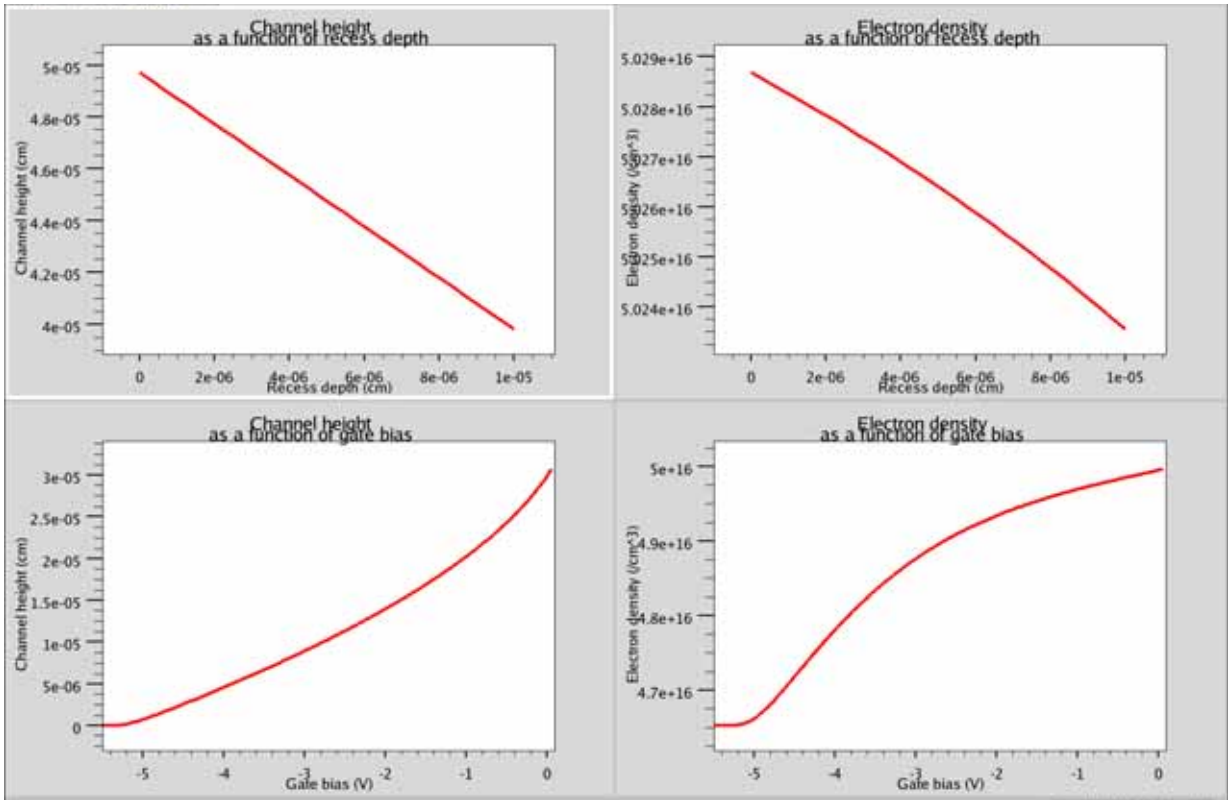


Figure 19-6: Channel Height and Electron Volume Density

The look-up tables are generated with the command

```
POISSON LOOK_UP VG_MIN=-2.5 VG_MAX=0.0 VD_MIN=0.0 VD_MAX=1.0
```

The parameters VG_MIN and VG_MAX are used to define the range of gate voltages. The parameters VD_MIN and VD_MAX are used to define the range of drain voltages. These ranges are needed to ensure that the gate look-up table is calculated over a sufficient range.

19.4.2 Channel Simulation

The main channel simulation is essentially one-dimensional. The device is divided into slices from the source to the drain. The transport equations are derived around the parallelogram surrounding a grid point (see Figure 19-7).

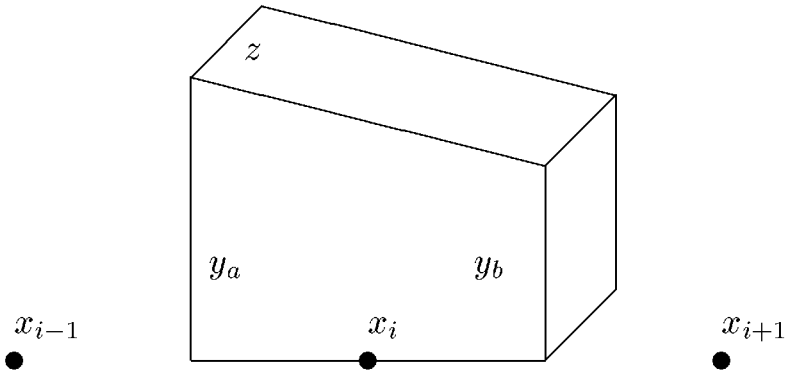


Figure 19-7: The section of the channel around the grid point xi.

Here, y_a and y_b are the heights of the channel (rather than the device). The volume of this parallelogram is

$$V = \frac{(x_{i+1} - x_{i-1})}{2} \cdot \frac{(y_a + y_b)}{2} \cdot z = 0.25 \cdot (x_{i+1} - x_{i-1}) \cdot (y_a + y_b) \cdot z \quad \text{19-6}$$

The resultant equations are similar to the standard set of equations used in the drift diffusion model.

Poisson's Equation

Mercury's version of Poisson's equation is derived by analyzing the electric fields on the surface of a section of the channel.

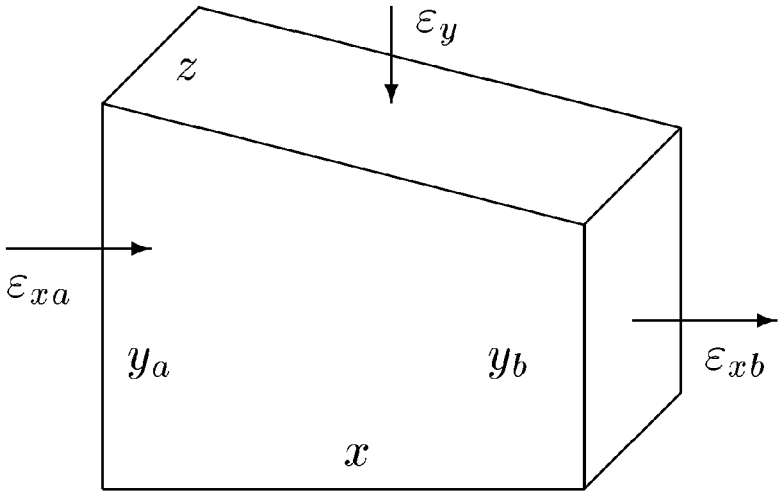


Figure 19-8: The electric fields around a section of the channel.

Start with Gauss's law

$$\int \varepsilon \cdot dA = \frac{Q}{\varepsilon} \qquad 19\text{-}7$$

Q is the total charge contained within the surface. Initially, $\varepsilon_{xa}=0$ and $\varepsilon_{xb}=0$.

Therefore

$$-\varepsilon_y \cdot x \cdot z = \frac{Q_0}{\varepsilon} \qquad 19\text{-}8$$

The relationship between Q_0 and ε_y is calculated in the charge-control simulation. Therefore, when you add the X direction electric fields you have

$$\varepsilon_{xb} \cdot y_b \cdot z - \varepsilon_{xa} \cdot y_a \cdot z - \varepsilon_y \cdot x \cdot z = \frac{Q}{\varepsilon} \qquad 19\text{-}9$$

Replace the ε_y term with the values from the charge control

$$\varepsilon_{xb} \cdot y_b \cdot z - \varepsilon_{xa} \cdot y_a \cdot z = \frac{Q - Q_0}{\varepsilon} \qquad 19\text{-}10$$

Write the charge in terms of the carrier density (where q is the charge on a hole, V is the volume of the parallelogram, and ρ is the density of carriers).

$$Q = q \cdot V \cdot \rho \qquad 19\text{-}11$$

Therefore, p is the density of holes, n is the density of electrons, N_D^+ is the density of ionized donors, and N_A^- is the density of ionized acceptors.

$$\rho = p - n + N_D^+ - N_A^- \quad \text{19-12}$$

If the ionization density is a constant, then

$$\rho - \rho_0 = (p - p_0) - (n - n_0) \quad \text{19-13}$$

Therefore

$$\varepsilon_{xb} \cdot y_b \cdot z - \varepsilon_{xa} \cdot y_a \cdot z = \left(\frac{q \cdot V}{\varepsilon} \cdot [(p - p_0) - (n - n_0)] \right) \quad \text{19-14}$$

In the limit $x \rightarrow 0$ ($y_a = y_b$), this becomes Poisson's equation. Mercury, however, implements the discrete version.

Continuity Equation

Mercury's version of the continuity equation is derived by analyzing the currents on the surface of a section of the channel.

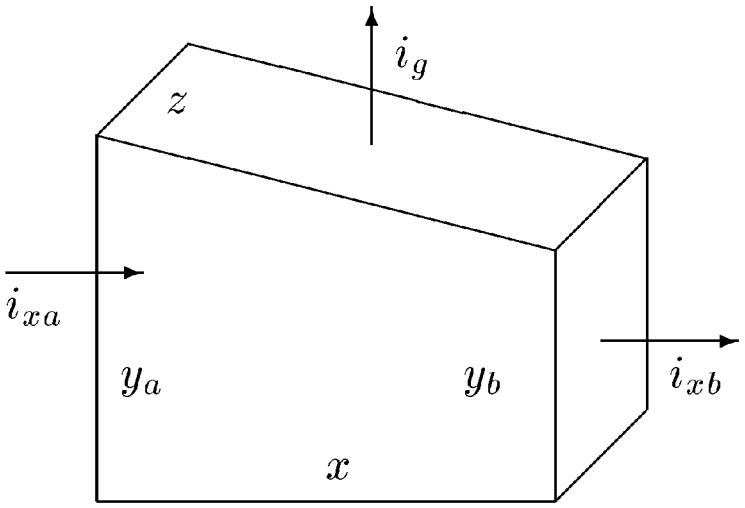


Figure 19-9: The currents around a section of the channel.

A current is a flow of positive charge. Therefore, the change in the number of carriers in a small time δt due to a current i is (s is the sign of the carriers)

$$\delta C = s \cdot \frac{1}{q} \cdot i \cdot \delta t \quad \text{19-15}$$

So in a small time δt, the change in the number of carriers in the parallelogram is

$$\delta C = s \cdot \frac{1}{q} \cdot i_{xa} \cdot \delta t - s \cdot \frac{1}{q} \cdot i_{xb} \cdot \delta t - s \cdot \frac{1}{q} \cdot i_g \cdot \delta t + genTerm \quad \text{19-16}$$

Here, *genTerm* is the number of carriers generated in the volume in the small time δt. Dividing through by δt

$$\frac{\delta C}{\delta t} = \frac{s}{q} \cdot i_{xa} - \frac{s}{q} \cdot i_{xb} - \frac{s}{q} \cdot i_g + \frac{genTerm}{\delta t} \quad 19\text{-}17$$

Writing in terms of the current density rather than the current and dividing by the volume

$$\frac{\delta C}{V \cdot \delta t} = \frac{s}{V \cdot q} \cdot J_{xa} \cdot y_a \cdot z - \frac{s}{V \cdot q} \cdot J_{xb} \cdot y_b \cdot z - \frac{s}{V \cdot q} \cdot J_g \cdot \frac{x_{i+1} - x_{x-1}}{2} \cdot z + \frac{genTerm}{V \cdot \delta t} \quad 19\text{-}18$$

Take the limit as $\delta t \rightarrow 0$ and get

$$\frac{dc}{dt} = \frac{s}{V \cdot q} \cdot J_{xa} \cdot y_a \cdot z - \frac{s}{V \cdot q} \cdot J_{xb} \cdot y_b \cdot z - \frac{s}{V \cdot q} \cdot J_g \cdot \frac{x_{i+1} - x_{i-1}}{2} \cdot z + U_c \quad 19\text{-}19$$

Here, c is now the density of the carriers and U_c is the generation rate of the carrier. In the limit as $x \rightarrow 0$, this becomes the standard continuity equation.

Channel Current

The channel current is described by the standard drift diffusion model written using the Bernoulli function

$$J = \frac{s \cdot k \cdot T \cdot \mu_c}{x_{i+1} - x_i} \cdot (B(-\Phi_{i,i+1}) \cdot c_i - B(-\Phi_{i,i+1}) \cdot c_{i+1}) \quad 19\text{-}20$$

with

$$\Phi_{i,i+1} = \frac{s \cdot q}{k \cdot T} \cdot (v_i - v_{v+1}) \quad 19\text{-}21$$

19.5 DC and Small-Signal AC

As mentioned in Section 19.2 “External Circuit”, the FET under simulation is embedded in an external circuit. But when investigating the DC or small-signal AC behavior, you will normally be interested in the characteristics of the bare FET. The **NETWORK** CLEAR command sets the external circuit so that the input voltage is applied directly to the gate and the output voltage is applied directly to the drain.

To calculate a set of DC-IV curves, you need to issue a set of commands like this (the DC-IV data would be saved to the file DC.LOG).

```
LOG OUTFILE=DC.LOG
SOLVE VINPUT=-0.0 VOUTPUT=0.0 VSTEP=0.5 VFINAL=5.0 NAME=output
SOLVE VINPUT=-0.5 VOUTPUT=0.0 VSTEP=0.5 VFINAL=5.0 NAME=output
SOLVE VINPUT=-1.0 VOUTPUT=0.0 VSTEP=0.5 VFINAL=5.0 NAME=output
SOLVE VINPUT=-1.5 VOUTPUT=0.0 VSTEP=0.5 VFINAL=5.0 NAME=output
SOLVE VINPUT=-2.0 VOUTPUT=0.0 VSTEP=0.5 VFINAL=5.0 NAME=output
```

In order to make a Mercury input deck as close to an Atlas input deck as possible, GATE is recognized as an alias for INPUT and DRAIN is recognized as an alias for OUTPUT. So the previous set of commands can been written as:

```
LOG OUTFILE=DC.LOG
SOLVE VGATE=-0.0 VDRAIN=0.0 VSTEP=0.5 VFINAL=5.0 NAME=drain
SOLVE VGATE=-0.5 VDRAIN=0.0 VSTEP=0.5 VFINAL=5.0 NAME=drain
SOLVE VGATE=-1.0 VDRAIN=0.0 VSTEP=0.5 VFINAL=5.0 NAME=drain
SOLVE VGATE=-1.5 VDRAIN=0.0 VSTEP=0.5 VFINAL=5.0 NAME=drain
SOLVE VGATE=-2.0 VDRAIN=0.0 VSTEP=0.5 VFINAL=5.0 NAME=drain
```

But be careful, VGATE is just an alias for VINPUT and VDRAIN is just an alias for VOUTPUT. Therefore, the command VGATE sets the voltage V_i (rather than the voltage V_g). The command VDRAIN sets the voltage V_o (rather than the voltage V_d). But if the external network has been cleared (as it usually will during a DC or small-signal AC simulation), setting V_i is equivalent to setting V_g and setting V_o is equivalent to setting V_d.

19.5.1 Small-Signal AC

In Mercury, the FET is always arranged in a common source configuration, port 1 is always the gate-source port, and port 2 is always the drain-source port. A typical set of commands for a small-signal AC simulation will be

```
SOLVE VGATE=-1.0 VDRAIN=3.0
LOG S.PARAM GAINS NOISE.V OUTFILE=AC.LOG
SOLVE AC NOISE FREQUENCY=1.0E6 FSTEP=1.2589 NFSTEP=40 MULT.FREQ
```

Mercury can calculate the small-signal noise of the device in addition to the small-signal AC behavior.

19.6 Harmonic Balance

Harmonic Balance simulates the behavior of a device when its input is large enough to cause a non-linear response.

The following shows the minimum needed to simulate the large-signal response in Mercury.

1. Solve at a dc bias point

```
SOLVE vinput=-0.5 voutput=2.0
```

2. Open a file to store the results.

```
log num.harmonic=3 outfile=hb.log
```

The harmonic balance simulation is slow so it's a good idea to record a result when it calculates.

3. Use the **SOLVE** command to simulate the large-signal behavior. You define the frequency and the range of the size of the input signal at this frequency.

```
SOLVE hb vinput=1e-4 vfinal=1.0 nvstep=20 name=input mult.v
```

For more infomation about the **LOG** and **SOLVE** commands see Chapter 22 "Statements".

The SOLVE Command

The NUM_HARMONICS parameter on the **SOLVE** command defines the number of harmonics used in the harmonic balance simulation. That is, the number of frequency components used to describe the time domain signal. The number of harmonics should be high enough to model the time domain signal with sufficient fidelity. But there should be no more harmonics than necessary: the more harmonics the slower the simulation.

The LOG Command

The following data is stored in the harmonic balance log file:

- the DC bias point,
- a list of the frequency components of the input and output voltages and currents,
- the input power at the fundamental,
- the output power at the fundamental and each harmonic.

The NUM_HARMONICS parameter on the **LOG** command defines how many harmonics to output the data for. The complex numbers can output as a real/imaginary pair or a magnitude/phase pair.

19.6.1 Non-Linearity

Figure 19-10 shows a schematic of a two-port circuit.

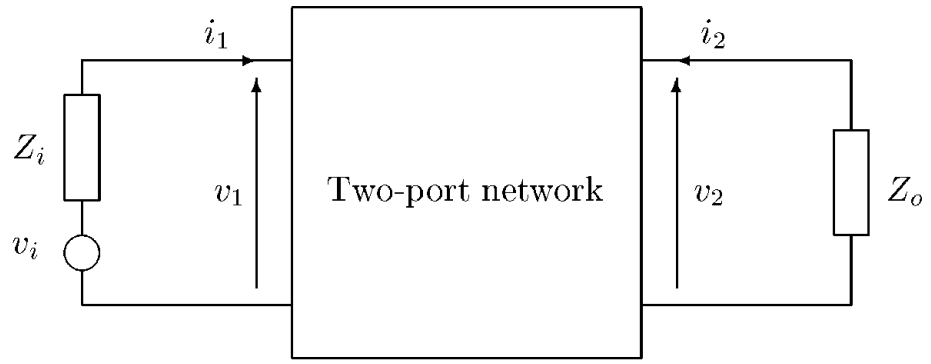


Figure 19-10: Two-Port Circuit

A signal is input into port 1 of the circuit. It is then transmitted through the two-port to the output impedance on port 2. In reality, the input signal will be irregular. But for analysis, the input signal is usually assumed to be sinusoidal at a frequency of ω. The frequency ω is called the fundamental frequency.

This section focuses on the steady-state output from the circuit to a periodic input signal. It does not focus on the transient response of the circuit to non-periodic changes to the input.

A Linear Circuit

In a linear circuit, the shape of the output signal is a perfect reproduction of the shape of the input signal. (The amplitude of the output signal is likely to be different from the input: almost all circuits amplify or attenuate. And there may be a delay between a signal appearing at the input and being transmitted to the output.)

If the input is a sinusoid at a frequency of ω, then the output will be a sinusoid at a frequency of ω. But probably with a different amplitude and phase than the amplitude and phase of the input sinusoid.

The behavior of this circuit can be summarized by a set of two-port parameters.

$$\begin{pmatrix} v_1(\omega) \\ v_2(\omega) \end{pmatrix} = \begin{pmatrix} Z_{11}(\omega) & Z_{12}(\omega) \\ Z_{21}(\omega) & Z_{21}(\omega) \end{pmatrix} \cdot \begin{pmatrix} i_1(\omega) \\ i_2(\omega) \end{pmatrix} \qquad 19\text{-}22$$

The Z-parameters depend only upon the frequency of the input signal. They don't depend upon the magnitude of the input signal.

A Non-linear Circuit

In a non-linear circuit, the shape of the output signal is not the same as the shape of the input signal.

If the input is a sinusoid at a fundamental frequency of ω, then the output is a signal that is a sum of a component at the fundamental frequency and additional components at the integer harmonics of this fundamental frequency.

$$v_0 \equiv v_{0,1}(\omega) + v_{0,2}(2 \cdot \omega) + v_{0,3}(3 \cdot \omega) + \ldots \quad \text{19-23}$$

In principle, this series is infinite but in practice we need only the first few terms. The bigger the magnitude of the input signal the more terms you need to include in the calculation of the output signal.

Real Devices

In reality, all devices and components are non-linear. If the input signal is small enough, only the first term in the expansion of the output is required and the circuit will be effectively linear.

19.6.2 Harmonic Balance

Time Domain and Frequency Domain

Any measured signal is real (rather than complex). A periodic real signal that is described by *n* harmonics can be written as

$$v(t) = V_0 + V_1 \cdot \exp(j \cdot \omega \cdot t) + V_2 \cdot \exp(2 \cdot j \cdot \omega \cdot t) + \ldots + V_n \cdot \exp(n \cdot j \cdot \omega \cdot t) + V_1^* \cdot \exp(-j \cdot \omega \cdot t) + V_2^* \cdot \exp(-2 \cdot j \cdot \omega \cdot t) + V_n^* \cdot \exp(-n \cdot j \cdot \omega \cdot t) \quad \text{19-24}$$

The "positive frequencies" correspond to sinusoidal waves moving in the negative direction. The "negative frequencies" correspond to sinusoidal waves moving in the positive direction.

The signal $v(t)$ is completely described by the set of coefficients V_0 to V_n. This is 2.n+1 real numbers (V_0 is real and the coefficients V_1 to V_n are complex). This representation of v(t) is called the frequency domain description.

Correspondingly, the signal $v(t)$ is completely described by a set of 2.n+1 values equally spaced over one period. The period of $v(t)$ is dictated by the period of the fundamental. The period T is such that

$$v(t) = v(t + T) \quad \text{19-25}$$

Therefore

$$T = \frac{2 \cdot \pi}{\omega} \quad \text{19-26}$$

We define the time step t_s as

$$t_s = \frac{T}{2 \cdot n + 1} \quad \text{19-27}$$

The time domain representation of $v(t)$ is the set of points

$$v_0 = v(0), v_1 = v(t_s), v_2 = v(2 \cdot t_s), \ldots v_{2 \cdot n} = v(2 \cdot n \cdot t_s) \quad \text{19-28}$$

These two representations of the signal $v(t)$ are equivalent. The discrete Fourier transform is used to translate between them.

The Problem With Large-Signal Simulation

The equations that describe the behavior of non-linear components are mostly written in the time domain. Therefore, they must be solved in the time domain. The time domain behavior must be solved at 2.n+1 points over one period of the signal.

A linear circuit almost certainly contains resistance, capacitance, and inductance and has an RCL time constant. The output from a time domain simulation of a linear circuit will initially contain a transient part in addition to the steady state part. For the transient part of the result to decay and leave only the steady state part, the simulation must continue for several multiples of the RCL time constant. If the time constant is longer than the period of the input signal, a time domain simulation may have to run for many hundreds of periods before reaching the steady state solution.

This is a problem when trying to solve a circuit that contains both linear and non-linear elements. The non-linear elements must be solved in the time domain. Solving the linear elements, however, in the time domain is very inefficient.

Harmonic Balance Method

While the steady state response of a linear circuit is difficult to calculate in the time domain, this response can be directly calculated in the frequency domain. This is the key to the Harmonic Balance method.

The circuit is divided into the linear parts and the non-linear parts (see Figure 19-11). Attention is focused on the ports where the two circuits connect.

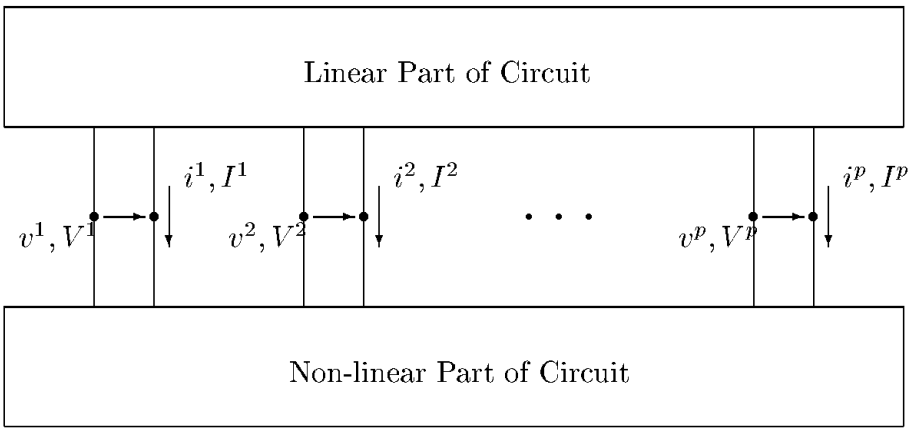


Figure 19-11: A Circuit divided into Linear and Non-linear Parts

The Harmonic Balance algorithm is as follows:

1. Mercury makes an initial guess of the time domain voltage at the ports.
2. Mercury uses the non-linear circuit to calculate the resultant time domain current at the ports.
3. Mercury uses the Fourier transform to translate each time domain current into a frequency domain current.
4. Mercury uses the linear circuit to calculate the resultant frequency domain voltage at the ports.
5. Mercury uses the Fourier transform to translate each frequency domain voltage into a time domain voltage.
6. If the new time domain voltage is close enough to the time domain voltage from the previous iteration, Mercury has solved the response of the circuit. If the new time domain voltage is significantly different from the time domain voltage in the previous iteration, Mercury returns to step 2 and performs another iteration.

19.7 NETWORK

NETWORK allows you to define the external network surrounding the FET in a Mercury simulation. This command is valid only when using the Mercury module.

Syntax

```
NETWORK [CLEAR] [HARMONIC=<n>] [dc_circuit] [rf_circuit]
```

Parameter	Type	Default	Units
CLEAR	Logical	False	
HARMONIC	Integer		
CD.RF	Real	0.0	F
CG.RF	Real	0.0	F
CI.11.RF	Real	0.0	F
CI.12.RF	Real	0.0	F
CI.21.RF	Real	0.0	F
CI.22.RF	Real	0.0	F
CI.RF	Real	0.0	F
CO.11.RF	Real	0.0	F
CO.12.RF	Real	0.0	F
CO.21.RF	Real	0.0	F
CO.22.RF	Real	0.0	F
CO.RF	Real	0.0	F
CS.RF	Real	0.0	F
GI.11.DC	Real	0.0	Mho
GI.11.IM.RF	Real	0.0	Mho
GI.11.RE.RF	Real	0.0	Mho
GI.12.DC	Real	0.0	Mho
GI.12.IM.RF	Real	0.0	
GI.12.RE.RF	Real	0.0	
GI.21.DC	Real	0.0	
GI.21.IM.RF	Real	0.0	
GI.21.RE.RF	Real	0.0	
GI.22.DC	Real	0.0	Ohm
GI.22.IM.RF	Real	0.0	Ohm

GI.22.RE.RF	Real	0.0	Ohm
HO.11.DC	Real	0.0	Ohm
HO.11.IM.RF	Real	0.0	Ohm
HO.11.RE.RF	Real	0.0	Ohm
HO.12.DC	Real	0.0	
HO.12.IM.RF	Real	0.0	
HO.12.RE.RF	Real	0.0	
HO.21.DC	Real	0.0	
HO.21.IM.RF	Real	0.0	
HO.21.RE.RF	Real	0.0	
HO.22.DC	Real	0.0	Mho
HO.22.IM.RF	Real	0.0	Mho
HO.22.RE.RF	Real	0.0	Mho
LD.RF	Real	0.0	H
LG.RF	Real	0.0	H
LI.11.RF	Real	0.0	H
LI.12.RF	Real	0.0	H
LI.21.RF	Real	0.0	H
LI.22.RF	Real	0.0	H
LI.RF	Real	0.0	H
LO.11.RF	Real	0.0	H
LO.12.RF	Real	0.0	H
LO.21.RF	Real	0.0	H
LO.22.RF	Real	0.0	H
LO.RF	Real	0.0	H
LS.RF	Real	0.0	H
RD.DC	Real	0.0	Ohm
RD.RF	Real	0.0	Ohm
RG.DC	Real	0.0	Ohm
RG.RF	Real	0.0	Ohm
RI.11.RF	Real	0.0	Ohm
RI.12.RF	Real	0.0	Ohm

RI.21.RF	Real	0.0	Ohm
RI.22.RF	Real	0.0	Ohm
RI.DC	Real	0.0	Ohm
RI.RF	Real	0.0	Ohm
RO.11.RF	Real	0.0	Ohm
RO.12.RF	Real	0.0	Ohm
RO.21.RF	Real	0.0	Ohm
RO.22.RF	Real	0.0	Ohm
RO.DC	Real	0.0	Ohm
RO.RF	Real	0.0	Ohm
RS.DC	Real	0.0	Ohm
RS.RF	Real	0.0	Ohm
SI.11.DC	Real	0.0	
SI.11.IM.RF	Real	0.0	
SI.11.RE.RF	Real	0.0	
SI.12.DC	Real	0.0	
SI.12.IM.RF	Real	0.0	
SI.12.RE.RF	Real	0.0	
SI.21.DC	Real	0.0	
SI.21.IM.RF	Real	0.0	
SI.21.RE.RF	Real	0.0	
SI.22.DC	Real	0.0	
SI.22.IM.RF	Real	0.0	
SI.22.RE.RF	Real	0.0	
SO.11.DC	Real	0.0	
SO.11.IM.RF	Real	0.0	
SO.11.RE.RF	Real	0.0	
SO.12.DC	Real	0.0	
SO.12.IM.RF	Real	0.0	
SO.12.RE.RF	Real	0.0	
SO.21.DC	Real	0.0	
SO.21.IM.RF	Real	0.0	

SO.21.RE.RF	Real	0.0	
SO.22.DC	Real	0.0	
SO.22.IM.RF	Real	0.0	
SO.22.RE.RF	Real	0.0	
YD.IM.RF	Real	0.0	Mho
YD.RE.RF	Real	0.0	Mho
YG.IM.RF	Real	0.0	Mho
YG.RE.RF	Real	0.0	Mho
YI.IM.RF	Real	0.0	Mho
YI.RE.RF	Real	0.0	Mho
YO.IM.RF	Real	0.0	Mho
YO.RE.RF	Real	0.0	Mho
YS.IM.RF	Real	0.0	Mho
YS.RE.RF	Real	0.0	Mho
ZD.IM.RF	Real	0.0	Ohm
ZD.RE.RF	Real	0.0	Ohm
ZG.IM.RF	Real	0.0	Ohm
ZG.RE.RF	Real	0.0	Ohm
ZI.11.DC	Real	0.0	Ohm
ZI.11.IM.RF	Real	0.0	Ohm
ZI.11.RE.RF	Real	0.0	Ohm
ZI.12.DC	Real	0.0	Ohm
ZI.12.IM.RF	Real	0.0	Ohm
ZI.12.RE.RF	Real	0.0	Ohm
ZI.21.DC	Real	0.0	Ohm
ZI.21.IM.RF	Real	0.0	Ohm
ZI.21.RE.RF	Real	0.0	Ohm
ZI.22.DC	Real	0.0	Ohm
ZI.22.IM.RF	Real	0.0	Ohm
ZI.22.RE.RF	Real	0.0	Ohm
ZI.IM.RF	Real	0.0	Ohm
ZI.RE.RF	Real	0.0	Ohm

ZO.11.DC	Real	0.0	Ohm
ZO.11.IM.RF	Real	0.0	Ohm
ZO.11.RE.RF	Real	0.0	Ohm
ZO.12.DC	Real	0.0	Ohm
ZO.12.IM.RF	Real	0.0	Ohm
ZO.12.RE.RF	Real	0.0	Ohm
ZO.21.DC	Real	0.0	Ohm
ZO.21.IM.RF	Real	0.0	Ohm
ZO.21.RE.RF	Real	0.0	Ohm
ZO.22.DC	Real	0.0	Ohm
ZO.22.IM.RF	Real	0.0	Ohm
ZO.22.RE.RF	Real	0.0	Ohm
ZO.IM.RF	Real	0.0	Ohm
ZO.RE.RF	Real	0.0	Ohm
ZS.IM.RF	Real	0.0	Ohm
ZS.RE.RF	Real	0.0	Ohm

Description

CLEAR	Sets all contact impedances to 0, sets the input matching network to G11=0, G12=-1, G21=1, and G22=0, and sets the output matching network to H11=0, H12=1, H21=-1, and H22=0.
HARMONIC	Specifies which harmonic, in a harmonic balance simulation, the RF circuit is applied to. If HARMONIC is not set, then the RF circuit will be applied to all harmonics.

DC Contact Resistances

Note: At DC all circuits are assumed to be real.

RD.DC	Defines the DC drain resistance.
RG.DC	Defines the DC gate resistance.
RI.DC	Defines the DC input resistance.
RO.DC	Defines the DC output resistance.
RS.DC	Defines the DC source resistance.

DC Input Matching Network

GI.11.DC, **GI.12.DC**, **GI.21.DC**, **GI.22.DC**	Defines the DC input matching network as a set of g-parameters.
SI.11.DC, **SI.12.DC**, **SI.21.DC**, **SI.22.DC**	Defines the DC input matching network as a set of s-parameters.
ZI.11.DC, **ZI.12.DC**, **ZI.21.DC**, **ZI.22.DC**	Defined the DC input matching network as a set of z-parameters.

DC Output Matching Network

HO.11.DC, **HO.12.DC**, **HO.21.DC**, **HO.22.DC**	Defines the DC output matching network as a set of h-parameters.
SO.11.DC, **SO.12.DC**, **SO.21.DC**, **SO.22.DC**	Defines the DC output matching network as a set of s-parameters.
ZO.11.DC, **ZO.12.DC**, **ZO.21.DC**, **ZO.22.DC**	Defines the DC output matching network as a set of z-parameters.

RF Contact Impedances

RD.RF, **CD.RF**, **LD.RF**	Defines the RF drain impedance as a series RCL circuit.
RG.RF, **CG.RF**, **LG.RF**	Defines the RF gate impedance as a series RCL circuit.
RI.RF, **CI.RF**, **LI.RF**	Defines the RF input impedance as a series RCL circuit.
RO.RF, **CO.RF**, **LO.RF**	Defines the RF output impedance as a series RCL circuit.
RS.RF, **CS.RF**, **LS.RF**	Defines the RF source impedance as a series RCL circuit.
YD.RE.RF, **YD.IM.RF**	Defines the RF drain impedance as an admittance.
YG.RE.RF, **YG.IM.RF**	Defines the RF gate impedance as an admittance.
YI.RE.RF, **YI.IM.RF**	Defines the RF input impedance as an admittance.
YO.RE.RF, **YO.IM.RF**	Defines the RF output impedance as an admittance.
YS.RE.RF, **YS.IM.RF**	Defines the RF source impedance as an admittance.
ZD.RE.RF, **ZD.IM.RF**	Defines the RF drain impedance as an impedance.
ZG.RE.RF, **ZG.IM.RF**	Defines the RF gate impedance as an impedance.
ZI.RE.RF, **ZI.IM.RF**	Defines the RF input impedance as an impedance.
ZO.RE.RF, **ZO.IM.RF**	Defines the RF output impedance as an impedance.
ZS.RE.RF, **ZS.IM.RF**	Defines the RF source impedance as an impedance.

RF Input Matching Network

RI.11.RF, **CI.11.RF**, **LI.11.RF**, **RI.12.RF**, **CI.12.RF**, **LI.12.RF**, **RI.21.RF**, **CI.21.RF**, **LI.21.RF**, **RI.22.RF**, **CI.22.RF**, **LI.22.RF**	Defines the RF input matching network, each parameter of the z-parameter two-port is defined as a series RCL circuit.
GI.11.RE.RF, **GI.11.IM.RF**, **GI.12.RE.RF**, **GI.12.IM.RF**, **GI.21.RE.RF**, **GI.21.IM.RF**, **GI.22.RE.RF**, **GI.22.IM.RF**	Defines the RF input matching network as a set of g-parameters.
SI.11.RE.RF, **SI.11.IM.RF**, **SI.12.RE.RF**, **SI.12.IM.RF**, **SI.21.RE.RF**, **SI.21.IM.RF**, **SI.22.RE.RF**, **SI.22.IM.RF**	Defines the RF input matching network as a set of s-parameters.
ZI.11.RE.RF, **ZI.11.IM.RF**, **ZI.12.RE.RF**, **ZI.12.IM.RF**, **ZI.21.RE.RF**, **ZI.21.IM.RF**, **ZI.22.RE.RF**, **ZI.22.IM.RF**	Defines the RF input matching network as a set of z-parameters.

RF Output Matching Network

RO.11.RF, **CO.11.RF**, **LO.11.RF**, **RO.12.RF**, **CO.12.RF**, **LO.12.RF**, **RO.21.RF**, **CO.21.RF**, **LO.21.RF**, **RO.22.RF**, **CO.22.RF**, **LO.22.RF**,	Defines the RF output matching network, each parameter of the z-parameter two-port is defined as a series RCL circuit.
HO.11.RE.RF, **HO.11.IM.RF**, **HO.12.RE.RF**, **HO.12.IM.RF**, **HO.21.RE.RF**, **HO.21.IM.RF**, **HO.22.RE.RF**, **HO.22.IM.RF**	Defines the RF output matching network as a set of h-parameters.
SO.11.RE.RF, **SO.11.IM.RF**, **SO.12.RE.RF**, **SO.12.IM.RF**, **SO.21.RE.RF**, **SO.21.IM.RF**, **SO.22.RE.RF**, **SO.22.IM.RF**	Defines the RF output matching network as a set of s-parameters.
ZO.11.RE.RF, **ZO.11.IM.RF**, **ZO.12.RE.RF**, **ZO.12.IM.RF**, **ZO.21.RE.RF**, **ZO.21.IM.RF**, **ZO.22.RE.RF**, **ZO.22.IM.RF**	Defines the RF output matching network as a set of z-parameters.

Examples

```
NETWORK CLEAR
```

Do this before DC and small-signal AC simulations to remove the input and output matching networks.

```
network zi.re.rf=50.0 zi.im.rf=0.0 zo.re.rf=50.0 zo.im.rf=0.0
```

This sets the input and output impedance to 50 Ohms at all harmonics.

19.8 POISSON

POISSON allows you to solve Poisson's equation in the Y direction of a FET in a Mercury simulation. This command is valid only when using the Mercury module.

Syntax

```
POISSON <cap simulation> <gate simulation> <look-up table calculation>
```

Parameter	Type	Default	Units
ASCII	Logical	False	
BINARY	Logical	False	
CAP	Logical	False	
COMPRESSED	Logical	False	
DE.DX	Real	0.0	V/cm2
DEBUG.CAP	Character		
DEBUG.GATE	Character		
E.SURF	Real	0.0	V/cm
GATE	Logical	False	
IN.FILE	Character		
LOOK.UP.TABLE	Logical	False	
NUM.RECESS.POINTS	Integer		
OUT.FILE	Character		
QUICK	Logical	False	
RECESS	Real	0.0	um
SAFE	Logical	False	
STANDARD	Logical	True	
V.GATE	Real	V	
V.SUBSTRATE	Real		V
V.TYPICAL	Real	0.0	V
VD.MAX	Real	0.0	V
VD.MIN	Real	0.0	V
VG.MAX	Real	0.0	V
VG.MIN	Real	0.0	V
VG.NUM	Real		
VG.START	Real		V

VG.STOP	Real		V
VG.TYPICAL	Real	0.0	V
VI.MAX	Real	0.0	V
VI.MIN	Real	0.0	V
VO.MAX	Real	0.0	V
VO.MIN	Real	0.0	V

Description

ASCII	Saves the data file in a standard, uncompressed format.
BINARY	Saves the data file in a binary format.
COMPRESSED	Saves the data file in a compressed ASCII format.
OUT.FILE	Specifies the name of the output file.
V.SUBSTRATE	Specifies the potential on the substrate contact.

Cap Poisson Solution

CAP	Defines that Poisson's equation is solved once, over an ungated slice of the device.
DE.DX	Specifies the gradient of the X direction electric field.
E.SURF	Specifies the Y direction electric field at the surface of the slice.
RECESS	Specifies the position of the surface of the slice.

Gate Poisson Solution

DE.DX	Specifies the gradient of the X direction electric field.
GATE	Defines that Poisson's equation is solved once, over a gated slice of the device. The recess depth is automatically set to the position of the gate specified in the device construction.
V.GATE	Specifies the potential on the gate contact.

Poisson Look-Up Table Calculation

DEBUG.CAP	Specifies a file to save the results of the cap look-up table calculation. This file is suitable for viewing in TonyPlot but has no other purpose in a device simulation.
DEBUG.GATE	Specifies a file to save the results of the gate look-up table calculation. This file is suitable for viewing in TonyPlot but has no other purpose in a device simulation.
IN.FILE	Specifies the file name of a previously calculated Poisson look-up table. If a table cannot be read, then you must calculate it.
LOOK.UP.TABLE	Defines that the Poisson look-up table must be calculated.
NUM.RECESS.POINTS	Specifies the number of points to use when calculating the cap look-up table.
QUICK	Specifies that a sparse look-up table should be calculated (16 points in the cap look-up table and 128 points in the gate look-up table).
SAFE	Specifies that a dense look-up table should be calculated (48 points in the cap look-up table and 384 points in the gate look-up table).
STANDARD	Specifies that a normal look-up table should be calculated (32 points in the cap look-up table and 256 points in the gate look-up table)
V.TYPICAL	Specifies a typical voltage that is used to initialize the Poisson solution. This should be a gate voltage that generates a noticable 2D sheet carrier density. But it doesn't generate too much additional charge at the surface.
VD.MAX	Specifies the maximum expected bias on the drain contact. This is used to calculate the range over which to solve the gate look-up table.
VD.MIN	Specifies the minimum expected bias on the drain contact. This is used to calculate the range over which to solve the gate look-up table.
VG.MAX	Specifies the maximum expected bias on the gate contact. This is used to calculate the range over which to solve the gate look-up table.
VG.MIN	Specifies the maximum expected bias on the drain contact. This is used to calculate the range over which to solve the gate look-up table.
VG.NUM	Specifies the number of points to use when calculating the gate look-up table.
VG.START	Sets the lower value of gate bias to use when calculating the gate look-up table.
VG.STOP	Sets the upper value of gate bias to use when calculating the gate look-up table.
VG.TYPICAL	This is a synonym for `V.TYPICAL`.
VI.MAX	This is a synonym for `VG.MAX`.

VI.MIN	This is a synonym for VG.MIN.
VO.MAX	This is a synonym for VD.MAX.
VO.MIN	This is a synonym for VD.MIN.

Examples

```
poisson cap recess=0.0 out.file=cap.str
```

This solves Poisson's equation for an ungated slice with the surface at y=0. The results are stored in a file cap.str.

```
poisson gate v.gate=0.0 out.file=gate.str
```

This solves Poisson's equation for a gated slice with a bias of Vg=0. The results are stored in a file gate.str.

```
poisson look.up vg.min=-1.5 vd.max=5.0 out.file=look_up.dat
```

The Poisson look-up table must be calculated before a channel simulation is attemped (basically before a **SOLVE** command). The simulation is going to have gate biases in the range of 0 to -1.5 V and drain biases in the range of 0 to 5 V.

19.9 SURFACE

SURFACE allows you to define the surface topography of a FET in a Mercury simulation. This command is valid only when using the Mercury module.

Syntax

```
SURFACE X=<x> Y=<y>
```

Parameter	Type	Default	Units
X	Real	0.0	um
Y	Real	0.0	um

Description

X	Gives the X coordinate of the point being defined.
Y	Gives the Y coordinate of the point being defined.

Examples

```
surface x=0.5 y=0.0
surface x=0.6 y=0.1
surface x=1.4 y=0.1
surface x=1.5 y=0.0
```

These commands define a 0.1 um deep recess on the surface of the device. The bottom of the recess is 0.8 um long and centered around x=1 um. The sides of the recess slope at 45°.

Chapter 20
MC Device

20.1 What is MC Device

MC Device is a module for performing full-band Monte Carlo simulation of silicon and strained silicon devices. MC Device is based on MOCA. The MOCA software was developed by the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign.

MC Device has two operating modes:

- A bulk mode, which is for fitting material parameters to experimental data. This mode provides a unipolar, space-homogeneous transport model that can be used to investigate femtosecond transients, impact-ionization coefficients, doping, field or stress-dependent mobilities, and many other bulk transport properties. This mode is intrinsically unipolar (i.e., only the behavior of electrons or holes can be simulated).
- A 2D mode, which is for performing device MC simulation with regions of different materials, ohmic and blocking contacts, and so on. A non-linear Poisson solver and Schroedinger solver are included. This mode shares the band structure and scattering models with the bulk mode.

Providing the two modes above in a single module has advantages of maintainability and consistency. For example, a scattering mechanism added to the bulk mode is available in the 2D mode. The complication of sharing some models but not other models between modes is tolerable. For example, position-dependent impurity scattering is necessary in the 2D mode because the impurity concentration is not uniform over the device. A different model that considers only one doping level is used for the bulk mode.

20.2 Using MC Device

The MC Device module can be used in all modes of operations of Atlas (see Section 2.3 “Modes of Operation”). We recommend that you run MC Device in the DeckBuild’s run-time environment as described in Sections 2.3.1 “Interactive Mode With DeckBuild” through 2.3.4 “Running Atlas inside Deckbuild”.

The MC Device module can also be used in batch mode without DeckBuild, although this is not recommended by Silvaco. Running in batch mode without DeckBuild can be used for reading input files in MOCA input format (see Section 20.3.3 “Migrating From MOCA Input Format”). If you don’t want the overhead of a DeckBuild window, then use DeckBuild with the no-windows mode (see Section 2.3.3 “No Windows Batch Mode With DeckBuild”). Many important features such as variable substitution, automatic interfacing to process simulation, and parameter extraction are only available inside the DeckBuild run-time environment.

To use the MC Device module in batch mode without DeckBuild, use the statement:

```
atlas <input file>
```

<input file> is the name of an input file that contains the simulation parameters. The input file can have any extension. Here, we will use `.in`. The following statement line runs MC Device using the input file `mcdeviceex01.in`:

```
atlas mcdeviceex01.in
```

The file `defaults.in` contains default values for all parameters; the input file overrides those values that need to be changed. For example, the default lattice temperature for the material is 300K. So you do not need to set the `TEMP` parameter to run a room-temperature simulation. Section 20.3 “Input Language and Syntax” describes the format of the input file. Appendix D: “MC Device Files” gives the complete default configuration file.

At startup, MC Device reads the `defaults.in` file (i.e., `$<S_INSTALL>/lib/atlas/<version>/common/mcdevice/defaults.in`). Next, MC Device reads your input file (i.e., `mcdeviceex01.in`). Simulation parameters defined in your input file override those in the `defaults.in` file. Next, upon processing a **SOLVE** statement, MC Device loads database files (i.e., band structure or scattering rate data or both) from a database directory (i.e., `$<INSTALL>/lib/mcdevice_db/<version>`).

You can modify the file MC Device `defaults.in` file in the directory where this file is installed (namely, `$<S_INSTALL>/lib/atlas/<version>/common/mcdevice`). But, we do not recommend this. Instead, if you need to modify the `defaults.in` file, we recommend that you set the `MCDEVICEDEFDIR` environment variable to a new directory. Copy `$<S_INSTALL>/lib/atlas/<version>/common/mcdevice/defaults.in` to that new directory. Then, make whatever changes you want to the copied `defaults.in` file. If the `MCDEVICEDEFDIR` environment variable is unset, it is automatically set in the start-up Atlas script to `$<S_INSTALL>/lib/atlas/<version>/common/mcdevice`. On Windows platforms, `MCDEVICEDEFDIR` begins with the driver letter and a colon (i.e,. `C:/`).

You can modify the MC Device database files in the directory where they are installed (namely, $<S_INSTALL>/lib/mcdeivce_db/<version>). But, we do not recommended this. Instead, if you need to modify the database files, we recommend that you set the MCDEVICEDIR environment variable to a new directory. Copy $<S_INSTALL>/lib/mcdeivce_db/<version> to that new directory. Then, make whatever changes you want to the copied database files. If the MCDEVICEDIR environment variable is unset, it is automatically set in the start-up Atlas script to $<S_INSTALL>/lib/mcdevice_db/<version>. On Windows platforms, MCDEVICEDIR begins with the driver letter and a colon (i.e,. C:/).

Upon processing **SOLVE** statements, MC Device writes a summary file that reports all the user-definable parameters that have been used for the solve. The name of this file is by default summary.out. You can change this name in the input file. See the description of the **OUTPUT** statement in Section 20.3.4 “Commonly-Used Statements”. You can permanently change this name by editing defaults.in. Most of the simulation parameters affect the physical models (e.g., optical-phonon energies) and will not be changed often. Others, like the number of iterations or the simulation mode, will be redefined at every run. There are also some numerical parameters that cannot be changed (e.g., $\hbar$, π, and the unstrained lattice constant of silicon).

20.3 Input Language and Syntax

20.3.1 The Input Language

MC Device reads input files in MC Device input format with data grouped on to statements. Each statement can have parameter settings or no parameter settings.

For example, the **BULK** statement initializes the bulk simulation parameters and looks like this:

```
BULK FIELD=(100e3,5e3) DOPING=1e17 \
     CONC=0.0
```

The body of the statement follows its name and is continued by using the backslash (\) at end of each line except the last line of the statement. The body of the statement may be empty or it may contain one or more settings of simulation parameters. For example, the (electric) field, the doping, and the (carrier) concentration are parameters on the **BULK** statement.

A parameter setting has the form:

```
<parameter>=value
```

The value can be an integer, a floating-point number, a string, or a vector of one of the three simple types. A vector is a list of values separated by commas and enclosed in parentheses. For example, the FIELD parameter on the **BULK** statement has a list of two floating-point numbers as its value: the first is the value for TIME<0; the second is the value for TIME>=0. In the example input above, we are simulating the sudden reduction of the electric field from 100e3 V/cm to 5e3 V/cm. Normally, you should provide the value in cgs units although lengths are given in um by default (see the **UNITS** statement in Section 20.5.1 "Device Geometry").

Note: Field values for TIME<0 and TIME>=0 field may be set equal. In this case, TIME<0 is used to improve the quality of the initial conditions used at TIME=0. Output averaging in MC Device is done for TIME>=0.

A vector should include values of the same type. MC Device can make conversions between real and integer types. For clarity, we recommend you use values with the same type as the parameter. All example input files and the defaults.in files demonstrate this approach. MC Device stops if you set a parameter using an incompatible type (e.g., a string for a floating-point parameter).

20.3.2 Statement allocation and usage

Memory associated with MC Device statements is either allocated once or is allocated for each occurrence. The allocation behavior of each statement relates to its usage. For example, memory associated with the **BULK** statement, shown above, is allocated once. This means that if the **BULK** statement occurs more than once in the input file before a **SOLVE** statement, then each occurrence of the **BULK** statement changes parameter settings associated with prior **BULK** statements beginning with the **BULK** statement in the defaults.in file. All MC Device statements that appear in the defaults.in file (except for the **MATDEF** statement) are allocated once. The **SOLVE** statement is also allocated once, although it does not appear in the defaults.in file.

The MC Device statements associated with mesh, regions, materials, and doping (see Section 20.5.1 “Device Geometry”) are allocated for each occurrence. All statements involving other types of regions (e.g., current regions and Schrödinger regions) are also allocated for each occurrence. The parameters on statements that are allocated for each occurrence are initialized to zero (or null). Therefore, non-zero settings from prior occurrences are not retained. All MC Device statements that are allocated for each occurrence (except the MATDEF) do not appear in the defaults.in file.

20.3.3 Migrating From MOCA Input Format

Although MC Device can read input files in MOCA input format as an aid to users migrating from MOCA to MC Device (e-mail support@silvaco.com for details), Silvaco strongly recommends that MOCA users transform their input files to MC Device input format.

Files in MOCA input format look like this:

```
bulk {
  field = {100e3,5e3};
  doping = 1e17;
  conc =0.0;
}
```

If you compare this **BULK** statement to the same statement definition (in MC Device input format) in Section 20.3.1 “The Input Language”, you can see the differences. Semicolons terminate parameter assignments rather than white space. Curly braces enclose statement bodies rather than using continuation characters. Curly braces enclose vectors rather than using parentheses.

To transform a file from MOCA input format into MC Device input format, you will need to make the following changes:

1. Add the line "MCDEVICE" as the first uncommented line of the input file.
2. Delete any semicolons.
3. Delete any curly braces that surround the body of a statement.
4. Use a continuation character (\) to continue statements to the next line. Move any comments that were inside a statement to outside of that statement.
5. Replace curly braces that surround parameter vectors with parentheses.
6. Remove any braces that surround the value of the n parameter on the **PARTICLE** statement. For example, particle {n={10000};}; becomes PARTICLE N=10000. Do likewise, for any other scalar parameter which unnecessarily used braces surrounding its value. Using array syntax on scalar parameters is allowed in the MOCA input format but is not allowed in the MC Device input format.
7. Replace any MOCA’s 1d statements with MC Device’s oned statements. In MC Device input format, statements names and parameter names may not begin with a number, so the 1d statement has been changed to ONED.
8. Replace any use of the p parameter on an **IMPACT** statement with the PARAM parameter. Both MC Device input format and MOCA input format have changed from using the p parameter to using the PARAM parameter to distinguish it from the symbol for phosphorus, P.

9. Insert the contents of the $MCDEVICEDIR/"bands2d.hole.in" file into an input file that uses carrier=h (for MC holes) on the **ALGO** statement. (The "bands2d.*.in" files are not used by MC Device, so this input must now reside in your input file.)
10. Insert the contents of the $MCDEVICEDIR/"bands2d.elec.ssi.in" file into an input file that uses TYPE=1 (for MC electrons in strained silicon) on the mmat statement. (The "bands2d.*.in" files are not used by MC Device, so this input must now reside in your input file.)
11. If you used the **IMPORT** statement, change the TYPE and filename parameters as needed. (See Section 20.5.8 "Importing Data".) The default in MC Device is TYPE=8 for Silvaco structure files.
12. If you have meshing or boundary specifications in centimeters in your input file, add the statement units length=cm. By default, MC Device uses units length=um. Alternatively, you may want to change values in cm to um by multiplying them by 1e4.
13. Add a **SOLVE** statement at the end of your input file. The **SOLVE** statement is not required in MOCA input format. In MOCA input format, all explicit **SOLVE** statements are ignored and a single implicit **SOLVE** statement is performed after processing all input. The **SOLVE** statement is required in MC Device input format. In MC Device input format, **SOLVE** statements are processed explicitly at the point at which they appear in the input file.

You can add the **GO** ATLAS and **QUIT** statements to the start and end of your input file. These statements tell DeckBuild to start and stop Atlas. These statements are ignored when you use them outside DeckBuild.

You not required to convert statement and parameter keywords to uppercase when converting an input file in MOCA input format to MC Device input format. Our convention in the body of this manual is to use uppercase for statement and parameter keywords to make these stand out in the input file. (Our defaults and examples are provided, however, using lowercase keywords.) Our convention is to use lowercase for keywords in MOCA input format, since UIUC MOCA only supported lowercase keywords. Uppercase, lowercase, and even mixedcase keywords are supported in MC Device. In MC Device, arbitaryly mixedcase keywords are supported in both the MC Device input format and the MOCA input format. Abbreviated keywords are supported in exclusively in MC Device input format.

20.3.4 Commonly-Used Statements

Many input statements are used to establish MC Device's default configuration and are not commonly used in input files. As mentioned in Section 20.2 "Using MC Device", the default values of most parameters are set in the defaults.in file. Any parameters that do not appear in the defaults file take the value 0|FALSE|NO, 0.0, or a null string, depending on the type of the parameter.

Note: MC Device uses 0, FALSE, and NO interchangeably. MC Device also uses 1, TRUE, and YES interchangeably. As part of MC Device input format, MC Device also supports the use of bare parameter names (i.e. **ALGO** RESTART) to set a boolean variable to true, and bare parameter names preceeded by the negation symbol, ^, (i.e. **ALGO**^RESTART) to set a boolean variable to false.

The following is a description of the most commonly used input statements and their parameters. For a more complete list of input statements, see Section 20.3.5 "List of Statements".

The **MCDEVICE** statement below must be included at the start of every MC Device input file or interactive input session.

```
MCDEVICE
```

The statement above must be the first non-comment line and indicates that the input that follows pertains to the MC Device module.

The **ALGO** statement contains the fundamental Monte Carlo algorithm parameters:

```
ALGO MODE=2 CARRIER=E DT=1e-15 ITER=6000 TRANS=4000 \
RESTART=YES SEED=65428
```

The MODE parameter controls the type of simulation to be run. Set MODE=0 perform a bulk simulation. Set MODE=2 to perform a 2D simulation. CARRIER is the Monte Carlo carrier type, Set CARRIER=E to simulate electrons with an MC transport model. Set CARRIER=H to simulate holes with an MC transport model. When MODE=0 (bulk simulation), only one carrier type is possible and it is set by the CARRIER parameter. When MODE=2 (2D), both carrier types are possible. The carrier type not specified by the carrier parameter will be treated with the constant quasi-Fermi level model (see "Regions" on page 966 and Section 20.7.1 "Self-Consistent Simulation"). The DT parameter is the time step for the free-flights. The default is 2×10^{-15} s. Use a lower value for high fields (100 kV/cm and up) or for relatively more accurate free-flight calculations. The ITER parameter is the number of iteration steps in a simulation and relates to the simulation time as TIME=ITER*DT. The TRANS parameter is the number of time steps (DT) taken as an initial transient. During the initial transient, no estimator averaging and estimator output is performed.

Note: MC Device uses parameters which begin with the letter D (DT, DX, and DE) to denote differential intervals of quantities (here time, space, and energy). Similarly, MC Device uses parameters, which end with the word STEP (TSTEP, XSTEP, and ESTEP) to denote integer multiples of the corresponding differential intervals. For example, TSTEP is an integer parameter, DT is a floating-point parameter (in seconds), TSTEP denotes an integer multiple of DT, and a time of TSTEP*DT seconds.

If RESTART is 0, as it is by default, a new simulation will be started. Otherwise, the carrier states (positions and momenta) are read from the "status.in" file.

SEED is the initial seed for the random-number generator. Since the outcome of the MC simulation depends on a sequence of random numbers, you can use the variable SEED to perform different simulations of the same physical system.

```
PARTICLE N=20000 NNORM=10000 CONC=0 DZ=-1.0
```

The **PARTICLE** statement defines parameters for the particle (electron or hole) ensemble. N is the maximum number of particles. If NNORM>0, then NNORM sets the number of particles to use at the beginning of the simulation to normalize the charge. NNORM must be less than or equal to N. If NNORM=0 (as it is by default), then NNORM=N is assumed.

If the device width, DZ, is greater than 0.0, then DZ is the device width in cm. If CONC>Ntot, where Ntot is the internally-calculated initial integrated particle concentration (cm^{-1}) and where CONC is a user-specified initial integrated particle concentration (cm^{-1}), then the device width is NNORM/CONC. (Ntot and CONC have the units cm^{-1} since they are volumetric concentrations integrated over *x* and *y*.) If neither of the two previous conditions are true, then the device width is determined as N/1.5/Ntot. If NNORM, CONC, and DZ are left at their default values of 0, 0.0, and -1.0 respectively, then the choice above for the device width

means that the initial particle number is (2/3)*N. This is normally the case. The fraction of 2/3 represents the fraction of initially-used particle memory compared to allocated particle memory. This choice allows the particle number to vary during the simulation.

Note: For the mcdevicex02.in example (Section D.3.2 "A 25-nm n-MOSFET: mcdeviceex02"), the device width and initial particle number are determined as they normally are where NNORM, CONC, and DZ are left at their default values of 0. In this example, the device width is N/1.5/Ntot=40,000/1.5/(1.76182E+09 cm^{-1})=1.51359E-05 cm and the initial particle number is (2/3)*N=(2/3)(40,000)=26,666.

You can use CONC to set a total charge in the device. This is useful for strong-inversion devices, where you need to help MC Device establish the amount of stored charge in the device. This may be necessary because the transient time of the charging is much longer than can realistically be simulated with MC Device.

```
MAT TEMP=77.0 CUT=(1.0,0.0,0.0)
```

This **MAT** statement defines material parameters (i.e., temperature and wafer cut or crystal orientation. TEMP is the lattice temperature in Kelvin. CUT is a triplet of real numbers that give the orientation of k-space with respect to real space. When properly normalized, they are the direction cosines of the X in the (k_x, k_y, k_z) space. For example, 1.0, 1.0, 0.0 means that the X-axis of real space (i.e., the direction of the electric field) is in the (110) direction of the crystal.

```
OUTPUT SUMMARYOUTFILE="summary.out" DIFFLOGFILE="diff.log" \
CURRENTLOGFILE="current.log" SOLSTRFILE="sol.str"
SCATOUTFILE="scat.out" \
TSTEP=100 XSTEP=(1,1,1) ESTEP=50 RESTART=500 \
WPSTEP=PINF INIT=NO HYDRO=NO HIST=YES QETA=NO OUTFILES=NO
```

The **OUTPUT** statement has parameter settings associated with output. The SUMMARYOUTFILE parameter sets the name of a file written for each SOLVE statement, which holds the input related to each solve statement. If you don't change the value of SUMMARYOUTFILE from its default value of summary.out, then after your simulation is complete, summary.out will hold the input used for your last solve statement. The DIFFLOGFILE parameter sets the name of the file used to hold time-averaged quantities for a bulk simulation (see diff.log in Section D.2.1 "Silvaco Output Files"). The CURRENTLOGFILE parameter sets the name of the file used to hold currents for a 2D simulation (see current.log in Section D.2.1 "Silvaco Output Files"). The SOLSTRFILE parameter sets name of the file used to hold the time-averaged and instanteous quantities related to a 2D solution (see sol.str in Section D.2.1 "Silvaco Output Files").

The SCATOUTFILE parameter sets the name of the file used to hold the scattering rates that were calculated and used during your simulation (see scat.out in Section D.2.4 "Scattering Output Files").

Note: The aliases SUMMARYFILE (for SUMMARYOUTFILE) and SCATFILE (for SCATOUTFILE) have been provided on the **OUTPUT** statement for backward compatibility with MOCA.

The TSTEP parameter is the number of time steps (DT on the ALGO statement) between output of observables (or estimators). The XSTEP parameter is the size of spatial bins (DX, DY, DZ) in units of mesh cells used for output. The ESTEP parameter is the size of energy bins in meV used for output. The RESTART parameter is the number of time steps between dumps of the particle status to the restart.out file. The WPSTEP parameter is the number of time steps between successive writes of the instantaneous electric potential (voltage) and the instantaneous MC carrier concentration. The init parameter indicates MC Device will write out the initial electrostatic potential (voltage) and initial MC carrier concentration at 5 points during the initialization process. The HYDRO parameter indicates MC Device should write out 2D estimators for hydrodynamic parameter extraction. The HIST parameter indicates that MC Device should write the energy histograms. The QETA parameter indicates that MC Device should write the quantum potential.

Note: The OUTFILES parameter on the **OUTPUT** statement is an important control variable in MC Device. The OUTFILES parameter enables/disables the writing of almost all *.out files generated by MC Device. By default, the outfiles parameter is set to 0|FALSE|NO so that *.out files are not written. (The summaryfile is treated as an exception and is always created.) Setting OUTFILES=FALSE results in no *.out files in the directory where the test is run and makes it easy to see the output files that can be used with other Silvaco tools, especially **TONYPLOT**. To enable the generation of all *.out files, set the OUTFILES parameter to 1|TRUE|YES. If other boolean parameters are specifically associated with a *.out file (like the init parameter above), then you must also set this parameter to 1|TRUE|YES if it is not true by default.

```
DEBUG NPICK=0
```

The **DEBUG** statement defines debugging parameters like a particle to track and holds the output parameter, which controls the generation of virtually all non-Silvaco-standard file formats.

If NPICK>0, MC Device tracks the corresponding particle index, dumping debugging data to various files. If NPICK<0, a large amount of debugging data is stored for the entire simulation (enough to fill up a 1 GB disk in a few hours). Also when NPICK≠0, many internal self-consistency checks are performed to catch bugs. The status of subroutine calls are then printed out at the console for each iteration.

```
SOLVE
```

If the **SOLVE** statement includes no parameters settings, it indicates that an MC solve should be performed for the biases specified by the input that precedes it (see "Biasing" on page 970).

```
SOLVE VGATE=1.0 VDRAIN=1.0
```

The **SOLVE** statement can also include one or more parameter settings of the form: V<NAME>=#. For these settings, # is a the applied voltage in V. <NAME> must be from the list: GATE, GG, DRAIN, DD, SOURCE, BULK, SUBSTRATE, EMITTER, EE, COLLECTOR, CC, BASE, BB, ANODE, CATHODDE, FGATE, CGATE, NGATE, PGATE, WELL, NWELL, PWELL, CHANNEL, and GROUND. Multiple SOLVE statements may be used in an input file. The V<NAME> settings are preserved from one SOLVE statement to the next. The V<NAME> settings are updated after the ramping **SOLVE** (see example below) so that they can be used in subsequent solves. The V<NAME> parameters default to 0 on the first **SOLVE**, so you only have to set non-zero biases. You can generate all output files when MC Device processes the **SOLVE** statement. To retain

output files from one **SOLVE** after executing a second **SOLVE**, change the output files using an **OUTPUT** statement between the **SOLVE** statements.

```
SOLVE NAME="drain" VSTEP=1.0 VFINAL=2.0
```

If you set VSTEP!=0 or NSTEPS>0 on the **SOLVE** statement, then MC Device performs a voltage ramp in 2D simulations. The voltage ramp is done on the contact specified by the NAME parameter. The voltage ramp completes after solving at a voltage of VFINAL. When USELOG=0|false|no (as it is by default), MC Device will use VSTEP for uniform spacing on a linear scale. When USELOG=1|true|yes, MC Device will use VSTEP for uniform spacing on a logarithmic scale. The USELOG parameter defaults to 0 on each **SOLVE** statement. Transient solves are performed for each bias condition beginning with those specified on the same or prior **SOLVE** statements.

The *current.log and *sol.str files produced during each voltage ramp step are saved by inserting the contact voltage after the base file name. For example, if the voltage on the drain were 1.0 V prior ot the solve above, then current_vdrain=1.log will pertain to the first ramp step. In this case, only one step is performed. The currents after this voltage step are saved in current_vdrain=2.log. Correspondingly, for this example, the solution structure files are saved as sol_vdrain=1.str and sol_vdrain=2.str. The NAME parameter must match a contact name as specified by the NAME parameter on a **REGION** statement (with TYPE=CONTACT). After the ramping **SOLVE** is complete, the corresponding V<GATE> parameter is automatically set to VFINAL so that this value will be used in subsequent solves. The VSTEP and VFINAL parameters are in V. The values of the voltage included in the filenames are also in V. The VSTEP, VFINAL, NAME, and NSTEPS parameters default to 0 or null on each SOLVE statement.

Note: MC Device checks that VSTEP takesw the voltage evenly to VFINAL. It stops with an error if it does not. Please choose VSTEP and VFINAL consistently or use NSTEPS instead of VFINAL or VSTEP.

The results in *current.log pertaining to the last simulation time for each voltage in the ramp are saved into the *current_ramp.log file. To specify the file name, use CURRENTRAMPLOGFILE parameter on the **OUTPUT** statement. Since this file contains both the voltage at the contact and the currents, you may use it to plot a current-voltage (I-V) curve.

```
SOLVE NSTEPS=3 FFINAL=1e6 USELOG=yes
```

If you set FSTEP!=0 or NSTEPS>0 on the **SOLVE** statement, then MC Device performs an electric field ramp in bulk simulations. The field ramp completes after solving with the field at FFINAL. When USELOG=0|false|no (as it is by default), MC Device will use FSTEP for uniform spacing on a linear scale. When USELOG=1|true|yes, MC Device will use FSTEP for uniform spacing on a logarithmic scale. The USELOG parameter defaults to 0 on each **SOLVE** statement. Transient solves are performed for each ramp step beginning with those specified by the FIELD parameter on the **BULK** statements (see Section 20.5.1 “Device Geometry”). To simplify solves during field ramping, FSTEP is added to both components of the FIELD vector (namely both the value for time<0 and the value for time>=0). If you want to control each component of the FIELD parameter on the **BULK** statement separately, perform the ramp steps as a sequence of separate solves by setting the FIELD parameter on the **ALGO** statement as desired. Then, change the DIFFLOG file parameter on the **OUTPUT** statement between the **SOLVE** statements.

Note: MC Device checks that FSTEP takes the field evenly to FFINAL. It stops with an error if it does not. Please choose FSTEP and FFINAL consistently or use NSTEPS instead of FFINAL or FSTEP.

The results in *diff.log files produced during each field ramp step are saved by inserting the field after the base file name. For example, if the field were 1000 V/cm prior to the solve above, then diff_field=1000.log will pertain to the first ramp step. In this case, then only two more iterations are performed at 1e4 V/cm and 1e5 V/cm. The results for these ierations are saved in diff_field=1e4.log and diff_field=1e5.log respectively. The FSTEP and FFINAL parameters are fields in V/cm. The values of the fields included in the filenames are also in V/cm. The FSTEP, FFINAL, and NSTEPS parameters default to 0 on each **SOLVE** statement.

The results in the *diff.log file pertaining to the last simulation time for each field ramp step will be saved into the the *diff_ramp.log file. To specify the file name, use the DIFFRAMPLOGFILE parameter on the OUTPUT statement. Since this file contains both the field and the bulk results (like mobilities), you may use this file to plot a mobility-field curve.

When you use the drift diffusion transport model to obtain the solution for a particular DC bias point, it is reasonable to think that you get the same solution for the same bias conditions even when the DC bias points that preceed the final point differ. This is normally not true (because normally each solution uses the prior solution as an initial conditions), but the solutions are close enough to each other that you can think of them as being the same.

When you use the Monte Carlo transport model (MC Device), small variations in the initial conditions can have a relatively larger effect on the solutions than they do for the drift diffusion model. To keep this behavior similar to what you are accustom to with drift diffusion, we have implemented the **SOLVE** statement in MC Device to exactly reproduce the results (on a particular hardware platform) whenever the bias conditions are exactly the same regardless what the preceeding bias point was. For example, this means that the solution you obtain as the last step of a ramp will be identical to the solution you obtain by a single solve for the final condition of the ramp. This means that you can add solution points to your results without re-running the entire ramp. The results you obtain for the additional single solves are identical to those you would obtain by re-running the entire ramp.

The **QUIT** statement stops execution at the point in the input where the **QUIT** statement appears. All input lines after the **QUIT** statement are ignored. The **QUIT** statement should be the last statement in your input file. The **QUIT** statement must be included when you run DeckBuild in batch mode and use DeckBuild's outfile command-line option. Otherwise, your output file will finish with an ATLAS> prompt and the END-OF-RUN message of MC Device will be omitted.

20.3.5 List of Statements

Table 20-1. Lists of Statements	
Statement	**Description**
AC	Acoustic phonon scattering. See Section 20.7.5 “Phonon Scattering”.
ALGO	Simulation mode parameters. See Section 20.3 “Input Language and Syntax”.
BANDS	Band structure information. See Section 20.7.2 “Band Structure”.
BIPOLAR	Pipe for interprocess communication. See Section 20.5.5 “Bipolar”.
BULK	Bulk simulation parameters. See Section 20.4 “Bulk Simulation”.
CREGION	Current estimation region. See Section 20.5.3 “Estimators”.
CROSSEST	Boundary crossing estimators. See Section 20.5.6 “Track Boundary Crossings”.
DEBUG	Toggle debugging on/off. See Section 20.3 “Input Language and Syntax”.
DEVICE	Device applied voltages. See Section 20.5.1 “Device Geometry”.
DOPING	Doping profile definition. See Section 20.5.1 “Device Geometry”.
DTREGION	Fine time-step region. See Section 20.5.7 “Time Step Regions”.
EBREGION	Eta barrier region definition. See Section 20.7.8 “Size Quantization”.
EEREGION	Energy estimator region definition. See “Energy Regions” on page 974.
ENHAN	Statistical enhancement. See Section 20.6 “Statistical Enhancement”.
EREGION	Statistical enhancement region. See Section 20.6 “Statistical Enhancement”.
ESTIM1D	1D averaging. See Section 20.6.6 “Statistical Enhancement Statements”.
FIELD	Initial field definition. See “Biasing” on page 970.
FFREGION	Fixed-field region definition. See Section 20.5.4 “Initial Conditions”.
FINAL	Final state selection. See Section 20.7.4 “Coulomb Interaction”.
GO	DeckBuild control. See Section 20.3.3 “Migrating From MOCA Input Format”.
ICLREGION	Initial concentration limit region. See Section 20.5.4 “Initial Conditions”.
IMPACT	Impact ionization parameters. See Section 20.7.6 “Impurity Scattering”.
IMPORT	Import initial guess. See Section 20.5.8 “Importing Data”.
INJECT	2D injection parameters. See Section 20.5.2 “Ohmic Contacts”.
LINBIAS	Additional linear biasing. See “Biasing” on page 970.
MAT	Material parameters. See Section 20.3 “Input Language and Syntax”.

Table 20-1. Lists of Statements	
Statement	**Description**
MATDEF	Material definition. See Section 20.5.1 “Device Geometry”.
MCDEVICE	Module specification. See Section 20.3.4 “Commonly-Used Statements”.
OUTPUT	File outputs. See Section 20.3 “Input Language and Syntax”.
PARTICLE	Number of particles. See Section 20.3 “Input Language and Syntax”.
PHON	High-energy phonon scattering. See Section 20.7.5 “Phonon Scattering”.
POISSON	Poisson solver parameters. See Section 20.7.1 “Self-Consistent Simulation”.
QREGION	Quantum region definition. See Section 20.7.8 “Size Quantization”.
QUANTUM	Quantum correction parameters. See Section 20.7.8 “Size Quantization”.
QUIT	Stop execution. Section 20.3.4 “Commonly-Used Statements”.
QSS	Surface charge parameters. See Section 20.5.4 “Initial Conditions”.
REFLECT	Properties for reflection. See Section 20.5.2 “Ohmic Contacts”.
REGION	Region definition. See Section 20.5.1 “Device Geometry”.
RESIST	Contact resistance information. See Section 20.5.2 “Ohmic Contacts”.
RIDLEY	Ridley model for impurity scattering. See Section 20.7.6 “Impurity Scattering”.
SCHRMESH	Schrödinger solver mesh definition. See Section 20.7.8 “Size Quantization”.
SCHRREGION	Schrödinger region definition. See Section 20.7.8 “Size Quantization”.
SEREGION	Scattering estimation region. See Section 20.7.4 “Coulomb Interaction”.
SMREGION	Smoothing region definition. See Section 20.6 “Statistical Enhancement”.
SOLVE	Perform MC solve. See Section 20.3.4 “Commonly-Used Statements”.
SSPARAM	Surface scattering parameters. See Section 20.7.6 “Impurity Scattering”.
SSREGION	Surface scattering region. See Section 20.7.6 “Impurity Scattering”.
TRACKSCAT	Track scattering region definition. See Section 20.7.6 “Impurity Scattering”.
TRAJ	Track particle trajectories. See Section 20.7.6 “Impurity Scattering”.
TTEMP	Transverse temperature calculation region. See Section 20.7.8 “Size Quantization”.
TUNNEL	Gate tunneling current calculator. See Section 20.7.7 “Tunneling”.
UNITS	Set units used in other statements. See Section 20.5.1 “Device Geometry”.
XL	X-L phonon scattering. See Section 20.7.5 “Phonon Scattering”.
XMESH	X-mesh definition. See Section 20.5.1 “Device Geometry”.

Table 20-1. Lists of Statements	
Statement	**Description**
XXF	X-X f-type phonon scattering. See Section 20.7.5 “Phonon Scattering”.
XXG	X-X g-type phonon scattering. See Section 20.7.5 “Phonon Scattering”.
YMESH	Y-mesh definition. See Section 20.5.1 “Device Geometry”.
ZSREGION	Zero-scattering region definition. See Section 20.7.6 “Impurity Scattering”.

20.3.6 Accessing The Examples

MC Device is supplied with standard examples that demonstrate the tool. We recommended that new users study the examples and use them as a starting point for creating their own simulations. Please learn how to access, load, plot, and run these examples by following the procedure below.

The examples are accessed from the menu system in DeckBuild.

1. Start DeckBuild with Atlas as the simulator as described in Section 20.2 “Using MC Device”.
2. Click the **MainControl** button to open this pull-down menu.
3. Click **Examples...** button to open the DeckBuild: Examples window (see Figure 20-1).

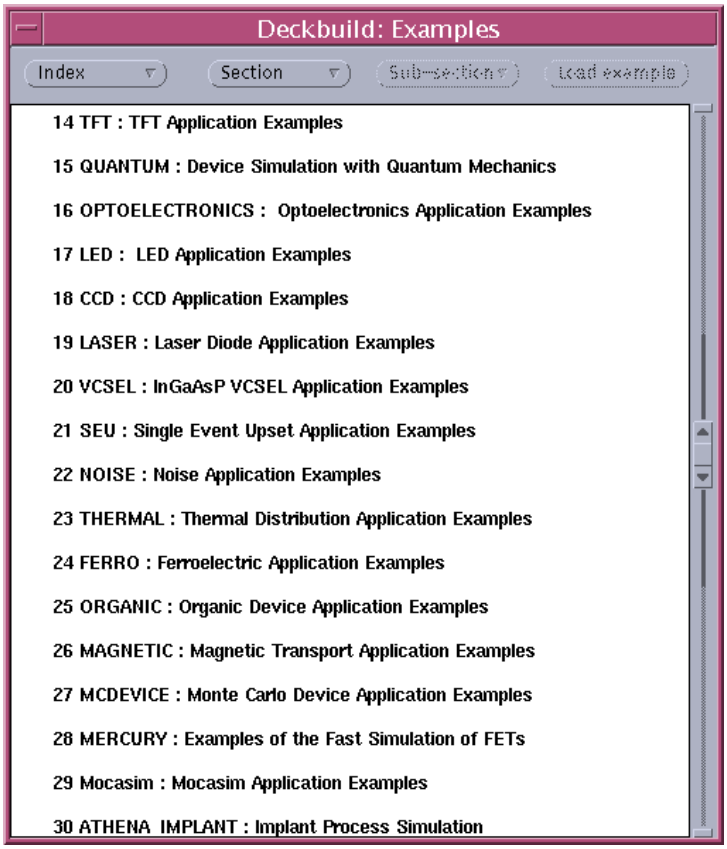


Figure 20-1: DeckBuild: Examples Window

4. Scroll down to show **MCDEVICE: Monte Carlo Appliation Examples** by clicking on the bottom arrow of the slider at the right of the window.
5. Choose MC Device by double clicking the MC Device line of the index. A list of examples will appear (see Figure 20-2). These examples illustrate different types of devices or different applications of MC Device.

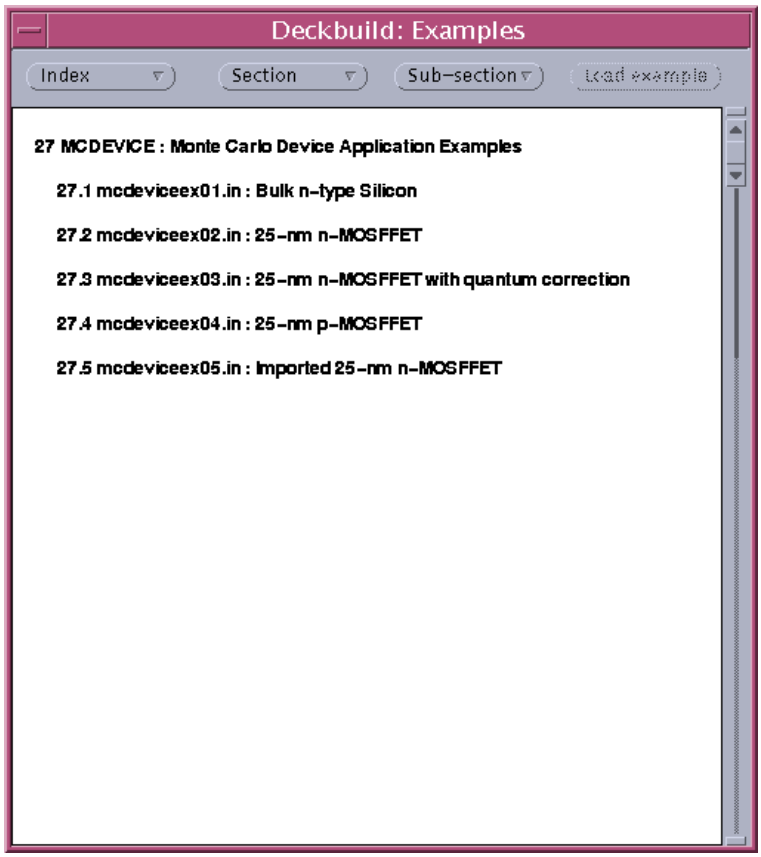


Figure 20-2: MC Device Examples Window

6. Choose each example by double clicking that item in the list. A description of the example will appear in the window. This description gives important details about how to use MC Device. Pleaes read this before proceeding. We recommend that you begin with example 1 and go in order since important concepts and statements are described only in the first example in which they appear.
7. Click on the **Load example** button. All files associated with that example are copied into the current working directory. The input command (`*.in`) file is loaded into DeckBuild. The sample output (`*_diff.log` or `*_current.log` and `*_sol.str`) files produced on x86-64-linux are copied to the working directory. TonyPlot set (`*.set`) files associated with each output file are also copied into the local working directory. DeckBuild will overwrite exiting files with fresh copies when you click the **load** button and click to confirm.
8. Look over the input file in DeckBuild. One or more `tonyplot` statements are included near the end of the file to plot some of the simulation results. You may view the sample results without running the example. To do this, highlighting the first `tonyplot` statement and click the **line** button to have DeckBuild to prepare to start at this line. Then, click the **next** button to execute the next line or click the **cont** button to continue with all remaining lines (to generate all the plots).
9. To run the entire example, click on the **run** button. The examples will finish by generating the plots in Tonyplot. Each `tonyplot` statement near the end of the input file will produce a separate TonyPlot window.

20.4 Bulk Simulation

Bulk simulation parameters are specified on the **BULK** statement.

```
BULK FIELD=(10e3,200e3) DOPING=1e18 CONC=4e17 MEMORY=NO MMFRAC=0.0
```

As mentioned in Section 20.3.1 “The Input Language”, the FIELD parameter sets the values of the electric field for TIME<0 and TIME>=0 in V/cm. Splitting the simulation into two parts serves two purposes. First, waiting for a transient time before computing time averages. Second, allowing the time-dependent study of carrier dynamics on the femtosecond or picosecond scale. The first case obviously applies only when the electric field before and after TIME=0 is the same. The second case applies when the field changes abruptly.

The DOPING parameter is the background doping level in cm^{-3}.

The CONC parameter allows you to specify the carrier concentration. This is relevant for a bulk simulation since it affects the screening of impurities and the scattering of carrier with ionized-impurities.

If you set the MEMORY parameter to 1|TRUE|YES, the status of the carrier ensemble will be maintained during the simulation. The energy and other quantities associated with MC carriers are recorded at time steps determined by the step field of the **OUTPUT** statement. This slows down MC Device but allows the time-dependent study of the microscopic histories of the particles. The file memory.out is printed out at the end of the run with statistics on impact ionization events.

For modeling the bulk behavior of strained silicon, the MMFRAC parameter sets the effective strain level in the bulk from 0.0 to 1.0. When modeling the bulk behavior of strained silicon layers on a relaxed $Si_{1-x}Ge_x$ substrate, MMFRAC refers to x, the fraction of Germanium in the substrate. When modeling the bulk behavior of strain silicon on a silicon substrate, MMFRAC refers to an effective strain which can vary from 0.0 (unstrained) to 1.0 (max strain of the fitting). For a full description of the strain model in MC Device, see Section 20.7.13 “Strain”.

20.5 Device Simulation

20.5.1 Device Geometry

Grid

MC Device solves the device equations on a rectangular non-uniform grid. Grid points are conventionally labeled by a pair of integers ranging from 1 to N_i, where i=x,y. The rectangular grid comprises $(N_x+1)(N_y+1)$ nodes and N_xN_y rectangular domains, or mesh cells. Cell (i,j) is bounded by the rectangle whose two opposite vertices are (i,j) and $(i+1,j+1)$.

The grid is specified using **XMESH** and **YMESH** statements in the input file in a manner similar to that used in Atlas. Each statement gives the coordinate of a node in the sequence of X or Y-coordinates. Node positions in between specified nodes are interpolated by geometric growth. The specification for a node layer in the X-direction is

```
XMESH (NODE=n LOC=l RATIO=r)
```

where NODE, LOC and RATIO are respectively the node index, the X-position, and the growth ratio. The format for the corresponding ymesh is the same.

Here, we will use the following terms:

- Grid lines: The loci $x=X_i$ or $y=Y_j$, where X_i and Y_j are grid coordinates.
- Grid point: The intersection of two grid lines. In other words, a point (X_i,Y_j).
- Grid segment: The segment joining two adjacent grid points in the X- or Y-direction. For example:

$$(X_i, Y_j) \quad - \quad (X_{i+1}, Y_j) \qquad \text{20-1}$$

or

$$(X_i, Y_j) \quad - \quad (X_i, Y_{j+1}) \qquad \text{20-2}$$

- Box: The rectangle surrounding a grid point, delimited by the lines bisecting grid segments.
- Mesh cell: The rectangle delimited by two pairs of adjacent grid lines. One pair in the X-direction and the other pair in the Y-direction.

The units of the values specified by the LOC parameter on the **XMESH** and **YMESH** statements are determined by the LENGTH parameter on the **UNITS** statement. The LENGTH parameter is set to UM for micrometers by default. You can change to LENGTH=CM for centimeters for backward compatibility with UIUC MOCA. The LENGTH parameter on the **UNITS** statement also defines the length units for all positional bound parameters (including all BOUNDP parameters) and the CHAR parameter on the **DOPING** statement.

Regions

A region is a rectangular area whose boundaries lie on mesh points. A region is geometrically defined by four integer numbers specifying the discrete indices of the first and last mesh nodes in the *X*- and *Y*-directions. (For your convenience, we recommend you define the boundaries using minimum and maximum positions along the X and Y axes, see the BOUNDP parameter on the **REGION** statement described below.) Regions are identified by number using the N parameter on the **REGION** statement. The N parameter should be set to a positive integer starting with 1. We recommend you additionally give each region a name which identifies its function in your device, see the NAME parameter on the **REGION** statement described below.

A region is assigned a material. Silicon (SI), polysilicon (POLY), silicon dioxide (SIO2), silicon nitride (SI3N4), aluminum (AL), gold (AU), platinum (PT), tungsten (W), and air (AIR) are supported.

The behavior of each region is specified by its type. Possible types are MC, CONTACT, BLOCK, and OUT. A MC region is assigned particles and its time-dependent behavior is determined during a solve. A CONTACT region is assumed to behave like a sample of the specified material at thermal equilibrium. It also functions as a boundary condition for other regions forcing a constant quasi-Fermi level during unipolar solves and a space-dependent electric potential. For example, if a silicon MC region neighbors a CONTACT, the equilibrium carrier concentration is computed for each point along the region boundary based on the doping concentration. Particles are then injected into MC regions to maintain that concentration. Also, spatially dependent boundary conditions for the Poisson equation are used, where the boundary conditions depends on the variations of the quasi-Fermi level. For a non-semiconducting material, such as SiO_2 or aluminum, the boundary concentration is computed from the known barrier height with respect to a MC region material and the uniform boundary condition for the electrostatic potential. Finally, regions of type BLOCK and OUT are insulator regions. These enters the simulation as part of the Poisson solve. The simulation depends on the dielectric constant of these regions as well as their size and location.

The definition of a device proceeds by successive definitions of regions, each overriding previous definitions. For example, to create a mesh structure, define a rectangular silicon region and then etching out two regions at the sides by defining additional regions. This will effectively define a T-shaped MC region. Insulator regions can be of two types: BLOCK and OUT. BLOCK and OUT regions are distinguished in order to simplify the handling of MC carrier reflections. BLOCK regions reflect particles towards the outside of their rectangular boundaries. OUT regions do not reflect particles. Instead, OUT regions are used to define insulating regions with boundaries defined implicitly by the boundaries of other regions. Usually, a large OUT region is defined as a global container for the entire device and regions that define the device structure are defined inside it. This allows complicated insulator shapes to be defined as multiple insulating BLOCK regions. At the same time, reflection of particles is ensured by the boundaries of the remaining regions. BLOCK regions are used in turn to make insulating parts out of the MC and contact regions. This scheme allows an arbitrary device structure without requiring the CPU-intensive algorithm of reflecting carriers along non-rectangular boundaries. Figure 20-3 shows an example.

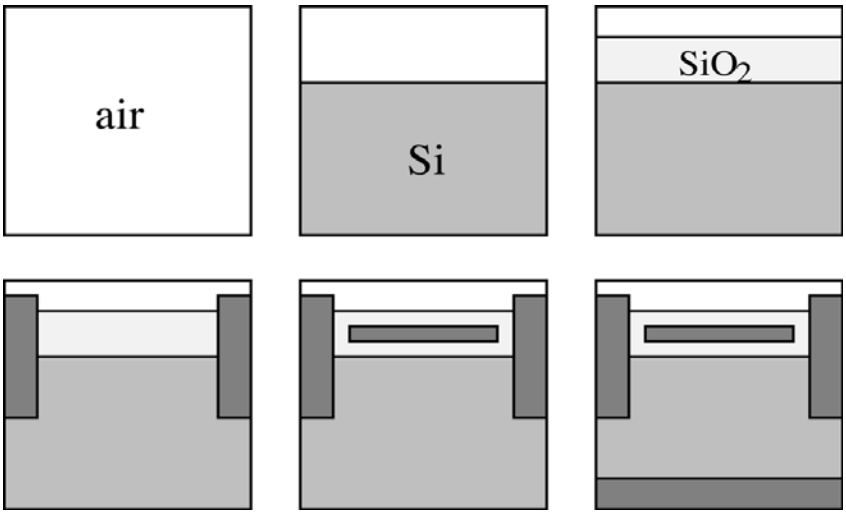


Figure 20-3: Example of device structure definition for a simple MOSFET. The successive steps define the simulation region (out), silicon substrate (MC), oxide (block), source/drain contacts (contact), gate (contact), and substrate contact (contact), respectively.

All combinations of material and type are allowed in principle, even though they may be unphysical (e.g., an aluminum insulator region). MC Device will complain about combinations for which it has no knowledge (e.g., a Si_3N_4 MC region). Otherwise, the material is independent of the way the region is simulated. For example, you can simulate a silicon region with the Monte Carlo method or treat it as a contact. Thus, assuming assigned potential and concentration along its edges. Similarly, you can directly simulate a polysilicon region using MC. It can also simply be a boundary condition for Poisson's equation.

You must apply a bias voltage to each contact region. MC Device does not allow you to have "floating shorts" in the device, which equalize potential along the edges without being connected to a voltage source. Also, non-MC, non-fixed charges in MC regions will be modeled assuming an uniform quasi-Fermi level propagating from the nearest contact region. The propagation algorithm is explained in Section 20.7.1 "Self-Consistent Simulation".

A region is specified by a **REGION** statement:

```
REGION N=n NAME="Region Name" MAT=AIR|SI|POLY|SIO2|SIN4|AL|PT|W|AU \
  TYPE=MC|CONTACT|BLOCK|OUT BOUNDP=(x1,x2,y1,y2) VBIAS=1.0 \
  USEFERMI=NO FERMI=0.0
```

N is an integer (>=1) and represents the region number.

NAME is the name of the region. You can see the name of regions in TonyPlot by selecting **Color by: Region** in the Tonyplot: Regions window. When using the **IMPORT** statement to import regions from a structure file, you can set the NAME parameter to the name of an imported region to redefine properties of that region. For example, you may need to specify a different spatial boundary for a contact region using the BOUNDP parameter. When using the import statement to import regions from a structure file, if the name of a contact region does not match a contact region in the import structure file, then a new contact region will be added to the imported structure. In the last case, the region number, N, is automatically set as the highest region number in the device (at the time that region is processed).

MAT is the material type. The appropriate electrical and transport parameters are taken from a library of materials provided in the defaults.in file. TYPE is the region type.

BOUNDP provides the positions of the rectangular region bounds using units specified by the LENGTH parameter on the **UNITS** statement (see Section 20.5.1 "Device Geometry"). *x1* is the minium position along x. x2 is the maximum position along x. y1 is the minimum position along y. y2 is the maximum position along y. The BOUND=(ix1 ix2 iy1 iy2) (not shown above) provides the rectangular region bounds in terms of integral mesh indices rather than positions.

The BOUND parameter is provided for backward compatibility with MOCA. We recommend using BOUNDP rather than BOUND since it enables you to change the mesh with little effect on your device geometry. BOUNDP is also more convenient to use when you import the device structure or mesh or both. See Section 20.5.8 "Importing Data" on using the IMPORT statement. If BOUND=(0 0 0 0) as it is by default, then BOUNDP is used.

VBIAS enables you to set the applied bias voltage on regions with TYPE=CONTACT. If the bias on a contact is set to a non-zero value using the VBIAS parameter on the REGION statement, then it will not be set by the VBIAS parameter on the **DEVICE** statement. But if the VBIAS parameter on the **REGION** statement is left at its default value of zero, then the VBIAS parameter on the **REGION** statement will be used to set the applied bias at the contact (see "Biasing" on page 970). When you are importing regions from a structure file, you must use the VBIAS parameter on the **REGION** statement rather than the VBIAS parameter on the **DEVICE** statement. Since the VBIAS parameter on the **REGION** statement can be used in all cases (including when you are importing regions), it is recommended over the VBIAS parameter on the **DEVICE** statement.

The USEFERMI and FERMI parameters enable you to obtain improved results for quantities related to the non-MC quasi-Fermi potential and a small correction to the terminal currents. By default, USEFERMI=0. As a result, the non-MC quasi-Fermi potential is set using the default method to the voltage applied at the contact within the doping well associated with the contact. If you set USEFERMI=1 on a **REGION** statement with TYPE=CONTACT, then MC Device will set the non-MC quasi-Fermi potential (V_n or V_p in Section 20.7.1 "Self-Consistent Simulation" for CARRIER=H and CARRIER=E on the ALGO statement, respectively) using the FERMI parameter within the doping well associated with that contact.

For example, let's consider the common case of simulating an *n*-MOSFET using CARRIER=E on the ALGO statement, a positive voltage on the drain, 0.0 V on the source, and a gate high voltage enough to create an inversion layer. In this common case, you can use USEFERMI=1 on the **REGION** statement associated with the drain contact to obtain an improved estimate for the hole quasi-Fermi potential in the drain's doping well (see examples of the USEFERMI=1 setting in the standard examples). USEFERMI=1 means that the hole quasi-Fermi potential within the drain's doping well will be set using the FERMI parameter (which is 0.0 V by default). Since we've assumed in this example that the source voltage is 0.0 V, the default of FERMI=0.0 correctly sets the hole quasi-Fermi level within the drain doping well to the applied bias at the source (0.0 V). You can then omit the setting FERMI=0.0 from your input since it matches the default setting. Since equlilbrium is assumed within each contact region, the electron and hole quasi-Fermi potentials are always equal to each other and equal to the voltage applied within a particular contact region. Therefore, the USEFERMI and FERMI parameter have no effect within contact regions themselves. In this example, the USEFERMI and FERMI parameters provide improved estimates for the hole quasi-Fermi level and the hole

concentration within the drain's doping well (neighboring the drain contact) and a small correction to the terminal currents.

Materials

The materials in listed in the description of the mat parameter on the region statement are predefined. You can override their default values and, thereby, create new materials using **MATDEF** statements. For example:

```
MATDEF N=n NAME="Material Name" EPS=e_r BARRIER=phi_B ROUGH=r
```

The barrier height is defined with respect to n-type silicon. It is used for determining the boundary conditions for the potential for a given applied bias. The ROUGH parameter specifies the probability of diffuse scattering due to surface roughness.

Note: Any parameters not explicitly listed in a **MATDEF** section will be set to 0|0.0|null. You cannot split settings for one material onto two separate statements (although, normally, this can be done). Instead, only the last statement for that material type in the input file is considered and all non-null settings must appear on that statement.

Doping

You can assign any part of the device a doping concentration and profile. Though, the simulator will complain if you try to dope a material, such as Aluminum. By default, there is no doping define for any material. Doping is assigned using the DOPING statement as follows:

```
DOPING DOPANT=B CONC=1e19 \
  BOUNDP=(0.0,0.095,0.0019,0.002) \
  CHAR=(0.0172,0.0172,0.016,0.016)
```

The DOPANT parameter can be either B (Boron), P (Phosphorus), In (Indium), or As (Arsenic). The CONC is concentration in atoms/cm^3 for the box defined by BOUNDP parameter. The BOUNDP parameter defines the coordinates of the box for the DOPING statement in units specified by the LENGTH parameter on the **UNITS** statement (see "Grid" on page 965). Again, a BOUND parameter for mesh indices rather than positions is provided for backward compatibility with UIUC MOCA. BOUNDP is used if BOUND=(0 0 0 0) as it is by default. Finally, the CHAR parameter provides the Gaussian distribution parameters in units specified by the LENGTH on the UNITS statement (see "Grid" on page 965). It works very similar to the Gaussian doping statement used by Atlas. The values specified are standard deviations in the directions respectively.

$-x, +x, -y, +y$ 20-3

Biasing

Assign any non-zero applied voltage biases using the **SOLVE** statement.

For backward compatibility with MOCA, you may assign the voltage for every contact defined using the VBIAS parameter on the **DEVICE** statement as follows:

```
DEVICE VBIAS=(Vi,Vj,Vk,Vl)
```

where each value of the VBIAS parameter is specified in Volts. The subscripts i, j, k, and l refer to the contacts in the order they are defined. In this case, first suppose the source and drain contacts are defined, followed by the gate and finally a substrate contact. Then, the statement above will assign voltages as follows:

```
DEVICE VBIAS=(Vs=Vi Vd=Vj Vg=Vk Vsub=Vl)
```

The **DEVICE** statement supports up to 16 contact definitions or 16 values on the VBIAS parameter.

Defining contacts and then assigning each one a potential is essentially setting the boundary conditions for the simulation. There are some cases where you must be careful. For example, if you define a MOS device with a very shallow body, but in reality when the device is fabricated, then there is a much larger distance to the substrate grounding potential. The simulation results then may not be accurate.

The preferred method is to set the applied biases with the **SOLVE** statement. MC Device will stop with an error if you use both the **DEVICE** statement and the **SOLVE** statement to set non-zero biases. Instead, you may only use the **DEVICE** statement or the **SOLVE** statement to do this. MC Device provides you ways to define fields over chosen regions. The **FIELD** statement lets you do this.

```
FIELD     MAXFIELD=(start,end)     RAMP=(start,end)     NEGFIELD=0|1
USEBARR=0|1
```

MAXFIELD defines the starting and ending maximum electric fields in [V/cm]. The RAMP parameter sets the iteration (ITER) on which to start ramping the field and the iteration on which to end ramping the field. NEGFIELD sets whether to use the negative of the field. USEBARR sets whether to use the material-dependent band-offset barriers in the field calculation.

The material-dependent band-offset barriers are determined by the AFFINITY parameter (for the conduction band barriers) and the AFFINITY and EGAP parameters (for the valence band barriers) on the **MATDEF** statements, which appear at the end of the defaults.in file. The material-dependent band-offset barriers are only used when using the multi-material model (when TYPE=1 on the MMAT statement).

MC Device also allows you to setup bias conditions that vary linearly between them. The **LINBIAS** statement allows you to choose a specific **REGION** and specify the boundary voltages of that region.

```
LINBIAS LBREGION=n LBX1=Vx1 LBX2=Vx2 LBY1=Vy LBY2=Vy2
```

The LBREGION parameter chooses which region this **LINBIAS** statement applies to. The remaining fields set the voltage for each of the 4 boundaries of the chosen region. LBX1 sets the potential at the left-hand x boundary. LBX2 sets the potential at the right-hand x boundary. LBY1 sets the potential at the top y boundary. LBY2 sets the potential at the bottom y boundary.

20.5.2 Ohmic Contacts

For metal-semiconductor contacts, the barrier height, ϕ_B, represents the difference between the metal work function and the electron affinity of the semiconductor, χ, namely

$$\phi_B = \Phi_m - \chi \quad \text{20-4}$$

and is taken from the material library. For silicon contacts, the barrier height is found using the surface electron concentration, $n_s(x)$ (where x is the position along the boundary), and the conduction band density of states, N_c (where n_s is found by assuming charge neutrality at each node on the contact). For the non-degenerate case:

$$\phi_B(x) = \frac{k_B T_L}{q} \ln\left(\frac{N_c}{n_s(x)}\right) \quad \text{20-5}$$

For the degenerate case, the logarithm above is replaced by an inverse Fermi-Dirac integral. In all cases, the surface potential, ϕ_s, is determined by the barrier height, ϕ_B, and the applied bias, V_a, as:

$$\phi_s(x) = V_a - \phi_B(x) \quad \text{20-6}$$

Regarding the MC simulation, ohmic contacts must obviously force a quasi-equilibrium concentration. We adopt the same algorithm as used by DAMOCLES [127].

The total charge along a contact is computed, and particles are injected to the mesh nodes where the depletion is maximum. It must be observed that, for the case of non-uniform doping or mesh spacing or both, this method can easily lead to unphysical non-uniformities of concentration. The only solution to this problem is to use small, uniform contacts.

In MC Device, the **INJECT** statement controls injection from the contacts and **RESIST** controls the resistance of the contact region.

```
INJECT DN=0.0 DP=0.0 HEMIMAX=NO OVERINJ=0.0 TSTEP=1
```

DN and DP allow the selection of the fraction of electron and holes injected. This is more useful in bipolar mode when both particle types are actually simulated simultaneously. HEMIMAX sets injection to be a hemi-Maxwellian distribution (or not). OVERINJ sets the allowed fraction of over-injection. So OVERINJ=0.01 allows 1% over-injection. TSTEP sets the injection time step relative to the simulation time step. The default in MOCA is to inject every time Poisson is solved.

```
RESIST OHM=0.0 CRNUM=-1 DIR=1 CONTNUM=-1 TSTEP=PINF
```

OHM defined the ohmic value of the contact in Omega-cm. CRNUM is the index of the region in the input file from which to take material properties. DIR is the direction primary injection direction for the contact, where 1 is for the x-direction and 2 is for the y-direction. CONTNUM is the index of the actual contact in the .in input file. TSTEP is the simulation step interval on which to update values.

You can also control reflection from boundaries and contacts in MOCA. The **REFLECT** statement is used to set the minimum distance travelled after reflection. By default, this value is set to -1 to turn off the function.

```
REFLECT MINDIST=-1E0
```

The value of MINDIST is given in cm.

Note: Unlike other regions, the information about a contact is not overwritten by subsequent regions. Once you specify a grid point to be a contact, it will remain that way regardless what follows in the input file.

20.5.3 Estimators

Current Regions

MC Device allows you to define one or more rectangular current regions (C-Regions) where you can calculate the current. A **CREGION** statement takes the form:

```
CREGION BOUNDP=(0.0275 0.0675 0.0 0.08)
```

The BOUNDP parameter contains the boundaries of the region as on the **REGION** statement (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

The total current is the sum of the conduction current and the displacement current. The conduction current is computed by assuming that carriers flow either along the X or along the Y direction (or through two opposite sides of each rectangular region). The displacement current is computed by considering the current across all four edges of each rectangular region.

Consider the case of the X-oriented current, flowing in the negative X-direction (the normal case for an *n*-MOSFET). Case a in Figure 20-4 illustrates how to calculate the current exiting from the left-hand side of a rectangle, such as at the source of an *n*-MOSFET. Case b in Figure 20-4 illustrates how to calculate the current entering from the right-hand-side of a rectangle, such as at the drain of an n-MOSFET.

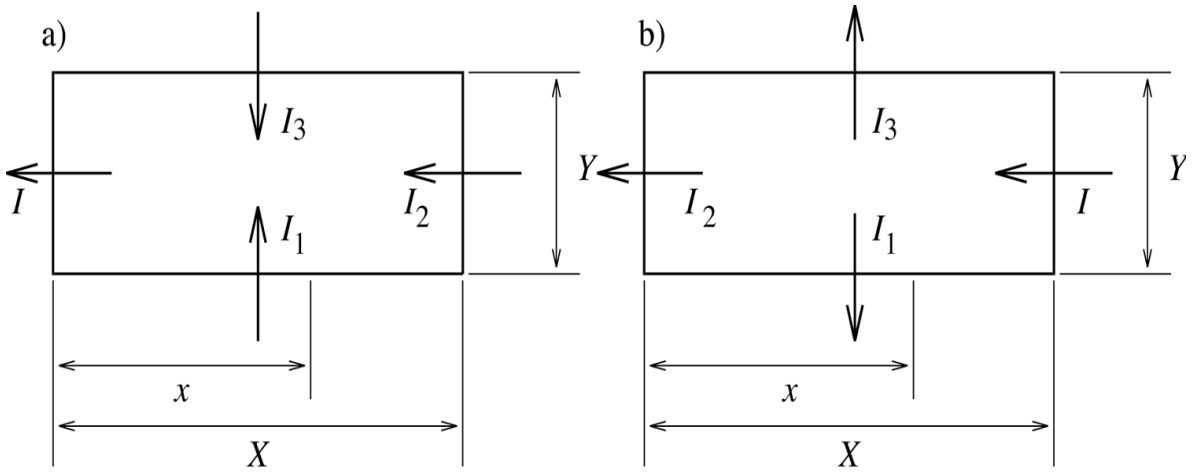


Figure 20-4: Rectangular regions are employed to compute the total current I for cases a and b, which pertain to the source and drain of an n-MOSFET.

Since the sum of the currents into any region is 0, the current I can be written, in both cases, as:

$$I = I_1 + I_2 + I_3 \quad \text{20-7}$$

where I, I_1, I_2, and I_3 refer to either to case a or case b collectively.

We will use the equation

$$I = \int J_{tot}\, dl = \int_0^L [\dot{D} + J]dl = \int_0^L [\varepsilon\dot{E} + J]dl \quad \text{20-8}$$

where the total current density, J_{tot}, is written as the sum of the displacement current density, $\dot{D}$, and the conduction current density, J, and where I represents a current per unit length in a 2D domain.

The source current I in case a of Figure 20-4 is then:

$$I = \int_0^x \varepsilon\dot{E}_y(x', 0)\mathrm{d}x' - \int_0^Y [\varepsilon\dot{E}_x(x, y) + J(x, y)]dy - \int_0^x \varepsilon\dot{E}_y(x', Y)dx' \quad \text{20-9}$$

We now consider the average current within the C-Region by integrating the RHS of Equation 20-9 from x=0 to x=X and by dividing by X, namely:

$$I = \frac{1}{X}\left[\varepsilon\int_0^X\int_0^x [\dot{E}_y(x', 0) - \dot{E}_y(x', Y)]dx'\mathrm{d}x - \int_0^X\int_0^Y J(x, y)\mathrm{d}y dx - \varepsilon\int_0^X\int_0^Y \dot{E}_x(x, y)\mathrm{d}y dx\right] \quad \text{20-10}$$

The first integral in Equation 20-10 is evaluated as:

$$\int_0^X\left(\int_0^x \dot{E}_y(x', \eta)dx'\right)dx = \int_0^X\left(\int_0^X \dot{E}_y(x', \eta)H(x - x')dx'\right)dx \quad \text{20-11}$$

$$= \int_0^X \dot{E}_y(x', \eta)dx'\int_0^X H(x - x')dx' \quad \text{20-12}$$

$$= \int_0^X \dot{E}_y(x, \eta)(X - x)dx \quad \text{20-13}$$

where H is the Heaviside or unit step function.

The expression for the source current I in case a in Figure 20-4 is then:

$$I = \frac{1}{X}\left[-\int_0^X\left(\int_0^Y J((x, y)\mathrm{d}y)\right)dx + \varepsilon\int_0^Y [\dot{\phi}(X, y) - \dot{\phi}(0, y)]dy + \varepsilon\int_0^X (X - x)[\dot{E}_y(x, 0) - \dot{E}_y(x, Y)]dx\right] \quad \text{20-14}$$

where the x component of the electric field, E_x, has been expressed in terms of the electrostatic potential, ϕ, using the fundamental theorem of calculus.

Similarly, the drain current I in case b in Figure 20-4 is:

$$I = -\int_0^X \varepsilon \dot{E}_y(x', 0)dx' - \int_0^Y [\varepsilon \dot{E}_x(x, y) + J(x, y)]dy + \int_x^X \varepsilon \dot{E}_y(x', Y)dx' \quad \text{20-15}$$

Again, we consider the average current in the C-Region by integrating the RHS of Equation 20-15 from x=0 to x=X and by dividing by X. The expression for the drain current I in case b in Figure 20-4 is then:

$$I = \frac{1}{X}\left[-\int_0^X \left(\int_0^Y J(x, y)\mathrm{dy}\right)dx + \varepsilon\int_0^Y [\dot{\phi}(X, y) - \dot{\phi}(0, y)]dy + \varepsilon\int_0^X x[\dot{E}_y(x, Y) - \dot{E}_y(x, 0)]dx\right] \quad \text{20-16}$$

In both Equations 20-14 and 20-16, the first term represents the conduction current. The second term represents the the displacement current flowing through the left (x=0) and right (x=X) boundaries. The third term represents the displacement current flowing through the lower (y=0) and upper (y=Y) boundaries.

In steady state, all displacement currents average to zero, so the second and third terms in Equations 20-14 and 20-16 drop out. So in steady state, the current in both cases is negative the integral over y of the X-average of the conduction current density.

During the MC simulation, the current will be averaged over time to reduce the noise and increase the accuracy of the estimates. The time derivatives of the potentials and electric fields do not need to be time-averaged. Instead, these can be directly calculated with a backward difference approximation to the derivatives. The conduction current density, however, is time averaged.

The sign convention for the current was adopted from MOCA where a net flow of electrons in the positive x direction (from source to drain in an n-MOSFET) was defined as positive current despite the fact this is contrary to the standard definition. Conversely, a net flow of holes in the positive x direction (from source to drain in a p-MOSFET) is defined as negative current. Since the currents calculated by a current regions is not necessarily associated with a terminal, the reference direction for the current is not related to the terminals. This means that the source and drain currents from current regions (normally the x-component of the current in each current region) will be positive for n-MOSFETs and negative for p-MOSFETs when these devices are biased in the linear or saturations regions.

Energy Regions

MC Device also allows you to define one or more rectangular regions used for the calculating the energy estimator (EE-Regions). A **EEREGION** statement takes this form:

```
EEREGION BOUNDP=(0.0051,0.0102,-0.0015,0.0102)
```

The BOUNDP parameter contains the boundaries of the region as on the **REGION** statement (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

The energy estimator is calculated for each **EEREGION** in a way similar to how the total current is calculated (see "Current Regions" on page 972).

20.5.4 Initial Conditions

MC Device allows you to customize many quantities used in the simulation. You can do the following:

- Defining regions over which the field is constant.
- Setting the initial particle concentration in a given region.
- Setting the particle injection rate.
- Setting a linear bias over a region.

FFREGION allows setting up a constant field region. The **FFREGION** statement has three parameters: FIELDX, FIELDY, and BOUNDP. FIELDX sets the x-directed field at the four edges of the region. FIELDY sets the y-directed field at the four edges of the region. BOUNDP defines the boundaries of the region (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

```
FFREGION \
  FIELDX=(Ex1,Ex2,Ey1,Ey2) \
  FIELDY=(Ex1,Ex2,Ey1,Ey2) \
  BOUNDP=(x1,x2,y1,y2)
```

You can use the **ICLREGION** statement to set the initial carrier concentration. This statement is useful for controlling the initial solution to Poisson's equation.

```
ICLREGION CONC=conc BOUNDP=(x1,x2,y1,y2)
```

CONC sets the initial concentration. BOUNDP sets the boundary wherein this concentration is applied (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

You can use the **QSS** statement to set a fixed surface charge density. This statement is useful for studying trapped oxide charge in MOS devices.

```
QSS QSCHARGE=conc BOUNDP=(x1,x2,y1,y2)
```

QSCHARGE sets the fixed surface charge density. BOUNDP sets the boundary wherein this concentration is applied (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

If you need to define a line charge, then set *y1* and *y2* to be equal to one another.

20.5.5 Bipolar

For devices where impact ionization is important, you can run MC Device in bipolar mode. The **BIPOLAR** statement sets the interprocess communication parameters.

```
BIPOLAR  COMM=0|1|2  MSFILE="msfile"  SMFILE="smfile"  NSLAVE=n
INITNLIN=conc \
 NFRACII=0.0
```

The COMM field sets the communication method, 0 is to turn bipolar mode off, 1 is for master, and 2 is for slave (unix pipe). If COMM is set to 1, the variable MSFILE is set to the master to slave pipe. If COMM is set to 2, then the variable SMFILE is set to the slave to master pipe. NSLAVE is the number of slave particles. INITNLIN is the initial slave concentration. NFRACII is the fraction at which to increase the number of particles.

20.5.6 Track Boundary Crossings

Along with using statements in the scattering section to track particles (see "Impact Ionization and other topics" on page 1011), there is **CROSSEST** to track particles crossing boundary regions. The **CROSSEST** statement specifies where region you want to track particles. Multiple **CROSSEST** regions may be defined.

```
CROSSEST CROSSREG=n
```

20.5.7 Time Step Regions

To precisely simulate some areas of a device where carriers move quickly, you can use **DTREGION** to set a small time step. Conversely, to efficiently simulate areas of a device where carriers move slowly, you can use **DTREGION** statements to set a larger time step.

Use the **DTREGION** statement to define the area over which to use the fine time step.

```
DTREGION BOUNDP=(x1 x2 y1 y2)
```

The region defined by **DTREGION** will be simulated every flight time step while the rest of the device will only be updated every Poisson time step. This way you can accurately simulate, say, the channel region with a flight time step of 0.01 fs while still keeping the Poisson step around 0.1 fs. The BOUNDP parameter defined the rectangular boundary of the region (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

20.5.8 Importing Data

MC Device allows you to import data from input data files. In particular, you can import the electric potential, the initial MC carrier concentration, doping profiles, regions and mesh. You can import all these data or any combination of them. All imports must be done from one file type, specified by the TYPE parameter on the **IMPORT** statement.

```
IMPORT TYPE=1 VOLT=FALSE AVGVOLT=FALSE \
  NCONC=FALSE AVGNCONC=FALSE PCONC=FALSE AVGPCONC=FALSE DOP=TRUE \
  MESH=FALSE REGIONS=FALSE \
  DIPSH=0 OFFSET={x,y} BOUNDP=(x1,x2,y1,y2) \
  DOPFILE="ref/dop.in"
```

The TYPE parameter allows you select from eight different file formats:

- TYPE=1 is for UIUC MOCA files.
- TYPE=2 is for PISCES version-2 files.
- TYPE=3 is for PISCES version-3 files.
- TYPE=4 is for ISE files.
- TYPE=5 is for rectangular grid files.
- TYPE=6 is for TIF files.
- TYPE=7 is for Well-Tempered MOSFET (WTM) files.
- TYPE=8 is for Silvaco structure files.

The TYPE parameter sets the file type for all imported files. The default value of TYPE=8 is for Silvaco Structure files.

VOLT, AVGVOLT, NCONC, AVGNCONC, PCONC, AVGPCONC, and DOP are boolean parameters that enable you to import the related quantities. By default, these parameters are FALSE so that no import occurs. Setting VOLT to 1|TRUE|YES will import the "Potential" as the electric potential. Setting AVGVOLT to 1|TRUE|YES will import the "Average Potential" as the electric potential. You may not set both VOLT=1 and AVGVOLT=1. When CARRIER=E on the **ALGO** statement, setting NCONC to 1|TRUE|YES will import the "Electrons" as the initial electron concentration and setting AVGNCONC to 1|TRUE|YES will import the "Average Electrons" as the initial electron concentration. You may not set both NCONC=1 and AVGNCONC=1. When CARRIER=H on the **ALGO** statement, setting PCONC to 1|TRUE|YES will import the "Holes" as the initial hole concentration and setting AVGPCONC to 1|TRUE|YES will import the "Average Holes" as the initial hole concentration. You may not set both PCONC=1 and AVGPCONC=1. Setting DOP=1|TRUE|YES will import the "Net Doping" if IONIZ=0|FALSE|NO on the POISSON statement (for complete ionization of dopants). Setting DOP=1|TRUE|YES will import "Boron", "Arsenic", "Indium", and "Phosphorus" if IONIZ=0|FALSE|NO on the POISSON statement (for incomplete ionization of dopants).

MESH and REGIONS are boolean parameters that enable you to import these structure elements from a Silvaco structure files (TYPE=8). Setting MESH=1|TRUE|YES will import the mesh from STRFILE and ignore **XMESH** and **YMESH** statements in your input file. Since MC Device supports only rectangular tensor product meshes, meshes of a type other than this are transformed to a rectangular tensor mesh which includes every x position and every y position in the imported mesh. Setting REGIONS=1|TRUE|YES will import regions from STRFILE. You must comment out any **REGION** statements in your input file unless you want to add additional regions or redefine regions which get imported. Since MC Device supports only rectangular regions, regions with a shape different than this are expanded to a rectangle which includes every mesh point in the imported region. You can not import contacts from a structure file using an **IMPORT** statement. Instead, when REGIONS=1|TRUE|YES, MC Device will continue to process any uncommented **REGION** statements in your input file. Those with TYPE=CONTACT will represent the contact regions for the device and any contacts in the structure file will be ignored. Contacts in your input file are added to the imported structure in the order they appear in the input file. So when you import a structure from strfile using TYPE=8, you will still need to use **REGION** statements with TYPE=CONTACT in your input file to create contacts for the imported structure.

The DIPSH parameter is set to 0 (off) by default. You may set DIPSH to 1 (on) to use ISE DIPSH with the import.

The OFFSET parameter is an (X,Y) offset for importing quantities. The OFFSET parameter is (0,0) by default and (X,Y) are specified in cm.

The BOUNDP=(x1 x2 y1 y2) parameter sets a rectangular boundary over which doping imports are applied. If any of x1, x2, y2, or y2 are left as their default values of 0, then these are set to the xmin, xmax, ymin, ymax of the device structure respectively, so that doping imports are applied to the entire device by default. Alternatively, you can set x1, x2, y1, and y2 to set to defined a window for importing the doping. Imports for the voltage and concentration are always done over the whole device, so these are not affected by the BOUNDP parameter (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

The VOLTFILE, CONCFILE, DOPFILE, GRIDFILE, SOLFILE, VERFILE, TRIFILE, SDFILE, RETROFILE, and STRFILE parameters provide the filenames of the data to be imported. The particular parameters that are used depend on the TYPE parameter.

For TYPE=1, VOLTFILE, CONCFILE, and DOPFILE are used, where VOLTFILE is a 2D voltage file, CONCFILE is a 2D MC carrier concentration file, and DOPFILE is a 2D doping file.

For TYPE=2|3, GRIDFILE and SOLFILE are used, where GRIDFILE is a 2D grid file and SOLFILE is a 2D solution file.

For TYPE=4|5|6, VERFILE, TRIFILE, DOPFILE, and SOLFILE are used, where VERFILE is a 2D vertex file (a 2D grid file when DIPSH=1), TRIFILE is a 2D triangle file (a 2D data file when DIPSH=1), DOPFILE is a 2D doping file, and SOLFILE is a 2D solution file.

For TYPE=7, SDFILE, RETROFILE, and VOLTFILE are used, where SDFILE is a 2D source-drain doping file, RETROFILE is a 1D retrograde doping file, and VOLTFILE is a 2D voltage file.

For TYPE=8, STRFILE is used, where STRFILE is a 2D Silvaco structure file.

For example, if you set volt=1|TRUE|YES and TYPE=1 (for the UIUC MOCA file type), then MC Device will import the electric potential (voltage) from the file specified by the VOLTFILE parameter (which is, by default, volt.in). Correspondingly, if you set CONC=TRUE and TYPE=1, then MC Device will import the MC carrier concentration from the file specified by the CONCFILE parameter (which is, by default, conc.in). Finally, if you set DOPING=TRUE and TYPE=1, then MC Device will import the doping profiles from the file specified by the DOPFILE parameter (which is, by default, dop.in).

20.6 Statistical Enhancement

The methods for statistical enhancement are controlled with the **ENHAN** statement.

```
ENHAN MODE=2 TSTEP=2 ESTEP=100 XSTEP=(0,0,0) \
  LOTHRE=1e-20 HITHRE=1e20 RELTHRE=0.05 RATIO=2. N=10
```

The default value of MODE=0 indicates that statistical enhancement is turned off. MODE=1 is deprecated and should not be used. MODE=2 selects the non-local gathering algorithm also known as phase-space balancing [232, 233]. MODE=3 selects the comb method (for bulk simulation only). In all cases, TSTEP is the interval in time steps between calls of the variance-reduction subroutine. This depends on a compromise between quality of the simulation (reduction of the noise) and CPU time spent in the balancing routine. Usually, a balancing step of less that 100 fs is needed to achieve significant reduction of the noise. ESTEP and XSTEP specify the partitioning of phase space in units of meV and mesh cells respectively. Time steps between application of the enhancement algorithm.

You can also specify only a sub-region of the device to perform the statistical enhancement. The BOUND parameter defines a rectangular boundary where to apply the statistical enhancement.

The non-local gathering method works by a two-phase process. First, the ensemble of particles will be scanned. A linked list is then created that contains all particles that do not contribute to the simulation or redundant particles. A particle is marked as redundant if its weight is less than RELTHRE times that of the largest particle in the same bin. Then, the average number of particles per bin is computed. Particles in under-populated bins are split until the occupation of phase space is balanced. Every splitting needs some storage for the newly-created particles. This storage is obtained by gathering particles. You can perform gathering on the redundant set if this is available. If more splittings are needed than redundant particles are available, some local gatherings will be performed in bins with higher particles.

The algorithm is completely controlled by only five variables:

- **LOTHRE**
- **HITHRE**
- **RELTHRE**
- **RATIO**
- **N**

These will be described in the following sections.

20.6.1 LOTHRE

The **LOTHRE** parameter is the single most important parameter in the non-local gathering method. It is the minimum weight allowed for particles in the simulation. That is, particles with weight below **LOTHRE** cannot be split further. This minimum threshold defines the accuracy of the simulation, since the weakest details will have to be resolved by very small particles. Usually, a minimum weight of 10^{-20} to 10^{-10} is adequate to resolve most important high-energy effects. Since this parameter is a double-precision number, you can specify a very small value. A small value for **LOTHRE** uses particles to populate a large number of bins. Therefore, it trades the overall accuracy of the simulation for a wider coverage of phase space.

20.6.2 HITHRE

The **HITHRE** parameter controls the maximum allowed size for particles in the simulation. It is used in two different ways. One way is to control the maximum total weight for particles in a gathering set if the redundant set is used. Another way is to control the maximum size of any particle in the gathering set if local gathering is used. A value of 1.0 will force a maximum unit weight for all particles, so that no particle will be larger than in the conventional situation. This will inhibit any gathering starting from a situation where all particles have unit weight. In this case, if you apply balancing to an initial status from a non-balanced simulation, then some extra particles will have to be allocated (NREF larger than the initial number of particles). Therefore, the algorithm will be able to expand the ensemble without gathering particles.

20.6.3 RELTHRE

The **RELTHRE** parameter defines the criterion for marking particles as redundant. A value of 10^{-3} means that a particle will be included in the redundant set only if there exists another particle in the same bin with a weight 1,000 times larger. While a value of 10^{-3} is safe because it only tries to “reuse” very small particles, it may be inefficient in cleaning the bins of particles that do not contribute to the simulation. A large value like 10^{-1}, however, may mark as redundant particles that actually contribute significantly to the estimators.

The optimum value of **RELTHRE** depends on the dynamic range of weights in any given bin. A simple rule of thumb can be derived from the exponential dependence of electron occupancy on energy. We can reasonably take the thermal-equilibrium distribution as the steepest possible distribution in a common device. If ΔE is the width of the energy bins, we have a possible dynamic range inside a bin of $exp\ \ (\Delta E/k_B T_L) \approx 50$ for 0.1 eV bins and room temperature.

In this case, you can see a value of 10^{-2} for **RELTHRE** is safely low enough. Also for space dependence, you can apply the same criterion, where the energy range is simply the maximum voltage drop across a space bin. For a typical bin width of 10 nm and an electric field of 200 kV/cm, we have an energy difference of 0.2 eV, which yields a dynamic range of about 2500. For this case, a value for **RELTHRE** below 4×10^{-4} is suggested by our rule of thumb.

Note: In high-field regions, you cannot assume a thermal-equilibrium distribution. But in high-field regions, you should not use the guidelines described above.

20.6.4 RATIO, N

These **RATIO** and **N** parameters define the gathering process. **RATIO** is the maximum dynamic range of weights for any gathering. n is the maximum number of particles in the gathering set. Let us assume that n particles are being gathered and the **RATIO** of weights between the largest and the smallest particle is **RATIO**. This means that if the smallest particle is retained, its weight will increase by a factor between **RATIO** and n×**RATIO**. If there is a great increase in weight, it will worsen the numerical noise. To estimate the maximum tolerable increase of weight, consider the case of gathering of redundant particles. For each redundant particle, there is another particle in the same part of phase space with a weight **RELTHRE**$^{-1}$ times larger. The increase will be tolerable if the retained particle remains quite smaller than the dominant particles in the same bin. In other words, if:

$$ratio \times n \ll relthre^{-1} \qquad 20\text{-}17$$

This gives us the rule of thumb:

$$ratio \ll \frac{relthre^{-1}}{n} \qquad 20\text{-}18$$

For example, with **RELTHRE**=10^{-2} and n=10, a dynamic range equal to 10 can be tolerated before an excessive increase of variance occurs. If you use a very aggressive criterion for marking redundant particles (e.g., **RELTHRE**=10^{-1} or more) then rescale **RATIO** accordingly. Otherwise, the sudden growth of the redundant particles will create large spikes in the estimated distribution function. The optimum value for **RATIO** depends on the number of particles used. For a large number of particles (20,000 or more), a value as low as 2 allows a very small increase in variance without impairing the performance. For small number of particles, a value of 10 or more can be necessary to collect a sufficient number of redundant particles for the gathering to be effective. If the constraint becomes too strict, lower **RELTHRE** instead of setting **RATIO** to a very low value.

20.6.5 Particle Compression

MC Device automatically sets a reference number of particles, N_{ref}, as slightly less than the number physically allocated in memory. In this way, it allows for some fluctuations of the number of computational number of particles. If the current number n of particles is larger than N_{ref}, MC Device can perform some gatherings with no splittings to reduce the total number. This step is called compression. If n<N_{ref}, the extra storage available is used to perform some “free” splittings with no gatherings. Under normal conditions, at the exit of the variance-reduction procedure the number of particles will be equal to N_{ref}. The introduction of N_{ref} solves a problem intrinsic with the manipulation of the particle states. The total number of particles in the simulation is related to the transit time of carriers through the device. If no variance reduction is used, this will be the physical transit time, so that the number of particle will be easily estimated and controlled. The introduction of variance reduction algorithms changes the distribution of MC particles in phase space, so that the computational transit time can be higher or lower than the physical one. This leads to unpredictable drifts and fluctuations in the storage required by the program. You can only use the compression step by a many-particle variance-reduction scheme [233]. This solves this problem by forcing an almost-constant number N_{ref} of particles.

20.6.6 Statistical Enhancement Statements

The following statements don't specifically reduce the noise during the simulation. These are only statements that smooth the output. **ESTIM1D** allows you to select a region and make a 1D cut through it and study quantities of that cut. **SMREGION** sets a region to perform the smoothing of the outputs.

```
ESTIM1D DIR=X/Y BOUNDP=(x1,x2,y1,y2)
```

DIR sets what direction to make the cut in. BOUNDP defines the boundary where to run the 1D estimator (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

```
SMREGION BOUNDP=(x1,x2,y1,y2) TYPE=n
```

The BOUNDP parameter contains the boundaries of the region, as on the **REGION** statement (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility. TYPE sets the quantity to smooth. Setting TYPE to 1 smooths the potential, ETA. Setting it to 2 smooths the quantum potential, QETA.

20.7 Physical Models

20.7.1 Self-Consistent Simulation

Background

Poisson's equation is solved on the same grid used to define the device. Assuming CARRIER=E on the **ALGO** statement, the non-linear Poisson equation takes the form

$$-\nabla \cdot (\varepsilon \nabla \phi) = e(-(n_{MC} + n(\phi)) + N_D^+(\phi) - N_A^-(\phi) + p(\phi)) . \quad 20\text{-}19$$

Here, ε is the permittivity. ϕ is the electrostatic potential. e is the fundamental electron charge. n_{MC} is the electron concentration computed using the Monte Carlo cloud-in-cell (CIC) scheme [127]. n_{MC} is zero in semiconductor contact regions and non-contact gate regions and non-zero in MC regions. $n(\phi)$ is the electron concentration computed using ϕ and N_c using Equation 20-23. N_C is the effective density of states of the conduction band. N_C and $n(\phi)$ are zero in MC regions. N_C and $n(\phi)$ are non-zero in semiconductor contact regions and non-contact gate regions and zero in MC regions. N_D^+ is the ionized donor concentration. N_A^- is the ionized acceptor concentrations. $p(\phi)$ is the hole concentration computed using ϕ and N_V using Equation 20-22. N_V and $p(\phi)$ are non-zero in all semiconducting regions.

All potential-dependent terms in the RHS of Equations 20-19 are determined self-consistently using the quasi-Fermi potentials in each mesh cell and the electrostatic potential, ϕ, at each grid point. For 1D simulation, the linear Poisson equation is solved and holes must be considered in the MC simulation.

For CARRIER=H on the **ALGO** statement, -n and p are switched in Equation 20-19.

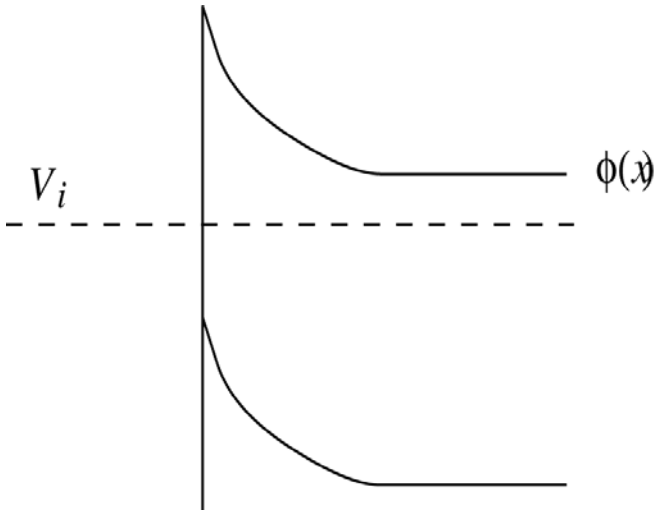


Figure 20-5: V_i is the contact bias at contact *i*. By default, $E_{fp}=-e*V_i$ is the hole quasi-Fermi energy. $\phi(x)=(E_g(Si)/2 - E_c(x))/e$ is the electrostatic potential.

Still assuming CARRIER=E on the **ALGO** statement, the hole density is computed as

$$p(\phi) = \int_0^\infty dE\ g_v(E) f_D((E - E_V + E_{fp})/k_B T_L) = \int_0^\infty dE\ g_v(E) f_D((E + e\phi(x) + E_g - eV_i)/k_B T_L) \quad 20\text{-}20$$

where $f_D(\xi) = f_D(\xi, g=1)$ is the Fermi-Dirac distribution and generalizes for a degeneracy g to

$$f_D(\xi, g) = \frac{1}{1 + g \cdot \exp(\xi)} \quad \text{20-21}$$

This is the Fermi-Dirac distribution including degeneracy. ξ is a unitless factor proportional to the energy, E_g is the bandgap energy. $g_v(E)$ is the energy-dependent valence-band density of states. E_{fp} is the hole quasi-Fermi energy, and V_i is the applied voltage at contact i.

By default, within contact region i, E_{fp}=-eV_i. Also by default, within the doping well of contact i, E_{fp}=-eV_i, where each contact is assumed to reside in a doping well of one type (n or p). This is the default implementation of the constant quasi-Fermi level method in MC Device.

The USEFERMI and FERMI parameters on the **REGION** statement provide you with a way to modify the default behavior of the constant quasi-Fermi level method. For example, for a *n*-MOSFET with an inversion channel and assuming Ohmic contacts, you can change the default choice for the hole quasi-Fermi energy of E_{fp}=-eV_{drain} in the well adjacent to the drain contact by setting E_{fp} using USEFERMI=1 and FERMI=<value> on the **REGION** statement for the drain (see "Regions" on page 966). For example, you may wish to set E_{fp}= -min(V_{source},V_{drain}) = -min(0.0 eV,eV_{drain}>0 V) = 0.0 eV. To do so, you can then use the default value of FERMI=0.0 eV by simply including USEFERMI=1 on the **REGION** statement for the drain contact. Similarly, for a *p*-MOSFET when V_{source}=0.0 V and V_{drain}<0 V and when using CARRIER=H on the ALGO statement, you can use USEFERMI=1 on the **REGION** statement for the drain contact to get E_{fn}=-max(V_{source},V_{drain})=FERMI=0.0 eV in the drain well instead of the default of E_{fn}=-eV_i=-e*V_{drain}.

For non-MC carrier concentrations:

$$p(\phi) = N_v F_{1/2}((E_V - E_{fp})/k_B T_L) = N_v F_{1/2}((-e\phi(x) - E_g + \mathrm{eV}_i)/k_B T_L) \quad \text{20-22}$$

$$n(\phi) = N_c F_{1/2}((E_{fn} - E_c)/k_B T_L) = N_c F_{1/2}((-\mathrm{eV}_i + e\phi(x))/k_B T_L) \quad \text{20-23}$$

where

$$F_{1/2}(\xi) = \int_0^\infty \xi^{1/2} f_D(\eta - \xi)\, d\eta \quad \text{20-24}$$

is the Fermi-Dirac integral of order 1/2 and where f_D is defined above. E_{fn} is the electron quasi-Fermi energy and is equal to negative the applied voltage at contact i, V_i, within each contact region i. For incomplete ionization of donors, E_{fn} is also assumed to be constant and equal to the contact value within each doping well surrounding contact i. E_{fn} is calculated from Equation 20-23 using the average electron concentration, <n>, and the average conduction band edge, <E_c>, and taking E_{fn}=0 wherever <n>=0. Although MC Device uses Equation 20-23 to calculate E_{fn} throughout semiconductor regions, MC Device treats electrons to be in quasi-equilibrium (or to have thermalized to a distribution based on the lattice temperature). Instead, E_{fn}, is simply an output quantity that you can plot in energy band

diagrams. E_{fn} obtained with MC Device compares closely with E_{fn} obtained with the drift diffusion model inside regions of your device where quasi-equilibrium holds. Conversely, E_{fn} obtained with MC Device diverts from E_{fn} obtained with the drift-diffusion model in regions of your device where quasi-equilibrium does not hold.

For the fixed charge, if E_{D} and E_{A} are the donor and acceptor impurity energies (with respect to each corresponding band edge), we have for donors:

$$N_D^+(\phi) = N_D f_D(-E_c + E_D + E_{fn}, g_D) = N_D f_D(e\phi + E_D - \mathrm{eV}_i, g_D) \quad \text{20-25}$$

and for acceptors:

$$N_A^-(\phi) = N_A f_D(E_c + E_A - E_g - E_{fp}, g_A) = N_A f_D(-e\phi + E_A - E_g + \mathrm{eV}_i, g_A) \quad \text{20-26}$$

where $g_D=2$ is the donor degeneracy factor for phosphorus and arsenic. $g_A=4$ is the acceptor degeneracy factor for boron and indium.

The ionization energies of donors and acceptors are assumed constant. No heavy-doping effect, such as the donor deionization effect [168], are considered.

The 1D simulation has no specific input statement. The solver is actually controlled by the `MODE` variable on the `ONED` statement.

Most of the energies and concentrations used in the Equations 20-19 through 20-26 are saved in `*sol.str` (structure) files when performing a 2D simulation. See Section D.2 “File Formats”. You can perform 1D cuts of the 2D structures to plot energy-band diagrams relating to your solution.

The 2D Poisson solver is controlled through the **`POISSON`** statement

```
POISSON TSTEP=5 ITER=100 TOL=(2e-4,1e-6) IONIZ=NO \
  START=1 FREEZE=PINF LINPOI=NO CUTOFF=1e25
```

`TSTEP` sets the number of flight time steps between each Poisson solution. `ITER` is the maximum number of Newton iterations. `TOL` is a vector of 2 values, which is the L_2-norm tolerance for the Newton and conjugate-gradient methods respectively.

`IONIZ` is used to switch on and off incomplete ionization of dopants. Two reasons exist to disable incomplete ionization. First, the response time of dopant levels is much longer than the time scale used in the MC simulation. Therefore, it is unrealistic to treat dopants in the “adiabatic” approximation (i.e., with instantaneous response). Second, for very high doping levels (above 10E18 cm^{-3}), it is approximately correct to assume that all dopant atoms are ionized [243]. In any case, the program automatically switches off the `IONIZ` flag whenever the quasi-Fermi energies are not constant across a non-depleted region (e.g., in a JFET or MESFET simulation) where a finite voltage is applied across a non-depleted channel.

The fields `START` and `FREEZE` define the time window during which the solver actually works. The field is frozen outside that window, either using a result from a previous simulation, or an average from the same simulation. In the example above, the field is kept frozen during the first 1000 time steps. The Poisson solver is then used until the end of the simulation. The keyword `PINF` is internally defined as the largest positive integer. The keyword `NINF` is internally defined for the largest negative integer.

If the first field of FREEZE is greater than one, then an initial fixed field is needed to start the simulation. By default, MC Device reads this field from the file volt.in. This file is also written at the end of each run for use in a subsequent simulation.

LINPOI selects whether the linear or non-linear Poisson equation is solved. If LINPOI=1|TRUE|YES, then a linear version of the Poisson equation is solved where complete ionization of dopants is assumed and where the opposite carrier type as set by the CARRIER parameter on the ALGO statement is assumed to be 0. If LINPOI=0|FALSE|NO (as it is by default), the non-linear poisson equation is solved. In this case, the IONIZ parameter may be used to include incomplete ionization and the Poisson equation includes a expression for the density of the opposite carrier type as set by the CARRIER parameter on the **ALGO** statement as a function of the unknown potential.

CUTOFF sets an upper limit for the complete ionization of dopants. The default value of CUTOFF is 1e25 cm^{-3}. Because the default value of CUTOFF is high compared to typical doping concentrations, the CUTOFF parameter typically has no effect when the default value is used. The CUTOFF parameter will only have an effect if it is reduced to below the peak of the net doping profile of your device and if IONIZ=0|FALSE|NO for complete ionization. Since using the CUTOFF parameter can change the net doping profile that will be used internally by MC Device for solving Poisson's equation and for treating impurity scattering, the effects of using the CUTOFF parameter can be significant. The CUTOFF parameter should not be reduced below 3e19 cm^{-3} since this may dramatically affect impurity scattering. The CUTOFF parameter also affects the doping in the gate, so a value of the CUTOFF parameter below the net doping in the gate will change the threshold voltage. Since there are many restrictions associated with proper use of the CUTOFF parameter, we recommend that you not include the CUTOFF parameter in your input files. Instead, we encourage you to choose doping levels at or below the doping ionization limit when using complete ionization of dopants. Note that CUTOFF has no effect when IONIZ=1|TRUE|YES for incomplete ionization of dopants.

In MOCA, the IPOLY parameter turned the model for the depletion of polysilicon on and off. In MC Device, the IPOLY parameter is ignored and the depletion of polysilicon gates (or of any semiconducting region where the TYPE is not MC) is considered automatically whenever they appear in a device. For example, in MC Device, you can define the entire gate as type CONTACT, and no depletion will occur on the gate. Instead, you can set the region immediately above the gate oxide to a MAT=POLY (or any semiconducting material) and TYPE=BLOCK, and you can define a contact above this semiconducting non-contact region. In this case, the non-linear Poisson equation including potential-dependent terms for both the electron and hole concentrations is solved within the non-contact portion of the gate to account for the depletion effect.

Microscopic Analysis of Noise

Here is a brief analysis of voltage fluctuations in a conductive medium under the hypothesis of weak screening. This amounts to assuming a non-collisional system because we compute the fluctuations within a screening sphere from the generating carriers. Since the result will be expressed as a power spectrum, the analysis is valid for frequencies above the plasma frequency. The result will be useful when the plasma frequency is low (i.e., for low-density regions).

Velocity Autocorrelation

We start with an elementary model of one-dimensional velocity autocorrelation in a non-interacting gas of classical particles:

$$C_v(t) = E\{v(t')v(t'+t)\} \approx v_{th}^2 e^{-|t|/\tau} \quad 20\text{-}27$$

where v_{th} is the thermal velocity and τ is the momentum-relaxation time. At thermal equilibrium:

$$v_{th}^2 = \frac{k_B T_L}{m^*} \quad 20\text{-}28$$

The corresponding power spectrum of the velocity is

$$S_v(f) = \int_0^\infty v_{th}^2 e^{-2\pi f t - |t|/\tau} dt = \frac{2v_{th}^2 \tau}{1 + (2\pi f \tau)^2} \quad 20\text{-}29$$

and is approximately white up to the frequency $1/(2\pi\tau)$, which is several THz. This frequency must be compared with the plasma frequency that characterizes the dynamic screening of the thermal oscillations by the electron gas.

In the following, we will assume a white spectrum with uniform power density $v_{th}^2\tau$ for each of the three components of the particle velocity, truncated at the autocorrelation frequency. Although we consider the simple, constant-relaxation-time model at thermal equilibrium, the non-equilibrium case with variable scattering rate can be analyzed by replacing v_{th} with an average RMS group velocity and τ with an average relaxation time.

Current-Density Fluctuations

The current density at a point *r* in space is defined as the limit.

$$\boldsymbol{J}(\boldsymbol{r}) = \lim_{\Omega \to 0} \frac{1}{\Omega} \sum_{x_i \in \Omega} e\boldsymbol{v}_i \quad 20\text{-}30$$

From this definition, the power spectrum of the fluctuations of the k-th component of the current density can be readily computed as

$$S_{J_k}(f) = \lim_{\Omega \to 0} \frac{e^2}{\Omega^2} \sum_{x_i \in \Omega} S_v(f) \quad 20\text{-}31$$

$$S_{J_k}(f) = \lim_{\Omega \to 0} \frac{2\tau n e^2 k_B T_L}{\Omega m^*} \quad 20\text{-}32$$

It is clear that, as Ω tends to zero, the fluctuation diverges. It follows that the fluctuation is not well defined for the current density but only for its average on a finite volume. This can be verified by a simple derivation, from Equations 20-21, of the familiar formula for the Johnson noise. Assuming for simplicity a cylindrical resistor of length *L* and area *A*, the total (conduction plus displacement) terminal current can be computed as

$$I = \frac{1}{L}\int_0^L \left(\int_0^A J \right) d^3\boldsymbol{r} \quad \text{20-33}$$

$$I = \frac{1}{L} \sum_{\boldsymbol{x}_i \in L \times A} e(v_i)_x \quad \text{20-34}$$

The short-current noise power spectrum is therefore

$$S_I(f) = \frac{A}{L} \times \frac{2\tau n e^2 k_B T_L}{m^*} \quad \text{20-35}$$

$$S_I(f) = \frac{2A\sigma k_B T_L}{L} = \frac{2k_B T_L}{R} \quad \text{20-36}$$

This is the usual short-circuit thermal-noise current formula for a resistance R, in the relaxation time hypothesis where the conductivity σ is given by $ne^2\tau/m^*$.

Potential Fluctuations

The main reason for the divergence for the current-density fluctuation is the finite size of the charge carriers. Let us now analyze the power spectrum of the electostatic potential. Each carrier of charge e generates a potential

$$V_i(\boldsymbol{r}) = \frac{e}{4\pi\varepsilon|\boldsymbol{r} - \boldsymbol{r}_i|} \quad \text{20-37}$$

In a given point r of space, the total potential, V, will be given by the sum of all contributions from nearby carriers:

$$V = \frac{e}{4\pi\varepsilon}\sum_i \frac{1}{r_i} \quad \text{20-38}$$

where the origin has been conveniently chosen at $\boldsymbol{r}$. Deriving Equations 20-22 with respect to time and using Equations 20-20, we obtain the power spectrum of the time-derivative of the potential:

$$S_{\dot{V}}(f) = \frac{e^2\tau k_B T_L}{8\pi^2\varepsilon^2 m^*}\sum_i \frac{1}{r_i^4} \quad \text{20-39}$$

The summation, for a large number of particles, can be approximated with an integral:

$$S_{\dot{V}}(f) = \frac{e^2\tau k_B T_L}{8\pi^2\varepsilon^2 m^*}\int_0^\infty 4\pi r^2 n \frac{1}{r^4}\mathrm{dr} \quad \text{20-40}$$

$$S_{\dot{V}}(f) = \frac{e^2\tau k_B T_L n}{2\pi\varepsilon^2 m^*}\int_0^\infty \frac{1}{r^2}\mathrm{dr} \quad \text{20-41}$$

Again, the integral presents a pole for vanishing distances, due to the 1/r dependence of the potential. We may remove the singularity by assuming a minimum characteristic distance, r_c. In a physical sense, that forces a discretization cutoff in Equations 20-25. With computer simulations, r_c is the size of the discretization mesh on which Poisson's equation is solved. Applying this minimum-distance cutoff, we have

$$S_{\dot{V}}(f) = \frac{e^2 \tau k_B T_L n}{2\pi\varepsilon^2 m^* r_c} \quad 20\text{-}42$$

We have found the high-frequency behavior of the fluctuations of the potential can be useful for estimating the oscillations of potential barriers. Unfortunately, we have assumed no correlation between positions of the carriers, which results in a perfectly white spectrum of the time-derivative of the potential. This implies an infinite power of the potential fluctuations (i.e., a "random walk" of the potential). In reality, Coulomb interaction between carriers limits the possible oscillation of the potential. You can get a rough estimate of the total amplitude of the potential fluctuations by limiting the bandwidth of its derivative to some characteristic frequency f_c. It's usually the inverse of the dielectric-relaxation time. The power spectrum of the potential is then

$$S_V(f) = \frac{e^2 \tau k_B T_L n}{2\pi\varepsilon^2 m^* r_c [(2\pi f)^2 + (2\pi f_c)^2]} \quad 20\text{-}43$$

Integrating over the entire spectrum, we have

$$P_V = \frac{e^2 \tau k_B T_L n}{8\pi^2 \varepsilon^2 m^* r_c f_c}.$$

It is evident that as $f_c \to \infty$, the fluctuations of the potential diverge without bound.

Even after removing the similarities, we see that the total power is proportional to the factor $e^2 n$. This means that the granularity of the charge carriers affect directly the noise. Only when the critical radius r_c is scaled along with the charge n, the limit remains finite. With computer simulations, this means the mesh size must decrease by the same factor as the weight of the particles. In a physical system, you can obtain a rough estimate of the microscopic noise using the dielectric relaxation frequency $\sigma/2\pi\varepsilon$ as the cutoff frequency. If the conductivity is approximated as

$$\sigma = \frac{e^2 \tau n}{m} \quad 20\text{-}44$$

we have

$$P_V = \frac{e^2 \tau k_B T_L}{4\pi\varepsilon m^* r_c \sigma} = \frac{k_B T_L}{e} \times \frac{e}{4\pi\varepsilon r_c} \quad 20\text{-}45$$

The standard deviation of the electrostatic potential is the geometric mean between the thermal voltage and the potential at a distance r_c from a charge e.

In Equation 20-25, the volume integral converges within a radius of a few times r_c from the center point. This means that most of the potential noise is local and thus electrostatic screen. Entering Equation 20-25 as a convergence factor exp(-$r\beta_s$) does not affect it as long as r_c is smaller than the screening length $1/\beta_s$.

You may want to estimate the fluctuations for the potential for the case of a two-dimensional system of particles with charge λ per unit length in the third dimension. The corresponding potential generated form the i-th particle is

$$V_i = \frac{\lambda \ln r_i}{2\pi\varepsilon} \qquad 20\text{-}46$$

Along the lines of the derivation of Equation 20-26, we have

$$S_{\dot{V}}(f) = \frac{\lambda^2 \tau k_B T_L}{2\pi^2 \varepsilon^2 m^*} \int_0^\infty 2\pi r n (2 v_{th}^2 \tau) \frac{dr}{r^2} \qquad 20\text{-}47$$

$$= \frac{\lambda^2 \tau k_B T_L n}{\pi \varepsilon^2 m^*} \int_0^\infty \frac{dr}{r} \qquad 20\text{-}48$$

This time there are two singularities. One arising from the short-range effect and another one from the very-long-range action of the two-dimensional Coulomb force. Again, you can remove the short-range singularity by assuming a finite size for the particles or a discretized solution of Poisson's equation. The long-range action will be effectively screened at low frequencies. Therefore, a convergence factor $exp(-r\beta i)$ can be introduced into the integral of Equations 20-29. This, however, will remain largely unscreened for high frequencies (in the THz range). Therefore, in the high-frequency range, there is no upper bound on the potential fluctuations. Unlike the three-dimensional case, the fluctuations are mainly of a non-local nature and depend on the physical size of the two-dimensional system. We can conclude that a two-dimensional model of fundamental noise cannot account quantitatively for the microscopic fluctuation in the physical three-dimensional system.

20.7.2 Band Structure

Band Structure of Silicon

MC Device models the band structure of silicon by using a numerical representation of 2 or 4 conduction bands and 3 valence bands. Figure 20-6 shows the silicon band structure as obtained from a local pseudopotential calculation [65].

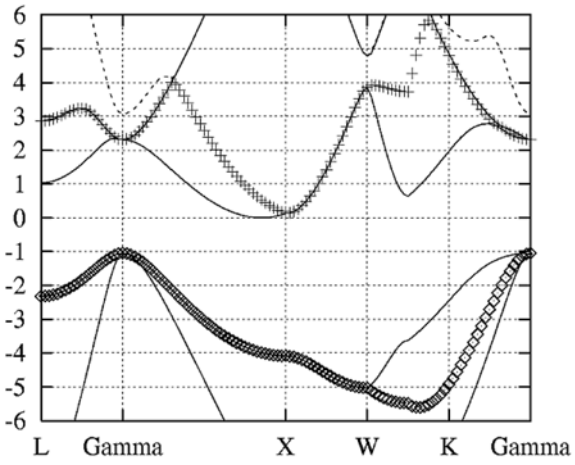


Figure 20-6: Silicon band structure.

The second and third conduction band of silicon are degenerate, or very close, from 2.3 eV to 4 eV over a large part of the Brillouin zone around Γ, so that electron kinematics at those energies is quite well described by the second band alone. Nevertheless, the third and fourth bands can affect energy gain through three “fast” Γ valleys, located at about 2.3 and 3.1 eV. Although these have a small density of states, they can work as effective “tunnels” to oxide-injection energies. Figures 20-7 and 20-8 show the density of states for the first 6 conduction bands and for the first 4 valence bands respectively.

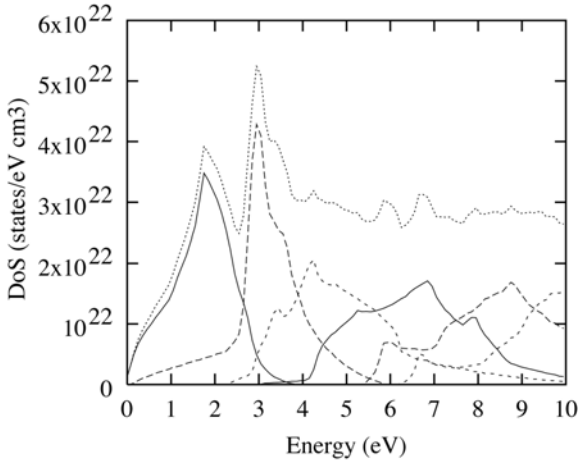


Figure 20-7: Density of states for the first 6 conduction bands of silicon.

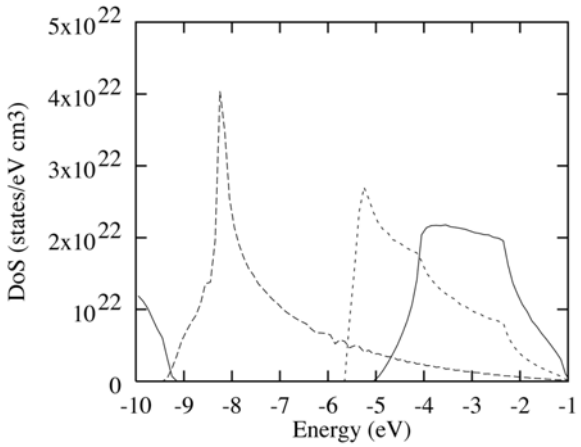


Figure 20-8: Density of states for the first 4 valence bands of silicon.

The **BANDS** statement controls the representation of the band structure:

```
BANDS N=2 NK=40 FILE="eder40_2.in" INTERP=0
```

Here, N is the total number of bands, NK is the number of points on Γ–*X*, and file is the ASCII file containing the band structure representation. These fields have two values each, one for the conduction bands and the second for the valence bands. Due to their different shapes, two independent representations are allowed for the conduction and valence bands. FILE is the file where the energies and velocities (as functions of the wave vector and band index) are stored (see Section D.2.7 "Band Structure Input Files"). INTERP is the interpolation algorithm (0 for tricubic, 1 for less-accurate (but faster) trilinear, 2 for more-accurate (but slower) trilinear). When using INTERP=0 on the **BANDS** statement and TYPE=1 on the **MAT** statement (for the strain model), the energies and velocities are corrected using the full quadratic correction factors. When using INTERP=1 or INTERP=2 on the **BANDS** statement and TYPE=1 on the **MAT** statement (for the strain model), the energies and velocities are approximately corrected using fitting parameters that depend on the mole fraction (or effective strain).

Band Degeneracies

To most accurately model charge transport in more than one band, you should model the behavior of electrons at degeneracy points. Two types of degeneracy points may occur: symmetry-induced degeneracies and accidental degeneracies. At symmetry-induced degeneracies, Bragg reflection occurs. At accidental degeneracies, the carrier's group velocity should be continuous during a free flight. Early MC programs [45] and DAMOCLES [176] neglect symmetry-induced degeneracies. MC Device neglects accidental degeneracies. This can be an important effect when transport occurs in more than one band since, for example, an electron can switch from the second to the third band around 3 eV, with a rapid increase in energy.

In research on this topic [231], the problem of ballistic interband transitions has been approached by assuming that the free flight of electrons can be described by a perturbation of the electron potential, leading to transitions between eigenstates of the total lattice potential. When an electron moves in ***k***-space, its band index varies according to the overlap integral between nearby states, which describes the matrix element for smooth perturbations.

The overlap integrals have been computed between the midpoints of all neighboring domains in a 40-point cubic discretization of the irreducible wedge. Since during a 2-fs time step an electron covers only 10^6 cm^{-1} of ***k***-space, the mesh step equal to about 2.9×10^6 cm^{-1} is large enough to ensure that transitions only occur between neighboring domains. For each cubic domain, the band-index transition with the largest overlap integral is stored in a look-up table. When an electron crosses the interface between domains the band-index changes accordingly. There is a discretization problem with this method that could lead to small violations of energy conservation. That is, if bands cross far from the boundary of the domain (Figure 20-9), there is an abrupt jump of energy when the particle crosses the boundary.

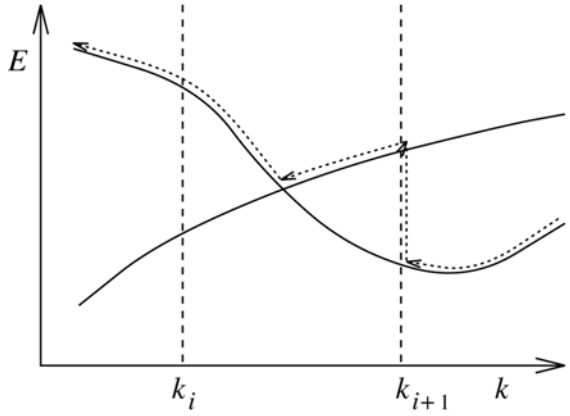


Figure 20-9: Lack of energy conservation during ballistic band switching.

Although the correct trajectory in ***k***-space is recovered after the crossing, energy is not conserved because the finite difference in energy cannot be accounted for by the electric field. An approximate solution to this problem can be obtained by placing the particle right at the center of the cube to minimize the energy step. To obtain a better solution to this problem, re-order the bands globally to achieve continuity of the Bloch functions. A separate program has been written to search the irreducible wedge for an optimal ordering of the bands [231]. This functionality, however, has not been integrated into MC Device.

This topic has been presented to make you aware of the issue of band degeneracies, to explain what research has been done on this topic, and to clarify that MC device modeling programs, including MC Device, are neglecting some aspects of band degeneracies. Particularly, MC Device neglects accidental degeneracies.

20.7.3 Electron Dispersion and Equations of Motion

MC Device can use either a tricubic or a trilinear interpolation of the electron dispersion in *k*-space. The tricubic interpolation yields a very good accuracy since it uses the “real” partial derivatives at mesh points to obtain continuity of the group velocity across cubic domains. On the other hand, the tricubic-interpolation subroutine takes a longer CPU time than the simpler trilinear formula. Such a high accuracy may not be necessary if a consistent method is used to integrate the equations of motion in real space.

Let us consider a sampling of the energy dispersion on a discrete set of points $\{(k_i,E_i)\}$. Any two interpolating functions $E'(k)$ and $E'(k)$. A particular case can be the true electron dispersion $E(k)$ (see Figure 20-10).

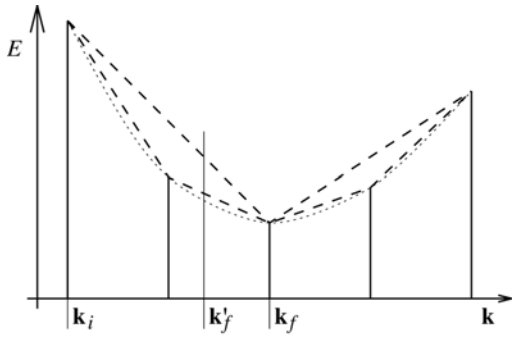


Figure 20-10: Different linear interpolations for the electron energy dispersion shown as a dotted line.

Let us assume that during a time T a given particle moves between two sampling points k_i and k_f, for a constant electric field F. The trajectory in k-space is a straight line. Therefore, we know exactly the final point in k-space and the final energy $E(k_f)$. We now assume that the basic equations of motion are consistent with the chosen interpolating dispersion $\tilde{E}$. This means that for any $\tilde{E}$ so that $\tilde{E}(k_i) = E_i$ and $\tilde{E}(k_f) = E_f$, the difference in real coordinate x_j is

$$\Delta x_j = \frac{1}{\hbar}\int_0^{\Delta t} \frac{\partial}{\partial k_j}\tilde{E}(k(t))dt \qquad 20\text{-}49$$

$$= \frac{1}{eF_j}\int_{(k_i)_j}^{(k_f)_j} \frac{\partial}{\partial k_j}\tilde{E}(k(k_j))dk_j \qquad 20\text{-}50$$

where we have used the linearity of the transformation between t and the components of k to express everything as an unambiguous function of k_j. Now we take our coordinates so that there is only the component F_j of the field is different from zero. In this case, the partial derivative with respect to k_j gives the total differential of energy, so that we can integrate and obtain

$$\Delta x_j = \frac{\tilde{E}_f - \tilde{E}_i}{eF_j} = \frac{E_f - E_i}{eF_j} \qquad 20\text{-}51$$

Now, nothing changes replacing $\tilde{E}$ with E', or E', or the true $E(k)$. Therefore, as long as the equations of motion are consistently solved, there will be conservation of energy and the final point in real space will remain. Only the interpolating trajectory in real and reciprocal space change, which should be irrelevant if you satisfy some basic accuracy requirements. If the starting or ending or both points of the trajectory are not sampling points for the electron dispersion, there will be an error, which is related to the accuracy of the interpolation scheme. The previous treatment, however, is valid for any (non-infinitesimal) trajectory. Therefore, the energy-conservation error is limited to the local error on energy. It does not grow as the particle moves. Instead, the error goes back to zero whenever the particle touches a sampling point. The maximum violation of energy conservation is in practice when the maximum value of the error in the interpolation scheme.

The real difference between different schemes is now that more accurate interpolation is more difficult to integrate in real space. For example, another final point k_f' in Figure 20-10. The two interpolations give different final energies, which the more finely spaced is certainly closer to the true dispersion. But, integration of the simpler trajectory is trivial, whereas the finer interpolation requires breaking the trajectory into two linear segments. Nevertheless, both schemes are consistent. That is, the difference between initial and final energy is exactly the energy taken by the electric field. The finer scheme yields a lower energy because the average group velocity is higher. Therefore, the energy loss is larger. This scheme fails if we assume a constant group velocity (as it is likely to happen if we want a fast real-space trajectory algorithm), while the more coarse interpolation gives no error in this respect.

Of course, these arguments do not allow us to say anything about what happens to the components of velocity *transverse* to the field. The analysis guarantees conservation of energy or invariance of the trajectory in the direction of field, which is important from a physical standpoint. Although transverse motion is critical for real-space transfer (coupling between carrier energy and diffusion properties), you only need a basic requirement of group-velocity accuracy to ensure an accurate modeling of such phenomena.

Summarizing, we can conclude that:

- Any interpolation scheme is consistent and physically sound as long as a consistent real-space trajectory is calculated.
- A lower-order interpolation allowing exact integration in real space can be more accurate than a higher-order, hard-to-integrate method.
- A coarse-mesh scheme can be more accurate than one based on a fine mesh if an approximate method for integrating the equations of motion is used.

Interpolation Scheme Comparisons

Figure 20-11 compares the average error achieved with the tricubic and the trilinear schemes.

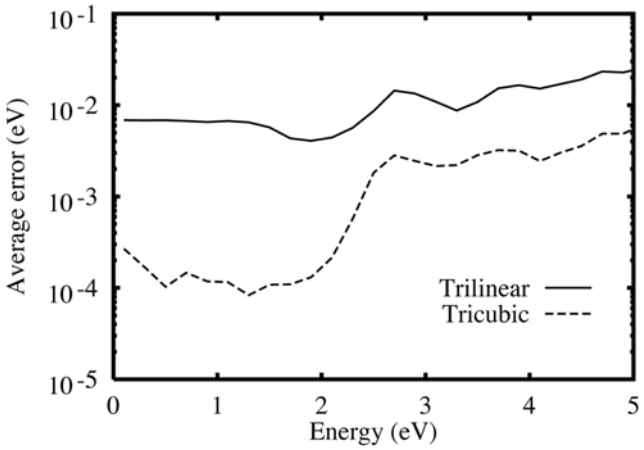


Figure 20-11: Tricubic vs. trilinear interpolation scheme, average error as a function of energy.

The averaging has been performed on the midpoints of all cubes making a 6281-point IW. The tricubic technique obviously performs much better than the simple trilinear scheme. The average error is often in the order of 0.1 meV vs. 5–10 meV for the trilinear interpolation. Figure 20-12 compares the maximum error instead of the average error.

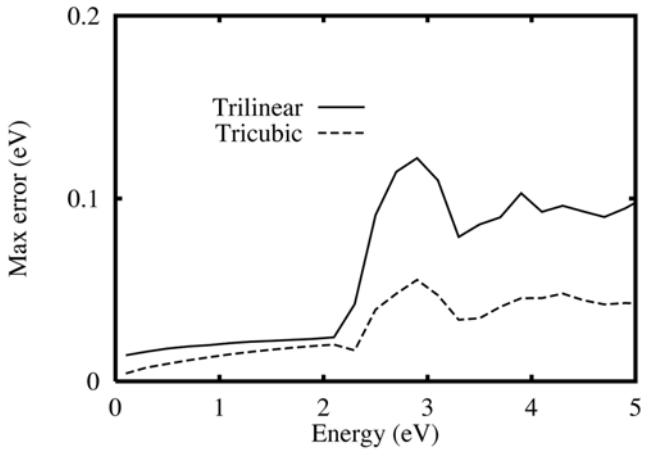


Figure 20-12: Tricubic vs. trilinear interpolation scheme, maximum error as a function of energy.

In the second case, the difference is not so marked. In fact, it may be argued, that a trilinear scheme with a finer mesh would give a smaller maximum error than the tricubic scheme. This fact is important since a Chebychev-norm constraint on the energy conservation is a good proof of physical robustness of the simulator. In any case, we see that the accuracy of both schemes drops dramatically at approximately 2.5 eV. This may be due to the presence of a sharp crossing between the second and third conduction band, which starts at about 2.3 eV. This crossing, however, occurs only on the Δ symmetry line. Therefore, it does not represent a serious limitation to either scheme.

To check the practical usefulness of the interpolation schemes, the same simulation has been performed in which carriers started at zero energy were accelerated by a field of 100 kV/cm. Acoustic inelastic scattering has been switched off, the temperature of intervalley phonons reduced to only 1 Kelvin, and collision broadening reduced by a factor of 20, so as to obtain a quasi-monoenergetic beam particles, but with a negligible probability of ballistic transport. Figure 20-13 shows the histogram of electrons at three points along this imaginary device (50, 100, and 150 nm). The deformation potential has been reduced accordingly to avoid an enormous scattering rate.

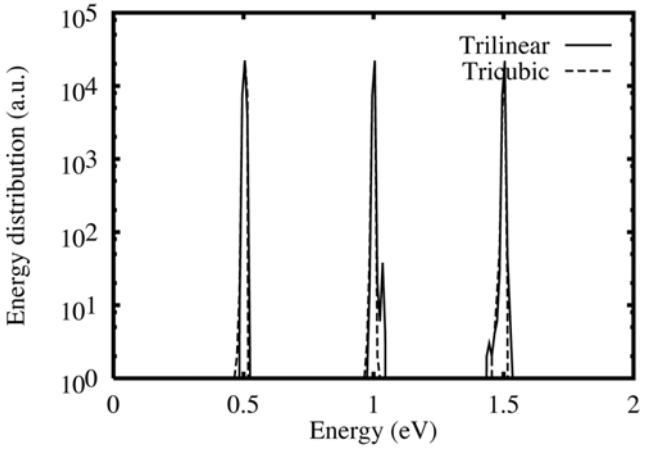


Figure 20-13: Monoenergetic beams of electrons simulated with both a trilinear and tricubic interpolation schemes.

The expected energies for the three beams are 0.5, 1.0, and 1.5 eV. It is evident that energy conservation is fully satisfied in both cases. Occasionally, only a few more electrons in the trilinear case are scattered to energies slightly above and below the expected value with respect to the tricubic case. This spread, however, is comparable with the “natural” spread due to collision broadening. In a real simulation, the broadening will be 20 times stronger and this effect will be irrelevant.

There are two problems that can occur with a simpler interpolation scheme. One, improper modeling of the transverse motion with respect to the electric field. Two, an error on the quantities which depend strictly on energy, notably the scattering rates. The first error is important for hot electrons when real-space transfer is a relevant phenomenon. The second error affects mostly regions of *k*-space where the scattering rate is a strong function of energy (i.e., the bottom of the conduction band). Figure 20-14 shows the various components of the room-temperature scattering rate at low energies (less than 100 meV).

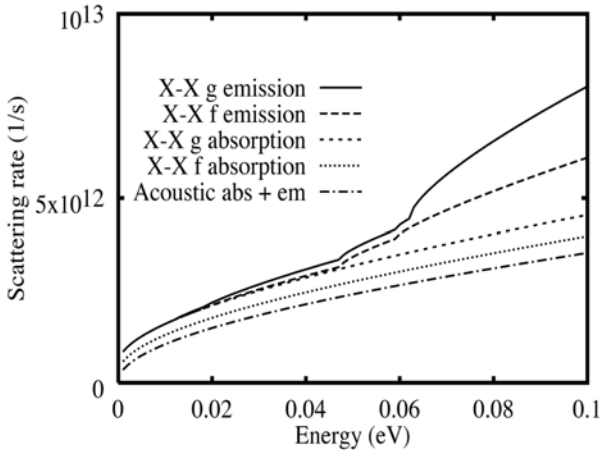


Figure 20-14: Low-energy scattering rate at 300 K.

The phonon-emission thresholds are clearly visible. At these critical points, we expect some inaccuracies of the trilinear scheme. But even for a smoothly varying scattering rate, a strictly linear interpolation for the energy can lead to some error in the time-dependent scattering rate. For example, if an electron starts a ballistic flight from the bottom of the first conduction band, its momentum increases linearly. Since the scattering rate is roughly proportional to the square root of the energy, which then is proportional to the absolute value of the momentum, we will expect the scattering rate also to increase linearly in time. But if in our linear interpolation scheme the electron energy is proportional to the wave vector, we have a linear increase in energy with time. Therefore, the scattering rate appears to grow as the square root of time. This was to be expected because we overestimated the energy during the flight and the scattering rate.

There is another class of problems that can arise with a linear interpolation for the electron dispersion. Since the algorithm for the choice of the final state uses a table of states sorted according to their energy, an electron after a scattering event will always have the exact energy as computed by the empirical-pseudopotential approximation. Generally, a tricubic scheme will give a very small error on the energy. Therefore, it can be considered as consistent with the final-state algorithm. The trilinear scheme, however, can lead to unphysical effects, such as the *k*-vector returned by the final-state routine can correspond, in the linear interpolation, to a very different energy. For example, an electron that moves after a

scattering event under the effect of the electric field. After a first free flight, the interpolation routine is called to calculate the new energy to determine the next scattering event. In the linear scheme, though, the new *k*-vector can correspond to a very different energy from that assumed by the final-state algorithm. This can lead to an artificial gain or loss of energy because the final-state and the interpolation algorithms use two different energy dispersions. Therefore when using a trilinear interpolation, you want to use the same k-space discretization for the choice of the final state. That way, an electron will always be scattered to a node of the trilinear-interpolation mesh, where the interpolation error is zero.

In most MC Device modeling tools including DAMOCLES [231] and MC Device, an analytical approximation is used to increase the accuracy of the final-state selection at low energies, where the *k·p* nonparabolic expansion is valid. This unfortunately contrasts with the requirement of using the same grid for the selection of the final states and for interpolation of the dispersion relation. This problem can be overcome in two ways. One, by using a nonparabolic approximation for the *E*-to-*k* relation. Two, by using the discrete grid for the selection of final states.

Although the second way is simpler and more consistent, it reproduces the low-field transport parameters very poorly. If you adopt the second way, you cannot accurately model contact regions. Therefore, MC Device uses the first way. Namely, MC Device uses the non-parabolic approximation for the E-to-k relation.

The standard non-parabolic expression in the Herring-Vogt starred *k*-space is

$$E(1+\alpha E) = \frac{h^2(k^*)^2}{2m^*} = \beta \qquad 20\text{-}52$$

$$E = \frac{\sqrt{1+4\alpha\beta}-1}{2\alpha} \qquad 20\text{-}53$$

and can be approximated to second order in β as

$$E = \beta - \alpha\beta^2 = \beta(1-\alpha\beta) \qquad 20\text{-}54$$

This approximation gives an error of less than 0.3 meV for all energies below 0.1 eV with respect to the full *k·p* formula and simultaneously it is very fast to compute. All the parameters (effective masses, non-parabolicity factor) have been derived directly from the pseudopotential calculation to avoid inconsistencies. Particularly, the non-parabolicity factor of 0.3 eV^{-1} has been found to minimize the average error for all states below 0.1 eV instead of the more common value of 0.5 eV^{-1}. The average error is kept within 5 meV for all energies less than 0.1 eV and 1 meV for states below 0.05 eV. The minimum of the conduction band has been identified, for the form factors given in [65], exactly at 0.850031. This is particularly important because even a small asymmetry of the band causes large errors in the calculation of the effective masses. These have been determined as m_t=0.194755 and m_l=0.912548. Note that the perfect parabolic, symmetrical band is only valid for very small energies. For energies of the order of 1 meV, the asymmetry is such that an accurate estimate of the effective masses is impossible. In fact, the asymmetry effect has been found to be stronger than the non-parabolicity effect. The correction is applied to all particles below 0.1 eV after the trilinear interpolation has been performed. Execution profiles show that this method increases the total time spent on the interpolation by only 20%, still ensuring low-

temperature and low-field accuracy. For example, the diffusion coefficients (a stringent test for low-field mobility) are correctly reproduced both at 300 Kelvin and at 77 Kelvin. Since the trilinear interpolation is used only as a rough check to separate high-energy from low-energy particles, the accuracy of the low-field behavior is semi-dependent on the *k*-space mesh step. A small error, of the order of 10% on the diffusion coefficient, is obtained for a very coarse mesh with just 40 points on $\Gamma - X$ for both final states and interpolation.

20.7.4 Coulomb Interaction

Types of Coulomb Interaction

Modeling the intrinsically three-dimensional Coulomb interaction by a two-dimensional simulation requires some care to conserve the main physical features of the system. Screening must be accounted for first. In a non-degenerate situation, we can think of screening as an exponential decay of the effective interaction potential, according to the space-dependent screening factor:

$$\beta_s^{-2} = \frac{\varepsilon(k_B T_e / e)}{en} \qquad 20\text{-}55$$

where T_e is the effective temperature of the carriers and n is their concentration. This picture holds up to plasma frequencies. For higher frequencies, the ensemble of carriers will not follow the perturbing potential quickly enough and anti-screening or no screening will occur.

To check the validity of the screening model from a thermodynamical point of view, the plasma frequency must be compared with the power spectrum of the carrier velocity. The auto-correlation of velocity extends up to about the frequency $S/2\pi$, where S is the scattering rate. For thermal carriers, this frequency is of the order of 10^{12} Hz. The plasma frequency ranges from 2×10^{12} Hz for $n=10^{17}$ cm^{-3} to about 10^{14} Hz for $n=3\times10^{20}$ cm^{-3}. Within this range of concentrations, you can assume all interactions as screened. For lower concentrations, part of the power spectrum will remain unscreened. In these cases, however, the microscopic Coulomb interactions can usually be neglected when compared to the macroscopic electric field. Moreover, the screening length is so large (13 nm at $n=10^{17}$ cm^{-3}) that unscreened interaction will occur in any case over a significant distance.

The general idea is to separate Coulomb interaction into two classes:

- **Short-range Coulomb interaction:** This occurs within a screening length. Since the wave packets representing the electrons can be assumed as overlapping, a simple space-homogeneous theory based on Bloch functions can be used to estimate the scattering rate for a given electron concentration. Given the average electron density, you can compute the full three-dimensional scattering probability per unit time. Screening of the Coulomb perturbation potential is included in the calculation if necessary. Electron-electron scattering is applied to carriers within a screening length in the two-dimensional simulation. The effect of short-range interaction is an exchange of energy between carriers. The position of the particles is not changed. Therefore, the potential energy remains constant and ensemble modes are not affected.
- **Long-range Coulomb interaction:** We can assume that beyond a few screening lengths, no interaction between single carriers occurs. Carrier-carrier is ineffective on these length scales because screening attenuates exponentially the mutual force. But, the perturbation has a large effect on the entire ensemble of carriers. In other words, beyond a screening length the individual carrier no longer sees other isolated carriers but rather a plasma. The

long-range interaction is thus basically an exchange of energy between a single carrier and the particle-field ensemble system.

You can model the cutoff between short-range and long-range interactions by choosing a Poisson-mesh spacing equal to the local screening length [177]. Assuming that the short-range interactions are correctly represented by a two-particle scattering rate, there remains the problem of representing the 3D electron-plasma interactions in a 2D model. The most physical approach that can be followed is the conservation of collisionality in the system.

Discretization and Collisionality

The long-range interaction can be pictured as exchange of energy between the single particles and the plasma modes. The latter are correctly modeled by a fine representation of the continuous plasma. Particularly, the mesh spacing must be fine enough to represent plasma modes that suffer Landau damping. The required maximum spacing is

$$h_{Landau} = \frac{v_{th} T_p}{2} \qquad 20\text{-}56$$

where v_{th} is the thermal energy and T_p is the plasma period. For example, for $v_{th}=2\times10^7$ cm/s and $T_p=10^{-13}$ s, we have $h_{\mathrm{Landau}}=1$ nm. But, the thermal agitation of the carriers depends on the granularity of the simulation (i.e., each computational particle represents a physical energy E_{3D} equal to its "computational energy" E_{2D}) times the number of carriers per unit length s that it represents. No perfect mapping is possible between a completely 2D model and a completely 3D physical system as the density of states for collective modes changes also with dimensionality. As a basic constraint, we can try to preserve the collisionality of the system. For systems of low collisionality, the kinetic energy of carriers dominates over the potential energy, and interaction with plasma modes does not change it very much. For systems at thermal equilibrium, the collisionality is given by the number of carriers within a Debye sphere. This quantity is roughly conserved in the transition between 2D and 3D if

$$s = \beta_s \qquad 20\text{-}57$$

This condition ensures that if Coulomb interaction is negligible in the physical system, it will also be in the 2D model and vice versa. Coupling between single carriers and collective modes will thus be qualitatively modeled. This criterion also minimizes the difference in short-range Coulomb interaction with respect to the 3D case [177]. It must be observed that a higher weight implies a higher physical kinetic energy, so that large particles will heat the plasma sea more than small particles. If all particles have the same size, the system will reach thermal equilibrium. If particles of different weights are plasma-coupled, there will be a transfer of energy from the large particles to the small particles.

Collisionality in Silicon Devices

The critical carrier concentration above, which an uniform sample of silicon is collisional, can be obtained as:

$$n\frac{4}{3}\pi\beta_s^{-3} = \frac{4}{3}\pi\left(\frac{\varepsilon k_L/e}{e}\right)^{3/2}\frac{1}{\sqrt{n}} = 1 \qquad 20\text{-}58$$

$$n = \frac{16\pi^2}{9}\left(\frac{(\varepsilon k_L)/e}{e}\right)^3 = 8\text{e}16\ \text{cm}^{-3} \qquad 20\text{-}59$$

You see that basically all electron concentrations with some significant conductivity (such as contact regions) give rise to collisional systems. Several remarks can be made on this result.

First, the collisionality criterion is correct for systems at thermal equilibrium. Hot carriers, of course, will be much less sensitive to plasma interactions. Some corrections, however, can be expected in the position of the high-energy tails, especially when the important effects are very sensitive with respect to energy (e.g., impact ionization, oxide injection). Therefore, the collisionality rule retains validity even for very high fields. In other terms, the comparison between kinetic and potential energy becomes a comparison between the lattice vibration energy and potential energy (i.e., marks the separation between dominant electron-phonon or electron-plasma (plasmon) interaction).

Talking about plasmons, for high carrier concentrations the plasmon energy becomes significant and quantization effects cannot be neglected. Since all plasma modes have the same frequency, this problem cannot be solved by removing some modes and Poisson-modeling the others.

For collisional systems there is less than one carrier within a screening sphere, which means that screening is statistical in nature and cannot be represented as a continuous distribution of carriers “filling up” the perturbing potential.

Another issue is the strict locality of the collisionality rule. Since the screening length is a function of position, no unique scaling factor will preserve collisionality for the entire system, and the transverse size of the device must be imagined as “shrinking” and “widening.” More importantly, it will be impossible to model correctly TRUE long-range effects between regions with different doping. For example, the plasma modes in the highly-collisional contacts might couple to the hot electrons in the channel more effectively than the weak, low-frequency channel modes. In this case, the same statistical weight should be used for channel and drain electrons. Finally, there remains an open question whether the transverse plasma modes can be qualitatively different from their projection on the simulation plane.

The Collisionality Criterion

We start from the Bogoliubov, Born, Green, Kirkwood, and Yvon (BBGKY) hierarchy of transport equations, which relates each *n*-particle distribution function f_n to itself and to f_{n+1}. The first equation is

$$\frac{\partial f}{\partial t} + (\boldsymbol{v}_g \cdot \nabla_r f) + \frac{1}{\hbar}\int d\boldsymbol{x}' d\boldsymbol{k}' F_{12} \nabla_k f_2 = \left(\frac{\partial f}{\partial t}\right)_{scat} \quad \text{20-60}$$

The collision term can be written as follows:

$$f_2(\boldsymbol{x}, \boldsymbol{x}') = f_1(\boldsymbol{x}) f_1(\boldsymbol{x}') + g(\boldsymbol{x}, \boldsymbol{x}') \quad \text{20-61}$$

Of course, the problem is minimized if short-range correlations can be neglected. In other words, if *g* is small with respect to the average-interaction term and to the collision term. The low-collisionality criterion can be considered in the following ways:

- **Absence of short-range correlation**: This is obtained if short-range interaction is smoothed out by solving Poisson's equation on a very coarse mesh. Particles do jerk around but plasma modes are unaffected. Of course, if g=0, particles "see through" each other.
- **Competition between phonon bath and e-e coupling**: Controls all correlations if scattering term is very strong because it will dominate both kinetic energy (by absorption/emission) and potential energy (by momentum relaxation). This happens when the thermal agitation energy is much larger than the interaction potential energy.
- **Kinetic vs. Potential Energy**: Same as above. Leads to the classical Debye-sphere criterion.
- **Energy of plasma modes**: Kinetic energy scales with the mass. Equipartition assigns the same amount of energy to all degrees of freedom. If particles are small enough, plasma modes will receive (and store) very little energy, and plasma-mediated exchange will be small.

General Scattering Statements

There are two MC Device statements that apply to all scattering events. **FINAL** sets parameters dealing with computation of final states. **SEREGION** defines the boundary for scattering estimation.

```
FINAL NK=40 FORMAT=0 N=100 FILE="name" CUTOFF=0.0
```

Final states are tabulated along with band structure in the `$MOCA/lib` directory. The default final states file is `eord100_*.in`. The * represents the number of conduction bands chosen to be used in **BANDS** (4 by default). See Section D.2.7 "Band Structure Input Files" for a description of these files. `NK` sets the number of points on the Gamma-X line. The default is 40 and the pre-calculated tables use 40 points. To use another value, you need to calculate new bandstructure data. `FORMAT` selects ASCII or binary (0 or 1) as the format of the final states file. The default is 0 for ASCII. `n` sets the number of final states points. The default is 100. Again, the tabulated data is based on the default values and setting a new value will require recalculation of bandstructure data. `CUTOFF` is to set the cutoff used for Lorentzian broadening, the default value is 0.0. To turn scattering estimation `ON`, insert the statement **SEREGION** in your input file.

```
SEREGION BOUNDP=(x1,x2,y1,y2)
```

The **SEREGION** statement defines the region over which scattering estimators are turned on. Generally, you can set the boundaries as equal to the boundaries of the entire device using the BOUNDP parameter (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

20.7.5 Phonon Scattering

Phonon Modes

The scattering rates used in MC Device are based on the energy dependent models first proposed by Jacoboni and Reggiani [143]. Both inelastic acoustic phonon modes and optical phonon modes are included to produce physical drift velocity versus field results. A wave vector dependent scattering was not implemented because the empirical models produced more physical results.

The derivation for the phonon scattering rates below is taken from [143]. The deformation potential method follows as

$$H' = E\frac{dy}{dr} \quad \text{20-62}$$

where dy/dr is the deformation of the lattice due to phonons and E is the tensor that describes the shift in the electron band per unit deformation.

Let us define phonon absorption and emission vectors respectively.

$$q = k' - k + G \quad \text{20-63}$$

$$q = k - k' + G \quad \text{20-64}$$

This fact states the conservation of crystal momentum from the point of view of the crystal subsystem.

For phonon creation (a_q) and annihilation (a_{-q}), the phonon densities for absorption and emission respectively are:

$$\left|\langle k'|a_q|k\rangle\right| = N_q \quad \text{20-65}$$

$$\left|\langle k'|a_{-q}|k\rangle\right| = N_q + 1 \quad \text{20-66}$$

where N_q is the phonon number.

Finally, Fermi's Golden Rule can be applied to obtain final transition probabilities that can be applied to all scattering mechanisms.

$$P(k,k') = \frac{\pi}{\rho V \omega_q}\begin{pmatrix} N_q \\ N_q + 1 \end{pmatrix} G\left|\Im_{ij} q_j \xi_i\right|^2 \delta[\varepsilon(k') - \varepsilon(k) \mp \hbar\omega_q] \quad \text{20-67}$$

Acoustic Phonons

For a non-degenerate band at the center of the Brillouin zone, the deformation potential, $\Im$, is diagonal and can be treated as a scalar. This simple situation, energy, and momentum conservation imply

$$q = \mp 2\left(k\cos\theta - \frac{m^* u_l}{\hbar}\right) \quad \text{20-68}$$

where u_l is the longitudinal velocity of sound, m^* is the effective of an electron, θ is the angle between k and q, and the + and - are for emission and absorption respectively. The second term above is the correction for the phonon involved in the scattering event. Therefore, for an elastic event, the equation will just be

$$q = \mp 2k\cos\theta \quad \text{20-69}$$

One important observation to make when tying this into a simulation is that when collisions are assumed elastic, the energy distribution is also assumed to be Maxwellian. Generally, this assumption is FALSE and cannot be used in Monte Carlo simulation. But, the only case when this assumption does not hold is for low fields and temperature. At high fields and high temperature, acoustic phonon scattering can be assumed to be elastic and is the same as optical phonons.

For realistic band structures, there will be separate probabilities for longitudinal and transverse modes. But the anisotropy is small and can be neglected by averaging the velocity of sound over the two directions as

$$u = \frac{1}{3}(2u_t + u_l) \quad \text{20-70}$$

The form of the final scattering probabilities are the same for inelastic acoustic phonon scattering. The main addition is that the δ function has phonon energy in it.

Keeping the same isotropic approximation for deformation potential coupling and using the Herring-Voigt transformation on k and q vectors, the magnitude of q becomes

$$q = q^*\left(\frac{m_l}{m_0}(\cos\theta^*)^2 + \frac{m_t}{m_0}(\sin\theta^*)^2\right) \quad \text{20-71}$$

$$q\left(\frac{m_l}{m_0}\right)^{1/2} \approx \left(\frac{m_l}{m_0}\right)^{1/2}, \quad \text{20-72}$$

where θ^* is the angle between $\boldsymbol{k}^*$ and the principal axis of the valley. This approximation is sufficient for Silicon.

Optical Phonons

For optical modes, the perturbation Hamiltonian to the lowest value of approximation is assumed to be proportional to the atomic displacement cite [143]. The continuous medium approximation from long-wavelength acoustic phonons would be inconsistent for optical modes. The scattering probability can be written by substituting an optical coupling constant, D, for potential. The energy associated with intravalley transitions can be assumed to be constant and given by $\hbar\omega_{\mathrm{op}} = k_B\theta_{op}$ where θ_{op} is the temperature of optical phonons. The resulting probability becomes

$$P(k,k') = \frac{\pi(D_tK)^2}{\rho V\omega_{\mathrm{op}}}\begin{pmatrix} N_{\mathrm{op}} \\ N_{\mathrm{op}}+1 \end{pmatrix}\delta[\varepsilon(k')-\varepsilon(k)\mp\hbar\omega_{\mathrm{op}}] \qquad 20\text{-}73$$

where N_{op} is replaced by N_{op} + 1 for emission. Note there is no angular dependence and therefore scattering is isotropic.

For the final state, it is random with constant probability over the energy-conserving sphere of k^*.

Intervalley Scattering

Electron transitions between states in two different equivalent valleys can be induced by both acoustic and optical phonons.

The phonon wave vector q involved in the transition remains close to the distance between initial and final minima, even for high electron energies. So the change in momentum and frequencies are approximately constant. Therefore, this scattering mechanism is treated exactly like intravalley scattering by optical phonons.

The coupling constant, $(D_tK)^2$, depends on the types of valleys involved in the scattering event and the branch of phonons involved.

In the case that electron energies are high and the particles can move up to bands of higher energy, the appropriate Δk in the first Brillouin zone must be considered along with the change in kinetic energy of the electron.

The integrated scattering probability for intervalley scattering is then

$$P_i(\varepsilon) = \frac{(D_tK)^2 m^{3/2} Z_f}{\sqrt{2}\pi\rho h^3\omega_i}\begin{pmatrix} N_i \\ N_i+1 \end{pmatrix}(\varepsilon\mp\hbar\omega_i-\Delta\varepsilon_{\mathrm{fi}})^{1/2} \qquad 20\text{-}74$$

where Z_f is the number of total number of final equivalent valleys for the scattering in consideration, N_i is the phonon number, and $\Delta\varepsilon_{fi}$ is the energy difference between the bottoms of the initial and final valleys.

Hole Intraband Scattering

There are both acoustic and optical types of this scattering for holes as there is for electrons. Due to the complexity of the valence band structure, use three deformation potential parameters. The characteristic wave function of holes also introduces an overlap factor.

To overcome analytical difficulties without losing the physics, use one coupling constant instead of the three deformation parameters. For a warped band, the resulting scattering probability is similarly to the electron acoustic scattering case.

$$P(k,k') = \frac{\pi q E^2}{\rho V u}\begin{pmatrix} N_q \\ N_q + 1 \end{pmatrix}\frac{1}{4}(1+3(\cos\theta)^2)\delta[\varepsilon(k')-\varepsilon(k)\mp\hbar q u] \quad \text{20-75}$$

Here, E is the coupling constant, and θ is the angle between k and k'. A numerical method is used to solve this integral in MC Device.

For optical hole intraband scattering, the rate can be described with one deformation potential parameter and has been found to be isotropic [143].

The integrated probability is then

$$P(k,k') = \frac{3\pi {d_0}^2}{2\rho V \omega_{op} a_0^2}\begin{pmatrix} N_q \\ N_q + 1 \end{pmatrix}\delta[\varepsilon(k')-\varepsilon(k)\mp\hbar\omega_{op}] \quad \text{20-76}$$

where d_0 is the deformation potential parameter. The resulting energy dependence of the final equation is analogous to that for optical scattering of electrons.

Hole Interband Scattering

When a multi-band model is considered, scattering rates should be used to account for the scattering (transitions) between the bands. In MC Device, only two bands (the heavy and light hole bands) of the three-bands hole scattering model are used. Since the calculations for the heavy-to-light and light-to-heavy interband scattering rates are somewhat complex, these rates are approximated by setting heavy-to-light rate equal to the light-to-light acoustic rate and by setting the light-to-heavy rate equal to the heavy-to-heavy acoustic rate. The file `scat.out` contains scattering rates for all possible processes. Any rates not calculated in MC Device (such as those involving the split-off band) are set to 0.0.

Density-of-States Scaling

The calculation of density of states is a basic ingredient of any bull-band Monte Carlo transport model to compute scattering rates and probabilities of final states. The density of states including both spin directions is

$$N(E) = \frac{1}{V_R}\left(\sum_{n,k' \in BZ}\delta[\varepsilon(k')-E]\right) \quad \text{20-77}$$

where n is the band index. The sum is over all cubes/tetrahedra in all bands. It is assumed that the energy $\varepsilon(k)$ is known at the corners of whatever shape is used to discretize the Brillouin zone (BZ).

Final State Selection

In MC Device, final state selection is based on the energy as opposed to being based on the momentum (or wave vector). To accomplish this, a relation for $k(E)$ is obtained. Since the bandstructure is stored in a table that is ordered by the wave vectors (k values). A final state table can be obtained by reordering the bandstructure table by energies, E.

The uncertainty principle is satisfied in MC Device by broadening the energy using the imaginary part of the self-energy (i.e., $(\hbar/(2\tau))$) and not using a one-to-one relationship between k and E. For each final energy, there are a range of final energies (and a corresponding range of momenta) that may be selected. Therefore, there is uncertainty in the momentum of the electron after a scattering event in accordance with the uncertainty principle.

Using Scattering Parameters

At the beginning of the simulation, scattering rates are either calculated or are loaded from data files. In either case, the scattering rate table is written to `scat.out`. The manner of obtaining the scattering rates and the number and type of the mechanisms, depends on whether the strain model is on (`TYPE=1|TRUE|YES` on the `mmat` statement) and whether the `TYPE=ANALYTIC` or `TYPE=FULLBAND` is used on the `ESCAT` statement. Below is a description of its contents of the `scat.out` file. Table 20-2 is for electrons and holes when the strain model is off or when the strain model is on and `TYPE=ANALYTIC`. Table 20-2 is for electrons when the strain model is on and when `TYPE=FULLBAND` on the `escat` statement.

Table 20-2. Form of scat.out (strain model off or TYPE=ANALYTIC)

Column	Format when CARRIER=E	Format when CARRIER=H
1	Energy	Energy
2	Total	Total
3	Ac abs	h-h AC abs
4	Ac emi	h-h AC emi
5	XX f abs (Tp=TF1)	l-l AC abs
6	XX f abs (Tp=TF2)	l-l AC emi
7	XX f abs (Tp=TF3)	s-s AC abs
8	XX g abs (Tp=TG1)	s-s AC emi
9	XX g abs (Tp=TG2)	h-h OP abs
10	XX g abs (Tp=TF3)	h-h OP emi
11	XX f emi (Tp=TF1)	l-l OP abs
12	XX f emi (Tp=TF2)	l-l OP emi
13	XX f emi (Tp=TF3)	s-s OP abs
14	XX g emi (Tp=TG1)	s-s OP emi
15	XX g emi (Tp=TG2)	h-l Interband

Table 20-2. Form of scat.out (strain model off or TYPE=ANALYTIC)		
16	XX g emi (Tp=TG3)	l-h Interband
17	XL abs (Tp=TL1)	h-s Interband
18	XL abs (Tp=TL2)	s-h Interband
19	XL abs (Tp=TL3)	l-s Interband
20	XL abs (Tp=TL4)	s-l Interband
21	XL emi (Tp=TL1)	Ionized impurity
22	XL emi (Tp=TL2)	Impact ioniz.
23	XL emi (Tp=TL3)	
24	XL emi (Tp=TL4)	
25	Ionized impurity	
26	Impact ioniz.	

Table 20-3. Form of scat.out (strain model on and TYPE=FULLBAND)	
Column	**Format when carrier=e**
1	Energy
2	Total
3	Longitudinal AC abs
4	Longitudinal AC emi
5	Transverse AC abs
6	Transverse AC emi
7	OP abs
8	OP emi
9	Impact ioniz.

The statements relating to phonon scattering deal with both acoustic (AC) and optical (OP) types, X-L and X-X, and f and g types of scattering events. The abbreviations abs and emi refer to absorption and emission. The abbreviations h, l, and s refer to heavy, light, and split-off. Tp refers to the phonon temperature, which can be one of 3 values for f-type and g-type transitions and one of 4 values for X-L transitions.

Acoustic scattering parameters are set by the AC statement as

```
AC DEFO=0.0
```

where DEFO is the desired deformation potential in [eV]. The default value is 9.0 eV. Optical, or high energy, phonon scattering is characterized by the **PHON** statement as

```
PHON ETH=0.0 MODEL=0 FILE="name"
```

where ETH is the energy threshold, in [eV], to activate this type of scattering. MODEL chooses which scattering model to use.

Currently, the only model implemented in MC Device is model=0 (Tang model). FILE chooses the density of states file. The default is dos_c.in, which is located in the $MCDEVICEDIR directory.

The intervalley scattering statements XL, XXF, and XXG set phonon temperatures and deformation potentials. The default values have been determined experimentally for silicon. As shown above, 4 values are used for X-L transitions and 3 values are used for f and g-type X-X transitions.

```
XL  TEMP=(672.0,634.0,480.0,197.0) DEFO=(2e8,2e8,2e8,2e8)
XXF TEMP=(220.0,550.0,685.0)       DEFO=(3e7,2e8,2e8)
XXG TEMP=(140.0,215.0,720.0)       DEFO=(5e7,8e7,11e8)
```

20.7.6 Impurity Scattering

For all materials, the lattice sites are many thousands of times heavier than electrons. Therefore, impurity scattering is essentially elastic in nature. Therefore, for the energy distribution to match reality, it must be accompanied by some dissipative mechanism.

The Brooks and Herring approach from [143] is described below.

For an ionized impurity, the scattering source is a screened Coulomb potential. The potential can be represented by a simple decaying exponential as

$$V(r) = \frac{Ze^2}{\kappa r}\exp(-\beta_s r) \qquad 20\text{-}78$$

where β_s^{-1} is the screening length, κ the dielectric constant, and Z the number of charge units in the impurity. The Debye formulation for nondegenerate statistics gives

$$\beta_s = \left(\frac{4\pi e^2 n_0}{\kappa k_B T}\right)^{1/2} \qquad 20\text{-}79$$

where n_0 is the free carrier density.

The Fourier transform of a screened Coulomb potential is given by

$$H'(q) = \frac{1}{(2\pi)^{3/2}}\int V(r)\exp(-iq\cdot r)dr = \frac{\sqrt{2}Ze^2}{\sqrt{\pi}\kappa}\frac{1}{\beta_s^2+q^2} \qquad 20\text{-}80$$

and leads to a scattering probability per unit time of

$$P(k,k') = \frac{2^5\pi^3 Z^2 n_{Ie}^4}{\hbar V\kappa^2} G\frac{1}{(\beta_s^2+q^2)^2}\delta[\varepsilon(k')-\varepsilon(k)] \qquad 20\text{-}81$$

where G is the overlap integral.

For electrons we can look at the case for spherical, parabolic bands or ellipsoidal, non-parabolic bands. For the first case, G is 1, and multiplying by the density of states $V/(2\pi)^3$ we get

$$P_{e,I}(\varepsilon) = \frac{2^{5/2}\pi n_I Z^2 e^4}{\kappa^2 \varepsilon_\beta^2 m^{1/2}} \frac{\varepsilon^{1/2}}{1 + 4\varepsilon/\varepsilon_\beta} \quad 20\text{-}82$$

where

$$\varepsilon_\beta = \frac{\hbar^2 \beta_2^2}{2m} \quad 20\text{-}83$$

For non-parabolic bands, The Herring-Voigt model is used. The results are given as

$$P_{e,I}(\varepsilon) = \frac{2^{5/2}\pi n_I Z^2 e^4}{\kappa^2 \varepsilon_\beta^{1/2} m_d^{1/2}} \frac{1 + 2\alpha\varepsilon}{1 + 4\frac{\gamma(\varepsilon)}{\varepsilon_\beta'}} \quad 20\text{-}84$$

where

$$\varepsilon_\beta' = \frac{\hbar^2 \beta_2^2}{2m_d} \quad 20\text{-}85$$

There are also two cases to account for impurity scattering of holes. There is the spherical parabolic band case and the warped band case. For the parabolic band, we can adapt the general scattering probability given above by accounting for the overlap factor. This gives

$$P(k,k') = \frac{2^5 \pi^3 Z^2 n_I e^4}{\hbar V \kappa^2} \frac{0.25(1 + 3\cos^2\theta)}{(\beta_s^2 + q^2)^2} \times \delta[\varepsilon(k') - \varepsilon(k)] \quad 20\text{-}86$$

and finally integrating over the solid angle we arrive at

$$P_{h,I} = \frac{2^{5/2}\pi n_I Z^2 e^4}{\kappa^2 \varepsilon_\beta^2 m^{1/2}} \frac{\varepsilon^{1/2}}{1 + 4\varepsilon/\varepsilon_\beta} F_{ov} \quad 20\text{-}87$$

where F_{ov} is the overlap factor. The overlap factor essentially works to reduce the efficiency of the impurity scattering mechanism and can be found in [143].

We will neglect the warped band case because it can be neglected as long as the warping is small. For treatment of the case, see [143].

Implementation of Impurity Scattering

In MOCA, the Ridley impurity scattering model is implemented. The scattering rate is calculated as follows:

$$\frac{1}{\tau_{imp}} = \frac{\sqrt{\frac{2\gamma}{m^*}}\frac{1}{A}}{1+2\alpha E} \times 1 - \exp\left[-\frac{ae^2 m^* k_B T(1+2\alpha E)}{8\pi\varepsilon\hbar^2\gamma\left(1+\frac{Ne^2\hbar^2}{8\varepsilon m^* k_B T\gamma}\right)}\right] \quad \text{20-88}$$

Final state selection is for the Ridley model are calculated including the factor 2π from Chandrashekar.

$$F_1 = \frac{e^2 m_d k_B T}{8\pi^2\varepsilon\hbar^2 N}$$

$$F_2 = \frac{Ne^2\hbar^2}{8m_d\varepsilon k_B T} \quad \text{20-89}$$

$$F_3 = \frac{-2\pi^2 N^2}{3}$$

These factors are used in density of states scaling. See "Impact Ionization and other topics" on page 1011 for more information.

MOCA includes the option to modify Ridley parameters using the **RIDLEY** statement.

```
RIDLEY SCALE=1.0 TSTEP=n
```

Here, SCALE is a scaling factor. TSTEP is the simulation time step where to update the concentration.

Impact Ionization and other topics

The other impurity-related scattering statements are: **IMPACT**, **SSPARAM**, **SSREGION**, **TRACKSCAT**, **TRAJ**, and **ZSREGION**. These are described below.

The **IMPACT** statement is for impact ionization.

```
IMPACT MODEL=0|1|2|3 PARAM=(i,j,k) ETH=(E1,E2,E3) \
  WEIGHT=NO FINAL=YES sec=NO CSEC=NO \
  START=1000 RANDSEC=NO RESPOS=(min,max)
```

The models for impact ionization that are available are KELYDYSH (0), BUDE (1), KANE (2), and CARTIER (3). PARAM allows you to specify parameters for 3 terms in the rate. You can set three threshold energies with ETH corresponding to the three parameters in PARAM.

Here are the following parameters:

- 1 = eta
- 2 = e-field in x-direction
- 4 = carrier concentration
- 5 = carrier velocity

- 7 = energy
- 8 = impact ionization rate
- 9 = generation

WEIGHT either sets weighting off or doubles weighting. **FINAL** determines whether to set final energy of the initial particle to zero (NO) or 1/3 of its starting energy (YES). SEC toggles generation of secondary particles, and CSET does the same for complimentary secondary particles. START defines a transient period at the start of the simulation during which impact ionization is not to be calculated. RANDSEC chooses whether to randomly place secondary particles. RSPOS defines a range over which to place these particles (min and max from the location of the ionized impurity scattering event).

SSREGION and **SSPARAM** statements are for surface scattering.

```
SSPARAM SEXPL=i SEXPD=j SSFIELD=n FILE="name" SURPHON=(p,e)
```

SEXPL and SEXPD are factors used in calculating surface scattering. The default values are set for best fit with experimental mobility data.

SSFIELD is the global surface scattering field. You can choose the output file using FILE. Finally, SURFPHON sets the prefactor and power for surface scattering calculations.

The **TRACKSCAT** statement is for outputting scattering estimators to a file called trackscat.out.

```
TRACKSCAT BOUNDP=(x1,x2,y1,y2) DIR=(x,1)
```

As with all statements, BOUNDP defines the boundaries over which to track scattering (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility. DIR sets the direction and the dimensionality of the tracking.

The TRAJ statement tracks the trajectories of particles.

```
TRAJ MODE=0 ZEROSCAT=NO BOUNDP=(x1,x2,y1,y2)
```

By including this statement, MOCA outputs a file called traj.out. MODE toggles this feature on and off. A value of 0 turns the feature OFF. Setting it to be 1 tracks particle trajectories and sets them to be initially evenly distributed in the bound box. Setting the value to 2 sets the feature ON and writes all the detailed information. ZEROSCAT chooses whether to zero the scattering rates in the simulation. Finally, BOUNDP determines the region where to track particle trajectories (see "Regions" on page 966). A BOUND parameter, which uncharacteristically takes arguments of position in cm is provided for backward compatibility.

The **ZSREGION** statement enables you to turn scattering off within more or more rectangular regions.

```
ZSREGION BOUNDP=(x1,x2,y1,y2)
```

The BOUNDP parameter contains the boundaries of the region, as on the **REGION** statement (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

20.7.7 Tunneling

No clear knowledge exists to date on the transmission mechanism from silicon into silicon dioxide. In principle, the full quantum problem should be approached, solving the Schrödinger equation for the given arrangement of silicon and oxygen atoms. The large number of unknown parameters in this problem has so far made this approach unfeasible. We will briefly illustrate two possible approaches to an analysis of the phenomenon, which rely on simple relationships between the silicon and silicon dioxide band structures.

Conservation of Parallel Momentum

The first approach, widely used in the literature, relies on all the following assumptions:

- **Conservation of parallel momentum**: The in-plane (or parallel) component, $k^{||}$, of the wave vector remains unchanged in the transition from Si to SiO_2 [164]. This corresponds to assuming that the band structure remains consistent in the transition, so that the interface can only supply an out-of-plane (normal) momentum. This hypothesis is sensible if a 10–20 Angstrom epi-oxide layer exists and if no significant image-force lowering of the potential barrier occurs. Thus, the epi-oxide layer can be thought as an extension of the Si lattice with a thin, abrupt dipole causing a discontinuity in the band edge.

  Since the band structure of the epi-oxide is unknown, carrier tunneling through the epi-oxide must be analyzed by considering only the single, direct conduction band of amorphous SiO_2.

- **Conservation of energy**: We assume a sharp discontinuity of E_B in the potential energy between Si and SiO_2:

$$E_{ox}(0^+) = E_{Si}(0^-) - E_B \qquad 20\text{-}90$$

  In a sense, this hypothesis, combined with the previous one, implies a “disappearing” of all Si bands except the fourth conduction band, starting at 3.1 eV.

- **Isotropic dispersion in the oxide**: The dispersion relation in SiO_2 is assumed:

$$E_{ox} = \frac{h^2}{2m^*_{ox}}(k_x^2 + k_y^2 + k_z^2) \qquad 20\text{-}91$$

  Again, this may not be a good approximation for the strained crystalline-oxide layer that we are considering. Since the parallel momentum is assigned, we obtain the normal momentum in the oxide as *ktox1*:

$$k_{ox}^{\perp}(x) = \sqrt{\frac{2m_{ox}E_{ox}(x)}{h^2} - (k_{Si}^{||})^2} \qquad 20\text{-}92$$

  For simplicity, we assume a parabolic (non-Franz) dispersion with equal effective masses for both propagating and evanescent states. We note that for “true” tunneling states ($E_{ox}<0$) both terms inside the square root are negative. The parallel-wavevector conservation can force some extent of tunneling even if the states have energies above the barrier ($E_{ox}>0$).

- **WKB approximation**: The transmission coefficients should in principle take into account the discontinuity in the band structures (i.e., the overlap integrals between Tamm states in the silicon and the oxide). We do not have this information, however, we extend the consistent-lattice approach and we assume that no further reflection is due to the discontinuity in band structure. The error level is now so high that using the WKB approximation will do no harm to the results. Although for energies close to the top of an abrupt barrier, the WKB formula can lead to large errors. In any case, we use a modified version of the WKB formula that joins smoothly with a transmission coefficient of 1 for non-tunneling states:

$$T = \exp\left(2i\int_{x_1}^{x_2} k_{ox}^{\perp}(x)dx\right) \quad 20\text{-}93$$

  where x_1 and x_2 are the two classical inversion points where $k_{ox}^{\perp} = 0$.

- **Constant field in the oxide**: This amounts to evaluating Equations 20-91 with $E_{ox}(x) = E_{Si}(0) - E_B + eF_{ox}x$,.

This evaluation yields:

$$x_1 = 0 \quad 20\text{-}94$$

$$x_2 = \frac{1}{eF_{ox}}\left(E_B - E_{Si} + \frac{\hbar^2(k_{Si}^{\parallel})^2}{2m_{ox}^*}\right) \quad 20\text{-}95$$

$$k_{ox}^{\perp} = \sqrt{\frac{2m_{ox}^*}{\hbar^2}(E_{Si} - E_B + eF_{ox}x) - (k_{Si}^{\parallel})^2} \quad 20\text{-}96$$

The result is

$$T_1 = \exp\left[-\frac{4}{3}\sqrt{\frac{2m_{ox}^*}{\hbar^2}}\frac{1}{eF_{ox}}\left(E_B - E_{Si} - \frac{\hbar^2(k_{Si}^{\parallel})^2}{2m_{ox}^*}\right)^{3/2}\right] \quad 20\text{-}97$$

Band-Structure Approach

The second approach neglects totally momentum conservation in the transition from Si to SiO_2. But it assumes that the silicon band structure controls the amount of crystal momentum that can be used in the transition. We use the following assumptions:

- **Ballistic flight in silicon**: The entire difference in energy is assumed to be sustained within the silicon. From the initial state in *k*-space, a minimum-energy state is found in the same band that conserves parallel momentum. The energy E_{si}–E_{min}–E_B is the difference between the minimum kinetic energy that the energy can achieve by ballistic emission and the barrier energy. It is taken as the kinetic energy of the emitted electron. An undetermined amount of parallel momentum is considered to be supplied by the

heterojunction during the change of band structure to bring the electron to the Γ valley of the oxide.

- **Normal incidence in the oxide**: Since the silicon band structure has taken care of the kinetic aspects of the emission process, all remaining momentum is assumed to be available as normally-oriented kinetic energy in the oxide. Note that the energy not "used" for the ballistic emission is considered as "lost." In other words, known in the silicon but unknown, and irrelevant, in the oxide. Therefore, the normal wavevector of the electron in the oxide is taken as

$$k_{ox}^{\perp}(x) = \sqrt{\frac{2m_{ox}}{\hbar^2}(E_{Si} - E_{min} - E_B + eF_{ox}x)} \qquad 20\text{-}98$$

- **WKB approximation**: As before, the classical inversion points x1 and x2 are computed. The WKB formula (Equations 20-91) is used, where the dependence on $k_{Si}^{||}$ is neglected. The result is

$$T_2 = \exp\left[-\frac{4}{3}\sqrt{\frac{2m_{ox}}{\hbar^2}}\frac{1}{eFox}(E_B - E_{Si} + E_{min})^{3/2}\right] \qquad 20\text{-}99$$

Comparison and Discussion

Figures 20-15 and 20-16 compare the results from the two approaches with the usual WKB approximation for parabolic bands.

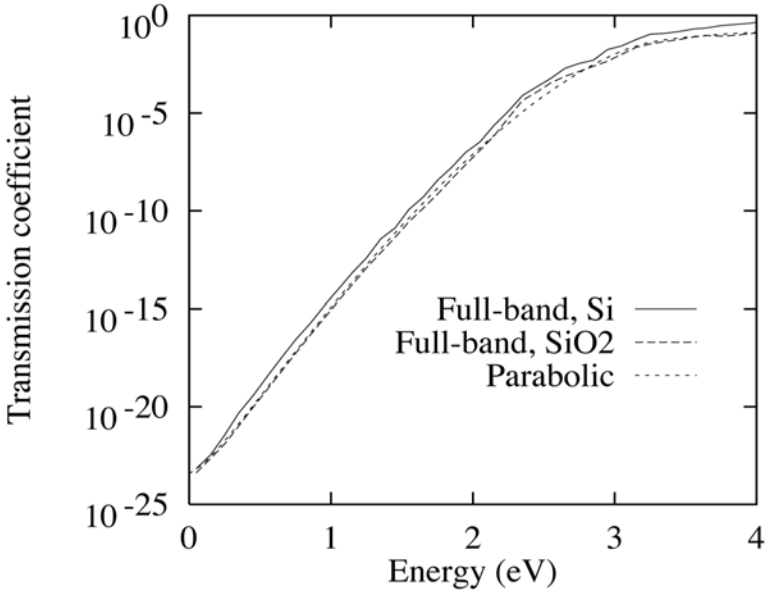


Figure 20-15: Comparison of formulae #1 and #2 with the parabolic WKB approximation.

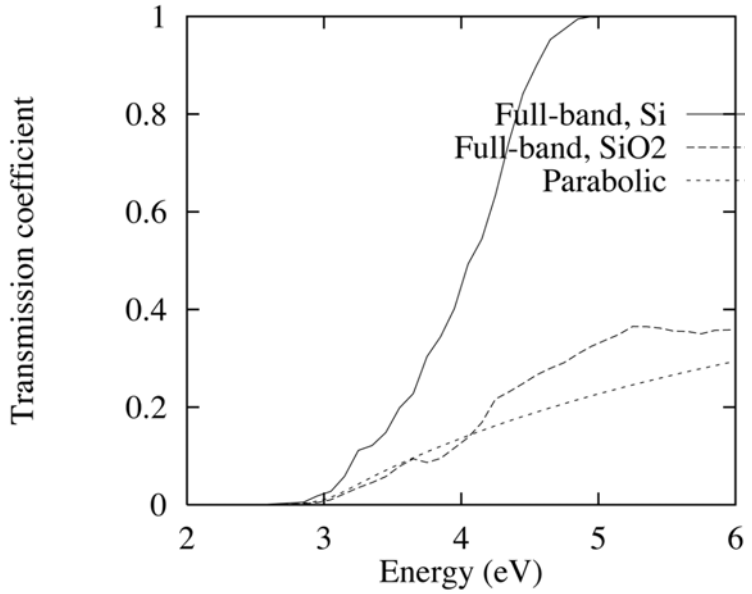


Figure 20-16: Comparison of formulae #1 and #2 with the parabolic WKB approximation for energies around the thermionic-emission threshold.

The transmission probabilities are plotted as a function of electron energy, assuming isotropic distribution of the incident particles. It is clear that the energy dependence of Equations 20-92 is very similar to a pure isotropic, parabolic electron dispersion. This is because the band structure of silicon is effectively neglected, and enters the calculation only through the density of states as a function of angle in k-space. But, the transmission coefficient obtained neglecting conservation of parallel momentum (Equations 20-95) is much higher. For low energies, the difference is on the order of a factor of 3, which is consistent with the arrangement of the conduction valleys in k-space.

Using the TUNNEL Statement

Oxide injection is calculated by default. The **TUNNEL** statement is used to calculate direct gate oxide tunneling on top of the carriers injected into the oxide.

```
TUNNEL MODE=0|1|2
```

MODE sets which calculation algorithm to use. MODE=0 turn off the calculation. MODE=1 chooses a simple triangular barrier. MODE=2 chooses a more accurate numerical integration with Schottky image force lowering.

20.7.8 Size Quantization

Background

In certain devices, such as inversion MOSFETs, as their dimensions become smaller, size quantization of charge layers occurs. The high fields in the device pull charge very strongly to the Si–oxide interface. Therefore, the depth of the charge layer becomes on the order of a deBroglie wavelength. Because the potential in the silicon changes by much more than k_{BT} over a carrier wavelength, the allowed energy levels become discrete and we must account for this effect.

The major effect of quantization is to alter the location of charge. In the MOSFET, destructive interference of the wave functions occurs at the Si–oxide interface because of the abrupt change in potential. This leads to a repulsion of carriers from the interface, which occurs both in accumulation and inversion. You can set the location of the charge density maximum back from the interface by 1 to 2nm, unlike the classical case where the maximum occurs directly at the interface. One of the important consequences of this is that the oxide thickness is effectively increased.

A secondary effect of charge quantization is an increase in phonon scattering. With quantization comes a sort of confinement, therefore the uncertainty in position is reduced. By the uncertainty principle, this implies an increase in momentum uncertainty. The delocalization in momentum space allows a range of phonon momenta to scatter between momentum eigenstates. Because conservation of momentum is relaxed, the density of final states for phonon scattering is increased. The consequence of this phenomenon is decreased mobility at medium field strengths (at high field strengths, surface scattering dominates).

The Schrödinger Equation

The 1D effective mass Schrödinger equation is [355]:

$$-\frac{\hbar^2}{2}\frac{\partial}{\partial z}\left(\frac{1}{m_z^*}\frac{\partial\phi}{\partial z}\right) + V(z)\phi = E\phi \qquad 20\text{-}100$$

where $V(z)$ is the macroscopic potential, ψ the envelope wave function for the macroscopic potential, m_z^* is the effective mass in the direction of the quantization, and E is the eigenenergy. For silicon, this equation is solved twice to account for the two-fold degeneracy in longitudinal valleys and four-fold degeneracy in transverse valleys. The Schrödinger equation is solved for all eigenvalues by using a box-integration finite difference scheme [355].

Once the wave functions ψ_k and energies E_k are obtained, the particle concentration, n, is found by the following equations:

$$N_{\mathrm{dos}}^{2D} = \frac{m_{\mathrm{xy}}^* k_B T}{\pi\hbar^2} \qquad 20\text{-}101$$

$$F_0(x) = \int_0^\infty (1 + exp(\mathrm{x}))^{-1}\mathrm{dx} \qquad 20\text{-}102$$

$$n(z) = N_{\text{dos}}^{2D}\sum_{k} F_0\left(\frac{E_F - E_k}{k_B T}\right)\left|\psi_k(z)\right|^2 \quad \text{20-103}$$

where N^{2D} is the total density of states of the 2D gas that arises from applying a parabolic dispersion relation to the constant density of states in k-space. The density of states effective mass, m^*, is the mass in the plane perpendicular to the 1D solution. For example, in Si, the mass in longitudinal valleys is m_t^* and for transverse valleys is $\sqrt{m_l^* m_t^*}$. To find the density of occupied states in the conduction band, the density of states is multiplied by the Fermi factor and integrated over the entire band. In this procedure, F_0 is the Fermi integral of order zero. Finally, $n(z)$ is the quantum effect on the density.

Monte Carlo Schrödinger Formulation

The approach used in MC Device is to periodically solve the Schrödinger equation during a simulation along the direction perpendicular to transport using the self-consistent electrostatic potential as input. The quantum potential in this method is found using the exact energies and wave functions. The correction to the Monte Carlo transport potential is found by linking the quantum density to the potential through

$$V_{\text{qc}}(z) = -k_{\text{B}}T\log(n_q(z)) - V_p(z) + V_0 \quad \text{20-104}$$

Here, V_{qc} is the quantum correction. z is the direction normal to the interface. n_q is the quantum density from the Schrödinger equation or equivalently the converged Monte Carlo concentration. V_{p} is the potential from the solution of Poisson. V_0 is a reference potential determined by the fact that the correction should go to zero in a bulk-like region.

In Equation 20-104, the Schrödinger concentration profile goes in an arbitrary multiplicative constant. For the purpose of filling the energy levels, only the relative populations are significant. The particles simulated by Monte Carlo, once the potential correction has been added, move in a potential environment that shapes the space distribution as in the result of the Schrödinger equation. The non-equilibrium full band Monte Carlo model determine the magnitude of the particle distribution at different points along the transport path and momentum space trajectories.

MC Device chooses the Fermi level to fill the energy levels according to the Maxwell-Boltzmann distribution. The setting of the Fermi level, E_F, also fixes the potential V_0. Explicitly, this can be seen by

$$n_q(z) = exp^{-\left(\frac{V_p(z) + V_{qc}(z)}{\text{kT}}\right)} exp^{\frac{V_0}{\text{kT}}} \quad \text{20-105}$$

$$exp^{-\left(\frac{V_0(z) - V_p(z)}{\text{kT}}\right)} \rightarrow N_{\text{eff}} exp^{-\left(\frac{V_p(z) - E_F}{\text{kT}}\right)} \quad \text{20-106}$$

$$V_0 = kT\log(N_{eff}) + E_F \quad \text{20-107}$$

N_{eff} is the effective density of states and E_F is the specified Fermi level. You can see that $n_q \rightarrow n_c$ as $V_{qc} \rightarrow 0$.

Using the Maxwell-Boltzmann relation in the formulation for the correction should be questioned since degeneracy is not explicitly taken into account. Degeneracy can be important for filling the levels in highly inverted MOS structures. But, despite using non-degenerate statistics to relate voltage and concentration in the correction, note that sufficient accuracy is retained even in the highly degenerate regime for all practical cases tested. This can be explained by the similar shape of the first three sub-bands in the region where the quantum correction is important. The correction only begins to break down at non-physical oxide fields where the higher subbands become degenerate.

Using the Quantum Correction

Using the quantum correction proceeds in a similar fashion to using the Monte Carlo simulation. First, quantum parameters are defined using the **QUANTUM** statement. Then, simulation regions are defined using **QREGION** and **SCHRREGION**. Finally, you build a sub-mesh for the solution of the Schrödinger equation using **SCHRMESH**.

Default values for the quantum parameters are set in the default parameters file. These parameters have been optimized and you should not change them for simulation of practical devices. The defaults for the **QUANTUM** statement are

```
QUANTUM RESTART=NO OPTION=(NO,NO) DECAY=(100e-15,2000e-15) \
  MINCONC=(1e10,1e10) MAXQV=-1.0 MAXQF=-1.0 \
  STRENGTH=1.0 Q1D=(-1,-1,-1) FREEZE=PINF SIGMA=8e-8 \
  RADIUS=3 LADDERS=(1,0,0) ENERGY=NO
```

In the example, quantum-corrected device input file is given in Section D.3 “Examples”. The **QUANTUM** statement is not included because the defaults are adequate. If you want to restart from an old quantum correction solution, change RESTART to YES. DECAY is the decay in time averaging. MINCONC is for setting the minimum Monte Carlo concentration and the minimum quantum concentration. MAXQV sets the maximum allowed correction to the potential. MAXQF does the same for the field. STRENGTH is only a multiplicative factor. Q1D is for 1D averaging and is turned off if the value is -1. FREEZE is to specify a freezing iteration. An option not included in the default configuration is SMOOTH1D and is for 1D smoothing. SIGMA is the smoothing standard deviation, and RADIUS is the integration radius for smoothing. LADDERS is the number of ladders to use, and ENERGY is whether quantum potential should contribute to carrier energy. One last option that is also not included in the defaults file is NOBARR. This is for choosing whether to perform barrier fitting.

The first statement you need in order to use the quantum correction is QREGION.

```
QREGION MODE=SCHROEDINGER RAMP=(0,0) \
  BOUNDP=(x1,x2,y1,y2) ABOUNDP=(x1,x2,y1,y2) \
  FIT=(-1,-1) FBOUNDP=(x1,x2,y1,y2) \
  SHIFT=-1 SHIFTDIST=0 DIR=Y \
  TSTEP=4000 XSTEP=1 ZERO=0
```

MODE is to choose the type of correction to use, such as FEYNMAN and WIGNER. RAMP is to specify a transient period where to ramp up the correction added to the Monte Carlo potential. BOUNDP is the region where to calculate the correction (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility. ABOUNDP is a region where to calculate attenuation. ABOUND is provided for backward compatibility. FIT is to specify a fitting function to fit the correction potential to. By setting the field to -1, it turns off the fitting function. FBOUNDP is the region where to fit the correction. FBOUND is provided for backward compatibility. SHIFT and SHIFTDIST are for specifying a grid to shift for interpolation and the distance to shift the grid respectively. DIR specifies the direction for the 1D correction, for a horizontal MOS device. This direction will be y. TSTEP is to specify the simulation step interval where Schrödinger is to be solved. XSTEP is the space step where to perform the calculation. Finally, the ZERO parameter specifies a grid line (>=1) where the quantum correction should go to zero. Leaving the ZERO parameter set to 0 (or -1 for backward compatibility with MOCA) lets MC Device calculate this on its own positive integral value for the grid line at which the quantum correction will go to to zero.

The **SCHRREGION** statement works very similar to the REGION statement. Any material chosen in **SCHRREGION** must have an entry in schroedinger.material.in. A quantum region is specified as follows.

```
SCHRREGION N=n MAT=AIR|SI|SIO2|SI3N4 BOUNDP=(x1,x2)
```

N is an integer (where N>=1) and represents the quantum region number. **MAT** is the material type. BOUNDP is the minmum and maximum position along the quantization axis (the *y* axis). A BOUND parameter, which uncharacteristically also uses positions rather than mesh indicies, is provided for backward compatibility. The mesh for the quantum regions is defined using **SCHRMESH**. It is used in exactly as **XMESH** and **YMESH**. See the following.

```
SCHRMESH MODE=n LOC=x-pos RATIO=r
```

The definitions for **SCHRMESH** are the same as for the regular mesh statements. Because the correction is only calculated along a single direction, only a single grid needs to be defined.

To complete the quantum correction section of the input file, include the **EBREGION** statement for an energy barrier.

```
EBREGION BOUNDP=(x1,x2,y1,y2)
```

The BOUNDP parameter defines a boundary and is typically set to the boundaries of the whole device (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

The statement associated with the Schrödinger quantum correction is the calculation of transverse carrier temperature. You can calculate transverse temperature, T_t, with **TTEMP**.

```
TTEMP RESTART=NO SIGMA=0 SMOOTH=NO BOUNDP=(x1,x2,y1,y2)
```

The **TTEMP** statement allows you to define a region over which to calculate the transverse carrier temperature, T_t. The RESTART parameter, as with other statements, allows you to start using a previous solution file. SIGMA is set to 0 to run in normal mode or to 1 to run in best-fit mode. The SMOOTH parameter allows you to smooth the transverse temperature prior to writing it out. The BOUNDP parameter sets the boundary where to calculate the transverse temperature (see "Regions" on page 966). Again, a BOUND parameter is provided for backward compatibility.

20.7.9 Schottky Barrier Injection

Background

The Schottky barrier is a junction between a metal and semiconductor where the work function of the metal is greater than the electron affinity of the semiconductor. When in contact, these two materials then form a fixed energy barrier of constant height but varying width.

Although a MOSFET with Schottky source and drain regions looks similar to a bulk MOSFET, the behavior between the two is very different. Carrier flow from the source contact to the conduction channel takes place mainly by tunneling through the Schottky barrier at the silicon/silicide interface. The width of the barrier is spatially modulated by the gate voltage, and conduction occurs when the barrier is sufficiently thin [354]. When the gate voltage is small, the barrier between source and channel is sufficiently wide so that appreciable tunneling does not occur. A certain amount of leakage current due to thermionic emission from the metal may be present, originating from the tail of the quasi-equilibrium in the silicide. For higher gate voltages, the Schottky barrier just below the Si–oxide interface may become sufficiently thin to allow tunneling. A Schottky barrier is also present at the drain to channel interface and may impede the flow of carriers at low drain biases.

Transport in the channel remains semi-classical because scattering in silicon is strong enough to break quantum coherence. This justifies a coupled quantum injection – Monte Carlo particle transport approach.

The Airy Function Approach

The injection between contacts is handled by transmission probabilities, which are solutions of the Schrödinger equation along paths perpendicular to the silicon/silicide interface. The transmission probability is calculated using a standard Airy function approach based on the 1D Schrödinger equation. The Airy approach was chosen over WKB because of its accuracy with thermionic reflection and particles with energy near the top of the tunneling barrier.

For an equation of the form:

$$\frac{\partial^2 \psi}{\partial z^2} = z\psi \qquad \text{20-108}$$

the solution looks like

$$\psi(z) = a\mathrm{Ai}(z) + b\mathrm{Bi}(z) \qquad \text{20-109}$$

where Ai*(z)* and Bi*(z)* are the Airy functions given by

$$\mathrm{Ai}(z) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{s^3}{3} + sz\right)\mathrm{d}s \qquad \text{20-110}$$

$$\mathrm{Bi}(z) = \frac{1}{\pi}\int_0^\infty \left[exp^{-\frac{s^3}{3} + sz} + sin\left(\frac{s^3}{3} + sz\right)\right] ds \qquad \text{20-111}$$

The direct solution of the Schrödinger equation also has the added benefit of seamlessly treating thermionic emission and tunneling over the energy range.

To compute the current injected by the silicide contacts, a transmission coefficient $T(E_{inc},k_{||})$ is defined as the ratio of the transmitted and incident probability current densities.

$$T(E_{inc}, k_{||}) = \frac{J_{\text{transmitted}}}{J_{\text{incident}}} = \frac{|\psi|^2 k_2}{m_2^*} \cdot \frac{m_1^*}{|\psi|^2 k_1} \quad \text{20-112}$$

Here, the subscripts 1 and 2 refer to the silicide and silicon regions respectively. $|\psi|^2$ represents the wave function probability density for the carriers. At each iteration, a table of transmission probabilities is generated for each mesh location along the contact interface. Once the quasi-equilibrium distribution in the silicide is determined in terms of incident energy, the injected current density is obtained by integrating the product between carrier density and transmission probability.

Usage of Schottky Contacts

Compared to typical MOS devices, simulation of Schottky devices require few particles. Since the behavior of the silicide contacts can be shown by injection boundary conditions, depending on a quasi-equilibrium distribution, the entire ensemble of simulated particles is applied to transport in the conduction channel. This greatly improves performance and resolution of the method and allows you to easily probe the subthreshold regime.

For Schottky barrier devices, you can set the value for N on the **PARTICLE** statement to around 10000 without losing accuracy. To create a Schottky contact, you can use the **REGION** statement in the same way. Two silicides are provided, platinum silicide PTSI, and erbium silicide ERSI. To set the barrier height for your Schottky contact, use the **MATDEF** statement and modify the BARRIER field.

For simulation of Schottky barrier devices, there is also a secondary input file. The file inj_2.in is read in by MC Device to interpret the Schottky-specific statements. The only thing is that the Poisson barrier height in the inj_2.in file must match the value for BARRIER set in the **MATDEF** statement for the Schottky material.

20.7.10 Gather/Scatter Algorithm

The identification of particles that need special handling in the simulation is performed as follows:

1. The predictor step is computed, yielding the *k*-vector and the position at the end of the time step. The first-order formula is used in *k*-space. The second-order formula in real space.
2. The time of the next scattering is computed for all particles as the ratio between the remaining normalized lifetime and the total scattering rate. This can be less than zero for particles that should have scattered in the previous time step, or greater than *DT* for particles that do not scatter.
3. The *k* and *x*-vectors at the end of the free flight are computed. The mesh and region indices are also computed. These state variables have the suffix NEW: XNEW, KNEW, IXNEW, and so on. A forward difference is used for the *k*-vector, assuming that the electric field doesn't change much during one time step. A second-order formula is used to compute the final position, in case the velocity varies during the free flight (e.g., for very string electric fields, or long time steps).

4. If a particle changes region during the predicted time step, the time of boundary crossing is computed. A straight-line approximation is used for the flight trajectory. This reduces the calculation to a simple geometrical problem, speeding up the calculation considerably. Since region crossings are relatively rare events, this approximation does not affect accuracy very much. If the time of crossing is less than the scattering time, the particle is added to the appropriate reflection-handling list. The time of crossing and the side of the rectangular region where the crossing occurs are stored.
5. Particles that scatter are moved to their scattering position XNEW. A final state is selected and used as a new initial state for the remainder of the time step. The new predictor step is computed.
6. For each scattered particle, the region index is recomputed at the end of the time step, and compared with that at the beginning of the time step. The intermediate region index is not used in case the particle touches the corner of a region while scattering. If they differ, the particle is added to the reflection list.
7. For each rectangular region, the reflection is performed for all particles in the associated list. Boundary check is performed again and particles are added to the reflection lists. The step is repeated until all particles maintain the same region index during the last subflight in the time step.

20.7.11 Interpolation Schemes

Trilinear Interpolation

Consider a trilinear function in the form:

$$f(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z + a_4 yz + a_5 xz + a_6 xy + a_7 xyz \quad (20\text{-}113)$$

where the following definitions are made:

$$f(0, 0, 0) = f_{000} \quad (20\text{-}114)$$

$$f(1, 0, 0) = f_{100} \quad (20\text{-}115)$$

$$f(0, 1, 0) = f_{010} \quad (20\text{-}116)$$

$$f(1, 1, 0) = f_{110} \quad (20\text{-}117)$$

$$f(0, 0, 1) = f_{001} \quad (20\text{-}118)$$

$$f(1, 0, 1) = f_{101} \quad (20\text{-}119)$$

$$f(0, 1, 1) = f_{011} \quad (20\text{-}120)$$

$$f(1, 1, 1) = f_{111} \quad (20\text{-}121)$$

Equations 20-113, 20-114, 20-115, 20-116, and 20-118 give:

$$a_0 = f_{000} \quad (20\text{-}122)$$

$$a_1 = f_{100} - a_0 \quad (20\text{-}123)$$

$$a_2 = f_{010} - a_0 \quad (20\text{-}124)$$

$$a_3 = f_{001} - a_0 \quad (20\text{-}125)$$

Equations 20-113, 20-116, and 20-120 yield:

$$f_{011} = a_0 + a_2 + a_3 + a_4 = f_{010} + a_3 + a_4 \quad (20\text{-}126)$$

Therefore, by symmetry:

$$a_4 = f_{011} - f_{010} - a_3 \quad (20\text{-}127)$$

$$a_5 = f_{101} - f_{001} - a_1 \quad (20\text{-}128)$$

$$a_6 = f_{110} - f_{100} - a_2 \quad (20\text{-}129)$$

Equations 20-113 and 20-121 yield:

$$f_{111} = \sum_{j=0}^{7} a_i = f_{011} + a_1 + a_5 + a_6 + a_7 \quad (20\text{-}130)$$

and therefore:

$$a_7 = f_{111} - f_{011} - a_1 - a_5 - a_6 \quad (20\text{-}131)$$

An efficient implementation of the interpolation algorithm can be achieved by factoring Equation 20-113.

$$f(x, y, z) = a_0 + x(a_1 + a_5 z + y(a_6 + a_7 z)) + y(a_2 + a_4 z) + a_3 z \quad (20\text{-}132)$$

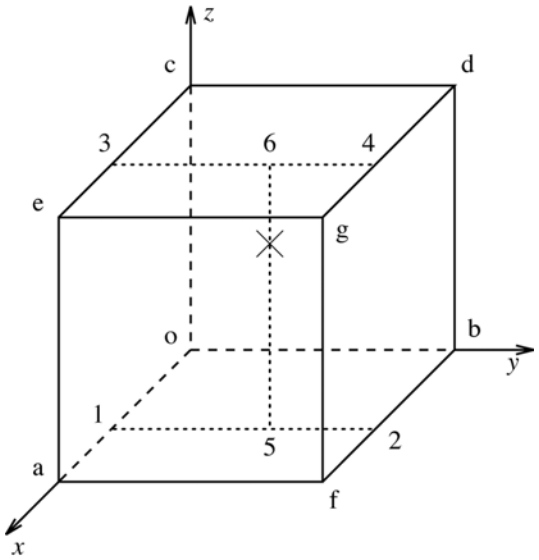


Figure 20-17: Tricubic interpolation scheme. The letters denote the vertices of the cube and lower-order interpolation points. The numbers denote the edges of the cube and higher-order (cubic) interpolation points.

Tricubic Interpolation

The tricubic interpolation is achieved by a sequence of 7 simple cubic interpolations (see Figure 20-17). The 1D cubic interpolation consists in finding a function of the form:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \quad (20\text{-}133)$$

where the following definitions are made:

$$f(0) = f_0 \quad (20\text{-}134)$$

$$f'(0) = f'_0 \quad (20\text{-}135)$$

$$f(1) = f_1 \quad (20\text{-}136)$$

$$f'(1) = f'_1 \quad (20\text{-}137)$$

Substituting them, we have:

$$a_0 = f_0 \quad (20\text{-}138)$$

$$a_1 = f'_0 \quad (20\text{-}139)$$

$$a_3 = f'_1 - 2f_1 + f'_0 + 2f_0 \quad (20\text{-}140)$$

$$a_2 = f_1 - a_3 - a_1 - a_0 \quad (20\text{-}141)$$

The complete evaluation of one tricubic point requires 7×3 = 21 multiplications.

20.7.12 WKB Formula

The Schottky-lowered barrier can be described by the formula:

$$E(x) = W - \mathrm{F}x - \frac{e}{16\pi\varepsilon x} \quad (20\text{-}142)$$

Here, W is an effective barrier (i.e., the ideal (non-lowered) band-edge discontinuity, minus the normal component of the kinetic inside the oxide). F is the electric field and ε is the effective image-force dielectric constant [88]. We consider lowering at one of the ends of the barrier (Figure 20-18).

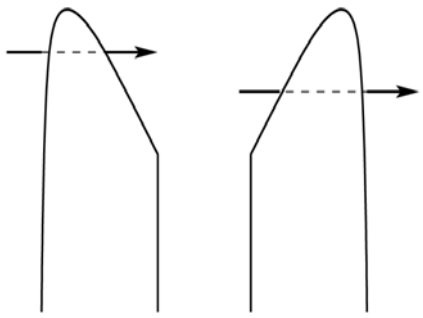


Figure 20-18: Schottky Lowering.

The entry and exit points x_{in} and x_{out} are found for $E(x)$=0.

$$x_{in,\,out} = \frac{W}{2F}\left(1 \pm \sqrt{1 - \frac{\mathrm{eF}}{4\pi\varepsilon W^2}}\right) \quad (20\text{-}143)$$

For very low fields or very high barriers or both, the second term within the square root can be much smaller than one. In such cases, a much more accurate expression for $x_{in,out}$ is obtained by a series expansion of the square root:

$$x_{in} = \frac{e}{16\pi\varepsilon W} \quad (20\text{-}144)$$

$$x_{out} = \frac{W}{F} \quad \text{20-145}$$

Finally, the WKB integral is performed numerically on a very fine mesh. Since a typical barrier is at most of the order of 1 Ry and the effective mass in the oxide is about $0.5m_0$, the attenuation factor is the order of a fraction of a_0^{-1}, where a_0 is the Bohr radius. A mesh step of $a_0/2$ has been chosen.

20.7.13 Strain

You can use the strain model in MC Device to account for the effect of strains on the transport of electrons and holes in bulk and 2D simulations. This model can be applied to different materials and is also known as the multi-material model [355]. In its present form, the model can treat semiconductors that have 16-fold degeneracy in their band structures and where the irreducible wedge (IW) falls in the domain where $k_x \geq 0$, $k_y \geq 0$, $k_z \geq 0$, and $k_y \geq k_x$. In this version of MC Device, fittings are provided for electron transport in silicon under biaxial tensile strain. When used in 2D, you may define any number of rectangular strain regions. In each strain region, the strain varies linearly from the value of the effective strain (or the effective mole fraction) you provide at the corners of the rectangle.

Band Structure

When you perform a bulk simulation, you need to consider just one value of the strain that pertains to the bulk. But when you perform device simulations, you often need to consider strain that varies over the device. For either type of simulation, it is convenient if your strain model uses one set of strain fittings that pertain to a wide range of strain levels so that these cases can be modeled with little change in your input. This is the case with the strain model in MC Device. To test different bulk cases for a given strain type, you change the effective strain for each case. To test different strain levels within a device, you define the strain over your device structure. For supported strain types, you can simply set the level of strain to use the model. In such cases, you do not need to characterize the effect of strain on the band structure and on carrier scattering since it has already been set.

In Monte Carlo electronic transport modeling, two operations related to the semiconductor band structure are performed repeatedly. The first operation is called the forward (or direct) evaluation. In the forward evaluation, the modeling tool determines the carrier energy, E, and carrier velocity, $\boldsymbol{v}$, of each MC carrier based on its band index, n, and its wave vector, $\boldsymbol{k}$. The second operation is call the reverse (or indirect) evaluation. In the reverse evaluation, the modeling tool determines the band index, n, and the wave vector, $\boldsymbol{k}$, after each MC carrier has scattered. Figure 20-4 shows a flowchart that depicts the forward and reverse evaluations done in MC Device when the strain model is active. (The evaluations done in MC Device when the strain model is not active amount to the top two-thirds of Figure 20-4.) Although not shown in Figure 20-4, the strain model also provides strain corrections for each component of the carrier velocity. Like the strain correction for the energy, the strain correction for each velocity component depends quadratically on the effective strain, m. For example, the strain correction for the x-component of the velocity involves a term that depends linearly with the effective strain (and uses the coefficients a_x) and a term that depends quadratically on the effective strain (and uses the coefficient b_x).

For the forward evaluation, the strain model uses the unstrained band structure along with fitting coefficients of the quadratic correction terms, $a(n,\boldsymbol{k})$ and $b(n,\boldsymbol{k})$. A quadratic correction was found to be reasonably accurate for modeling the band structure of strained silicon on relaxed SiGe over the entire Brillouin zone for both valence and conduction bands and for both the carrier energy and for the carrier velocity components [355]. Since two coefficients are needed for the correction for every point in the irreducible wedge and for every band, the amount of data needed to characterize a strain-dependent band structure is three times that of the unstrained band structure.

For the reverse evaluation, the strain model performs a search to obtain the strained band index, n_m, and strained wave vector, $\boldsymbol{k}_m$, of each MC carrier. In this case, the reverse lookup table that is used to obtain the final states based on the energy is extended to include the two quadratic fit parameters, $a(n,\boldsymbol{k})$ and $b(n,\boldsymbol{k})$. First, the final energy of the carrier is searched in the energy-ordered lookup table based on the energy of that state in unstrained silicon. This search yields an unstrained state $(n_0,\boldsymbol{k}_0)$. The location of the strain-dependent final state $(n_m, \boldsymbol{k}_m)$ can be reduced to a small local region in the table around the unstrained state $(n_0,\boldsymbol{k}_0)$. The local region is small because the modification of the energy of the strained state is small relative to the energy range of the table. The size of the local region is chosen so that it will always contain the strained state.

Second, the strained state $(n_m,\boldsymbol{k}_m)$ is determined by randomly choosing a state within the local region and checking if the strained energy, E_m, at the randomly chosen state matches the known final energy of the carrier, E_f.

Note: This search will correctly sample the states within the local region according to the density of states (DOS).

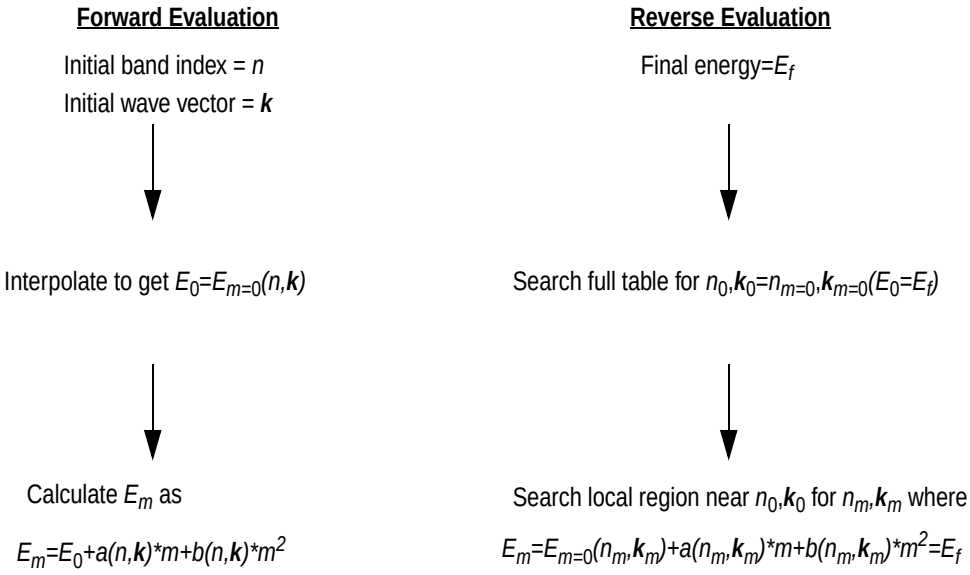


Figure 20-19: Forward and reverse evaluations for the strain model. *m* pertains to the effective strain (or to the effective mole fraction). When the strain model is inactive, the last step of the evaluation is skipped.

Phonon Scattering

MC Device models the transport of charge carriers but not the transport of phonons. In this context, the term phonon scattering refers to carrier-phonon scattering or the scattering of charge carriers with phonons. The phonon scattering rates in MC Device are determined at the start of the simulation by one of two methods. The first method is to calculate the rates using analytic formulas and is selected by setting TYPE=ANALYTIC on the ESCAT statement. The second method is to load the rates from data files and is selected by setting TYPE=FULLBAND on the ESCAT statement.

The first method (TYPE=ANALYTIC) is more flexible and supports a more complete set of mechanisms in MC Device. For this method, the phonon scattering mechanisms include: intravalley acoustic, intervalley X-X f-type (at 3 phonon temperatures), intervalley X-X g-type (at 3 phonon temperatures), and intervalley X-L (at 4 phonon temperatures). Both absorption and emission of phonons are considered for all carrier phonon scattering mechanisms. Therefore, there are a total of (1+3+3+4)*2=22 carrier-phonon scattering mechanisms. Anisotropy in the final state selection is included by having different mechanisms (e.g., f and g-types) and through the density of states (DOS). The formulas for these rates are scaled by the full-band DOS at high energies to obtain accurate values for the rates at high energies. The rates for these mechanisms are described in Section 20.7.5 "Phonon Scattering". The format of the scat.out file produced when TYPE=ANALYTIC is given in Table 20-2.

The second method (TYPE=FULLBAND) enables you to calculate the scattering rates using full band structure although it must be done outside of MC Device. For this method, the phonon scattering mechanisms include: longitudinal and transverse acoustic scattering and one type of optical scattering. Both absorption and emission of phonons are considered for all carrier-phonon scattering mechanisms. Therefore, there are a total of 3*2=6 carrier-phonon scattering mechanisms. In this case, carrier phonon scattering is anisotropic only through the density of states. In this case, the rates do not need to be scale by the full-band DOS at high energies, since they were calculated with full band structure. Table 20-3 shows the format of the scat.out file produced when TYPE=FULLBAND.

Bulk Simulations

Bulk simulations were performed with MC Device and compared with experimental data to verify the model. The model was originally developed to treat transport in silicon under biaxial tensile strain, so much of the testing has been performed for this type of strain. For this case, the low-field electron mobility versus effective mole fraction is shown in Figure 20-20. Here, the TYPE=FULLBAND is used on the ESCAT statement and FIELD=(1e3 1e3) on the **BULK** statement. The plot of the field-based electron mobility estimate shows that the strain model in MC Device obtains good agreement with the experimental results obtained by Welser [345] and with the experimental results obtained by Ismail at x=0.3 [142].

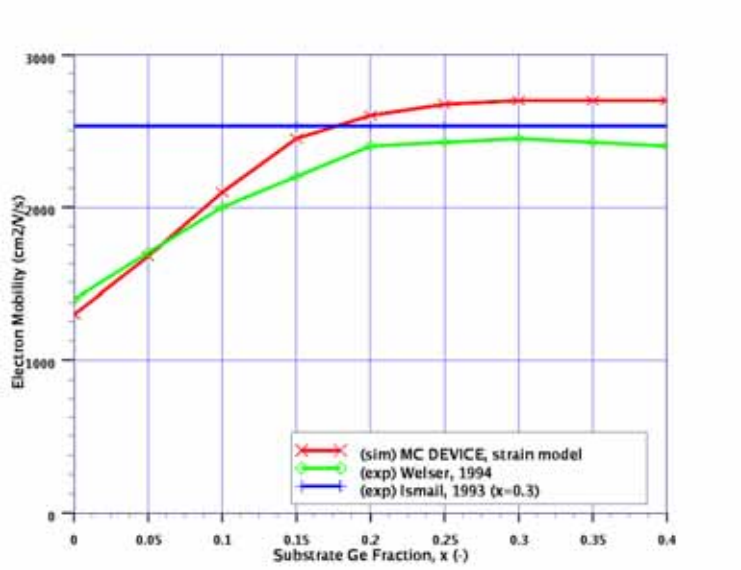


Figure 20-20: Electron mobility versus Substrate Ge fraction, *x*.

The mobility enhancement saturates at approximately x=0.2 (20% Ge in the substrate). This indicates that the splitting of the X valleys of the conduction band is sufficiently large at this mole fraction (strain level) that the entire population of electrons occupies the lower-lying X valleys where the effective mass is lower. Further splitting of the X valleys provides little increase in the electron mobility, since all electrons already reside in the lower-lying X valleys at *x*=0.2.

Note: The strain model in MC Device predicts the impact of strain on band structure and carrier scattering. The strain model in MC Device does not predict the impact of defects that can occur at high strain levels and the corresponding impact on carrier transport.

To use the strain model in MC Device for bulk modeling, you must use some or all of the following input statements: MMAT, ESCAT, and **BULK**.

The MMAT statement is used to turn on the strain model.

```
MMAT TYPE=1
```

If the TYPE parameter is set to 1|TRUE|YES, the strain mode is active. By default, the TYPE parameter is set to 0|FALSE|NO, so the strain model is off.

The ESCAT statement is used to control carrier-phonon scattering (see "Phonon Scattering" on page 1028).

```
ESCAT TYPE=ANALYTIC FBPROC=7 FBBAND=4 ANPROC=24 ANBAND=1 \
    FPNCPLFILE="phcpl_c.in" PHNFACFILE="phon_fac_c_uni_300k.in"\
    PLOTFILE="scatfl_c_uni.in"
```

If the TYPE=ANALYTIC (as it is by default), then the scattering rates are calculated inside MC Device using analytic formulas. If the TYPE=FULLBAND, then scattering rates are loaded from files in the $MCDEVICEDIR directory as specified by the PHNCPLFILE, PHNFACFILE, and PLOTFILE parameters.

FBPROC sets the number of scattering processes used for TYPE=FULLBAND. FBBAND sets the number of bands used for TYPE=FULLBAND. anproc sets the number of scattering processes used for TYPE=ANALYTIC. anband sets the number of bands used for TYPE=analtyic.

FHNCPLFILE and PHNFACFILE set two filenames related to the phonon scattering rates when TYPE=FULLBAND. PHNCPLFILE sets the file which contains the phonon coupling parameters. PHNFACFILE sets the file which contains the sum prefactor in the phonon scattering formulas.

PLOTFILE sets the file from which the phonon scattering rates will be loaded when TYPE=FULLBAND. Column 1 in PLOTFILE is the carrier energy in eV. Columns 2-7 are the 6 phonon scattering rates. The 6 phonon scattering rates pertain to columns 3-8 of scat.out produced for this case (see Table 20-3). The rates must be scaled by the squared phonon coupling parameters (in "phcpl_c.in") and scaled by the sum prefactors (in "phon_c_uni_300k.in") in order to get units of 1/s. This calculation is done inside MC Device after the data are loaded and the results can be written to "scat.out" (see Table 20-3).

Note: Fittings for TYPE=FULLBAND are only provided for simulating MC electrons in strained silicon (CARRIER=E on the ALGO statement). These are not provided for simulating MC holes (CARRIER=H on the ALGO statement).

The **BULK** statement is used to set parameters associated with a bulk simulation (see Section 20.4 “Bulk Simulation”).

```
BULK MMFRAC=0.05
```

The MMFRAC parameter sets an effective strain (or effective mole fraction) that pertains to the bulk. In the case of biaxial tensile strained silicon, this pertains to the mole fraction, *x*, of an assumed underlying substrate of relaxed $Si_{1-x}Ge_x$. For other types of strain, MMFRAC pertains to an effective strain.

Device Simulations

The strain model was tested in 2D by simulating a 50 nm well-tempered *n*-MOSFET in saturation with and without strain. The strained *n*-MOSFET was based on a relaxed $Si_{1-x}Ge_x$=$Si_{0.7}Ge_{0.3}$ substrate, while the unstrained (relaxed) *n*-MOSFET was based on a pure silicon substrate. Although no attempt was made to fit the surface scattering or the threshold voltage shift, relative comparisons of the results can be used to analyze the qualitative effects of strain. The strained channel device showed a 60% increase in the drive current relative to the unstrained case. This result agrees with the low-field mobility enhancement expected at low drain bias. A large velocity enhancement is seen in the strained *n*-MOSFET, which reflects the increased low-field mobility [355].

To use the strain model in MC Device in device simulations, you must use some or all of the following input statements: MMAT, ESCAT, and MMREGION.

The MMAT and the ESCAT statements are described in Section 20.4 “Bulk Simulation”. The MMAT and ESCAT statements are used the same in 2D device simulations as they are in bulk simulations.

The MMREGION statements are used to define the effective strain (or effective mole fraction) in rectangular regions.

```
MMREGION BOUNDP=(0.025,0.175,-0.062,0.12) FRAC=(0.3,0.3,0.3,0.3)
```

The BOUNDP parameter gives the boundary of the region within which the strain model will be applied (see “Regions” on page 966). A BOUND parameter is provided for backward compatibility. The FRAC parameter gives a list of four values for the effective strain to be used at the corners of each rectangular region. These four values pertain to the values in the lower-left, the lower-right, the upper-right, and the upper-left corners, respectively. (These directions are based on the directions of the axes in TonyPlot where $\hat{x}$ is to the right and where $\hat{y}$ is up.) The effective strain inside the rectangle is linearly interpolated based on the values at the corners.

20.8 Tips

1. Use the RESTART parameter when performing a series of simulations:

   There is a switch for using RESTART option on the algo statement. If you set RESTART = 1|TRUE|YES, the simulator reads in the particle states from the status.in file. A corresponding status.out file can be written at the end of a simulation and you may copy status.out to status.in. The way it helps when performing a series of simulations is as follows. MC Device initially places particles with the aim of achieving charge neutrality everywhere. Therefore, for any new simulation the bulk of the particles are places in the source and the drain region initially. These particles are then redistributed in the channel region. Therefore, when the restart option is used, the particles are already in the channel satisfying the boundary conditions of the previous bias conditions and the time that is needed for achieving this steady state is reduced or eliminated

2. Try changing the error bound for the inner loop for Poisson solver (see TOL=(2e-4,1e-5) parameter on the **POISSON** statement):

   The TOL parameter takes two values which correspond to the two tolerances used by the Poisson solver in MC Device. Since it is an iterative solver, it uses a preconditioner (inner loop) and then a linear search routine (outer loop) to get the solution. The two values for the TOL parameter are for the maximum error in the linear search routine and the tolerance of the preconditioner. Making the tolerance of preconditioner (the second value) more smaller (more strict) may increase the number on inner loops but reduce the number of outer loops. Thus, reducing the overall time. There is a trade off here. Depending on the device, you may want to adjust the solver for the fastest execution time while you keep the tolerance of the result (tolerance of linear search routine) the same.

3. Real time between Poisson solutions:

   It has been observed that solutions are not as accurate when the real time step between successive Poisson solutions is more than 0.2 fs. Therefore, it is advised that the real time step between any two Poisson solutions should be 0.2 fs or less. For example, if your transport time step (on the **ALGO** statement) is 0.02 fs, then the Poisson TSTEP (on the **POISSON** statement) should not be greater than 10.

Chapter 21
Numerical Techniques

21.1 Overview

This chapter describes the overall process of obtaining a numerical solution, the subtasks involved, and the options and defaults available in Atlas.

You don't need to master this material in order to use Atlas. Section 2.8 "Choosing Numerical Methods" presents the information about numerical techniques that is needed. This chapter provides additional information that will mainly be of interest to advanced users.

21.2 Numerical Solution Procedures

Semiconductor device operation is modeled in Atlas by a set of anywhere from one to six coupled, non-linear, partial differential equations (PDEs). Atlas produces numerical solutions of these equations by calculating the values of unknowns on a mesh of points within the device. An internal discretization procedure converts the original, continuous model to a discrete non-linear algebraic system that has approximately the same behavior. The set of PDEs, the mesh and the discretization procedure determine the non-linear algebraic problem that must be solved.

The non-linear algebraic system is solved using an iterative procedure that refines successive estimates of the solution. Iteration continues until the corrections are small enough to satisfy convergence criteria, or until it is clear that the procedure is not going to converge. The non-linear iteration procedure starts from an initial guess. The corrections are calculated by solving linearized versions of the problem. The linear subproblems are solved by using direct techniques or iteratively.

Different solution procedures exhibit different behavior with respect to convergence, accuracy, efficiency, and robustness. The two main aspects of convergence are whether a solution is obtained and how rapidly it is approached. Accuracy is how closely the computed solution approximates the true solution. Efficiency is the time required to produce a solution. Robustness is the ability to converge for a wide range of structures, using meshes and initial guess strategies that are not optimum.

When solving general systems of non-linear equations, there are no guarantees that any particular method will always work. It is also the case that different methods can work better for different problems. Fortunately, there is now a lot of practical experience concerning the numerical techniques that are effective for device simulation. This practical experience has been captured in Atlas in the form of default methods and parameters that work well in almost all circumstances. This chapter provides advanced information of interest to users who want to change the defaults.

21.3 Meshes

The specification of meshes involves a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine mesh that can resolve all significant features of the solution. Numerical efficiency requires a coarse mesh that minimizes the total number of grid points. This trade-off between accuracy and numerical efficiency is frequently a source of problems for beginners. Fortunately, enough experience to define reasonable meshes is soon acquired.

Atlas uses triangular meshes. Some triangulations yield much better results than others. Mesh generation is still an inexact science. Guidelines and heuristics, however, for defining satisfactory meshes exist. Good triangulations have the following features:

- They contain enough points to provide the required accuracy.
- They do not contain too many unnecessary points that impair efficiency.
- They avoid, or at least minimize, the number of obtuse triangles. Obtuse triangles tend to impair accuracy, convergence, and robustness.
- They avoid, or at least minimize, the number of long, thin triangles. These triangles also tend to impair accuracy, convergence, and robustness.
- They allow the average size of triangles to change smoothly in transition from a region where very small triangles must be used to a region where the use of much larger triangles is acceptable.

The error associated with a mesh can be investigated systematically by repeating a calculation using a sequence of finer meshes. This is very time consuming and is hardly ever done. The typical approach is to adequately resolve structural features, including doping, with an initial or base mesh, and then add nodes as required to resolve significant features of the solution. The insertion of additional nodes (regridding) is normally done by the program using user-specified criteria.

The initial mesh used by Atlas can be specified in several ways. It can be inherited from Athena. It can be constructed using DevEdit. It can be specified using the Atlas command language. Meshes can be refined using Atlas commands, or using DevEdit. The remainder of this section will focus on the capabilities available using Atlas commands. The capabilities provided by DevEdit are documented in the DevEdit User's Manual.

There are limits on the maximum number of nodes that can be specified. Two-dimensional Atlas simulations may have up to 20,000 nodes. Three-dimensional simulations may have up to 200,000 nodes, 400,000 elements, with no more than 20,000 in a single plane and a maximum of 200 planes in the Z direction. Most devices can be adequately simulated in two dimensions using meshes that contain from several hundred to around 3000 nodes.

21.3.1 Mesh Regridding

The REGRID statement supports refinement of regions of the mesh according to specified criteria. Refinement can occur when a specified solution variable exceeds some value, or when the change in that variable across a triangle exceeds a value. The variable can be any of the key quantities in a problem, such as potential, carrier concentration, doping concentration, or electric field.

The regrid algorithm searches the initial grid for triangles that meet the criterion specified for refinement. Each triangle that is identified is divided into four congruent subtriangles. Grid quantities (doping, potential, carrier concentrations, and so forth) are interpolated onto the new nodes, using linear or logarithmic interpolation, as appropriate for that quantity. The initial grid is referred to as being “on level 0" and the new triangles are referred to as “on level 1". After all level 0 triangles have been examined, the same procedure is applied to level 1 triangles, and any subtriangles of level 1 become “level 2" triangles. The grid is checked for consistency at each level and is updated to avoid abrupt changes of size from one triangle to the next. The regrid process continues until no more triangles meet the refinement criteria, or until a specified maximum level of refinement is reached. Grids used in practice are often coarser than is required to meet desirable refinement criteria. Therefore, the maximum level is the key factor in determining the size of the grid after refinement.

The MAX.LEVEL parameter of the **REGRID** statement is used to limit the amount of refinement at each step. By default, Atlas sets the maximum level equal to one more than the highest level in the existing mesh. To update a coarse region without regridding the finer regions, after a mesh has already been refined several times, set the maximum level below the level of the finer regions in the existing grid.

If several levels of regrid are performed in immediate succession, interpolated data is used to make the refinement decisions at higher levels. Since semiconductor problems are non-linear, this interpolation may not produce satisfactory results. It is often a good idea to calculate a new solution between regrid operations. In other words, to regrid only one level at a time and obtain a new solution after each regrid operation.

Two popular choices of quantities to be used for mesh refinement are potential and doping. Ideally, variations of electrostatic and quasi-Fermi potentials across an element would be limited to no more than *kT/q*, and variations in doping would be no more than a factor of 5 or so. In practice, the refinement criteria are often significantly coarser: around 10-20 kT/q for potential, and two to three orders of magnitude for doping. In high level injection situations, it is a good idea to regrid when the value of minority carrier concentration exceeds the local doping concentration.

21.3.2 Mesh Smoothing

Although every step of grid generation can introduce obtuse triangles, two steps in particular can cause problems. The first is that distorting a rectangular mesh introduces a very large number of obtuse elements. The second is that when regridding a rectangular grid that contains triangles with an aspect ratio of 4:1 or greater, very obtuse triangles are created in the transition region between high and low grid density. The **REGRID** statement allows several procedures to be used when dealing with poorly shaped elements such as obtuse triangles.

The two techniques are node smoothing and triangle smoothing. With node smoothing, several iterative passes are carried out during, which each node is moved to a position and improves the angles of the triangles surrounding it. Node smoothing should only be used for grids that are already irregular. If node smoothing is used for nearly rectangular grids, it may significantly degrade the quality of the mesh.

With triangle smoothing (which is also referred to as diagonal flipping), each adjoining pair of triangles is examined. If appropriate, the diagonal of the quadrilateral is flipped to stabilize the discretization. The diagonal is never flipped when two elements are composed of different materials. When elements are of the same material but have different region numbers, you can specify whether or not to flip the diagonals.

Triangle smoothing is desirable in almost all cases, and should be performed on both the initial grid and on subsequent regrids. The only exception to this rule arises from an undesirable interaction of three elements: regrid, high aspect ratio triangles, and smoothing. This situation frequently occurs in gate oxide regions that involve long, thin triangles. In these cases, smoothing may produce large triangles surrounded by many smaller triangles, giving the appearance of a hole in the mesh. To overcome this, use the smoothing command SMOOTH=4 to limit the formation of the large triangles.

21.4 Discretization

21.4.1 The Discretization Process

The discretization process yields relationships between the variables defined at mesh points [252]. In order to be useful, a discretization must be consistent (i.e., it must limit to the underlying equation in the limit that the mesh size tends to zero). All of the discretizations used in Atlas have this property. Different discretizations can have different properties with respect to accuracy. The most important measure of accuracy is the order of the scheme (i.e., how errors scaled as the difference between mesh points tends to zero). Discretization schemes used in device simulation are often second order. In other words, as the mesh becomes very fine the discretization error varies as the square of the separation between mesh points.

The discretizations implemented in Atlas uses the Box Integration Method [329] to approximate differential operators on a general triangular grid. Each equation is integrated over a small polygon which encloses each node. The set of all polygons completely covers the solution domain. The integration procedure equates fluxes into a polygon with sources and sinks inside the polygon, which means that quantities that are conserved physically are also conserved by the numerical solution.

The fluxes must be discretized carefully for the carrier continuity and energy balance equations. Otherwise, nonphysical oscillations and negative carrier concentrations and temperatures may arise. Scharfetter and Gummel [276] introduced approximations for current densities that overcome this problem. Generalizations of this approach are used in Atlas for the discretization of current densities and energy fluxes.

21.5 Non-Linear Iteration

The non-linear solution method, and associated parameters such as iteration and convergence criteria, are specified in the **METHOD** statement. Non-linear iteration solution methods are specified in the **METHOD** statement using the NEWTON, GUMMEL, or BLOCK parameters. Combinations of these parameters may also be specified. In order to understand the effect of these parameters, it is helpful to briefly review how numerical solutions are obtained.

The non-linear algebraic system that results from discretization on a mesh is solved iteratively starting from an initial guess [270,352,9,47,44]. Linearized subproblems are set up and solved. These provide corrections that are used to update the current estimate of the solution. Different sequences of linear subproblems correspond to different non-linear iteration strategies. Iteration continues until convergence criteria are met, in which case the solution is accepted. Iternation also continues until a preset maximum allowable number of iterations is reached, in which case a different technique is tried or the solution procedure is abandoned. When a solution fails to converge, a user normally tries a different grid, a different initial guess strategy, or a different non-linear iteration technique.

21.5.1 Newton Iteration

Each iteration of the Newton method solves a linearized version of the entire non-linear algebraic system. The size of the problem is relatively large, and each iteration takes a relatively long time. The iteration, however, will normally converge quickly (in about three to eight iterations) as long as the initial guess is sufficiently close to the final solution. Strategies that use automatic bias step reduction in the event of non-convergence loosen the requirement of a good initial guess. Newton's method is the default for drift-diffusion calculations in Atlas. There are several calculations for which Atlas requires that Newton's method is used. These are DC calculations that involve lumped elements, transient calculations, curve tracing, and when frequency-domain small-signal analysis is performed.

The Newton-Richardson Method is a variant of the Newton iteration that calculates a new version of the coefficient matrix only when slowing convergence demonstrates that this is necessary. An automated Newton-Richardson method is available in Atlas, and improves performance significantly on most problems. To enable the automated Newton-Richardson Method, specify the AUTONR parameter of the **METHOD** statement.

If convergence is obtained only after many Newton iterations, the problem is almost certainly poorly defined. The grid may be very poor (i.e., it contains many obtuse or high aspect ratio triangles), a depletion region may have extended into a region defined as an ohmic contact, or the initial guess may be very poor.

21.5.2 Gummel Iteration

Each iteration of Gummel's method solves a sequence of relatively small linear subproblems. The subproblems are obtained by linearizing one equation of the set with respect to its primary solution variable, while holding other variables at their most recently computed values. Solving this linear subsystem provides corrections for one solution variable. One step of Gummel iteration is completed when the procedure has been performed for each independent variable. Gummel iteration typically converges relatively slowly, but the method will often tolerate relatively poor initial guesses. The Gummel algorithm cannot be used with lumped elements or current boundary conditions

Two variants of Gummel's method can improve its performance slightly. These both limit the size of the potential correction that is applied during each Gummel loop. The first method, called damping, truncates corrections that exceed a maximum allowable magnitude. It is used to overcome numerical ringing in the calculated potential when bias steps are large (greater than 1V for room temperature calculations). The maximum allowable magnitude of the potential correction must be carefully specified: too small a value slows convergence, while too large a value can lead to overflow. The DVLIMIT parameter of the **METHOD** statement is used to specify the maximum allowable magnitude of the potential correction. By default, the value of this parameter is 0.1 V. Thus, by default Gummel iterations are damped. To specify undamped Gummel iterations, specify DVLIMIT to be negative or zero.

The second method limits the number of linearized Poisson solutions per Gummel iteration, usually to one. This leads to under-relaxation of the potential update. This "single-Poisson" solution mode extends the usefulness of Gummel's method to higher currents. It can be useful for performing low current bipolar simulations, and simulating MOS transistors in the saturation region. It is invoked by specifying the SINGLEPOISSON parameter of the **METHOD** statement.

21.5.3 Block Iteration

Atlas offers several block iteration schemes that are very useful when lattice heating or energy balance equations are included. Block iterations involve solving subgroups of equations in various sequences. The subgroups of equations used in Atlas have been established as a result of numerical experiments that established, which combinations are most effective in practice.

In non-isothermal drift-diffusion simulation, specifying the BLOCK method means that Newton's method is used to update potential and carrier concentrations, after which the heat flow equation is solved in a decoupled step.

When the carrier temperature equations are solved for a constant lattice temperature, the BLOCK iteration algorithm uses Newton's method to update potential and concentrations. The carrier temperature equation is solved simultaneously with the appropriate continuity equation to update the carrier temperature and again carrier concentration.

When both the heat flow equation and the carrier temperature equations are included, the BLOCK scheme proceeds as described previously for the carrier temperature case. It then performs one decoupled solution for lattice temperature as a third step of each iteration.

21.5.4 Combining The Iteration Methods

You can start with the GUMMEL scheme. Then switch to BLOCK or NEWTON if convergence has not occurred within a certain number of iterations. One circumstance where this can be very helpful is that Gummel iteration can refine initial guess to a point from which Newton iteration can converge. The number of initial GUMMEL iterations is limited by GUM.INIT.

You may want to use BLOCK iteration and switch to NEWTON if there is no convergence. This is the recommended strategy for calculations that include lattice heating or energy balance. The number of initial BLOCK iterations is limited by NBLOCKIT.

Any combination of the parameters: GUMMEL, BLOCK, and NEWTON can be specified on the **METHOD** statement. Atlas will start with GUMMEL if it is specified. If convergence is not achieved within the specified number of iterations, it will then switch to BLOCK if BLOCK is specified. If convergence is still not achieved the program will then switch to NEWTON.

21.5.5 Solving Linear Subproblems

The linear subproblems generated by non-linear iteration can be solved by direct or iterative methods. Direct methods produce solutions in a predictable number of arithmetic operations. Solutions are affected by roundoff error but are otherwise exact. Iterative methods obtain solutions by making a series of corrections to an initial guess. Iteration proceeds until the calculated corrections satisfy specified convergence criteria. The results are not exact to within roundoff error, and convergence is not guaranteed for general problems. Iterative methods, however, can be more efficient than direct methods for large linear systems, and generally require less memory.

There are mathematical proofs regarding the types of linear system for which iterative techniques converge. All common iterative schemes converge for linear systems that are symmetric positive definite (SPD). The linearized form of Poisson's equation is of this type. The continuity equations are not SPD. But they can be reliably solved by modern, advanced iterative methods.

The size and structure of the coefficient matrix of the linear system plays an important role in the choice of direct or iterative methods. The overall problem size is determined by the number of variables per node (m) and the number of nodes (n). The number of unknowns is $m \times n$. The linear sub problems associated with Newton iteration have a coefficient matrix with $(m \times n)^2$ elements. Each linearized subproblem, which is used in Gummel iteration has a coefficient matrix with n^2 elements.

For practical 2D device simulation problems, the number of elements in the coefficient matrix is typically between 10^5 and 10^8. Fortunately, the matrices are sparse (i.e., most of the entries are zero, and you do not to store them). The sparsity arises because the variables at each node are coupled to only a few neighboring nodes. Special direct techniques are available for solving sparse matrices.

Direct techniques are preferred for relatively small problems. They provide reliable solutions that are exact to within roundoff error in a time that is predictable. Iterative techniques are preferred for very large problems because they are faster and require less memory. Direct techniques for solving sparse matrices have a competitive performance for the problem sizes typically encountered in 2D device simulation, and are used by Atlas to solve most of the linear subproblems that arise.

21.5.6 Convergence Criteria for Non-linear Iterations

After a few non-linear iterations, the errors will generally decrease at a characteristic rate as the iteration proceeds. Non-linear iteration techniques typically converge at a rate that is either linear or quadratic. The error decreases linearly when Gummel iteration is used (i.e., it is reduced by about the same factor at each iteration). For Newton iteration, the convergence is quadratic. In other words, small errors less than one are approximately squared at each iteration. The non-linear iteration is terminated when the errors are acceptably small. The conditions required for termination are called convergence criteria. Much effort has gone into developing reliable default convergence criteria for Atlas. The default parameters work well for nearly all situations, and most will never need to change them.

The main technique in Atlas is Gaussian Elimination (or LU Decomposition) with a minimum degree of reordering applied to the nodes [81].

21.5.7 Error Measures

A single positive number that characterizes error is obtained by taking a norm of the errors associated with each unknown. The quantity that Atlas tries to reduce to zero is the difference between the left and right hand sides of the equation. It is natural to use this quantity as the measure of the error. The associated error norm is called the right hand side (RHS) norm. The units of the RHS norm are C/μm for the Poisson equation, and A/μm for the continuity equations.

Carrier Concentrations and CLIM.DD (CLIMIT)

Another measure of error is provided by the size of the calculated corrections for each unknown. Since the updates are the unknown "Xs" at each step, this is called the X norm. Potential updates are measured in units of *kT/q*. Updates to carrier concentrations are measured relative to the previous value at the point. This relative εrror (ε_C) is defined as:

$$\varepsilon_C = \max_m \frac{C_m^{K+1} - C_m^K}{max(C_0, C_m^K)} \quad \text{20-1}$$

where $C = n$ or p, for electrons and holes respectively. m is the node identifier. $C0$ is a characteristic concentration. K is the iteration number. C_0 is specified as or as CLIM.DD or as CLIM·c* and

$$\text{CLIM.DD} = \text{CLIMT} \cdot c^* \quad \text{20-2}$$

where:

$$c^* = \sqrt[4]{N_c N_v} \quad \text{20-3}$$

CLIM.DD (or CLIMIT) is specified in the **METHOD** statement.

It is difficult to specify a reasonable default value for the CLIM.DD parameter in all situations. In many difficult cases, the round-off numerical errors do not allow for the resolution of very low concentrations. The default value of CLIMIT is set at 10^4 (the corresponding default value for CLIM.DD in Silicon is $4.5 \cdot 10^{13} cm^{-3}$). In simulation of breakdown, a lower value of CLIM.DD (~$10^8 cm^{-3}$ for Silicon diodes) should be specified. Otherwise, a "false" solution may be obtained.

Discussion of CLIM.EB

To estimate errors in the lattice temperature equation and the energy balance equations, corresponding RHS norms and X norms are calculated in Atlas. Updates to temperature are measured relative to some characteristic value of temperature. The CLIM.EB parameter can be viewed as a regularization parameter for the case of very small electron or hole densities in the energy balance equations. The CLIM.EB parameter specifies the minimum value of concentration for which the relaxation term in the energy balance equation will be properly resolved. The temperatures for points where the concentration is much less than CLIM.EB are equal to the lattice temperature. The units of CLIM.EB are cm^{-3} and the default is 0.0.

21.5.8 Terminal Current Criteria

Another qualification for convergence is derived from the relative changes in terminal currents and the satisfaction of total current conservation. This qualification can be expressed as the simultaneous satisfaction of the following conditions:

$$\left|I_i^{K+1} - I_i^K\right| \leq E_1\left|I_i^{K+1}\right| + E_2 \quad \text{20-4}$$

and

$$\left|\Sigma I_i^{K+1}\right| < 0.01\ maxi\ (\left|I_i^{K+1}\right|, E_2)$$

$$I = 1, nc \quad \text{20-5}$$

where

- I_i is the current through contact i
- K is the iteration number
- E_1 and E_2 are specified tolerances
- nc is the number of contacts

21.5.9 Convergence Criteria

A summary of the termination criteria that is enough for most purposes will now be given. Detailed reference information is provided in the next section.

The non-linear iteration is terminated when one of the following four criteria is satisfied.

1. The X norm for every equation falls below a specified tolerance. The specified tolerances for X norms are:
   - P_{tol}^x for potential equation.
   - C_{tol}^x for concentration equations.
   - TL_{tol}^x for lattice temperature equation.
   - TC_{tol}^x for carrier temperature equation.
2. The RHS for all equations and X norms for energy balance falls below a specified tolerance. The specific tolerances are:
   - P_{tol}^r for potential equations.
   - C_{tol}^r for concentration equations.
   - TL_{tol}^r for lattice temperature equations.
   - TC_{tol}^r for carrier temperature equations.
3. For every equation either the X norm or the RHS norm falls below a specified tolerance. In this case, both the XNORM and RHSNORM parameters must be specified true.

4. If either 1 or 2 or 3 criterion is fulfilled for weaker values of tolerances. In other words, for specified tolerances multiplied by the W parameter and current criteria 20-4 and 20-5 are satisfied.

To exclude the X-norm criterion, specify ^XNORM in the **METHOD** statement. To exclude RHS-norm criterion, specify ^RHSNORM. To exclude the current criterion, make E_1 and E_2 very small. You can change all the above mentioned tolerances simultaneously by specifying the relaxation factor TOL.RELAX in the **METHOD** statement.

Table 21-1 User-Specifiable Parameters for Convergence Criteria

Symbol	Statement	Parameter	Default
P_{tol}^{x}	**METHOD**	PX.TOL	10^{-5}
C_{tol}^{x}	**METHOD**	CX.TOL	10^{-5}
TL_{tol}^{x}	**METHOD**	TLX.TOL	10^{-5} (a)
TL_{tol}^{x}	**METHOD**	TOL.TEMP	10^{-3} (b)
TC_{tol}^{x}	**METHOD**	TCX.TOL	10^{-5}
P_{tol}^{r}	**METHOD**	PR.TOL	5.0×10^{-26}
C_{tol}^{r}	**METHOD**	CR.TOL	5.0×10^{-18}
TL_{tol}^{r}	**METHOD**	TLR.TOL	100
TC_{tol}^{r}	**METHOD**	TCR.TOL	100
E_1	**METHOD**	IX.TOL	2.0×10^{-5}
E_2	**METHOD**	IR.TOL	5.0×10^{-15}
W	**METHOD**	WEAK	200
TOL.RELAX	**METHOD**	TOL.RELAX	1
XNORM	**METHOD**	XNORM	TRUE
RHSNORM	**METHOD**	RHSNORM	TRUE
CLIMIT	**METHOD**	CLIMIT	10^4
CLIM.DD	**METHOD**	C_0	4.5×10^{13}
CLIM.EB	**METHOD**	CLIM.EB	0

Notes a) This refers to the NEWTON and GUMMEL method.
b) This refers to the BLOCK method.

21.5.10 Detailed Convergence Criteria

Only in very difficult situations is more detailed information concerning error estimation and the specification of convergence criteria needed. The material is organized by algorithm.

Convergence Criteria For Gummel's Algorithms

Relative update errors are defined as follows.

For potential:

$$\varepsilon_v = \frac{\max_m \left|\varphi_m^{K+1} - \varphi_m^K\right|}{max(1, \varphi_{mmax}^{K+1})}, \quad \text{20-6}$$

where *mmax* is the node where $\left|\varphi_m^{K+1} - \varphi_m^K\right|$ has its maximum value.

For electrons:

$$\varepsilon_n = \frac{\max_m \left|n_m^{K+1} - n_m^K\right|}{max(C_0, n_m^K)} \quad \text{20-7}$$

For holes:

$$\varepsilon_p = \frac{\max_m \left|p_m^{K+1} - p_m^K\right|}{max(C_0, P_m^K)} \quad \text{20-8}$$

For lattice temperature:

$$\varepsilon_{T_L} = \frac{(T_L)_{nmax}^{K+1} - (T_L)_{nmax}^K}{(T_L)_{nmax}^{K+1}} \quad \text{20-9}$$

where *nmax* is the node where $(T_L)_i^{K+1}$ has its maximum value.

For carrier temperature:

$$\varepsilon_{T_c} = max(\varepsilon_{T_n}, \varepsilon_{T_p}) \quad \text{20-10}$$

where

$$\varepsilon_{T_n} = \frac{\max_m \left|(T_n)_m^{K+1} - (T_n)_m^K\right|}{\max_m (T_n)_m^{K+1}} \quad \text{20-11}$$

$$\varepsilon_{T_p} = \frac{\max_m \left|(T_p)_m^{K+1} - (T_p)_m^K\right|}{\max_m (T_p)_m^{K+1}} \quad \text{20-12}$$

For drift diffusion, iterations are terminated if the following criteria are satisfied:

$$\varepsilon_v \leq P_{tol}^x \quad \text{20-13}$$

$$\varepsilon_n \leq c_{tol}^x \quad \text{20-14}$$

$$\varepsilon_v \leq C_{tol}^x \quad \text{20-15}$$

In non-isothermal drift diffusion, iterations are terminated if Equations 20-10 to 20-12 are satisfied, the current convergence criteria in Equations 20-4 and 20-5 are met and:

$$\varepsilon_{T_L} < TL_{tol}^x \quad \text{20-16}$$

In Gummel's method with energy balance equations, NITGUMM iterations, which only the non-linear Poisson equation is solved, will always be done. Gummel's method with energy balance equations is terminated if Equations 20-16 is fulfilled, the carrier temperature convergence criteria

$$\varepsilon_{T_c} \leq TC_{tol}^x \quad \text{20-17}$$

is achieved, and one of the following conditions is valid:

- NITGUMM < K+1 ≤ NITGUMM+NT1 and Equations 20-13 is fulfilled,
- NT1 + NITGUMM < K+1 ≤ NITGUMM + NT3 and

$$\varepsilon_n \leq P_{tol}^x \cdot w \quad m \quad \text{20-18}$$

where w = 10

- NT1 + NITGUMM < K + 1 ≤ NITGUMM + NT3, the current convergence criteria (Equations 20-4 and 20-5) are satisfied, and inequality (Equations 20-18) is valid for w=100.
- NT1 + NITGUMM < K+1 and (Equations 20-18) is valid for w=100.
- NT1 + NITGUMM < K+1 the current convergence criteria (Equations 20-4 and 20-5) are satisfied, and (Equations 20-18) is valid for w = 500.

The default values of the iteration parameters are NITGUMM=5, NT0=4, NT1=10, and NT3=100.

Convergence Criteria For Newton's Algorithm

Relative update errors are defined as follows.

For potential

$$\varepsilon_v = \max_m \left| \varphi_m^{K+1} - \varphi_m^K \right| \quad \text{20-19}$$

For electron concentration:

$$\varepsilon_n = \frac{\max_m \left| n_m^{K+1} - n_m^K \right|}{max(C_0, n_m^K)} \quad \text{20-20}$$

For hole concentration:

$$\varepsilon_p = \frac{\max_m \left| p_m^{K+1} - p_m^K \right|}{max(C_0, p_m^K)} \quad \text{20-21}$$

For lattice temperature and carrier temperature:

$$\varepsilon_{T_L} = \frac{\max_m \left| (T_L)_m^{K+1} - (T_L)_m^K \right|}{T_{scale}}, \quad \text{20-22}$$

$$\varepsilon_{T_n} = \frac{\max_m \left| (T_n)_m^{K+1} - (T_n)_m^K \right|}{T_{scale}}, \quad \text{20-23}$$

$$\varepsilon_{T_p} = \frac{\max_m \left| (T_p)_m^{K+1} - (T_p)_m^K \right|}{T_{scale}}, \quad \text{20-24}$$

where the scaling temperature, *Tscale*, is by default equal to 300K.

To define the RHS norms used in Atlas, first represent the non-linear equations obtained after discretization at every node as

$$F_\alpha(\overrightarrow{\chi}) = 0 \quad \text{20-25}$$

where α can be ψ, n, p, T_L, T_n, and T_p for the potential equation, electron continuity equation, hole continuity equation, lattice temperature equation, electron temperature equation and hole temperature equation, respectively. $\vec{\chi}$ represents the vector of unknowns.

The RHS norms in Atlas is then defined as follows:

For the potential equation:

$$\varepsilon_v^g = \max_m |F\psi| \quad \text{20-26}$$

For the electron and continuity equations:

$$\varepsilon_n^g = \max_m |F_n| \cdot C_T \quad \text{20-27}$$

$$\varepsilon_p^g = \max_m |F_p| \cdot C_T \quad \text{20-28}$$

where CT = 10-4 $4\sqrt{N_c N_v}$ kT

For the lattice temperature and carrier temperature equations:

$$\varepsilon_{T_L}^g = \max_m |F_{T_L}| \quad \text{20-29}$$

$$\varepsilon_{T_n}^g = \max_m |F_{T_n}| \quad \text{20-30}$$

$$\varepsilon_{T_p}^g = \max_m |F_{T_p}| \quad \text{20-31}$$

Newton iterations are terminated if one of the following criteria is satisfied.

The XNORM parameter is true and:

$$\varepsilon_v \le P_{tol}^x \cdot w, \quad \text{20-32}$$

$$\varepsilon_n \le c_{tol}^x \cdot w, \quad \text{20-33}$$

$$\varepsilon_p \le P_{tol}^x \cdot w, \quad \text{20-34}$$

$$\varepsilon_{T_L} \le TL_{tol}^x \cdot w, \quad \text{20-35}$$

$$\varepsilon_{T_n} \le TC_{tol}^x \cdot w, \quad \text{20-36}$$

$$\varepsilon_{T_p} \le TL_{tol}^x \cdot w, \quad \text{20-37}$$

where $w = 1$.

The RHSNORM parameter is true, conditions for Equations 20-36 and 20-37 are satisfied for w = 1 and:

$$\varepsilon_v^g \le P_{tol}^r \cdot w_1, \quad \text{20-38}$$

$$\varepsilon_n^g \le C_{tol}^r \cdot w_1, \quad \text{20-39}$$

$$\varepsilon_p^g \le P_{tol}^r \cdot w_1, \quad \text{20-40}$$

$$\varepsilon_{T_L}^g \le TL_{tol}^r \cdot w_1, \quad \text{20-41}$$

$$\varepsilon_{T_n}^g \le TC_{tol}^r \cdot w_1, \quad \text{20-42}$$

$$\varepsilon_{T_p}^g \le TL_{tol}^r \cdot w_1, \quad \text{20-43}$$

Both the XNORM and RHSNORM parameters are true. The convergence criteria for Equations 20-36 and 20-37 for carrier temperature are fulfilled for the inequalities (Equations 20-32 and 20-38), (Equations 20-33 and 20-39), (Equations 20-34 and 20-40), (Equations 20-35 and 20-41), and one of the conditions for every pair.

If the current convergence criteria are satisfied, then condition a) or condition b) or condition c) is fulfilled for `w = w1 = WEAK`.

Convergence Criteria For Block Iteration

For the potential equation and the continuity equations, Equations 20-19 to 20-21 and Equations 20-26 to 20-28 define the X norms and RHS norms for the Newton method. With *s* as an index that denotes the number of block iterations, update errors between successive pairs of block iterations for lattice and carrier temperature are defined by the following expressions.

For lattice temperature:

$$\varepsilon_{T_L}^B = \frac{\left|(T_L)_{mmax}^{s+1} - (T_L)_{mmax}^{s}\right|}{(T_L)_{mmax}^{s+1}} \quad \text{20-44}$$

where *mmax* is the number of the node where $(TL)_m$ has its maximum value.

For electron temperature:

$$\varepsilon_{T_n}^B = \frac{\left|(T_n)_{mmax}^{s+1} - (T_n)_{mmax}^{s}\right|}{(T_n)_{mmax}^{s+1}} \quad \text{20-45}$$

For hole temperature:

$$\varepsilon_{T_p}^{B} = \frac{\left|(T_p)_{mmax}^{s+1} - (T_p)_{mmax}^{s}\right|}{(T_p)_{mmax}^{s+1}} \quad 20\text{-}46$$

Block iterations are terminated if:

$$\varepsilon_{T_L}^{B} \leq TL_{tol}^{x} \quad 20\text{-}47$$

$$\varepsilon_{T_n}^{B} \leq TC_{tol}^{x} \quad 20\text{-}48$$

$$\varepsilon_{T_p}^{B} \leq TL_{tol}^{x} \quad 20\text{-}49$$

and one of the following criteria is fulfilled:

- the XNORM parameter is true and Equations 20-32, 20-33, and 20-34 are valid for $w = 1$.
- the RHSNORM parameter is true and Equations 20-38, 20-39, and 20-40 are valid for $w_1 = 1$.
- both XNORM and RHSNORM are true and for the pairs of inequalities (Equations 20-32 and 20-38), (Equations 20-33 and 20-39), (Equations 20-34 and 20-40) one of the conditions is satisfied for each pair.
- the current convergence criteria are satisfied and condition 1, condition 2, or condition 3 is fulfilled for `w = w1 = WEAK`.

21.6 Initial Guess Strategies

Non-linear iteration starts from an initial guess. The quality of the initial guess (i.e., how close it is to the final solution) affects how quickly the solution is obtained and whether convergence is achieved. Atlas users aren't required to specify an initial guess strategy. If no strategy is defined, Atlas follows certain rules that implement a sensible, although not necessarily optimum strategy.

There is some interaction between the choice of non-linear iteration scheme and the initial guess strategy. Decoupled iteration usually converges linearly, although perhaps slowly, even from a relatively poor initial guess. Newton iteration converges much faster for a good initial guess, but fails to converge if started from a poor initial guess.

One very simple initial guess strategy is to use the most recent solution as the initial guess. Of course, there is no previous solution for the first calculation in a series of bias points. In this case, an initial solution is obtained for equilibrium conditions. There is no need to solve the current continuity equations at equilibrium, and a solution of Poisson's equation is quickly obtained.

You can also modify the initial guess in a way that makes some allowance for the new bias conditions. Typical strategies include:

- Using two previous solutions and interpolation to project a new solution at each mesh point.
- Solving a form of current continuity equation with carrier concentrations held constant. This strategy yields an improved estimate of new potential distribution.
- Modifying the majority carrier quasi-Fermi levels by the same amount as the bias changes.

You can use the parameters on the SOLVE statement to specify an initial guess strategy. Six initial guess strategies are available.

- INITIAL starts from space charge neutrality throughout the device. This choice is normally used to calculate a solution with zero applied bias.
- PREVIOUS uses the currently loaded solution as the initial guess at the next bias point. The solution is modified by setting a different applied bias at the contacts.
- PROJECTION takes two previous solutions whose bias conditions differ at one contact and extrapolates a solution for a new applied bias at that contact. This method is often used when performing a voltage ramp.
- LOCAL sets the applied bias to the specified values, and changes the majority carrier quasi-Fermi levels in heavily doped regions to be equal to the bias applied to that region. This choice is effective with Gummel iteration, particularly in reverse bias. It is less effective with Newton iteration.
- MLOCAL starts from the currently loaded solution and solves a form of the total current continuity equation that provides an improved estimate of the new potential distribution. All other quantities remain unchanged. MLOCAL is more effective than LOCAL because it provides a smooth potential distribution in the vicinity of p-n junctions. It is usually more effective than PREVIOUS because MLOCAL provides a better estimate of potential. This is especially true for highly doped contact regions and resistor-like structures.

- NOCURRENT assumes there is negligible current flow in the device and solves the non-linear Poisson equation to arrive at the initial guess. If the assumption of negligible current density is indeed correct, then the initial guess should be the solution to the current continuity equations too. You can use NOCURRENT to avoid ramping a bias to attain the solution at high bias.

When a regrid is performed, the solution is interpolated from the original grid onto a finer grid. This provides an initial guess that can be used to start the solution of the same bias point on the new grid. Although the initial guess is an interpolation of an exact solution, this type of guess does not provide particularly fast convergence.

21.6.1 Recommendations And Defaults

It is not normally required to specify any initial guess parameter.

The INITIAL parameter is normally used only to obtain a thermal equilibrium solution. The PREVIOUS, PROJECT, LOCAL, and MLOCAL parameters are used for other bias points. PREVIOUS and PROJECTION are the most frequently used. PROJECTION is normally preferred to PREVIOUS when it is available (i.e., when there are two previous solutions differing in the bias applied to the appropriate terminal). PREVIOUS is required for transient simulations, and for simulations that use current boundary conditions. LOCAL and MLOCAL tend to work well for reverse-biased devices, and are especially efficient when trying to increase very large voltage increments. By default, PROJECTION is used whenever two appropriate solutions are available. Otherwise, the PREVIOUS guess is used, unless there is no previous solution, in which case INITIAL is used.

21.7 Globally Convergent Schemes

Atlas has a vector of independent variables, x, and a vector of functions of these independent variables, $F(x)$. A solution x_s is sought such that $F(x_s)=0$. A "residual" function, $G(F(x))$, is defined. This residual is a single number that gives some measure of how close $F(x)$ is to the solution. The residual is zero at the solution, and positive everywhere else. The smaller the residual, the "closer" x is to the solution x_s.

The iterative techniques start with the Taylor expansion of $F(x)$ around some vector x_i (which is an initial guess to x_s or some previous attempt to calculate x_s).

$$F(x_i + \delta x_i) = F(x_i) + \frac{dF(x_i)}{dx} \cdot \delta x_i + \frac{d^2F(x_i)}{dx^2} \cdot \frac{\delta x_i^2}{2} + \ldots \quad \text{20-50}$$

The δx_i^2 and above terms are ignored. The new x is made the solution, $x_i+?x_i = x_s$. Therefore, $F(x_i + ?x_i) = 0$ so

$$0 = F(x_i) + \frac{dF(x_i)}{dx} \cdot \delta x_i \Rightarrow \frac{dF(x_i)}{dx} \cdot \delta x_i = -F(x_i) \Rightarrow J(x_i) \cdot \delta x_i = -F(x_i) \quad \text{20-51}$$

where $J(x_i)$ is the Jacobian. This is the equation solved during the iterative schemes to find the next value of $?x_i$.

Because the Taylor expansion was truncated after the first two terms, the function $F(x_i + ?x_i)$ is modeled as linear in x. This is a good approximation close to x_i but will probably become increasingly bad the further away from x_i we are. Specifically, at a distance $?x_i$ away from x_i the linear approximation may be invalid and therefore

$$F(x_i) + \frac{dF(x_i)}{dx} \cdot \delta x_i \neq F(x_i + \delta x_i) \quad \text{20-52}$$

This is why Atlas often has to do more than one iteration to find the vector x_s.

Sometimes, the residual $G(F(x_i + ?x_i))$ is bigger than the residual $G(F(x_i))$, which means we are jumping to a point that is further away from the solution than the current point. The globally convergent schemes try to ensure that every step taken moves us closer to the solution.

21.7.1 Overview

When

$$J(x_i) \cdot \delta x_i = -F(x_i) \quad \text{20-53}$$

is solved, we get a step ?x_i. A variable k and a function $g(k)$ are introduced so that

$$g(k) = G(F(x_i + k \cdot \delta x_i)) \quad \text{20-54}$$

thus $g(0)$ is the residual at the current point x_i, and $g(1)$ is the residual at the full Newton step x_i+?x_i. The function $g(k)$ is expected to be a continuous function over the range [0,1]. The aim of the globally convergent schemes is to find a k_i so that $g(k_i)$ is sufficiently smaller than $g(0)$ and then set the next step as

$$x_{i+1} = x_i + k_i \cdot \delta x_i \quad \text{20-55}$$

Globally convergent scheme usually have two parts. A way to decide if a particular $g(k)$ is sufficiently small and an algorithm to calculate a new value for k if the current $g(k)$ is not sufficiently small. A particular k is accepted if

$$g(k) \leq g(0) - k \cdot D \quad \text{20-56}$$

and the purpose of the second term is often to reject steps where the reduction in the residual is insignificantly small, as such D is often not very big. However, the residual is expected to decrease therefore we need D to be positive.

We cannot guarantee there is an acceptable minimum along the Newton step. If there is not, then the algorithm to calculate k would continue for ever. In order to avoid such situations, a maximum number of inner iterations needs to be specified.

There is often some numerical overhead when using a damping scheme (the function $F(x)$ may need to be calculated several times). It is not always necessary to start damping on the first Newton iteration. The number of Newton iterations to skip before we start damping can also be specified.

21.7.2 Line Search Damping

In line search damping, we want

$$g(k_i) \leq g(0) + k_i \cdot \alpha \cdot g'(0) \quad \text{20-57}$$

In this case

$$D = -\alpha \cdot g'(0) \quad \text{20-58}$$

The gradient of $g(k)$ at k=0 is negative: if an infinitesimal step along the direction ?x does not reduce the residual, then the Newton iteration solution has failed. Therefore, ? should be positive, but ? can be small, 10^{-4} by default.

In this algorithm, $g(0)$, $g'(0)$, and $g(1)$ are initially calculated. If $g(1)$ is sufficiently small, then the full Newton step is accepted. If $g(1)$ is not sufficiently small, then a quadratic is fitted through the two points and the initial gradient

$$f_2(y) = a + b \cdot y + c \cdot y^2 \Rightarrow f_2'(y) = b + 2 \cdot c \cdot y \quad \text{20-59}$$

with

$$f_2(0) = g(0), \quad f_2'(0) = g'(0), \quad f_2(1) = g(1) \quad \text{20-60}$$

The minimum in $f_2(y)$ is found and the corresponding y is chosen as the first guess, k_1. If $g(k_1)$ is sufficiently small, then this value of k is accepted. If $g(k_1)$ is not sufficiently small, there are now three points and an initial gradient. The two smallest non-zero k are denoted k_A and k_B. Initially, k_A=k_1 and k_B=1. Thus, we fit a cubic through the three points and one gradient

$$f_3(y) = a + b \cdot y + c \cdot y^2 + d \cdot y^3 \Rightarrow f_3'(y) = b + 2 \cdot c \cdot y + 3 \cdot d \cdot y^2 \quad \text{20-61}$$

with

$$f_3(0) = g(0), \quad f_3'(0) = g'(0), \quad f_3(k_A) = g(k_A), \quad f_3(k_B) = g(k_B) \quad \text{20-62}$$

The minimum in $f_3(y)$ is found and the corresponding y is chosen as the next guess, k_k. If $g(k_k)$ is sufficiently small, then this value of k is accepted. If $g(k_k)$ is not sufficiently small, then k_A and k_B are redefined to the two smallest non-zero k that have been calculated and the cubic is recalculated.

Line search damping is activated with the LS.DAMPING parameter on the **METHOD** statement and deactivated with the LSD.OFF parameter. The coefficients are specified with the following in Table 21-2.

Table 21-2 User-Specifiable Parameters for Line Search Damping

Symbol	Statement	Parameter	Default
	METHOD	LSD.SKIP.ITER	5
	METHOD	LSD.MAX.ITER	5
α	**METHOD**	LSD.ALPHA	10^{-4}

LSD.SKIP.ITER is the number of Newton iterations that occur before line search damping is used. LSD.MAX.ITER is the maximum number of line search inner iterations, during a single Newton iteration, to find a suitable value for k.

21.7.3 Bank-Rose Damping

In Bank-Rose damping, we want $g(k_i)$ to be smaller than

$$g(k_i) \leq g(0) - k_i \cdot \delta \cdot g(0) \quad \text{20-63}$$

in this case

$$D = \delta \cdot g(0) \quad \text{20-64}$$

The residual has to be positive; therefore, ? should be positive as well. If ? were bigger than unity then (for some k at least) we would only accept negative residuals (which are impossible). The default value for ? is 10^{-4}.

In this algorithm, $g(0)$ has been initially calculated. The trial k is

$$k = \frac{1}{1 + K \cdot g(0)} \quad \text{20-65}$$

If $g(k)$ is sufficiently small, then this value of k is accepted. If $g(k)$ is not sufficiently small, then K is increased which will reduce k. The initial value of K is zero, which is increased to unity after the first failed iteration. If K is greater than zero, it is increased by a factor of ten (K?$K \times 10$).

If a value of k is accepted, the corresponding value of K is decreased by a factor of ten (K?K /10) and this value is used as the initial K to damp the subsequent Newton iteration.

Bank-Rose damping is activated with the BR.DAMPING parameter on the **METHOD** statement and deactivated with the BRD.OFF parameter. The coefficients are specified with the following in Table 21-3.

Table 21-3 User-Specifiable Parameters for Bank-Rose Damping

Symbol	Statement	Parameter	Default
	METHOD	BRD.SKIP.ITER	5
	METHOD	BRD.MAX.ITER	5
K	**METHOD**	BRD.KAPPA	0.0
δ	**METHOD**	BRD.DELTA	10^{-4}

BRD.SKIP.ITER is the number of Newton iterations that occur before Bank-Rose damping is used. BRD.MAX.ITER is the maximum number of Bank-Rose inner iterations, during a single Newton iteration, to find a suitable value for k. BRD.KAPPA sets the initial value of K for use in the first Bank-Rose damping iteration.

21.7.4 Explicit Minimum Damping

In explicit minimum damping, we look for the minimum in the $g(k)$ curve. This is computationally inefficient and should be used only as a last resort.

In this algorithm, $g(0)$, $g(0.5)$, and $g(1)$ are initially calculated. The minimum is "bracketed" by calculating the residual at the mid-points and keeping the three adjacent values that contain the minimum residual (if possible the minimum is the middle point of the three).

This algorithm has no check on how good the residual is, so we bracket the minimum a fixed number of times and return the best k we have found.

Explicit minimum damping is activated with the EM.DAMPING parameter on the **METHOD** statement and deactivated with the EMD.OFF parameter. The coefficients are specified with the following in Table 21-4.

Table 21-4 User-Specifiable Parameters for Explicit Minimum Damping

Symbol	Statement	Parameter	Default
	METHOD	EMD.SKIP.ITER	5
	METHOD	EMD.MAX.ITER	5

EMD.SKIP.ITER is the number of Newton iterations that occur before explicit minimum damping is used. EMD.MAX.ITER is the number of explicit minimum inner iterations, during a single Newton iteration, to find a suitable value for k. Because explicit minimum is looking for the actual minimum residual, it always does this for many inner iterations.

21.8 The DC Curve-Tracer Algorithm

Tracing I-V curves for complicated device phenomena, such as breakdown or latchup, can be very difficult using conventional methods. Atlas includes a special purpose DC curve-tracing algorithm that overcomes these problems. This algorithm is based on a dynamic load-line technique that adapts the boundary conditions for each step. The approach implemented in Atlas for curve tracing is based on the work describe in [104].

The key idea is that bias conditions can evolve smoothly between the limits of pure voltage control and pure current control. This is achieved using external resistors that adapt dynamically to the shape of the I-V curves to ensure that at each point the load line is perpendicular to the local tangent of the trace. With this value for the external resistor, the solution is projected to the next operating point by stepping the external voltage. Once the solution has converged, a new external resistance is calculated based on the new tangent information and the process repeats itself.

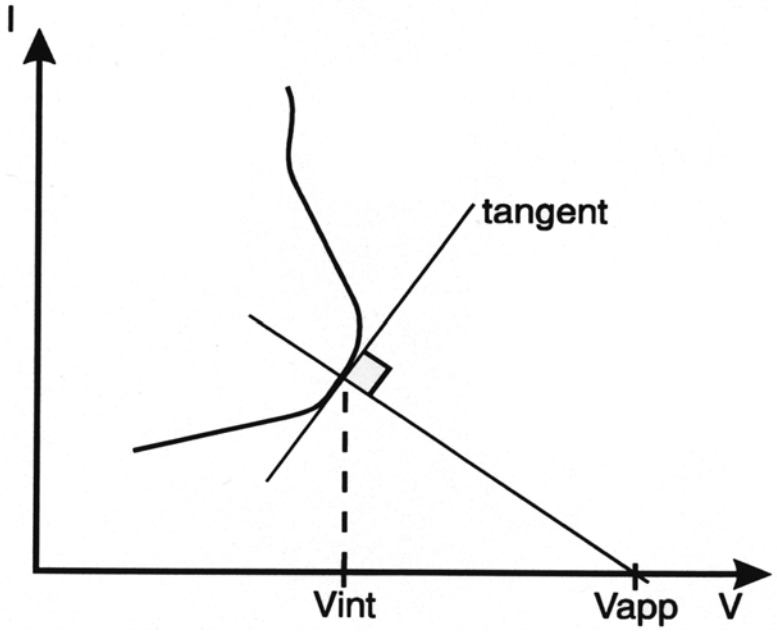


Figure 21-1: Load the algorithm used in the Curve Tracer

The curve tracing capability is activated by specifying the CURVETRACE parameter in the **SOLVE** statement. Prior to this, the **CURVETRACE** statement is used to set the parameters for the tracing algorithm. These parameters are the name CONTR.NAME of the ramped electrode (which will be referred to as a control electrode), the initial voltage increment STEP.INIT, the upper bound of the tracing curve, and additional parameters if they differ from the default values. The Upper bound parameter, END.VAL, is used to stop tracing. If the VOLT.CONT parameter is specified, END.VAL is a voltage. If the CURR.CONT parameter is specified, END.VAL is a current.

The applied voltage at each step is altered in accordance with the slope of the I-V curve. The resistor between the applied voltage and the semiconductor is also changed dynamically to ensure the voltage at the semiconductor (VINT) is smoothly varied along the I-V curve. If STEP.CONT is specified, the number of operational points on a trace will not exceed specified parameter STEPS.

21.9 Transient Simulation

When transient simulation is performed, the carrier continuity equations are integrated in the time domain. Time integration schemes differ in their accuracy, in the number of previous time levels they employ, and in their stability properties.

Accuracy is usually referred to as being “nth order”, where n is usually an integer between 1 and 4. In the limit of a small timestep, the magnitude of the local truncation error (LTE) introduced by the time integration scheme is proportional to the nth power of the timestep. Schemes that require the storage of solutions at timesteps previous to the most recent one are unattractive due to storage requirements. Single step integration schemes that use the solution at only one previous time level have a maximum order of 2.

The continuity equations are “stiff” (i.e., they are impacted by phenomena with very short timescales). Stiffness imposes stringent requirements on the stability of time integration schemes. Two forms of stability, A-stability and L-stability, are important. A-stability guarantees that errors introduced at one time step will not increase at the next timestep. L-stability guarantees that errors will decay even for large time step values. A-stability is a requirement for any practical scheme. L-stability is extremely desirable to avoid non-physical “ringing”.

Most device simulation codes use a simple first-order (implicit) backward difference formula for time integration [97,73,172]. This scheme, which is known as BDF1, is both A-stable and L-stable. Unfortunately, the scheme is inaccurate for typical timesteps of interest. Second order accuracy is obtained using the trapezoidal rule (TR) for time integration. This scheme is A-stable, but it is not L-stable. This means that solutions exhibit non-physical “ringing”, unless very small timesteps (much smaller than those dictated by LTE considerations) are used. The BDF2 scheme is second order, and is both A-stable and L-stable. The scheme, however, uses solutions from two previous time levels and is less accurate than TR.

For drift-diffusion calculations, Atlas uses a composite TR-BDF2 scheme that was developed by Bank et. al.[21]. This method is one-step, second order, and both A-stable and L-stable. An estimate of the LTE is obtained at each timestep. This estimate is also used to automatically adapt the timestep.

Different schemes are available for transient solutions that include solving for lattice temperature or carrier temperatures or both. In Atlas2D, the full TR-BDF2 is available as either full Newton or a block iteration scheme. To enable the block iteration scheme, use the BLOCK.TRAN keyword on the METHOD statement. In Atlas3D, the TR-BDF2 is available as a full Newton scheme but the block iteration is unavailable. In Atlas3D, the absolutely stable half-implicit scheme [249] is available (specify HALFIMPLICIT on the **METHOD** statement). It is only available if a lattice temperature solution is requested. It is not for carrier temperature solutions. Automatic timestep selection with local error control is implemented in this case. You can specify tolerance using the TOL.TIME parameter in the **METHOD** statement.

Note: You normally specify only the initial timestep with the TSTEP parameter of the **SOLVE** statement. After this, time steps are derived from the LTE and will typically increase.

21.10 Small Signal and Large Signal Analysis

There are several ways to predict the small-signal and large-signal high-frequency properties of semiconductor devices (review of these different techniques was presented by Laux [175]). Frequency domain perturbation analysis can be used to determine the small-signal characteristics. Fourier analysis of transient responses, however, can be used for both small-signal and large signal response.

21.10.1 Frequency Domain Perturbation Analysis

Frequency-domain perturbation analysis of a DC solution can be used to calculate small-signal characteristics at any user-specified frequency [175]. This scheme works for both 2D and 3D. The calculation proceeds in the following manner:

1. Variables are represented as the sum of the known DC component and a small unknown sinusoidal AC component.
2. All equations are expanded.
3. Differentiation in time becomes multiplication by the value of ω ($\omega = 2\pi$ frequency).
4. Products of AC quantities are neglected since they are small with respect to other quantities
5. The DC solution is subtracted.

What remains is a complex linear system whose unknowns are the AC components of the solution. Solving this linear system with appropriate boundary conditions yields small-signal characteristics.

The coefficient matrix of the complex linear system is simply the Jacobian associated with the DC operating point with some terms on the leading diagonal supplemented by $j\omega$. The Jacobian is available 'for free' if the DC solution was calculated using Newton iteration. This is a very attractive feature of the Newton method. The resulting linear complex system can be solved by an iterative or a direct method. A form of successive over relaxation (SOR) works well for frequencies significantly below the cutoff frequency but fails at higher frequencies. More sophisticated iterative techniques can be used at higher frequencies. It is simpler and more reliable, however, to switch to direct sparse matrix methods if SOR fails to converge.

Frequency domain perturbation analysis is extremely attractive when the full Newton method is used to calculate a DC solution. The method works for all frequencies. It requires a predictable amount of computation, which is quite low with respect to other calculations, since only a single linear system is solved.

Frequency domain perturbation analysis is invoked in Atlas by specifying the appropriate parameters in the **SOLVE** statement. The AC.ANALYSIS parameter specifies that this analysis is to be performed. TERMINAL specifies the contact whose terminal voltage is to be perturbed. A full characterization requires that all but one of the device terminals is perturbed. FREQUENCY, FSTEP, and NFSTEPS determine the frequencies for which solutions are obtained. FREQUENCY specifies the initial frequency. FSTEP and NFSTEPS define a loop on frequencies. If MULT.FREQ is specified, the frequency is multiplied by FSTEP at each increment. This is useful for characterizing the small-signal response over several decades of frequency. The solution method can be specified using the SOR, DIRECT, and AUTO parameters. AUTO starts out using SOR but switches to DIRECT if convergence problems are encountered.

You can specify the relaxation parameter by using the S.OMEGA parameter on the **SOLVE** statement. Its default value is unity, which usually suffices. If any of the contacts in the device has a lumped element, however, then the SOR method will probably fail to converge unless S.OMEGA is set to a value of less than 1. A value of 0.75 is recommended on a first attempt. For higher frequencies, you need to use the DIRECT method.

To work properly small signal analysis requires a complete Jacobian. Some models may be non-local, which means they may produce questionable small signal results.

The INCOMPLETE model although "local" lacks rate expressions for carriers. This can also lead to questionable results, such as currents that do not balance.

21.10.2 Fourier Analysis Of Transient Responses

Frequency-domain perturbation analysis is a post-processing step that must be performed on a LOG file, which contains transient data. The **FOURIER** statement performs a Fast Fourier Transform (FFT) on the time domain data transforming it into the frequency domain. Table 21-5 shows the syntax of the **FOURIER** statement.

Table 21-5 User-Specifiable Parameters for Fourier Statement

Statement	Parameter	Default
FOURIER	INFILE	
FOURIER	OUTFILE	
FOURIER	T.START	
FOURIER	T.STOP	
FOURIER	FUNDAMENTAL	
FOURIER	MAX.HARMONIC	
FOURIER	NUM.SAMPLES	64
FOURIER	INTERPOLATE	FALSE
FOURIER	COMPLEX.VALUES	FALSE

The explanation for the **FOURIER** parameters are as follows:

- INFILE – input log file. This should contain data from a transient simulation.
- OUTFILE – file output file for the Fourier transform.

The following parameters are optional:

- T.START – start of time data to be used for the FFT. The default value is the first time point in the input log file.
- T.STOP – end of time data to be used for the FFT. The default value is the last time point in the input log file.
- FUNDAMENTAL – fundamental frequency. If this is not specified, then the fundamental frequency is set to 1/(T.STOP - T.START). If the fundamental frequency is specified then T.STOP is set to T.START + 1/FUNDAMENTAL.

- MAX.HARMONIC – maximum harmonic frequency that the FFT should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.
- NUM.SAMPLES – number of samples. This should be an integer power of 2 (i.e., 2^n) where n is a positive integer. The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case, the number of samples is set to the nearest integer power of 2, which will generate this frequency.
- INTERPOLATE – performs linear interpolation on input data with non-uniform timesteps. This interpolates the data on to uniform timesteps. Interpolation of data can introduce addition (inaccurate) harmonic values into the FFT, which would not occur if uniform time is taken. INTERPOLATE must be used if the log file contains non-uniform time steps.
- COMPLEX.VALUES – prints the real and imaginary components to file as well as the magnitude and phase.

Note: FFT works best with uniform time steps. Therefore, set DT.MIN and DT.MAX on the **METHOD** statement to the same value. The time step should be set to: time step = 1/(number of samples * fundamental frequency).

The FFT can then calculate harmonic frequencies up to:

```
(number of samples/2 - 1)*fundamental frequency.
```

You can obtain small-signal data at high frequencies by calculating the terminal current responses to terminal voltage perturbations. Then, Fourier analysis will be performed on currents and voltages. Their ratio at each frequency provides admittance data for that frequency. The voltage perturbations are normally selected to have an analytic form with a known Fourier transform. Care must be taken to self-consistently account for geometric capacitances when step function voltage perturbations are used. The advantages of this technique are that it can be used whenever transient calculations are possible. Each transient solution gives information over a broad range of frequencies. The main disadvantage is that transients become very long when low frequency effects are investigated.

21.10.3 Overall Recommendations

Use frequency-domain perturbation analysis when it is available. The method works for arbitrary frequencies and does not require transient calculations. Use Fourier analysis of transient responses for high frequencies when it is unavailable but transient calculations are possible.

21.11 Numerical Precision

The convergence and accuracy of a device simulation requires the numerical resolution of the quantities $(n-p)$ and $(n+p)$. The first of these quantities appears in the space charge density, which is used in Equations 3-1 to calculate the electrostatic potential, while the second is used implicitly in the calculation of current conservation.

In order to resolve these terms, the calculations must maintain at least

$$P \sim \frac{|\ln(n) - \ln(p)|}{\ln(2)} \qquad 21\text{-}66$$

significant bits of arithmetic precision. From the theory of carrier statistics we can then estimate that for a convergent and accurate simulation, we should have

$$P \geq \frac{1}{\ln(2)}\left(\frac{E_G}{kT_L}\right). \qquad 21\text{-}67$$

Therefore, the required precision P depends on both the lattice temperature and on the bandgap. In practice, Inequality 21-67 constitutes a loose upper bound, and we can frequently carry out a successful simulation with somewhat less precision than the relation indicates. This inequality also tells us that precision is more likely to be an issue with high-bandgap materials, and at low temperatures.

By default, Atlas uses a nominal arithmetic precision of 64 bits (i.e., double precision). In order to support the simulation of high-bandgap materials in particular, Atlas can now run at several levels of extended precision as well. The level of precision is selected by means of a command-line flag (which in DeckBuild would be passed as one of the arguments to the SIMFLAGS parameter of the **GO** statement). The arithmetic precision levels currently supported by Atlas are summarized in Table 21-6.

Table 21-6 Arithmetic Precision Levels in Atlas

Command-line flag	**(None)**	-80	-128	-160	-256
Nominal precision P_{nom} (bits)	64	80	128	160	256
Significant bits	53	64	104	128	209
Type size (bytes)	8	16	16	32	32

Floating-point numbers are composed of three parts: the sign, the exponent, and the significand [215]. As its name suggests, the size of the significand corresponds to the number of significant bits in the number. When selecting a precision level for a simulation, you should check Table 21-6 to verify that the number of significant bits is at least as large as the required precision P given by Inequality 21-67.

The nominal precision of a floating point number is larger than the number of significant bits, because it also counts the bits used for the sign and the exponent. In Atlas, the various flags controlling the precision levels of the simulator and of the solver always refer to the nominal precision.

For well-converged solutions, the run-time increases with the precision. The increase is especially significant at the highest precision levels. Memory requirements also increase with the precision; relative requirements are suggested by the type sizes listed in Table 21-6. On the other hand, certain simulations (e.g., of SiC devices) that have difficulty converging at the lower precision levels are likely to run *faster* if the precision level is increased. Accordingly, the optimum precision level for a particular problem is likely to be the minimum level that yields good convergence during the solution. Based on Table 21-6 and Inequality 21-67, Table 21-7 provides a guide showing the nominal precision that should be required to simulate common semiconducting materials at $T_L = 300\ \mathrm{K}$.

Table 21-7 Required Precision for Common Semiconductors

Material	E_G (eV)	P_{nom} (bits)	Material	E_G (eV)	P_{nom} (bits)
InSb	0.17	64	ScN	2.15	160
SnTe	0.18	64	AlAs	2.16	160
PbSe	0.26	64	3c-SiC	2.2	160
PbTe	0.29	64	GaP	2.27	160
InAs	0.36	64	ZnTe	2.28	160
PbS	0.37	64	(160-bit Limit)	2.294	160
Ge	0.66	64	AlP	2.43	256
GaSb	0.75	64	CdS	2.48	256
(64-bit Limit)	0.950	64	HgS	2.5	256
Si	1.08	80	BeTe	2.57	256
(80-bit Limit)	1.147	80	ZnSe	2.71	256
InP	1.35	128	6h-SiC	2.9	256
GaAs	1.424	128	4h-SiC	3.26	256
CdTe	1.43	128	ZnO	3.3	256
AlSb	1.63	128	GaN	3.4	256
CdSe	1.75	128	(256-bit Limit)	3.745	256
(128-bit Limit)	1.864	128	C (Diamond)	5.45	(Excessive)
			AlN	6.2	(Excessive)

Table 21-7 lists 80 bits as the nominal precision required to simulate silicon devices at 300 K. From long experience, we know that in most cases a precision of 64 bits is sufficient. However, the full 80 bits may be required for devices — such as bipolar junction transistors and TFT's — where the minority carrier current plays an important functional role.

21.12 Differences Between 2D and 3D Numerics

With respect to numerical techniques, there are several differences between 2D and 3D simulations.

First, with respect to the nonlinear iteration strategies, all three strategies, NEWTON, GUMMEL and BLOCK are supported in 2D simulation, whereas, only NEWTON and GUMMEL are supported for 3D simulations. Implementation of the BLOCK strategy is expected in a future release.

Second, solution of the linear subproblem is handled differently for 2D and 3D simulations. As previously noted, the computational burden of solving the linear subproblem increases with the size of the solution domain. For smaller problems, direct methods are quicker while for larger problems iterative methods are preferred. It turns out that the point at which the iterative methods become less burdensome roughly coincides with the transition between 2D and 3D domains. As such, the default method for 2D simulations is a direct solver. For 3D simulations, the default method is an iterative solver. By default, ILUCGS is applied to 3D simulations. ILUCGS is an acronym for incomplete lower upper decomposition conjugate gradient squared. Two alternative iterative solvers are also available for 3D simulations. BICGST (BICGST on the **METHOD** statement) is an acronym for biconjugate gradient squared stabilized. GMRES (GMRES on the **METHOD** statement) is an acronym for generalized minimum residual. Direct methods can be used for 3D simulation by specifying DIRECT on the **METHOD** statement. Practical experience shows that for 3D simulations either of the iterative methods are faster than the direct method. But in some cases, the accuracy produced by the iterative methods can prevent convergence in the nonlinear outer loop. For 2D simulations, only direct methods are supported.

Chapter 22
Statements

22.1 Input Language

This chapter contains a complete description (in alphabetical order) of every statement and parameter used by any of the Atlas products, except for MixedMode. MixedMode specific parameters are described in Chapter 13 “MixedMode: Mixed Circuit and Device Simulator”. MC Device specific parameters are described in Chapter 20 “MC Device”. Descriptions for the statements in this chapter are as follows:

- The statement name.
- The product for which the statement is applicable.
- The syntax of the statement.
- A list of all statement parameters, their type, default value, and units.
- A description of each parameter.
- An example of the correct usage of each statement.

Note: An error message will be generated if you attempt to specify a statement for a simulator that you haven’t purchased. For example, the **BEAM** statement can only be used if you have purchased Luminous or Luminous 3D.

22.1.1 Syntax Rules

An input deck line is referred to as a statement (or statement line). Since statements and parameters are not case sensitive, they can be entered using either uppercase or lowercase letters.

A statement is specified in the general format:

```
<STATEMENT> <PARAMETER>=<VALUE>
```

where `STATEMENT` is the statement name, `PARAMETER` is the parameter name, and `VALUE` is the parameter value. The space character is used to separate the parameters in a statement.

The words and numbers, which follow a statement, are parameters of that statement. A `word` is an alphanumeric string, which is terminated either by a space or by a carriage return. A `number` is a numeric or alphanumeric string which is terminated either by a space or by a carriage return. Numerical values may range from 10^{-38} to 10^{38}. A number may contain the symbols + (positive), - (negative), and/or E (decimal notation). For example:

```
+10 -1.234E5 .003 -.12E+10
```

Four types of parameters are used by the Atlas products. These are: real, integer, logical, and character. Table 22-1 explains these parameter types.

Table 22-1 Types of Parameters			
Parameter	**Description**	**Value Required**	**Example**
Character	Any character string	Yes	INFILE=NMOS.DOP
Integer	Any whole number	Yes	REGION=2
Logical	A true or false condition	No	GAUSSIAN
Real	Any real number	Yes	X.MIN=0.52

Any parameter that doesn't have a logical value must be specified in the form: PARAM=VAL, where PARAM is the name of the parameter, and VAL is the value of the parameter. Logical parameters must be separated from other parameters or commands by a space.

For example, in the statement:

```
DOPING UNIFORM CONCENTRATION=1E16 P.TYPE
```

the UNIFORM and P.TYPE parameters have logical values and the CONCENTRATION parameter has a value of $1x10^{16}$ (real).

Logical parameters can be turned off (switched from true to false) by placing a caret (^) in front of the logical parameter. For example, in the statement:

```
DOPING UNIFORM CONCENTRATION=1E16 ^P.TYPE
```

the P.TYPE parameter has been set to false.

Mnemonics

It is not always necessary to input the entire statement or parameter name. Atlas only requires that you input enough letters to distinguish that command or parameter from other commands or parameters. For example, DOP may be used to abbreviate the **DOPING** command. Excessive truncation is not recommended, since future Atlas syntax might make short abbreviations become ambiguous.

Continuation Lines

Since it may be necessary for a statement line to contain more than 256 characters, Atlas allows you to specify continuation lines. To continue a line, put a backslash (\) character at the end of the line that is to be continued. When Atlas encounters the backslash, it will interpret the next line to be a continuation of the current line. The Pisces-II continuation of using a (+) at the start of the subsequent line is not supported in Atlas.

Comments

Comments are indicated either by **COMMENT** command, or by a pound sign (#). All characters on a line, which follow a comment indicator (**COMMENT** or #) will not be analyzed by Atlas.

Synonyms

Some parameters have synonyms. These are parameters that have a different name but the same functionality. A parameter's synonym is listed in the parameter descriptions of the statements.

Pseudonyms

Throughout the statement descriptions, pseudonyms are used either to indicate a group of parameters or to indicate the value of a particular parameter. A < symbol indicates the start of a pseudonym(s). A > symbol indicates the end of a pseudonym(s). Pseudonyms will be separated from one another by a space character ( ). For example, `LOC` might indicate a group of location parameters, and `FILENAME` might indicate the name of a file that must be specified.

Symbols

The following symbols are used in the statement descriptions:

- < Indicates the start of a list of pseudonyms.
- > Indicates the end of a list of pseudonyms.
- | Separates parameters or pseudonyms which are mutually exclusive. Only one of these parameters may be used in a statement.
- [Indicates the start of an optional command, parameter, or pseudonym.
-] Indicates the end of an optional command, parameter, or pseudonym.

Expressions

Atlas does not support arithmetic expressions in the syntax. You can, however, evaluate and use expressions by using the **SET** or **EXTRACT** statements.

22.2 A.MESH, R.MESH, X.MESH, Y.MESH, Z.MESH

<n>**.MESH** specifies the location of grid lines along the <n>-axis in a rectangular mesh for 2D or 3D simulation. It also specifies radial and angular components for 3D cylindrical meshing (see Section 2.6.9 “Specifying 3D Cylindrical Structures”).

Note: The commands are equivalent in the X, Y, or Z directions.

Syntax

```
X.MESH LOCATION=<l> (NODE=<n> [RATIO=<r>])| SPACING=<v>
```

Parameter	Type	Default	Units
DEPTH	Real		μm
H1	Real		μm
H2	Real		μm
H3	Real		μm
LOCATION	Real		μm
N.SPACES	Integer		
NODE	Integer		
RATIO	Real	1	
SPACING	Real		μm (degrees)
WIDTH	Real		μm
X.MAX	Real		μm
X.MIN	Real	0.0	μm
Y.MAX	Real		μm
Y.MIN	Real	0.0	μm

Description

DEPTH	Specifies the extent of a mesh section in the Y direction.
H1	Specifies the spacing between consecutive mesh lines at the beginning of a mesh section.
H2	Specifies the spacing between consecutive mesh lines at the end of a mesh section.
H3	Specifies the spacing between consecutive mesh lines at the middle of a mesh section.
LOCATION	Specifies the location of the grid line.

N.SPACES	Specifies the number of mesh spacings within a given mesh section.
NODE	Specifies mesh line index. There is a limit of 120 mesh lines. These mesh lines must be assigned in increasing order.
RATIO	Specifies the ratio to use when interpolating grid lines between given locations. Spacing between adjacent grid lines will increase or decrease by the factor assigned to the RATIO parameter. A RATIO value of between 0.667 and 1.5 is recommended. RATIO should not be used if SPACING is specified.
SPACING	Specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, the NODE and RATIO parameters should not be specified. If the SPACING parameter is used to specify mesh spacings, the NX, NY, and NZ parameters of the **MESH** statement should not be specified. When the mesh spacings are specified using the SPACING parameter, the mesh size will be calculated.
WIDTH	Specifies the extent of a mesh section in the X direction.
X.MAX/Y.MAX	Specify the location of the end of a given mesh section (should not be specified if DEPTH/WIDTH is specified).
X.MIN/Y.MIN	Specify the location of the beginning of a given mesh section (should only be specified for the first section).

Setting Fine Grid at A Junction Example

This example shows how to space grid lines closely around a junction at y=0.85 microns.

```
Y.MESH  LOC=0.0  SPAC=0.2
Y.MESH  LOC=0.85 SPAC=0.01
Y.MESH  LOC=2    SPAC=0.35
```

22.3 BEAM

BEAM specifies an optical input signal in the form of a collimated beam of light. This statement is used with Luminous or Luminous 3D.

Syntax

```
BEAM <parameters>
```

Parameter	Type	Default	Units
ABS.EDGE	Logical	False	
AM0	Logical	False	
AM1.5	Logical	False	
AMBIENT.INDEX	Real	1.0	
ANLE.RES	Real	5.0	Degrees
ANGLE	Real	90.0	Degrees
ANISO	Logical	False	
BACK.FACET	Logical	False	
BIG.INDEX	Logical	False	
BOTTOM.FACET	Logical	False	
BACK.REFL	Logical	False	
BPM	Logical	False	
CIRCULAR	Logical	False	
CON.REFLECT	Logical	False	
COSINE	Logical	True	
CURR.AMPL	Real	1.0	
DEVICE	Character		
DIEL.FUNC	Logical	False	
DIFFUSIVE	Logical	False	
DIPOL.AMPL	Logical	1.0	See description
DIPOL.ELEC	Logical	True	
DIPOL.MAGN	Logical	False	
DT	Real	0.0	sec.
DX.PHOTOGEN	Real		μm
DY.PHOTOGEN	Real		μm

Parameter	Type	Default	Units
E.END	Real	0	eV
E.START	Real	0	eV
E.NUMBER	Integer	1	
ELLIPTICAL	Logical	False	
ERRCTR	Integer	(number of samples in a wavelength)	
EXTEND.ALL	Logical	False	
EXTEND.BACK	Logical	False	
EXTEND.BOTTOM	Logical	False	
EXTEND.FRONT	Logical	False	
EXTEND.LEFT	Logical	False	
EXTEND.RIGHT	Logical	False	
EXTEND.TOP	Logical	False	
FDTD.LUM	Logical	False	
F.IMAGE	Character		
F.INTENSITY	Character		
F.RADIATE	Character		
F.REFLECT	Character		
F3.RADIATE	Character		
FACET	Logical	False	
FDTD	Logical	False	
FDTD.AUTO	Logical	False	
FDTD.AUTO.MIN	Integer	2	
FDTD.LUM	Logical	False	
FILE.PHOTOGEN	Character		
FRONT.FACET	Logical	False	
FRONT.REFL	Logical	False	
GAUSSIAN	Logical	False	
HORIZONTAL	Real	-1.0	

Parameter	Type	Default	Units
INPUTRAYS	Character		
INSTATE	Character		
INTEGRATE	Logical	True	
ITERATION	Integer	0	
LEFT.FACET	Logical	False	
LENS.HEIGHT	Real	0.0	µm
LENS.INDEX	Real	1.0	
LENS.PLANE	Real	0.0	µm
LENS.RADIUS	Real	0.0	µm
LENS.WIDTH	Real		µm
LENS.X	Real	0.0	µm
LENS.XMAX	Real	0.0	µm
LENS.XMIN	Real	0.0	µm
LENS.XSAGS	Character		
LENS.Y	Real	0.0	µm
LENS.Z	Real	0.0	µm
LENS.ZMAX	Real	0.0	µm
LENS.ZMIN	Real	0.0	µm
LENS.ZSAGS	Character		
LID	Logical	False	
LONGIT.STEP	Real	0.0	
MAX.WINDOW	Real	1.0×10^{20}	µm
MC.ARES	Real	0.001	
MC.SEED	Integer	-10	
MONTE.CARLO	Logical	False	
MEAN	Character	0.0	µm
MEMORIZE	Character		
MESHONLY	Logical	False	
METAL.REFLECT	Real	False	

Parameter	Type	Default	Units
MIN.POWER	Real	0.0	W/cm^2
MIN.WINDOW	Real	-1.0×10^{20}	μm
NORMALIZE	Logical	False	
NO.ENV	Logical	False	
NOZLENS	Logical	False	
NSAMP	Integer	25	
NUMBER	Integer	1	
NX	Integer	10	
NY	Integer	10	
NZ	Integer	10	
N.VS.X	Logical	False	
N.VS.Y	Logical	False	
N.VS.Z	Logical	False	
OUT.BACK	Logical	False	
OUT.BOTTOM	Logical	False	
OUT.FRONT	Logical	False	
OUT.LEFT	Logical	False	
OUT.RIGHT	Logical	False	
OUT.TOP	Logical	False	
OUT.POWER	Character		
OUT.SAG	Character		
OUT.XSAG	Character		
OUT.ZSAG	Character		
OUTFILE	Character		
OUTSTATE	Character		
PACKET	Logical	False	
PARALLEL	Logical	True	
PERIODIC	Logical	False	
PHASE	Real	0.0	Degrees

Parameter	Type	Default	Units
PHI	Real	90.0	Degrees
PLANE	Logical	True	
PML.CROP	Logical	False	
POINT	Logical	False	
POLAR.PHASE	Real	0.0	Degrees
POLARIZE	Real	0.0	Degrees
POWER.FILE	Character		
POWER.SCAL	Real	1.0	
PROP.LENG	Real	0.0	μm
PULSE	Logical	False	
QUANTUM.EFF	Real	1.0	
RAY.CHECK	Logical	False	
RAYAREA	Real	0.0	μm(2D) μm^2(3D)
RAYS	Integer	1	
RAYTRACE	Character		
REFLECTS	Integer	0	
REL.POWER	Real	1.0	
REMEMBER	Character		
RIGHT.FACET	Logical	False	
S.BACK	Logical	False	
S.BOTTOM	Logical	False	
SCAT.BACK	Real	0.0	μm
SCAT.BOTTOM	Real	0.0	μm
SCAT.FRONT	Real	0.0	μm
SCAT.LEFT	Real	0.0	μm
SCAT.LENGTH	Real	0.0	μm
SCAT.RIGHT	Real	0.0	μm
SCAT.TOP	Real	0.0	μm

Parameter	Type	Default	Units
S.FRONT	Logical	False	
S.LEFT	Logical	False	
S.RIGHT	Logical	False	
S.TOP	Logical	False	
SAG.ITS	Integer	10	
SEG.LOG	Integer	0	
SEG.STR	Integer	0	
SEG.NFP	Integer	0	
SEG.FFP	Integer	0	
SINE	Logical	False	
STR.DECI	Integer	16	
STR.PRECIS	Integer	16	
SIGMA	Real	0.0	μm
STRUCTURE	Character		
SUB.EXTINCT	Real	0.0	
SUB.INDEX	Real	1.0	
SUBDX	Real		μm
SUBDY	Real		μm
SUBDZ	Real		μm
SUBNX	Integer		
SUBNY	Integer		
SUBNZ	Integer		
SUBSTRATE	Logical	False	
SX	Real	0.0	μm
SY	Real	0.0	μm
SZ	Real	0.0	μm
TAUC.DOS	Logical	False	
TBLACK	Real	0.0 (off)	K
TD.END	Logical	False	

Parameter	Type	Default	Units
TD.ERRMAX	Real	0.0	
TD.EVERY	Integer	10	
TD.FILE	Character		
TD.HARD	Logical	False	
TD.LIMIT	Integer		
TD.LOG	Character		
TD.MANY	Integer		
TD.ONE	Logical	False	
TD.PER	Integer		
TD.SOFT	Logical	False	
TD.SRATE	Integer	20	
TD.TFSF	Logical	False	
TD.WAVES	Integer		
TD.WIDTH	Real	0.0	s
TD.ZCUT	Real	0.0	μm
TE	Logical	False	
TFSF	Logical	True	
THETA	Real	0.0	Degrees
THINEST	Real	0.0	μm
TIMING	Logical	False	
TM	Logical	True	
TRANSV.STEP	Real	0.0	
TR.MATRIX	Logical	False	
TOP.FACET	Logical	False	
USER.SPECTRUM	Logical	False	
VERBOSE	Logical	False	
WAVE.SAMP	Integer	0	
WAVELENGTH	Real	0.623	μm
WAVEL.END	Real	0.0	μm

Parameter	Type	Default	Units
WAVEL.NUM	Integer	1	
WAVEL.SCAL	Real	1.0	
WAVEL.START	Real	0.0	µm
X0.PHOTOGEN	Real		µm
X1	Real	0.0	µm
X2	Real	0.0	µm
X.CENTER	Real	0.0	µm
X.FACET	Real		µm
X.GAUSSIAN	Logical	False	
X.MAX.CROP	Real		µm
X.MEAN	Real	0.0	µm
X.MIN.CROP	Real		µm
X.MIRROR	Logical	False	
X.ORIGIN	Real	0.0	µm
X.PERIODIC	Logical	False	
XPERIODS	Real	0.0	
X.PITCH	Real	0.0	
X.POINT	Real	0.0	µm
X.RADIUS	Real	0.0	µm
X.SEMIAXIS	Real	0.0	µm
X.SIGMA	Real	0.0	µm
XCENTER	Real	0.0	µm
Y.FACET	Real		0.0
XGAUSSIAN	Logical	False	
XMAX	Real	1.0×10^{20}	µm
XMEAN	Real	0.0	µm
XMIN	Real	-1.0×10^{20}	µm
XRADIUS	Real	0.0	µm
XSIGMA	Real	0.0	µm

Parameter	Type	Default	Units
Y.MAX.CROP	Real		µm
Y.MIN.CROP	Real		µm
Y0.PHOTOGEN	Real		µm
Y.ORIGIN	Real	0.0	µm
Y.POINT	Real	0.0	µm
Y.SEMIAXIS	Real	0.0	µm
Z1	Real	0.0	µm
Z2	Real	0.0	µm
Z.CENTER	Real	0.0	µm
Z.GAUSSIAN	Logical	False	
Z.MAX.CROP	Real		µm
Z.MEAN	Real	0.0	µm
Z.MIN.CROP	Real		µm
Z.ORIGIN	Real	0.0	µm
Z.PERIODIC	Logical	False	
Z.POINT	Real		µm
Z.RADIUS	Real	0.0	µm
Z.SEMIAXIS	Real	0.0	µm
Z.SIGMA	Real	0.0	µm
ZCENTER	Real	0.0	µm
Z.FACET	Real		µm
ZGAUSSIAN	Logical	False	
ZMEAN	Real	0.0	µm
ZMAX	Real	1.0×10^{20}	µm
ZMIN	Real	-1.0×10^{20}	µm
Z.MIRROR	Logical	False	
ZRADIUS	Real	0.0	µm
ZSIGMA	Real	0.0	µm

Description

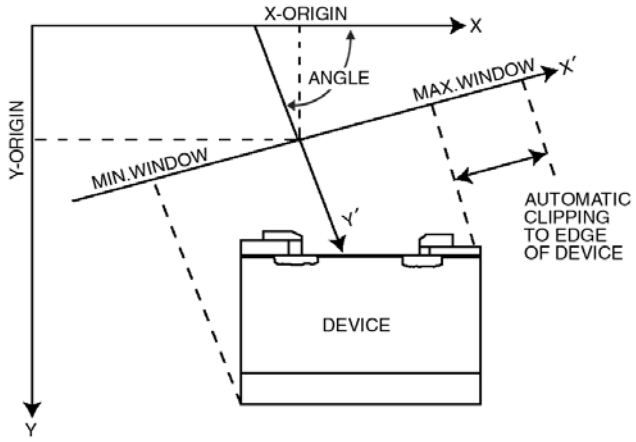


Figure 22-1: Luminous Optical Source Coordinate System

ABS.EDGE	Ensures that for energies (wavelengths) less than the bandgap of the material in question the imaginary part of the complext index is zero. By default, this parameter is FALSE.
AM0	Enables the built-in AM0 solar spectrum.
AM1.5	Enables the built-in AM1.5 solar spectrum.
AMBIENT.INDEX	Specifies the index of refraction of domain outside of all meshed regions.
ANLE.RES	Specifies the angular resolution in computing the far-field pattern.
ANGLE	This is the angle of propogation of the optical beam (see Figure 22-1). ANGLE=90 is vertical illumination from the top of the device. The synonym for this parameter is PHI.
ANISO	Signifies that the anisotropic solver should be applied to the problem. This incurs a roughly 2× increase in computation time.
BACK.REFL	Specifies that back side reflections are to be taken into account. When BACK.REFL is specified, the area outside the device domain is assumed to be a vacuum (i.e., n = 1.0, k = 0.0).
BPM (Luminous)	Use the Beam Propagation Method instead of Ray-Tracing for analysis of light propagation in the device. Use BPM when diffraction of light or coherent effects are important (see Section 11.4 "Beam Propagation Method in 2D"). **BEAM** parameters related specifically to ray tracing (such as RAYTRACE, RAYS, THINEST, MAX.WINDOW, MIN.WINDOW) are ignored when BPM is specified.
CIRCULAR	This is the synonym for **ELLIPTICAL**.
CURR.AMPL	Specifies the current density to be injected into the device. For an electric dipole, the unit is A.um^-2 and for magnetic dipole, the unit is V.um^-2.
CON.REFLECT	Specifies that all conductors are to be treated as perfect reflectors.

DEVICE	Specifies the name of a device in MixedMode to identify to which device the beam is directed. The synonym for this parameter is **STRUCTURE**.
DIEL.FUNC	Enables the plotting of the dielectric function for the Tauc-Lorentz Model (see Section 3.10.3 "Tauc-Lorentz Dielectric Function with Optional Urbach Tail Model for Complex Index of Refraction"). To plot the model, you should also specify the energy range using the LAM1 and LAM2 parameters of the **MATERIAL** statement with the number of samples, NLAM and the output file name OUT.INDEX.
DIFFUSIVE	Enables the diffusive transfer matrix method. This method permits modeling of both coherent (specular) and incoherent (diffusive) interactions at material interfaces (see Section 11.3.4 "Transfer Matrix with Diffusive Interfaces").
DIPOL.AMPL	Specifies the source's dipole moment in the case of a point source.
DIPOL.ELEC	Specifies an electric dipole source.
DIPOL.MAGN	Specifies a magnetic dipole source(i.e. when using POINT parameter). For an electric dipole, the units are C (2D) and C.um (3D). For a magnetic dipole, the units are A.um (2D) and A.um^2 (3D).
DX.PHOTOGEN **DY.PHOTOGEN**	Describe the spacings in the x and y directions for tabular photogeneration described in Section 11.6.4 "Tabular Photogeneration (Luminous 2D only)".
E.END	Specifies the maximum energy for uniform sampling of the spectrum in energy.
E.NUMBER	This is an alias for **WAVEL.NUM**.
E.START	Specifies the minimum energy for uniform sampling of the spectrum in energy.
ELLIPTICAL	Specifies that an elliptical source is to be used in Luminous 3D. When the ELLIPTICAL parameter is specified, the X.CENTER, Z.CENTER, X.RADIUS, and Z.RADIUS should also be specified. The synonym for this parameter is **CIRCULAR**..
ERRCTR	Specifies the number of time steps between error estimate updates. This only applies if NO.ENV is specified.
EXTEND.ALL	Allows the extension of the device from all directions using the refractive index profile of the line (or surface)
EXTEND.BACK	Allows the extension of the device from the back (along the z-axis) using the refractive index profile of the line (or surface)
EXTEND.BOTTOM	Allows the extension of the device from the bottom (along the y-axis) using the refractive index profile of the line (or surface)
EXTEND.FRONT	Allows the extension of the device from the front (along the z-axis) using the refractive index profile of the line (or surface)
EXTEND.LEFT	Allows the extension of the device from the left (along the x-axis) using the refractive index profile of the line (or surface)

EXTEND.RIGHT	Allows the extension of the device from the right (along the x-axis) using the refractive index profile of the line (or surface)
EXTEND.TOP	Allows the extension of the device from the top (along the y-axis) using the refractive index profile of the line (or surface)
FDTD.LUM	If specified, the luminous FDTD output analysis is executed. This includes mainly the scattered power angular distribution.
F.IMAGE	Specifies the file name containing a C-Interpreter function for specifying the 2D relative intensity versus offset at the source perpendicular to the direction of propagation for ray tracing in Luminous 3D.
F.INTENSITY	Specifies the file name containing a C-Interpreter fucntion for specifying the optical intensity as a function of position and wavelength.
F.RADIATE	Specifies the name of a file containing a C-Interpreter function for specifying generation rate as a function of position and optionally time. This function can be used to simulate single event upset (Luminous only).
F.REFLECT	Specifies the name of a file containing a C-Interpreter function for specifying reflection coefficient models as a function of wavelength, position, and angle incidence (Luminous only).
F3.RADIATE	This is the same as the F.RADIATE parameter but is applied in 3D. This is typically used for single event or photogeneration simulations with Luminous 3D.
FDTD.AUTO	Specifies that the structure will be meshed automatically so that locally the condition that the mesh spacing be less than the wavelength divided by the value of the parameter TD.STATE.
FDTD.AUTO.MIN	Specifies the minimum number of mesh lines used to resolve any region in any of the principal directions when FDTD.AUTO is also specified.
FILE.PHOTOGEN	Specifies the file name for tabular photogeneration described in Section 11.6.4 “Tabular Photogeneration (Luminous 2D only)”.
FRONT.REFL	Specifies that front side reflections are to be taken into account. When FRONT.REFL is specified, the area outside the device domain is assumed to be a vacuum (i.e.,. n=1.0, k= 0.0).
GAUSSIAN	This is the synonym for **X.GAUSSIAN**.
HORIZONTAL	Specifies the ratio of horizontal dipoles sources in a dipole orientation distribution 3D simulation.
INPUTRAYS	Specifies the filename of a file containing user-defined beam information. The file format is described in Section 11.6.1 “User-Defined Beams”.
INTEGRATE	Specifies whether the user-specified spectrum (contained in the file identified by the POWER.FILE parameter) should be numerically integrated and averaged over the wavelength sampling or if the samples should be interpolated directly from the table.

ITERATION	Sets or resets the iteration number passed to the C-Interpreter function F.CHARGING.
LENS.HEIGHT	Specifies an attribute of the composite lenslet (see Figure 11-16).
LENS.INDEX	Specifies the index of refraction of a lenslet (Luminous 3D only).
LENS.PLANE	Specifies the minimum y-cordinate of the lenslet sphere (Luminous 3D only).
LENS.RADIUS	Specifies the radius of the sphere defining a lenslet (Luminous 3D only).
LENS.WIDTH	Specifies an attribute of the composite lenslet (see Figure 11-16).
LENS.X	Specifies the X coordinate of the center of the sphere defining a lenslet (Luminous 3D only).
LENS.Y	Specifies the Y coordinate of the center of the sphere defining a lenslet (Luminous 3D only).
LENS.Z	Specifies the Z coordinate of the center of the sphere defining a lenslet (Luminous 3D only).
LENS.XMIN **LENS.XMAX** **LENS.ZMIN** **LENS.ZMAX**	Specify attributes of the composite lenslet (see Figure 11-16).

LENS.XSAGS
LENS.ZSAGS
	Specify the file names of aspheric lenslet Sag-h data files. These files must be in the following format.
	`number_of_pairs`
	`Sag_1 h_1`
	`Sag_2 h_2`
	. .
	. .
	. .
	`Sag_n h_n`
	Here, n is the number of pairs specified in the first line of the file. The pairs must be arranged in order of increasing h. The data modeling algorithm assumes that there is a sample with Sag=0 at h=0. Therefore, such a sample must not be included in the file.

LID	Specifies that the photogeneration rate of the beam will be applied only to the light induced defect generation model. It will not be included in the carrier continuity and temperature equations.
LONGIT.STEP	Sets the mesh size in the longitudinal direction when using BPM. The default value is WAVELENGTH/16.0.

MAX.WINDOW	Specifies the maximum x-value of the illumination window relative to the coordinate system of the optical beam (see Figure 22-1). The illumination window is always clipped to the device domain. The synonym for this parameter is XMAX.
MC.ARES	Specifies the angular resolution for angular distribution functions for Monte Carlo ray tracing in radians.
MC.SEED	Specifies the seed value for random number generation for Monte Carlo ray tracing.
MEAN	This is the synonym for **X.MEAN**.
METAL.REFLECT	Specifies that all metals are to be treated as perfect reflectors.
MIN.POWER	Specifies the minimum intensity relative to the source that a given ray will be traced. This is useful for limiting the numbers of rays traced.
MIN.WINDOW	Specifies the minimum x-value of the illumination window relative to the coordinate system of the optical beam. The synonym for this parameter is **XMIN**.
MONTE.CARLO	Specifies that Monte Carlo ray tracing will be used.
NO.ENV	Specifies that charge in intensity is used rather than field envelopes for error estimation. This saves about 50% in memory.
NORMALIZE	Indicates that the POWER.FILE spectral intensity needs to be normalized (integral over spectrum equals 1). In this case, beam intensity is set by the B<n> parameter on the **SOLVE** statement.
NSAMP	Specifies the number of samples used to represent the aspheric lenslet profile output using OUT.SAG.
NUMBER	Specifies the beam number (from 1 to 10). This number is used by the **SOLVE** statement to specify the relative intensity of different beams. You may specify beam numbers in any order that you desire.
NX	Specifies the number of rays traced along the source beam's X axis for Luminous 3D.
NZ	Specifies the number of rays traced along the source beam's Z axis for Luminous 3D.
N.VS.X **N.VS.Y** **N.VS.Z**	Indicate that index variation may occur in the associated direction relative to the source beam coordinate system.
OUT.POWER	Specifies the filename for output of multispectral samples, input using the POWER.FILE parameter, in a format suitable for display in TonyPlot.
OUT.SAG	Specifies the name of a file used to output an aspheric lenslet profile for subsequent display in TonyPlot. OUT.SAG allows a cutline along an arbitrary direction specified by two points in the XZ plane using the X1, Z1, X2, and Z2 parameters.

OUT.XSAG **OUT.ZSAG**	Specify the file names for output log files that can be displayed using TonyPlot. These files contain the data input from the LENS.XSAGS and LENS.ZSAGS files and the results of the data modeling.
OUTFILE	Specifies the name of a structure file for visualizing lenslet information and ray tracing. This can be distinguished from the RAYTRACE parameter, which will save ray trace information only without lenslets. The alias for this parameter is **TD.FILE**.
PACKET	Specifies a (time) Gaussian wave packet source in FDTD which is plane wave modulated by a Gaussian function. The width of the packet is given by the parameter TD.WIDTH and the central wavelength is given by WAVELENG.
PARALLEL	Specifies that multiple processors can be used in parallel during ray trace calculation.
PERIODIC	Specifies that for ray tracing, the structure is to be treated as periodic in the X and Z directions. Rays exiting the sides of the device are wrapped around to the other side of the device.
PHI	This is the synonym for **ANGLE**.
PML.CROP	Specifies that PMLs will be omitted from any FDTD structure file.
POLARIZE	Specifies the polarization of the optical beam at the origin. The polarization angle is the angle between the E vector and the incidence plane.
POLAR.PHASE	Specifies the polarization phase in FDTD 3D between transverse field components (follows general Jones vector representation of EM elliptic polarization).
POWER.SCAL	Specifies a scale factor. This factor is multiplied by each of the relative powers in the spectrum file when multi-spectral simulations are performed. The POWER.SCAL parameter can be used to perform unit conversions.

POWER.FILE	Specifies the filename of a spectrum file. The spectrum file must be in the following format.
	`number of pairs`
	`wavelength_1  power_1`
	`wavelength_2  power_2`
	`. . .  . . .`
	`. . .  . . .`
	`wavelength_n  power_n`

PULSE	Specifies a (time) Gaussian pulse source in FDTD to simulate broadband excitations.

Note: The power file must contain at least two power/wavelengths pairs.

QUANTUM.EFF	This is a quantum efficiency factor which specifies the number of carrier pairs generated per photon absorbed.
RAYTRACE	Specifies the name of a file where the results of a ray trace are saved. The ray trace may be viewed using TonyPlot3D (Luminous 3D only). This deos not include lens information. For lens information, use the **OUTFILE** parameter.
RAY.CHECK	Enables diagnostic printing during ray tracing.
RAYAREA	This is a parameter that specifies area in μm^2 (thickness in 2D in um) of each ray in the **BEAM** statement. This parameter is used when the specified INPUTRAYS file does not contain ray area information.
RAYS	Specifies the number of rays you want to split the optical beam. Luminous 2D will automatically split the beam into enough rays to resolve the geometry. Use of the RAYS parameter will cause further splitting of the optical beam (Luminous only).
REFLECTS	Specifies the number of reflections that will be traced. When the value of the REFLECTS parameter is increased, the total number of rays traced increases non-linearly. We recommend that this parameter be used wisely. For example, a single ray incident on three material layers will produce 4 rays if REFLECTS=0 is specified, 10 rays if REFLECTS=1 is specified, and 24 rays if REFLECTS=2 is specified.
REL.POWER	Specifies the relative power in the beam when mono-spectral simulations are performed. This factor is multiplied by the power parameters specified in the **SOLVE** statement to give the total optical power in the beam.
SAG.ITS	Specifies the number of iteration used to model the aspherical lenslet Sag-h data using non-linear least squares fitting.
SCAT.BACK	Specifies the length of the back scattering region in FDTD.
SCAT.BOTTOM	Specifies the length of the bottom scattering region in FDTD.
SCAT.FRONT	Specifies the length of the front scattering region in FDTD.
SCAT.LEFT	Specifies the length of the left scattering region in FDTD.
SCAT.LENGTH	Specifies the length of the scattering region in all directions in FDTD.
SCAT.RIGHT	Specifies the length of the right scattering region in FDTD.
SCAT.TOP	Specifies the length of the top scattering region in FDTD.
SEG.LOG	Specifies the number of name sequenced log plot files to be saved.
SEG.STR	Specifies the number of name sequenced structure plot files to be saved.
SEQ.NFP	Specifies the number of name sequenced near-field pattern plot files to be saved.
SEQ.FFP	Specifies the number of name sequenced far-field pattern plot files to be saved.

SIGMA	This is the synonym for **X.SIGMA**.
STRUCTURE	This is a synonym for **DEVICE**.
SUB.EXTINCT	Specifies the extinction coefficient used at the exit of a layer stack for analysis using the matrix method (see Section 11.3 "Matrix Method").
SUB.INDEX	Specifies the index of refraction at the exit of a layer stack for analysis using the matrix method (see Section 11.3 "Matrix Method").
SUBSTRATE	Indicates that the transfer matrix method in Luminous will treat the final layer as the substrate (i.e., it is treated as being infinitely thick).
TAUC.DOS	Specifies that midgap density of states will be calculated from the absorption coefficient and output with absorption as specified by the OUT.INDEX parameter of the **MATERIAL** statement.
TBLACK	Specifies the black body temperature of a 1 steradian target for multi-spectral analysis.
TD.WIDTH	Specifies the (time) width of the Gaussian pulse.
TFSF	Specifies a Total-Field-Scattered-Field (TFSF) plane wave source.
THETA	Specifies the angle of rotation for the source beam direction of propogation relative to the XY plane.
THINEST	Specifies the width of the thinnest ray to be traced (Luminous only).
TR.MATRIX	Specifies that optical absorption and photogeneration analysis in Luminous 2D will be done using the Matrix Method as described in Section 11.3 "Matrix Method".
TRANSV.STEP	Sets the mesh size in the transverse direction when using BPM. The default value is WAVELENGTH/16.0.
USER.SPECTRUM	Specifies that sampling in wavelength used for simulation is the same as that contained in the POWER.FILE.
VERBOSE	Enables a higher level of diagnostic run-time printing.
WAVELENGTH	Specifies the optical wavelength of the source beam (in the vacuum) for mono-spectral simulations.
WAVEL.END	Specifies the maximum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.
WAVEL.NUM	Specifies the number of wavelengths which will be used when multi-spectral simulations are performed. Spectral illumination is selected when the WAVEL.NUM parameter is greater than 1. Once multi-spectral simulations are selected, you must specify the POWER.FILE, WAVEL.START, and WAVEL.END parameters.
WAVEL.SCAL	Specifies the scale factor for the wavelengths which are used in multi- spectral simulations. Each of the wavelengths in the spectrum file is multiplied by this scale factor.

WAVEL.START	Specifies the minimum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.
X0.PHOTOGEN **Y0.PHOTOGEN**	Describe the origin of the tabular photogeneration described in Section 11.6.4 "Tabular Photogeneration (Luminous 2D only)".
X1, **X2**, **Z1**, and **Z2**	Specify the coordinates of two points in the XZ plane for extraction of an aspheric lenslet profile along the cutline defined by the two points. The profile is output to the file specified by the OUT.SAG parameter for subsequent display in TonyPlot.
X.CENTER	Specifies the X coordinate of the center of an elliptical source related to the beam coordinate system (which is centered at the beam origin) for Luminous 3D. Note that to enable elliptical sources, the ELLIPTICAL parameter should also be specified. The synonym for this parameter is **XCENTER**.
X.FACET **Y.FACET** **Z.FACET**	Specify the location of a plane intersecting perpendicular the associated axis used for extraction of near and far field data.
X.GAUSSIAN	Specifies that a Luminous 3D source is to have a Gaussian intensity profile in the X direction related to the beam coordinate system (which is centered at the beam origin). When X.GAUSSIAN is specified, X.MEAN and X.SIGMA should also be specified. The synonyms for this parameter are **XGAUSSIAN** and **GAUSSIAN**.
X.MEAN	Specifies the location of the mean value of a Gaussian source in the X direction related to the beam coordinate system (i.e., from the X.CENTER position). When X.MEAN is specified, X.GAUSSIAN and X.SIGMA should also be specified. The synonyms for this parameter are **XMEAN** and **MEAN**.
X.MIRROR	Indicates that the extreme boundaries of the simulation domain in the beam referenced coordinate system X coordinate are to be treated as mirrors for ray tracing.
X.ORIGIN	Specifies the X coordinate of the optical beam origin (see Figure 22-1). The beam must originate outside all device regions.
XPERIODS	Specifies the number of period repetitions along x of the structure defined (this defines the length of the illuminated region of the device).
X.PITCH	Specifies the length of the unit cell of a periodic system (smallest period of the structure) along x.
X.RADIUS	Specifies the X axis radius of an elliptical source related to the beam coordinate system (which is centered at the beam origin) for Luminous 3D. Note that to enable elliptical sources, the ELLIPTICAL should also be specified. The synonym for this parameter is **XRADIUS**.
X.SEMIAXIS **Y.SEMIAXIS** **Z.SEMIAXIS**	Specify attributes of the elliptical lense (see Equation 11-79).

X.SIGMA	Specifies the standard deviation in the X direction of the beam coordinate system for a Gaussian source in Luminous 3D. When X.SIGMA is specified, X.GAUSSIAN and X.MEAN should also be specified. The synonym for this parameter is **XSIGMA**.
XMAX	Specifies the maximum X coordinate in the source beam coordinate system for ray tracing in Luminous 3D.
XMIN	Specifies the minimum X coordinate in the source beam coordinate system for ray tracing in Luminous 3D.
Y.ORIGIN	Specifies the Y coordinate of the optical beam origin (see Figure 22-1). The beam must originate outside all device regions.
Z.CENTER	Specifies the Z coordinate of the center of an elliptical source related to the beam coordinate system (which is centered at the beam origin). Note that to enable elliptical sources, the ELLIPTICAL parameter should also be specified. The synonym for this parameter is **ZCENTER**.
Z.GAUSSIAN	Specifies that a Luminous 3D source is to have a Gaussian intensity profile in the Z direction related to the beam coordinae system (which is centered at the beam origin). When Z.GAUSSIAN is specified, Z.MEAN and Z.SIGMA should also be specified. The synonym for this parameter is **ZGAUSSIAN**.
Z.MEAN	Specifies the location of the mean value of a Gaussian source in the Z direction related to the beam coordinate system. When Z.MEAN is specified, Z.GAUSSIAN and Z.SIGMA should also be specified. The synonym for this parameter is **ZMEAN**.
Z.MIRROR	Indicates that the extreme boundaries of the simulation domain in the beam referenced coordinate system Z coordinate are to be treated as mirrors for ray tracing.
Z.RADIUS	Specifies the X axis radius of an elliptical source related to the beam coordinate system. When Z.MEAN is specified, Z.GAUSSIAN and Z.SIGMA should also be specified. The synonym for this parameter is **ZRADIUS**.
Z.SIGMA	Specifies the standard deviation in the Z direction of the beam coordinate system for a Gaussian source in Luminous 3D. When Z.SIGMA is specified, Z.GAUSSIAN and Z.MEAN should also be specified. The synonym for this parameter is **ZSIGMA**
ZMAX	Specifies the maximum Z coordinate in the source beam coordinate system for ray tracing in Luminous 3D.
ZMIN	Specifies the minimum Z coordinate in the source beam coordinate system for ray tracing in Luminous 3D.
Z.ORIGIN	Specifies the Z coordinate of the optical beam origin. The beam must originate outside all device regions (Luminous 3D only).

FDTD Parameters

BIG.INDEX	Specifies that the parameters TD.SRATE and PROP.LENS will be multipled by the maximum real part of the index of refraction.
COSINE	Specifies that the FDTD source is modeled as a cosine wave.
BACK.FACET **BOTTOM.FACET** **FRONT.FACET** **LEFT.FACET** **RIGHT.FACET** **TOP.FACET**	These are logical parameters for extracting near and far field patterns at the surfaces of the FDTD domain.
DT	Specifies the uniform time step for FDTD analysis.
FACET	Specifies the root name of files for output of near and far field patterns.
FDTD	Enables FDTD modeling of the beam propogation.
INSTATE	Specifies the root file name of the FDTD "state" previously saved to disk using the OUTSTATE parameter. INSTATE will load the previous state if available.
MEMORIZE	Specifies a user-defined word to be used as a key for retrieval of FDTD state from memory using the REMEMBER parameter.
MESHONLY	Specifies that only mesh structure is loaded when INSTATE is specified.
NOZLENS	Specifies that lens variation in the z direction is ignored. The x, y characteristics are taken at z=0.
NX	Specifies the number of uniformly spaced grid lines to be used in the FDTD mesh in the X direction.
NY	Specifies the number of uniformly spaced grid lines to be used in the FDTD mesh in the Y direction.
NZ	Specifies the number of uniformly spaced grid lines to be used in the FDTD mesh in the Z direction.
OUT.BACK **OUT.BOTTOM** **OUT.FRONT** **OUT.LEFT** **OUT.RIGHT** **OUT.TOP**	Indicate which PMLs are included in calculation of output coupling for LED analysis.
OUTSTATE	Specifies the root name of files written to disk saving the current FDTD "state" for later retrieval using the INSTATE parameter.
PHASE	Specifies the initial phase of the FDTD source.
PLANE	Specifies that the FDTD source is a plane wave located at y'=0.0.
POINT	Specifies that the FDTD source is a point source. The location is specified by X.POINT, Y.POINT, and Z.POINT.

PROP.LENG	Specifies the distance limit for which the light will be propagated before FDTD simulation is concluded.
REMEMBER	Specifies a user-defined word that is associated with the "state" of the FDTD saved in memory by the MEMORIZE parameter. The REMEMBER parameter will restore the previous state is available.
S.BOTTOM	Specifies that the plane source is located at the bottom (maximum Y coordinate) of the device.
S.BACK	Specifies that the plane source is located at the back (minimum Z coordinate) of the device.
S.FRONT	Specifies that the plane source is located at the front (maximum Z coordinate) of the device.
S.LEFT	Specifies that the plane source is located at the left (minimum X coordinate) of the device.
S.RIGHT	Specifies that the plane source is located at the right (maximum X coordinate) of the device.
S.TOP	Specifies that the plane source is located at the top (minimum Y coordinate) of the device.
SINE	Specifies that the FDTD source is modeled as a sine wave.
STR.DECI	Specifies the number of decimal digits of precision to maintain when writing structure files.
STR.PRECIS	This is alias for **STR.DECI**.
SUBDX **SUBDY** **SUBDZ**	Specify the spacing in microns that the FDTD mesh will be sampled at for subsampled output of FDTD meshes.
SUBNX **SUBNY** **SUBNZ**	Specify the number of uniform samples that the FDTD mesh will be sampled at for subsampled output of FDTD meshes.
SX	Specifies the spacing of uniformly spaced grid lines to be used in the FDTD mesh in the X direction.
SY	Specifies the spacing of uniformly spaced grid lines to be used in the FDTD mesh in the Y direction.
SZ	Specifies the spacing of uniformly spaced grid lines to be used in the FDTD mesh in the Z direction.
TE	Specifies that the FDTD source is a transverse electric wave.
TD.END	Specifies that a time-domain structure is written at the end of the number of specified iterations (TD.LIMIT or TD.WAVES) or when the estimated error is reduced below the specified maximum (TD.ERRMAX). This file will have a root name given by the parameter TD.FILE with a ".str" suffix.

TD.ERRMAX	Specifies the maximum residual (estimated) error for termination of FDTD time stepping.
TD.EVERY	Specifies how often FDTD time periodic saves are performed. A value of 1 indicates that files are saved during every time step.
TD.FILE	Specifies the root name of an output file saved periodically during the time domain simulation of FDTD. The alias for this parameter is **OUTFILE**.
TD.HARD	Specifies that a hard optical source will be used for FDTD.
TD.LIMIT	Specifies the maximum number of timesteps to simulate in FDTD analysis. If TD.LIMIT is unspecified, the time domain simulation will continue indefinitely.
TD.LOG	Specifies a log file for capture of time domain data for FDTD.
TD.MANY	Specifies how many FDTD files are saved during periodic time domain saves.
TD.ONE	Specifies that only one file name will be written for periodic FDTD saving. Each time a save is performed it is written over the previous save.
TD.PER	Specifies the number of structure files to output during a cycle at the source in FDTD time stepping.
TD.SOFT	Specifies that a soft optical source will be used for FDTD.
TD.SRATE	Defines the spatial sampling rate in terms on the number of samples per wavelength (in vacuum).
TD.TFSF	Specifies that a total field-scattered field optical source will be used for FDTD.
TD.WAVES	Specifies the maximum number of cycles (wavelengths) at the source for FDTD time stepping.
TD.ZCUT	Specifies that structure files output by specification of TD.FILE in Luminous 3D will be output at 2D cutplanes. The z location of the 2D cutplane is specified by the value of TD.ZCUT.
TM	Specifies that the FDTD source is a transverse magnetic wave.
TIMING	Enables the output of timing summaries for the solution process of FDTD.
WAVE.SAMP	This is alias for **TD.SRATE**.
X.MAX.CROP **X.MIN.CROP** **Y.MAX.CROP** **Y.MIN.CROP** **Z.MAX.CROP** **Z.MAX.CROP**	Provide cropping (truncation) of structure files output by TD.FILE along the principal axes.

X.POINT **Y.POINT**	Specify the location of a point source for FDTD.
X.PERIODIC	Specifies that for ray tracing, the structure is to be treated as periodic in the X direction. Rays exiting the sides of the device are wrapped around to the other side of the device.
Z.PERIODIC	Specifies that for ray tracing, the structure is to be treated as periodic in the Z direction. Rays exiting the sides of the device are wrapped around to the other side of the device.

Monochromatic Beam Example

This beam has a monochromatic spectrum with a wavelength of 0.6 μm. The beam originates at x=0.5 and y=-2.0. It has a 90 degree propogation angle and a beam width of 0.2 μm which is centered at the beam origin.During the ray trace calculation the rays will be terminated when the power level along the ray falls to 5% of the original power.

```
BEAM NUM=1 WAVELENGTH=0.6 X=0.5 Y=-2.0 ANG=90.0 MIN=-0.1 MAX=0.1 \
     MIN.POWER=0.05
```

Multi-spectral Beam Example

A multi-spectral beam (at a 45 degree angle) which originates at x=0.0 and y=-1.0. The multi-spectral source is imported from the spectrum file, `source.spc`. The spectrum is discretized into our wavelengths between 0.4 μm and 0.6 μm.

```
BEAM NUM=2 X=0.0 Y=-1.0 ANG=45.0 \
     POWER.FILE=SOURCE.SPC WAVEL.START=0.4 \
     WAVEL.END=0.6 WAVEL.NUM=4
```

Luminous 3D Lens Example

```
BEAM NUM=1 X.ORIGIN=2.5 Y.ORIGIN=-1.0 Z.ORIGIN = 2.5 ANG=90.0 WAVEL=0.6 \
     NX=10 NZ=10 LENS.X=2.5 LENS.Y=-0.5 LENS.Z=2.5 \
     LENS.INDEX=2.03 LENS.RADIUS=0.25 LENS.PLANE=-0.5
```

Gaussian Intensity Profile Example

The following beam statement will define a beam window 2 μm wide that is centred at (`X.ORIGIN`, `Y.ORIGIN`) with a Gaussian peak of 0.01 W/cm^2 with a standard deviation of 0.05 μm.

```
beam   num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=0.6  \
       xmin=-1   xmax=1 GAUSSIAN   MEAN=0   SIGMA=0.05 RAYS=200
       SOLVE B1=1E-2
```

Note: It is recommended that you use the RAYS parameter to define a large number of rays across the beam to ensure that the Gaussian profile is adequately reproduced. The rays are evenly spaced across the beam so it is necessary to use a large number of them.

22.4 CHARACTERIZE

The **CHARACTERIZE** statement initiates optical characterization. Characterization involves the capture of absorption versus depth data at each subsequent solution.

Syntax

```
CHARACTERIZE ABSORB=<string> [<parameters>]
```

Parameter	Type	Default	Units
ABSORPTION	Character		
BEAM	Integer		
NDEP	Integer		
DEP1	Real		microns
DEP2	Real		microns
S.DEP1	Real		microns
S.DEP2	Real		microns
SEQ.ABS	Integer	0	

Description

ABSORPTION	Specifies the file name root string for capture of absorption versus depth. A characterization file will be written at each subsequent solution.
BEAM	Specifies the beam index that associated with the beam to be characterized.
NDEP	Specifies the number of uniform samples to be taken with depth.
DEP1	Specifies the minimum y coordinate in microns for data capture.
DEP2	Specifies the maximum y coordinate in microns for data capture.
S.DEP1	Specifies the sample spacing at the minimum y coordinate for data capture in microns.
S.DEP2	Specifies the sample spacing at the minimum y coordinate for data capture in microns.
SEQ.ABS	Specifies the maximum number of seqeuenced files to capture.

22.5 COMMENT,

COMMENT allows comments to be placed in an Atlas input file. Atlas will print and display comment lines.

Syntax

```
COMMENT [<string>]
      # [<string>]
```

string is any alphabetic, numeric, or alphanumeric sequence of characters. The synonym for this parameter for #.

Example

```
COMMENT  ATLAS is a copyright of Silvaco International
#        ATLAS is a copyright of Silvaco International
```

Note: The $ was allowed as a comment character in previous versions of Atlas. This should be avoided and replaced by the **#** or **COMMENT** statement.

22.6 CONTACT

CONTACT specifies the physical attributes of an electrode.

Note: If the **CONTACT** statement is not used for a given electrode, the electrode is assumed to be charge-neutral (Ohmic).

Syntax

```
CONTACT NUMBER=<n>|NAME=<ename>|ALL [<wfp>] [<bc>] [<lcr>] [<link>]
```

Parameter	Type	Default	Units
ALL	Logical	False	
ALPHA	Real	0	cm
ALUMINUM	Logical	False	
BARRIER	Logical	False	
BETA	Real	1.0	
CAPACITANCE	Real	0	F/μm
CGTUNN	Logical	True	
COMMON	Character		
CON.RESIST	Real	0	$\Omega \cdot cm^2$
CURRENT	Logical	False	
DDMS.TUNNEL	Logical	False	
DE.TUNNEL	Real	10^{-4}	
DEVICE	Character		
DOS.NEGF	Real	0.1	1/eV
E.TUNNEL	Logical	False	
E.HIGHK	Real	0.0	eV
ELE.CAP	Integer		
EXCLUDE_NEAR	Logical	False	
EXT.ALPHA	Real	0	$W/(cm^2K)$
EXT.TEMP	Real	300	K
F.ETUNNEL	Character		
F.WORKF	Character		
FACTOR	Real	0	

Parameter	Type	Default	Units
FARADS	Logical	False	
FLOATING	Logical	False	
FG.CAP	Real	0.0	F/μm
GAMMA	Real	1.0	
H.TUNNEL	Logical	False	
H1SRV	Real	∞	cm/s
H2SRV	Real	∞	cm/s
HENRYS	Logical	False	
HIGHK.AL	Logical	False	
HIGHK.AU	Logical	False	
HIGHK.HF	Logical	False	
HIGHK.MG	Logical	False	
HIGHK.MO	Logical	False	
HIGHK.NI	Logical	False	
HIGHK.PT	Logical	False	
HIGHK.TA	Logical	False	
HIGHK.TIN	Logical	False	
HIGHK.W	Logical	False	
IFL.AL203	Logical	False	
IFL.HF02	Logical	False	
IFL.SI3N4	Logical	False	
IFL.SI02	Logical	False	
IFL.ZR02	Logical	False	
INDUCTANCE	Real	0	H·μm
ME.TUNNEL	Real	1.0	
MH.TUNNEL	Real	1.0	
MO.DISILICIDE	Logical	False	
MOLYBDENUM	Logical	False	
MULT	Logical	False	

Parameter	Type	Default	Units
NAME	Character		
NEGF.TUNNEL	Logical	False	
NEUTRAL	Logical	True	
N.POLYSILICON	Logical	False	
NSURF.REC	Logical	False	
NUMBER	Integer		
OHMS	Logical	False	
ON.AL2O3	Logical	False	
ON.HFO2	Logical	False	
ON.SIO2	Logical	False	
ON.SI3N4	Logical	False	
ON.ZRO2	Logical	False	
P.POLYSILICON	Logical	False	
PARABOLIC	Logical	False	
PIPINYS	Logical	False	
PSURF.REC	Logical	False	
QEX.BARRIER	Real	1	
QN.BARRIER	Real	1	
QP.BARRIER	Real	1	
Q.THERMIONIC	Real	1	
REF.ELEC	Character		
REF.ENUM	Integer		
REFLECT	Logical	False	
REFLECT	Real		
RESISTANCE	Real	0	Ω.μm
RESONANT	Logical	False	
S.HIGHK	Real	0.0	
SCOM.QUALITY	Real	1	
SCOM.RANGE	Real	1	

Parameter	Type	Default	Units
SCOTT.MALLIARAS	Logical	False	
SHORT	Logical	False	
STRUCTURE	Character		
SURF.REC	Logical	False	
QTUNN.CMASS	Real	1.0	
QTUNN.VMASS	Real	1.0	
THERM	Logical	True	
TU.DISILICIDE	Logical	False	
TUNGSTEN	Logical	False	
VSURFN	Real	see description	cm/s
VSURFP	Real	see description	cm/s
WORKFUN	Real	0	V

Description

ALL	Defines the same properties for all electrodes.
DDMS.TUNNEL	Specifies Schottky contact in DDMS model.
DEVICE	Specifies which device the CONTACT statement applies to in MixedMode. The synonym for this parameter is STRUCTURE.
DOS.NEGF	Sets the value of metal density of states to be used in contact self-energy of Schottky contact.
NAME	Specifies the name of a previously defined electrode. See ELECTRODE for more information.
NEGF.TUNNEL	Specifies Schottky contact in NEGF model.
NUMBER	Specifies the contact number to be defined. It must be the number of a previously defined electrode. It is recommended that electrode names be used rather than numbers.
STRUCTURE	This is a synonym for DEVICE.
wfp	This is one of the work function parameters described below. It is permitted to either specify the name of a material or a work function value (WORKFUN parameter).

bc	This is one or more of the boundary condition parameters.
lcr	This is one or more of the external parasitic element parameters.
link	This is one or more of a set of parameters that allow you to associate two or more electrodes electrically.

Workfunction Parameters

ALUMINUM	Specifies aluminum as the contact material for the electrode. This sets the workfunction to 4.10V. Note that this parameter should not be set if an Ohmic contact is required.
F.WORKF	Specifies the name of a file containing a C-Interpreter function describing the workfunction as a function of the electric field.
MOLYBDENUM	Specifies molybdenum as the contact material for the electrode. This sets the work function of the electrode to 4.53V.
MO.DISILICIDE	Specifies molybdenum disilicide as the contact material for the electrode. This sets the work function of the electrode to 4.80V.
NEUTRAL	Specifies that the electrode is Ohmic. This is the default characteristic of an electrode.
N.POLYSILICON	Specifies n+ doped polysilicon as the contact material for the electrode. This sets the work function to 4.17V.
NSURF.REC	Enables finite surface recombination velocity for electrons.
P.POLYSILICON	Specifies p+ polysilicon as the contact material for the electrode. This sets the work function to 4.17V + Eg(Si).
PARABOLIC	Enables the parabolic Schottky field emission model as given in Equation 3-180.
PSURF.REC	Enables finite surface recombination velocity for holes.
THERM	Enables the thermionic emission model described in Equation 3-182.
TU.DISILICIDE	Specifies tungsten disilicide as the contact material for the electrode. This sets the work function to 4.80V.
TUNGSTEN	Specifies tungsten as the contact material for the electrode. This sets the work function to 4.63V.
WORKFUN	Specifies the work function of the electrode in V. This parameter must be specified in the form `WORKFUN=n` where `n` is a real number. This specification is absolute workfunction and not workfunction difference to the semiconductor.

Note: If no **WORKFUN** or material type parameter is specified, the electrode is assumed to be an Ohmic contact.

Boundary Conditions

CURRENT	Specifies current boundary conditions. If specified, CAPACITANCE, CON.RESIST, INDUCTANCE, or RESISTANCE may not be specified.
FLOATING	Specifies a charge boundary condition. This parameter is used to specify the floating gate in EPROM devices. This parameter can only be used for insulated contacts. If specified, CAPACITANCE, CON.RESIST, INDUCTANCE, or RESISTANCE may not be specified, but special syntax exists for adding capacitances to a floating contact. See EL<n>.CAP and FG<n>.CAP. The synonym for this parameter is CHARGE.
ALPHA	Specifies the linear, dipole lowering coefficient. This parameter has no effect unless the BARRIER parameter has been specified (See Equation 3-176).
BARRIER	Turns on the barrier lowering mechanism for Schottky contacts.
BETA	Specifies a prefactor in the barrier lowering (See Equation 3-176).
CGTUNN	Includes Floating Gate to Control Gate Current (FGCG) in charging model. Only works on a FLOATING contact.
DE.TUNNEL	Specifies the fraction of the barrier height to use in numerical integration of the tunneling probability for the Parabolic Schottky model.
E.TUNNEL	Specifies that the Schottky tunneling model for electrons will be used. E.TUNNEL will also enable the SURF.REC boundary condition which models the thermionic emission in a Schottky contact.
H.TUNNEL	Specifies that the Schottky tunneling model for holes will be used. H.TUNNEL will also enable the SURF.REC boundary condition which models the thermionic emission in a Schottky contact.
H1SRV	Atomic hydrogen surface recombination velocity at electrode boundary.
H2SRV	Molecular hydrogen surface recombination velocity at electrode boundary.
EXCLUDE_NEAR	Specifies that the contact should be excluded from the algorithm that finds the nearest electrode to a given point in the direct quantum tunneling model (See Section 3.6.7 “Gate Current Models”).
EXT.ALPHA	Specifies the inverse value of the thermal resistance that can be applied to a contact when performing energy balance simulations. Basically, the thermal resistance allows the carrier energy boundary condition at a contact to be a value other than the ambient temperature.
EXT.TEMP	Specifies the external temperature, or carrier energy, of an electrode when performing energy balance simulations.
F.ETUNNEL	Specifies the name of a file containing a C-Interpreter function that specifies electron tunneling at a Schottky contact.

GAMMA	Specifies a exponent in the linear barrier lowering term (See Equation 3-176).
ME.TUNNEL	Specifies the relative effective mass for use in the electron tunneling mode (see **E.TUNNEL**).
MH.TUNNEL	Specifies the relative effective mass for use in the hole tunneling mode (see **H.TUNNEL**).
PIPINYS	Enables the reverse bias phonon-assisted tunneling model for GaN Schottky diodes.
QEX.BARRIER	Specifies the "quality" of the interface for exciton thermionic emission.
QN.BARRIER	Specifies the "quality" of the interface for electron thermionic emission.
QP.BARRIER	Specifies the "quality" of the interface for hole thermionic emission.
Q.THERMIONIC	Specifies the “quality” of the interface for thermionic emission.
QTUNN.CMASS	Specifies the electron effective mass to use in the contact for a direct quantum tunneling model (See Section 3.6.7 “Gate Current Models”).
QTUNN.VMASS	Specifies the hole effective mass to use in the contact for a direct quantum tunneling model (See Section 3.6.7 “Gate Current Models”).
REF.ELEC	Specifies the name of a reference electrode for the Parabolic Schottky model.
REF.ENUM	Specifies the index of a reference electrode for the Parabolic Schottky model.
REFLECT (Logical)	Specifies that for Energy Balance and Hydrodynamic simulation using a Neumann (reflective) boundary condition for carrier temperature equations. By default, contacts are handled as Dirichlet boundaries with carrier temperature equal to lattice temperature. In case of Schrodinger or NEGF solution, the parameter will enforce von Neumann boundary conditions for potential in the contact.
REFLECT (Real)	Specifies the reflection coefficient for all rays incident on the boundaries of the associated electrode for Luminous. The alias for this parameter is L.REFLECT.
SCOM.QUALITY	The "quality" of the Scott-Malliaras boundary conditions. A quality of 1 gives the Scott-Malliaras boundary conditions, a quality of 0 gives Schottky boundary conditions.
SCOM.RANGE	The fraction of the bandgap (centered on the middle of the bandgap) that the effective barrier height can lie in.
SCOTT.MALLIARAS	Turns on the Scott-Malliaras boundary conditions for a Schottky contact.

SURF.REC	Specifies that finite surface recombination velocities are used at the respective contact. This parameter must be specified in the form SURF.REC [VSURFN=<n>] [VSURFP=<p>], where n and p are real numbers.
VSURFN	Specifies the actual surface recombination velocities for electrons (V_{sn}). If this parameter is not specified, its default value is calculated by Equation 22-1. $$V_{sn} = \frac{\text{ARICHN} \cdot T_L^{\,2}}{qN_C} \quad (22\text{-}1)$$ where ARICHN is the effective Richardson constant for electrons. This constant accounts for quantum mechanical reflections and tunneling.
VSURFP	Specifies the actual surface recombination velocities for holes (V_{sp}). If this parameter is not specified, its default value is calculated by Equation 22-2. $$V_{sp} = \frac{\text{ARICHP}T_L^{\,2}}{qN_V} \quad (22\text{-}2)$$ where ARICHP is the effective Richardson constants for holes. This constant accounts for quantum mechanical reflections and tunneling.

Contact Parasitics

CAPACITANCE	Specifies a lumped capacitance value to be attached to the contact.
CON.RESIST	Specifies a distributed contact resistance. You cannot specify both CON.RESIST and RESIST.
FARADS	Specifies that the CAPACITANCE value will be in the units of Farads.
HENRYS	Specifies that the INDUCTANCE value will be in the units of Henrys.
INDUCTANCE	Specifies an external inductance which is related to the specified electrode. A synonym is L.
OHMS	Specifies that the RESISTANCE value will be in the units of ohms.
RESISTANCE	Specifies a lumped resistance value. You may not specify both RESISTANCE and CON.RESIST.

Note: There are restrictions on the allowed numerical methods and types of analysis possible when any form of parasitic element is attached to a contact. See Chapter 2 "Getting Started with Atlas" for details.

RESONANT	Only applies to small signal AC analysis. If set, it models the lumped elements as forming a parallel resonant circuit. The resistor and inductor are joined in series, while the capacitor is in parallel with them. The driving voltage used in the perturbation analysis is then applied to the circuit. If you do not set RESONANT, then the capacitor will be connected to AC ground (see Figure 3-5).

High-K Parameters

E.HIGHK	Specifies the charge neutrality workfunction used in Equation 3-204.
HIGHK.AL, **HIGHK.AU**, **HIGHK.HF**, **HIGHK.MG**, **HIGHK.MO**, **HIGHK.NI**, **HIGHK.PT**, **HIGHK.TA**, **HIGHK.W** **HIGHK.TIN**	Specify material dependent default metal workfunctions as given in Table 3-33.
IFL.AL203, **IFL.HF02**, **IFL.SI02**, **IFL.SI3N4**, **IFL.ZR02**	Specify material dependent charge neutrality workfunction and slope parameters for an interfacial layer as described in Table 3-35.
ON.AL2O3, **ON.HFO2**, **ON.SIO2**, **4ON.SI3N4**, **ON.ZRO2**	Specify material dependent default charge neutrality workfunction and slope paramaters as given in Table 3-34.
S.HIGHK	Specifies the slope parameter used in Equation 3-204.

Electrode Linking Parameters

These parameters allow one electrode to be biased as a function of another electrode. This allows separate regions of the same physical contact to be linked together. For example, in the statement:

```
CONTACT NAME=MYDRAIN COMMON=DRAIN FACTOR=4.6
```

The electrode, MYDRAIN, is linked to the electrode, DRAIN. The bias on MYDRAIN will always be equal to the bias on the drain plus 4.6. If the optional MULT parameter had been specified, the bias on MYDRAIN would be equal to the bias on the drain multiplied by 4.6.

COMMON	Specifies the electrode name to which the contact refered to by NAME is linked. Although the electrodes are linked, separate currents will be saved for both electrodes unless SHORT is also specified. The electrode refered to in NAME should not appear on any **SOLVE** statments since its bias is now determined as a function of the electrode refered to by COMMON.

SHORT	Specifies that the electrode referred to by NAME is shorted to the electrode specified by the COMMON parameter. This implies that the two electrodes will be treated as one and only one value will be written to log files and in the run time output.
FACTOR	Specifies the constant offset voltage (or current) between the electrodes refered by NAME and COMMON. By default, FACTOR is added the defined voltage.
MULT	Specifies that FACTOR is a multiplier.

Floating Gate Capacitance Parameters

In some cases, you may want to simulate floating gate structures, in 2D, which have control gates that are longer in the unsimulated dimension than the floating gate. In these cases, specify the following parameters to account for addition capacitance between the floating gate and the control gate and other electrodes. Up to four extra capacitances are allowed so in the following <n> is an integer number between 1 and 4.

EL<n>.CAP (ELE.CAP)	Specifies the name of the electrode to which the extra capacitance is linked.
FG<n>.CAP (FG.CAP)	Specifies the additional capacitance per unit length to be added between the floating gate and electrode specified in EL<n>.CAP.

Schottky Barrier and Surface Recombination Example

This example defines all electrodes except number 2 (aluminum) to be neutral. Electrode number 2 also includes finite surface recombination velocities and barrier lowering. A definition in the second statement overrides that definition in the first statement.

```
CONTACT ALL NEUTRAL
CONTACT NUMBER=2 ALUMINUM SURF.REC BARRIER
```

Parasitic Resistance Example

This example attaches a lumped resistor with a value of $10^5 \Omega\mu m$ to the substrate. A distributed contact resistance of $10^{-6} \Omega \cdot cm^2$ is included on the drain.

```
CONTACT NAME=substrate RESISTANCE=1E5
CONTACT NAME=drain CON.RESIST=1E-6
```

Floating Gate Example

This syntax defines a floating contact with a workfunction equal to 4.17eV. An extra 1fF/um capacitance is added between this electrode and the electrode named cgate.

```
CONTACT NAME=fgate FLOATING N.POLY EL1.CAP=cgate
FG1.CAP=1e-15
```

Note: The **MODELS** PRINT command can be used to echo the back contact workfunction and parasitic elements settings to the run-time output.

22.7 CURVETRACE

CURVETRACE sets the parameters for the automatic curve tracing routine.

Syntax

```
CURVETRACE <params>
```

Parameter	Type	Default	Units
ANGLE1	Real	5	Degree
ANGLE2	Real	10	Degree
ANGLE3	Real	15	Degree
BEG.VAL	Real	0.0	V
CONTROLRES	Real	1.0×1.0^{10}	Ω
CONTR.ELEC	Integer		
CONTR.NAME	Character		
CURR.CONT	Logical		
END.VAL	Real		
MAXDV1	Real	0.1	V
MAXDV2	Real	1.0	V
MINCUR	Real		
MINDL	Real	0.1	
NEXTST.RATIO	Real	2.0	
NO.BACKTRACE	Logical	False	
STEP.CONT	Logical		
STEP.INIT	Real	0.1	V
STEPS	Real	0.1	
THETA.BACKTRACE	Real	0.01	Degree
TURNINGPOINT	Logical	False	
V.BACKTRACE	Real	1	V
VOLT.CONT	Logical	False	

Description

ANGLE1, **ANGLE2**, and **ANGLE3**	These are the critical angles (in degrees) affecting the smoothness and step size of the trace. If the difference in slopes of the last two solution points is less than ANGLE1, the step size will be increased for the next projected solution. If the difference lies between ANGLE1 and ANGLE2, the step size remains the same. If the difference is greater than ANGLE2, the step size is reduced. ANGLE3 is the maximum difference allowed, unless overridden by the MINDL parameter. ANGLE2 should always be greater than ANGLE1 and less than ANGLE3.
BEG.VAL	This is the value of the voltage at the starting point of the curve trace for the controlling electrode.
CONTR.ELEC	This is the number of the electrode that is designated as a control electrode.
CONTR.NAME	This is the name of the control electrode.
CURR.CONT	Denotes that a maximum current on the control electrode, specified by END.VAL is used as the upper bound on the trace.
CONTROLRES	Specifies the control resistance value to be set at the control electrode after the current at the control electrode becomes greater than the value specified by the MINCUR parameter. Using a large value of the control resistance allows you to set the value of the current (instead of the internal voltage) on the control electrode. A large value of the control resistance is suitable for electron devices where the shape of the characteristic curve is vertical or exhibits the presence of turning points (where it is required to switch the electrode control from voltage to current).
END.VAL	This is used to stop tracing if the voltage or current of control electrode equals or exceeds END.VAL.
MAXDV1	Specifies maximum value of the voltage bias step at the beginning of the CURVETRACER algorithm. This limitation is used until the current at the control electrode is greater than the value specified by the MINCUR parameter.
MAXDV2	Specifies maximum value of the voltage bias step to be used after the current at the control electrode becomes greater than the value specified by the MINCUR parameter.
MAXDV1 and **MAXDV2**	Control (together with NEXTST.RATIO) the maximum voltage step at each point of the IV curve. Using large values for them means that few points are used to trace the curve. This results in a faster simulation but can affect the stability of the tracing algorithm and decrease the accuracy of the IV curve.

MINCUR	This may be used to set a small current value in order to switch from internal control electrode bias ramping to external ramping with load resistor. This parameter is recommended for small current breakdown simulation.
MINDL	This is the minimum normalized step size allowed in the trace. Usually, you don't need to adjust this parameter. Increasing MINDL will reduce the smoothness of the trace by overriding the angle criteria, resulting in more aggressive projection and fewer simulation points. Reducing MINDL will enhance the smoothness and increase the number of points in the trace.
NEXTST.RATIO	Specifies which factor to use to increase the voltage step on the smooth parts of the I-V curve.
NO.BACKTRACE	Stops curve tracing if it detects back tracing.

Note: If you define the curvetrace using the CONTINUE parameter on the **SOLVE** statement and you specify _TMA on the command line, then the default value of NO.BACKTRACE is true.

STEPS	This is the number of operational points on a trace if STEP.CONT was specified.
STEP.CONT	Specifies that the trace will proceed for a certain number of simulation points.
STEP.INIT	Specifies initial voltage step size.

Note: To set a sweep of increasingly negative voltage in **CURVETRACE**, you only need to set STEP.INIT to be negative. Since all parameters are multiplier of STEP.INIT, the whole voltage sweep will be negative.

THETA.BACKTRACE	Specifies the angular difference at which the curve is considered to be backtracing.
TURNINGPOINT	Specifies that binary output solution files will be saved whenever the slope of the IV curve changes sign (i.e., there is a turning point). The name of the output file is soln.num, where num is the number of the current solution.
V.BACKTRACE	Specifies the (internal) voltage difference at which the curve is considered to be backtracing.
VOLT.CONT	Denotes that a maximum voltage on the control electrode, specified by END.VAL is used as the upper bound on the trace.

Diode Breakdown Example

To trace a diode breakdown curve using current value as a termination criteria, the following statement may be used:

```
CURVETRACE CURR.CONT END.VAL=0.01 CONTR.NAME=anode \
MINCUR=5E-12 NEXTST.RATIO=1.1  STEP.INIT=0.1
SOLVE CURVETRACE
```

22.8 DATASET

The **DATASET** statement provides a common interface to import distributed data. The interface includes a common interface to allow distortions such as rescale, windowing, stretch, and rotation.

Syntax

```
DATASET INFILE=<string> NAME=<string> <filetype> [<parameters>]
```

Parameter	Type	Default	Units
ABS.CHAR	Logical	False	
AXIS.XYZ	Logical	True	
AXIS.XZY	Logical	False	
AXIS.YXZ	Logical	False	
AXIS.YZX	Logical	False	
AXIS.ZXY	Logical	False	
AXIS.ZYX	Logical	False	
DATA.XYZ	Logical	True	
DATA.XZY	Logical	False	
DATA.YXZ	Logical	False	
DATA.YZX	Logical	False	
DATA.ZXY	Logical	False	
DATA.ZYX	Logical	False	
FX1	Real	0.0	um
FY1	Real	0.0	um
FZ1	Real	0.0	um
FX2	Real	0.0	um
FY2	Real	0.0	um
FZ2	Real	0.0	um
INFILE	Character		
NAME	Character		
TX1	Real	0.0	um
TY1	Real	0.0	um
TZ1	Real	0.0	um

Parameter	Type	Default	Units
TX2	Real	0.0	um
TY2	Real	0.0	um
TZ2	Real	0.0	um
XMULT	Real	1.0	
YMULT	Real	1.0	
ZMULT	Real	1.0	

Description

ABS.CHAR	Specifies that the file contains photogeneration rate data extracted using the CHARACTERIZE statement.
AXIS.XYZ AXIS.XZY AXIS.YXZ AXIS.YXZ AXIS.ZXY AXIS.ZYX	Specify how the data in the file is to be ordered. In this case X, then Y, then Z. Since the file is sequential, the first character (X) represents the fastest varying ordinate, the second represents Y, and the third represents Z. Specifying AXIS.ZYX would reverse this order.
DATA.XYZ DATA.XZY DATA.YXZ DATA.YZX DATA.ZXY DATA.ZYX	These are similar to the AXIS.XYZ except now we are referring how the data sets are to be interpreted. In this case, we are specifying that the first set is the direction cosine with respect to X, the second one with respect to Z, and the last data set with respect to Y.
INFILE	This is the name of the source data file.
FX1, FX2, FY1, FY2 (FZ1, FZ2)	Specify the coordinates corresponding to rows and columns (and pages) of data in the source file. In this case, there are 71×21 data points in the data. We use FX and FY since we specified AXIS.XY(Z) first. The letter F is meant to indicate "FROM". The numbers 1 and 2 refer to the specification of two points in 2 (3) space describing the limits of the what subset of the data contained in the file are mapped.
NAME	This is a convenient tag that we can refer to later in the MODELS statement
TX1, TX2, TY1, TY2 (TZ1, TZ2)	Specify the corners in the device coordinate where the data is to be mapped. Scaling is automatic.
XMULT, YMULT, and ZMULT	These are multipliers times each of the data sets.

22.9 DBR

The **DBR** statement is used to define Distributed Bragg Reflectors (DBR). DBRs are periodic structures composed of alternating layers of two different materials. The alias for this statement is SUPERLATTICE.

Syntax

```
DBR <parameters>
```

Parameter	Type	Default	Units
BOTTOM	Logical	False	
CALC1.STRAIN	Logical	False	
CALC2.STRAIN	Logical	False	
HALF.CYCLES	Real	2	
LAYERS	Real	2	
LED1	Logical	False	
LED2	Logical	False	
MAT1	Character		
MAT2	Character		
N1	Real	0	
N2	Real	0	
NA1	Real	0.0	cm^{-3}
NA2	Real	0.0	cm^{-3}
NAME1	Character		
NAME2	Character		
ND1	Real	0.0	cm^{-3}
ND2	Real	0.0	cm^{-3}
NU12	Integer	5	
NU21	Integer	5	
POLARIZ1	Logical	False	
POLARIZ2	Logical	False	
POLAR1.CHARG	Real	0	cm^{-2}
POLAR2.CHARG	Real	0	cm^{-2}
POLAR1.SCALE	Real	1.0	

POLAR2.SCALE	Real	1.0	
QWELL1	Logical	False	
QWELL2	Logical	False	
SPA1	Real	0.0	μm
SPA2	Real	0.0	μm
STRAIN1	Real	0.0	
STRAIN2	Real	0.0	
TH12	Real		
TH21	Real		
THICK1	Real	0.0	μm
THICK2	Real	0.0	μm
TOP	Logical	False	
WELL1.CNBS	Integer	1	
WELL2.CNBS	Integer	1	
WELL1.FIELD	Logical	True	
WELL2.FIELD	Logical	True	
WELL1.GAIN	Real	1.0	
WELL2.GAIN	Real	1.0	
WELL1.NX	Integer	10	
WELL2.NX	Integer	10	
WELL1.NY	Integer	10	
WELL2.NY	Integer	10	
WELL1.VNBS	Integer	1	
WELL2.VNBS	Integer	1	
X1.COMP	Real	0.0	
X2.COMP	Real	0.0	
Y.START	Real	0.0	μm
Y.FINISH	Real	0.0	μm
Y.FINAL	Real	0.0	μm

Y.END	Real	0.0	μm
Y1.COMP	Real	0.0	
Y2.COMP	Real	0.0	

Description

The **DBR** statement is a short cut for specifying a set of **REGION** statements, which specifies a stack of alternating layers of two materials. The material compositions of the layers are specified by the MAT1, MAT2, NA1, NA2, ND1, ND2, STRAIN1, STRAIN2, X1.COMP, X2.COMP, Y1.COMP, and Y2.COMP parameters. The number of layers are specified by the HALF.CYCLES parameter. The locations of the layers are specified by the Y.START, Y.FINISH or the TOP/BOTTOM parameters. The meshing of the layers is specified by the N1, N2 or the SPA1, SPA2 parameters.

For more information about DBR, see "Specifying Distributed Bragg Reflectors" on page 601.

BOTTOM	Specifies that the DBR is to be added at the bottom (starting at the maximum previously specified Y coordinate and extending in the positive Y direction) of the structure.
CALC1.STRAIN **CALC2.STRAIN**	Specifies that the strain in the region/cycle is calculated from the lattice mismatch with adjacent regions.
HALF.CYCLES	Specifies the total number of layers. Note that if HALF.CYCLES is odd, there will be more layers of one material than the other. Its alias is LAYERS.

Note: Don't confuse HALD.CYCLES with fractions of the optical emisson wavelength. HALF.CYCLES relates to layers.

LAYERS	This is an alias for **HALF.CYCLES**.
LED1 **LED2**	Specifies that the region/cycle is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 12 "LED: Light Emitting Diode Simulator".
MAT1	Specifies the material name for layers of material 1.
MAT2	Specifies the material name for layers of material 2.
N1	Specifies the integer number of mesh lines per layer of material 1. Note that if N1 is specified, then SPA1 shouldn't be specified.
N2	Specifies the integer number of mesh lines per layer of material 2. Note that if N2 is specified, then SPA2 shouldn't be specified.
NAME1 **NAME2**	Specify the names of the regions composed of material 1 and material 2. You can conveniently use these names to modify material parameters and models on the **MATERIAL**, **MODELS**, **IMPACT**, and **MOBILITY** statements.

NA1	Specifies the ionized acceptor concentration in layers of material 1.
NA2	Specifies the ionized acceptor concentration in layers of material 2.
ND1	Specifies the ionized donor concentration in layers of material 1.
ND2	Specifies the ionized donor concentration in layers of material 2.
NU12	This is the number of grid lines in the graded region between the 1st and 2nd half cycle in the direction specified by the BOTTOM or TOP parameters.
NU21	This is the number of grid lines in the graded region between the 2nd and 1st half cycle in the direction specified by the BOTTOM or TOP parameters.
POLARIZ1 **POLARIZ2**	Enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. See Section 3.6.11 "Polarization in Wurtzite Materials".
POLAR1.CHARG **POLAR2.CHARG**	Specify polarization charge densities to replace those calculated using Equations 3-606 and 3-590.
POLAR1.SCALE **POLAR2.SCALE**	Specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the DBR statement. See Section 3.6.11 "Polarization in Wurtzite Materials".
QWELL1 **QWELL2**	Specifies that the region/cycle is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.
SPA1	Specifies the spacing between mesh lines in layers of material 1. Note that this should be smaller than or equal to THICK1.
SPA2	Specifies the spacing between mesh lines in layers of material 2. Note that this should be smaller than or equal to THICK2.
STRAIN1	Specifies the strain in the layer of material 1. (Negative strain is compressive).
STRAIN2	Specifies the strain in the layer of material 2. (Negative strain is compressive).
TOP	Specifies that the DBR is to be added at the top (starting at the minimum previously specified Y coordinate and extending in the negative Y direction) of the structure.
TH12	This is the thickness of grading region between 1st half cycle and 2nd half cycle going in the direction specified by the BOTTOM or TOP parameters.
TH21	This is the thickness of grading region between 2nd half cycle and 1st half cycle going in the direction specified by the BOTTOM or TOP parameters.
THICK1	Specifies the thickness of each layer of material 1.
THICK2	Specifies the thickness of each layer of material 2.

WELL1.CNBS **WELL2.CNBS** **WELL1.VNBS** **WELL2.VNBS**	Specify the number of bound states retained for calculation of radiative recombination or gain if the region/cycle is treated as a quantum well as specified by the QWELL1, QWELL2 parameters.
WELL1.FIELD **WELL2.FIELD**	When enabled, specify that the calculations of bound state energies should include the effects of the local field.
WELL1.GAIN **WELL2.GAIN**	Specify a constant scale factor multiplied by the calculated gain to give the net gain used for certain optoelectronic calculations.
WELL1.NX **WELL2.NX** **WELL1.NY** **WELL2.NY**	Specify the number of slices (WELL#.NX) and the number of samples per slice (WELL#.NY) are used in the solution of Schrodinger's equation to obtain the local bound state energies for calculation of radiative recombination or gain or both for certain optoelectronic models.
X1.COMP	Specifies the composition fraction x in layers of material 1.
X2.COMP	Specifies the composition fraction x in layers of material 2.
Y.START	Specifies the starting location of a DBR that begins with a layer of material 1 and alternates materials in an increasing Y direction.
Y.FINISH	Specifies the starting location of a DBR that begins with a layer of material 1 and alternates materials in an decreasing Y direction.
Y.FINAL **Y.END**	These are aliases for **Y.FINISH**.
Y1.COMP	Specifies the composition fraction y in layers of material 1.
Y2.COMP	Specifies the composition fraction y in layers of material 2.

Note: **DBR** statements can be intermixed with **X.MESH**, **Y.MESH**, and **REGION** statements.

22.10 DEFECTS

DEFECTS activates the band gap defect model and sets the parameter values. This model can be used when thin-film transistor simulations are performed using the TFT product.

Syntax

```
DEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
AMPHOTERIC	Logical	False	
CONTINUOUS	Logical	False	
DEVICE	Character		
DFILE	Character		
EGA	Real	0.4	eV
EGD	Real	0.4	eV
EP.AMPHOTERIC	Real	1.27	eV
EPT0.AMPHOTERIC	Real	1.3×10^{6}	s
EPBETA.AMPHOTERIC	Real	0.5	s
EU.AMPHOTERIC	Real	0.2	eV
EV0.AMPHOTERIC	Real	0.056	eV
F.DEFECTS	Character		
F.TFTACC	Character		
F.TFTDON	Character		
FILE.AMPHOTERIC	Character		
FILE.INT	Logical	False	
FILEX.AMPHOTERIC	Character		
FILEY.AMPHOTERIC	Character		
FILEZ.AMPHOTERIC	Character		
HCONC.AMPHOTERIC	Real	5×10^{21}	cm^{-3}
ID	Real		
INT_LIM1	Real	0	eV
INT_LIM2	Real	0	eV
LIMIT1	Real	0.0	eV

Parameter	Type	Default	Units
LIMIT2	Real	0.0	eV
MATERIAL	Character		
MODIFY	Logical	False	
NA.MIN	Real	1.0×10^{13}	cm^{-3}
ND.MIN	Real	1.0×10^{13}	cm^{-3}
NGA	Real	5.0×10^{17}	cm^{-3}/eV
NGD	Real	1.5×10^{18}	cm^{-3}/eV
NSISI.AMPHOTERIC	Real	2×10^{23}	cm^{-3}
NTA	Real	1.12×10^{21}	cm^{-3}/eV
NTD	Real	4.0×10^{20}	cm^{-3}/eV
NUM.AMPHOTERIC	Real	20	
NUMBER	Real	All	
NUMA	Real	12	
NUMD	Real	12	
NV0.AMPHOTERIC	Real	NV300	cm^{-3}
REGION	Real	All	
SIGGAE	Real	1.0×10^{-16}	cm^2
SIGGAH	Real	1.0×10^{-14}	cm^2
SIGGDE	Real	1.0×10^{-14}	cm^2
SIGGDH	Real	1.0×10^{-16}	cm^2
SIGMA.AMPHOTERIC	Real	0.19	eV
SIGN0.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGNP.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGP0.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGPN.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGTAE	Real	1.0×10^{-16}	cm^2
SIGTAH	Real	1.0×10^{-14}	cm^2

Parameter	Type	Default	Units
SIGTDE	Real	1.0×10^{-14}	cm^2
SIGTDH	Real	1.0×10^{-16}	cm^2
STRUCTURE	Character		
TAT.TRAP	Logical	False	
T0.AMPHOTERIC	Real	300	K
TFILE	Character		
WGA	Real	0.1	eV
WGD	Real	0.1	eV
WTA	Real	0.025	eV
WTD	Real	0.05	ev
X.MIN	Real		μm
X.MAX	Real		μm
Y.MIN	Real		μm
Y.MAX	Real		μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

The **DEFECTS** statement is used to describe the density of defect states in the band gap. You can specify up to four distributions, two for donor-like states and two for acceptor-like states. Each type of state may contain one exponential (tail) distribution and one Gaussian distribution.

AFILE	Specifies the file name where the acceptor state density distribution, as a function of energy, will be stored. You can examine this file by using TonyPlot.
AMPHOTERIC	Specifies the amphoteric defect model will be used.
CONTINUOUS	Specifies that the continuous defect integral model will be used.
DEVICE	Specifies which device the statement applies in mixed mode simulation. The synonym for this parameter is **STRUCTURE**.
DFILE	Specifies the file name where the donor state density distribution, as a function of energy, will be stored. You can examine this file by using TonyPlot.

EGA	Specifies the energy that corresponds to the Gaussian distribution peak for acceptor-like states. This energy is measured from the conduction band edge.
EGD	Specifies the energy that corresponds to the Gaussian distribution peak for donor-like states. This energy is measured from the valence band edge.
EP.AMPHOTERIC	Specifies the most probable potential defect energy.
EPT0.AMPHOTERIC	Specifies the time constant used in time amphoteric reflect generation model.
EPBETA.AMPHOTERIC	Specifies the exponential constant used in the amphoteric defect generation model.
EU.AMPHOTERIC	Specifies the defect electron correlation energy.
EV0.AMPHOTERIC	Specifies the characteristic energy.
F.DEFECTS	Specifies the name of a file containing a C-Interpreter function, describing the **DEFECTS** statement parameters as a function of position.
F.TFTACC	Specifies the name of a file containing a C-Interpreter function, describing the distribution of acceptor state densities as a function of energy.
F.TFTDON	Specifies the name of a file containing a C-Interpreter function, describing the distribution of donor state densities as a function of energy.
FILE.AMPHOTERIC	Specifies the file name where the dangling bond density of states distribution, as a function of energy, will be stored. If you specify FILEX.AMPHOTERIC, FILEY.AMPHOTERIC, and FILEZ.AMPHOTERIC, then the file will be saved at the coordinate closest to the one specified by FILEX.AMPHOTERIC, FILEY.AMPHOTERIC, and FILEZ.AMPHOTERIC.
FILE.INT	Specifies the integrated density of states with respect to energy will be stored in the AFILE, DFILE, and TFILE files for discrete defects.
FILEX.AMPHOTERIC	Specifies the X coordinate used in FILE.AMPHOTERIC.
FILEY.AMPHOTERIC	Specifies the Y coordinate used in FILE.AMPHOTERIC.
FILEZ.AMPHOTERIC	Specifies the Z coordinate used in FILE.AMPHOTERIC.
HCONC.AMPHOTERIC	Specifies the density of hydrogen.
ID	Specifies a reference number for use with the CIgetNodalVoid C-Interpreter callback function.
INT_LIM1	Specifies the lower limit for the numerical integration of the CONTINUOUS method.

INT_LIM2	Specifies the upper limit for the numerical integration of the CONTINUOUS method.
LIMIT1	Lower energy limit in TAT.TRAP integration range .
LIMIT2	Upper energy limit in TAT.TRAP integration range.
MATERIAL	Specifies which material from the table in Appendix B "Material Systems" will apply to the **DEFECTS** statements. If a material is specified, then the regions defined as being composed of that material will be affected.
MODIFY	Specifies that a previously defined DEFECTS statement will be modified. MODIFY must be specified along with the ID parameter and only applies to CONTINUOUS defects. The previously defined DEFECTS statement with the same ID number will be modified with the new values. Note: NUMA and NUMD cannot be modified.
NA.MIN	Specifies the minimum value of the acceptor trap DOS that will be used in the discrete **DEFECTS** model.
ND.MIN	Specifies the minimum value of the donor trap DOS that will be used in the discrete **DEFECTS** model.
NGA	Specifies the total density of acceptor-like states in a Gaussian distribution.
NGD	Specifies the total density of donor-like states in a Gaussian distribution.
NSISI.AMPHOTERIC	Specifies the density of Si-Si bonds.
NTA	Specifies the density of acceptor-like states in the tail distribution at the conduction band edge.
NTD	Specifies the density of donor-like states in the tail distribution at the valence band edge.
NUM.AMPHOTERIC	Specifies the number of energy levels used in the DOS integration for amphoteric defects.
NUMBER or **REGION**	Specifies the region index to which the **DEFECTS** statement applies.
NUMA	Specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.
NUMD	Specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.
NV0.AMPHOTERIC	Specifies the density of states of the valance-band tail exponential region extrapolated to the valence-band edge. If this parameter is not specified, then the valence band density of states (NV300 on the **MATERIAL** statement) will be used instead.

SIGGAE	Specifies the capture cross-section for electrons in a Gaussian distribution of acceptor-like states.
SIGGAH	Specifies the capture cross-section for holes in a Gaussian distribution of acceptor-like states.
SIGGDE	Specifies the capture cross-section for electrons in a Gaussian distribution of donor-like states.
SIGGDH	Specifies the capture cross-section for holes in a Gaussian distribution of donor-like states.
SIGMA.AMPHOTERIC	Specifies the defect pool width.
SIGN0.AMPHOTERIC	Specifies the electron capture cross-section for neutral defects.
SIGNP.AMPHOTERIC	Specifies the electron capture cross-section for positive defects.
SIGP0.AMPHOTERIC	Specifies the hole capture cross-section for neutral defects.
SIGPN.AMPHOTERIC	Specifies the hole capture cross-section for negative defects.
SIGTAE	Specifies the capture cross-section for electrons in a tail distribution of acceptor-like states.
SIGTAH	Specifies the capture cross-section for holes in a tail distribution of acceptor-like states.
SIGTDE	Specifies the capture cross-section for electrons in a tail distribution of donor-like states.
SIGTDH	Specifies the capture cross-section for holes in a tail distribution of donor-like states.
STRUCTURE	This is a synonym for **DEVICE**.
TAT.TRAP	Specifies that the **DEFECTS** will only be used in the Ielmini trap assisted tunneling models. Applies to insulators and semiconductors. With this flag, you must specify either NGA, EGA,WGA for acceptors or NGD, EGD, WGD for donors.
TFILE	Specifies the file name where the acceptor and donor state density distributions, as a function of energy referenced from E_v, will be stored. You can examine this file by using TonyPlot.
T0.AMPHOTERIC	Specifies the freeze-in temperature.
WGA	Specifies the characteristic decay energy for a Gaussian distribution of acceptor-like states.
WGD	Specifies the characteristic decay energy for a Gaussian distribution of donor-like states.

WTA	Specifies the characteristic decay energy for the tail distribution of acceptor-like states.
WTD	Specifies the characteristic decay energy for the tail distribution of donor-like states.
X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN**, and **Z.MAX**	Specify the bounding box for the **DEFECTS** statement.

TFT Example

The following statement lines specify distributed defect states which would typically be used for polysilicon.

```
DEFECTS NTA=1.E21 NTD=1.E21 WTA=0.033 WTD=0.049 \
        NGA=1.5E15 NGD=1.5E15 EGA=0.62 EGD=0.78 \
        WGA=0.15 WGD=0.15 SIGTAE=1.E-17 \
        SIGTAH=1.E-15 SIGTDE=1.E-15 SIGTDH=1.E-17 \
        SIGGAE=2.E-16 SIGGAH=2.E-15 SIGGDE=2.E-15 \
        SIGGDH=2.E-16
```

22.11 DEGRADATION

DEGRADATION specifies parameters for MOS device degradation modeling.

Syntax

```
DEGRADATION <params>
```

Parameter	Type	Default	Units
F.NTA	Character		
F.NTD	Character		
F.SIGMAE	Character		
F.SIGMAH	Character		
GF.BARREMI	Real	0.8	eV
GF.BARRPASS	Real	0.8	eV
GF.EB	Real	1.5	eV
GF.DEBDF	Real	-5.6×10^{-8}	cm
GF.HBAROMEGA	Real	0.075	eV
GF.KTHRM	Real	0.0	Hz
GF.NUEMI	Real	1×10^{12}	Hz
GF.NUPASS	Real	1×10^{12}	Hz
GF.NUPHONON	Real	1×10^{11}	Hz
KC.AE0	Real	1.5	eV
KC.AMBIENTH	Real	1.0	cm^{-3}
KC.BETA	Real	0.0	
KC.BETAPERP	Real	0.0	cm/V
KC.BETAPARL	Real	0.0	cm/V
KC.COUPLED	Logical	False	
KC.DELTPARL	Real	0.0	[Q=1] cm
KC.DELTPERP	Real	0.0	[Q=1] cm
KC.E.HCCOEF	Real	0.0	cm^2/A
KC.E.FNCOEF	Real	0.0	cm^2/A
KC.E.RHOHC	Real	1.0	
KC.E.RHOFN	Real	1.0	

Parameter	Type	Default	Units
KC.H.HCCOEF	Real	0.0	cm^2/A
KC.H.FNCOEF	Real	0.0	cm^2/A
KC.H.RHOHC	Real	1.0	
KC.H.RHOFN	Real	1.0	
KC.KF0	Real	10^{-5}	s
KC.KR0	Real	3.0×10^{-9}	s
KC.NIT0	Real	0.0	cm^2
KC.RHOPARL	Real	1.0	
KC.RHOPERP	Real	1.0	
KC.SIHTOT	Real	10^{12}	cm^2
KC.T0	Real	300.0	Kelvin
KC.VOLUME	Real	0.0	cm^3
NTA	Real	1.0×10^{11}	cm^{-2}
NTA.MP	Real	0.0	cm^{-2}
NTA.SP	Real	0.0	cm^{-2}
NTD	Real	1.0×10^{10}	cm^{-2}
NTD.MP	Real	0.0	cm^{-2}
NTD.SP	Real	0.0	cm^{-2}
PL.E.ALPHA	Real	0.5	
PL.E.EA	Real	0.25	eV
PL.E.NU0	Real	1.0	s^{-1}
PL.H.ALPHA	Real	0.5	
PL.H.EA	Real	0.25	eV
PL.H.NU0	Real	1.0	s^{-1}
RD.AE	Real	1.5	eV
RD.AESLOPE	Real	-5.6×10^{-8}	[Q=1] cm
RD.AEVAR	Real	0.1	eV
RD.COUPLED	Logical	True	

Parameter	Type	Default	Units
RD.E.HCCOEF	Real	0.0	cm^2/A
RD.E.FNCOEF	Real	0.0	cm^2/A
RD.E.OFFSET	Real	0.0	eV
RD.H.HCCOEF	Real	0.0	cm^2/A
RD.H.FNCOEF	Real	0.0	cm^2/A
RD.H.OFFSET	Real	0.0	eV
RD.INVHCOEF	Real	0.0	cm^3
RD.KF0	Real	10^{-5}	s^{-1}
RD.KR0	Real	3×10^{-9}	cm^3s^{-1}
RD.NIT0	Real	0.0	cm^2
RD.SIHTOT	Real	10^{12}	cm^2
SIGMAE	Real	1.0×10^{-17}	cm^2
SIGMAH	Real	1.0×10^{-16}	cm^2

Description

F.NTA	Specifies the file name for a C-Interpreter function that specifies arbitrary density distribution of the acceptor-like traps on the interface.
F.NTD	Specifies the file name for a C-Interpreter function that specifies arbitrary density distribution of the donor-like traps on the interface.
F.SIGMAE	Specifies the name of a file containing a C-Interpreter function specifying the distribution of acceptor trap cross-sections.
F.SIGMAH	Specifies the name of a file containing a C-Interpreter function specifying the distribution of donor trap cross-sections.
NTA	Specifies the uniform acceptor-like trap density on the interface.
NTD	Specifies the uniform donor-like trap density on the interface.
SIGMAE	Specifies the acceptor-like trap capture cross section.
SIGMAH	Specifies the donor-like trap capture cross section.

General Framework Model

GF.BARREMI	Hydrogen emission energy barrier.
GF.BARRPASS	Hydrogen capture energy barrier.
GF.EB	Passivated bond binding energy.
GF.DEBDF	Field dependence of passivated bond binding energy.
GF.KTHRM	Thermal attempt frequency.
GF.HBAROMEGA	Phonon energy.
GF.NUPHONON	Phonon attempt frequency.
GF.NUEMI	Hydrogen emission attempt frequency.
GF.NUPASS	Hydrogen capture attempt frequency.
NTA.MP	Many particle model acceptor interface trap density.
NTA.SP	Single particle model acceptor interface trap density.
NTD.MP	Many particle model donor interface trap density.
NTD.SP	Single particle model donor interface trap density.

Kinetic Model

KC.AE0	Activation energy.
KC.AMBIENTH	Scaling hydrogen density in repassivation model.
KC.BETA	Energy parameter for kinetic model.
KC.BETAPERP	Energy scaling parameter.
LKC.BETAPARL	Energy scaling parameter.
KC.COUPLED	Enables strong implicit coupling scheme between equations.
KC.DELTPARL	Energy scaling parameter.
KC.DELTPERP	Energy scaling parameter.
KC.E.HCCOEF	Hot electron current scaling factor.
KC.E.FNCOEF	Tunnel electron current scaling factor.
KC.E.RHOHC	Hot electron current scaling exponent.
KC.E.RHOFN	Tunnel electron current scaling exponent.
KC.H.HCCOEF	Hot hole current scaling factor.
KC.H.HCCOEF	Tunnel hole current scaling factor.
KC.H.RHOHC	Hot hole current scaling exponent.

KC.H.RHOFN	Tunnel hole current scaling exponent.
KC.COUPLED	Enable strong coupling in kinetic model.
KC.KF0	Depassivation rate constant.
KC.KR0	Repassivation rate constant.
KC.NIT0	Initial density of depassivated Si-H bonds.
KC.RHOPARL	Energy scaling exponent.
KC.RHOPERP	Energy scaling exponent.
KC.SIHTOT	Total passivated and depassivated dangling bond density.
KC.T0	Effective temperature.
KC.VOLUME	Repassivation volume.

Reaction-Diffusion Model

RD.AE	Median activation energy for depassivation process.
RD.AESLOPE	Coefficient for field dependent contribution to activation energy.
RD.AEVAR	Energy width of depassivation activation energy distribution.
RD.COUPLED	Enables strong implicit coupling scheme between equations.
RD.E.HCCOEF	Coefficient for electron hot channel current.
RD.E.FNCOEF	Coefficient for electron Fowler-Nordheim current.
RD.E.OFFSET	Energy offset for electron hot channel current.
RD.H.HCCOEF	Coefficient for hole hot channel current.
RD.H.FNCOEF	Coefficient for hole Fowler-Nordheim current.
RD.H.OFFSET	Energy offset for hole hot channel current. \\
RD.INVHCOEF	Coefficient of inversion hole contribution to depassivation rate.
RD.KF0	Depassivation rate constant.
RD.KR0	Repassivation rate constant.
RD.NIT0	Initial density of depassivated Si-H bonds.
RD.SIHTOT	Total passivated and depassivated dangling bond density.

Power-Law Model

PL.E.ALPHA	Exponent for power law dependence of degradation lifetime of electron traps.
PL.E.EA	Activation energy for degradation rate.
PL.E.NU0	Rate constant for degradation rate.
PL.H.ALPHA	Exponent for power law dependence of degradation lifetime of hole traps.
PL.H.EA	Activation energy for degradation rate.
PL.H.NU0	Rate constant for degradation rate.

MOS Interface State Example

```
DEGRADATION NTA=1.E-12 SIGMAE=5.E-18
```

This syntax defines a density of acceptor states uniformly distributed along the silicon-oxide interface. The trapping cross section is also defined. Traps will be filled by gate current in transient mode simulations leading to a shift in device parameters.

22.12 DEVDEGBULKTRAP

DEVDEGBULKTRAP specifies parameters for traps created by Stress Induced Leakage Current.

Syntax

```
DEVDEGBULKTRAP <params>
```

Parameter	Type	Default	Units
ELEC.DEPTH	Real	-999.0	eV
HOLE.DEPTH	Real	-999.0	eV
NT.N	Real	0.0	cm^{-3}
NT.P	Real	0.0	cm^{-3}
SIGMAP.N	Real	1.0e-14	cm^2
SIGMAN.P	Real	1.0e-15	cm^2
SIGMAT.N	Real	1.0e-16	cm^2
SIGMAT.P	Real	1.0e-14	cm^2
TAU.N	Real	1.0e300	s
TAU.P	Real	1.0e300	s

Description

ELEC.DEPTH	Energy below conduction band of acceptor traps.
HOLE.DEPTH	Energy above valence band of donor traps.
NT.N	Sets the uniform density of electron traps (acceptor-like) in the gate insulator.
NT.P	Sets the uniform density of hole traps (donor-like) in the gate insulator.
SIGMAP.N	Coefficient of term giving capture rate of conduction band electrons into donor-like trap.
SIGMAN.P	Coefficient of term giving capture rate of valence band holes into acceptor-like trap.
SIGMAT.N	Coefficient of term giving capture rate of conduction band electrons into acceptor-like trap.
SIGMAT.P	Coefficient of term giving capture rate of valence band electrons into donor-like trap.
TAU.N	Lifetime for electron de-trapping from acceptor-like traps to conduction band.
TAU.P	Lifetime for hole de-trapping from donor-like traps to valence band.

Example

```
DEVDEGBULKTRAP NT.P=1.0E16  NT.N=1.0E17  TAU.N=10000.0 TAU.P=10.0 \
               SIGMAT.N=1.0E-15 \
               SIGMAT.P=1.0E-15  ELEC.DEPTH=1.5 HOLE.DEPTH=1.5
```

22.13 DOPING

DOPING specifies doping profiles either analytically or from an input file. The alias for this statement is PROFILE.

Syntax

```
DOPING              <prof>[<psp>][<bound>][<loc>][<sprea>>][OUTFILE=
<fn>][<trps>]
```

Parameter	Type	Default	Units
1D.PROC	Logical	False	
2D.ASCII	Logical	False	
A.MAX	Real	360.0	Degrees
A.MIN	Real	0.0	Degrees
ACCEPTOR	Logical	False	
ACTIVE	Logical	True	
ALUMINUM	Logical	False	
ANTIMONY	Logical	False	
ARSENIC	Logical	False	
ASCII	Logical	False	
ASPECT.RATIO	Logical		
ATHENA	Logical	False	
ATHENA.1D	Logical	False	
BACKDOPE	Real	none	cm^{-3}
BORON	Logical	False	
C.MULT	Real	1.0	
CHARACTERISTIC	Real		μm
CHEMICAL	Logical	False	
CONCENTRATION	Real	0	cm^{-3}
DEGEN.FAC	Real		
DEVICE	Character		1
DIRECTION	Character	y	
DONOR	Logical	False	
DOP.OFFSET	Real	0	cm^{-3}

Parameter	Type	Default	Units
DOP.SEED	Integer	-10	
DOP.SIGMA	Real	0.0	cm^{-3}
DOP.XMIN	Real		
DOP.XMAX	Real		
DOP.YMIN	Real		
DOP.YMAX	Real		
DOSE	Real		cm^{-2}
ERFC	Logical	False	
ERFC.LATERAL	Logical	False	
E.LEVEL	Real		eV
F.COMPOSIT	Character		
F.DOPING	Character		
F3.DOPING	Character		
FILE	Character		
GAUSSIAN	Logical	False	
IMATER	Character	None	
IN.FILE	Character		
INDIUM	Logical	False	
INFILE	Character		
INAME	Character	None	
INT.LIN	Logical	False	
INT.LOG	Logical	True	
INT.OPTM	Logical	False	
IREGION	Character	None	
JUNCTION	Real		μm
LAT.CHAR	Real		μm
MASTER	Logical	False	
MATERIAL	Character		
METAL	Logical	False	

Parameter	Type	Default	Units
MODIFY	Logical	False	
N.COLUMN	Integer		
N.TYPE	Logical	False	
N-TYPE	Logical	False	
N.OFFSET	Real	0.0	cm^{-3}
N.PEAK	Real	0	cm^{-3}
NAME	Character		
NET	Logical	False	
NOROLLOFF	Logical	False	
NOXROLLOFF	Logical	False	
NOYROLLOFF	Logical	False	
NOZROLLOFF	Logical	False	
OUTFILE	Character		
OUTSIDE	Logical	False	
OX.CHARGE	Logical	False	
PEAK	Real		
PHOSPHORUS	Logical	False	
P.COLUMN	Integer		
P.TYPE	Logical	False	
P-TYPE	Logical	False	
R.MAX	Real	Device Radius	μm
R.MIN	Real	0.0	μm
RATIO.LATERAL	Real	0.7	
RESISTI	Real		Ω·cm
REGION	Integer	All	
SIGN	Real		cm^2
SIGP	Real		cm^2
SLICE.LAT	Real		μm
SPECIES1	Logical	False	

Parameter	Type	Default	Units
SPECIES2	Logical	False	
SPECIES3	Logical	False	
START	Real	0	
STRUCTURE	Character		
SUPREM3	Logical	False	
TMA.SUPREM3	Logical	False	
TAT.TRAP	Logical	False	
TAUN	Real		
TAUP	Real		
TRAP	Logical	False	
UNIFORM	Logical	False	
WIDTH	Real	width of structure	μm
X1	Real	0	μm
X2	Real	0	μm
XY	Logical	True	
XZ	Logical	False	
X.CHAR	Real		μm
X.COLUMN	Integer		
X.COMP	Logical	False	
X.DIR	Logical	False	
X.ERFC	Logical	False	
X.FLIP	Logical	False	
X.LEFT	Real	left of structure	μm
X.MAX	Real		μm
X.MIN	Real		μm
X.SCALE	Logical	True	
X.RIGHT	Real	right of structure	μm
XERFC.LAT	Logical	False	
XY.RATIO	Real	0.7	

Parameter	Type	Default	Units
Y.BOTTOM	Real	bottom of structure	μm
Y.CHAR	Real		μm
Y.COLUMN	Integer		
Y.FLIP	Logical	False	
Y.JUNCTI	Real		μm
Y1	Real	0	μm
Y2	Real	0	μm
YX	Logical	True	
YZ	Logical	False	
Y.COMP	Logical	False	
Y.DIR	Logical	False	
Y.MAX	Real		μm
Y.MIN	Real		μm
Y.SCALE	Logical	True	
Y.TOP	Real	top of structure	
Z1	Real	0	μm
Z2	Real	0	μm
ZY	Logical	False	
ZX	Logical	False	
Z.BACK	Real		
Z.DIR	Logical	False	
Z.FRONT	Real		
Z.MAX	Real		μm
Z.MIN	Real		μm
Z.SCALE	Logical	True	
ZERFC.LAT	Logical	False	
ZLAT.CHAR	Real		
ZRATIO.LAT	Real	0.7	
ZSLICE.LAT	Real		

Description

The **DOPING** statement is used to define doping profiles in the device structure. Typically a sequence of **DOPING** statements is given each building on the others.

MODIFY	Specifies that a previously defined DOPING statement will be modified. The previous values of doping that were defined on previous DOPING statements will be discarded and replace with the specified value or value from a C-Interpreter file.
OUTFILE	Specifies the name of an output file for use with **REGRID**. The first **DOPING** statement should use this parameter to specify a filename. All doping information from the first **DOPING** statement and all subsequent **DOPING** statements in the input file are saved to this file. The **REGRID** statement can read this file and interpolate doping on the new grid.

Note: The file from OUTFILE cannot be used in TonyPlot or in the **MESH** statement.The **SAVE** command should be used after all of the **DOPING** commands required to save a file for plotting the doping profile.

Statement Applicability Parameters

DEVICE	Specifies which device the statement applies to in the MixedMode simulation. The synonym for this parameter is **STRUCTURE**.
MATERIAL	Restricts the applicability of the statement to regions of the specified material.
NAME	Restricts the applicability of the statement to regions with the specified name.
REGION	Restricts the applicability of the statement to regions with the specified region number.
STRUCTURE	This is a synonym for **DEVICE**.

Note: If you don't specify the DEVICE, MATERIAL, NAME and REGION parameters, the **DOPING** statement will apply to all regions.

Interface Doping Profile Location Parameters

You can use these parameters with the parameters NAME, MATER, and REGION to describe interfaces between regions to be doped with an analytic profile. The NAME, MATER, and REGION parameters are used to describe the regions that receive the doping while the INAME, IMATER, and IREGION parameters describe regions incident on the regions receiving the doping where the doping is to be placed (i.e., along the interface).

The locations of of these interface profiles may also be restricted by the location parameters X.MAX, X.MIN, Y.MAX, and Y.MIN.

IMATER	Specifies the material of all neighboring regions where an interface profile is placed.

INAME	Specifies the name of a neighboring region where an interface profile is placed.
IREGION	Specifies the region number of the a neighboring region where the interface profile is placed.
OUTSIDE	Specifies that interface regions will placed at all interfaces with the outside of the simulation domain.

Analytical Profile Types

These parameters specify how Atlas will generate a doping profile from analytical functions.

DOP.SIGMA	Specifies the variance for random gaussian dopant distribution.
DOP.SEED	Specifies a seed value for random gaussian dopant distribution.
ERFC	Specifies the use of a ERFC analytical function to generate the doping profile. If ERFC is specified, the following parameters must also be specified: • Polarity parameters N.TYPE or P.TYPE • One of the following groups of profile specifications: • Group 1:CONCENTRATION and JUNCTION • Group 2:DOSE and CHARACTERISTIC • Group 3:CONCENTRATION and CHARACTERISTIC
GAUSSIAN	Specifies the use of a gaussian analytical function to generate the doping profile. If GAUSSIAN is specified, the following parameters must also be specified: • Polarity parameters N.TYPE or P.TYPE • One of the following groups of profile specifications: • Group 1:CONCENTRATION and JUNCTION • Group 2:DOSE and CHARACTERISTIC • Group 3:CONCENTRATION and CHARACTERISTIC
UNIFORM	Specifies the use of uniform (constant) analytical functions to generate the doping profile. If UNIFORM is specified, the N.TYPE, P.TYPE, and CONCENTRATION parameters must be specified. Doping is introduced into a box defined by the boundary parameters (see "Boundary Conditions" on page 1104). The box by default includes the entire region.
F.COMPOSIT	Specifies the name of a file containing a C-Interpreter function specifying the spatial distribution of composition fractions.
F.DOPING	Specifies the name of a file containing a C-Interpreter function specifying the spatial distribution of dopants.
F3.DOPING	Specifies the name of a file containing a C-Interpreter function specifying the spatial distribution of dopants for a 3D device.

File Import Profile Types

These parameters specify how Atlas will generate a doping profile from a file. Files can be user-defined or from process simulation.

1D.PROC	This is an alias for **TMA.SUPREM3**.
2D.ASCII	Specifies that a 2D doping profile, which is defined on a rectangular Cartesian grid, should be loaded from a file specified by INFILE. 2D.ASCII must be specified along with either the N.TYPE, P.TYPE or NET parameters. This first column of the file should contain the X coordinates. The second column should contain the Y coordinates. The third column should contain the doping data.

ASCII	There has two separate meanings. The first meaning is that it specifies the file type as ASCII when it's combined with other format parameters. The second meaning is when this parameter is used alone, it specifies ASCII data files containing concentration versus depth information. The alias for this parameter is **1D.PROC**.
	
	In the second meaning, this parameter must be written in the form:
	
	`ASCII INFILE=<filename>`
	
	where filename is the name of the ASCII input file. The data file must be in the following format:
	
	`depth concentration`
	`depth concentration`
	`depth concentration`
	`...`
	
	where depth is specified in μm and concentration is specified in cm^{-3}. An input file name, a dopant type, and boundary parameters must be specified. Positive concentrations are assumed to be n-type and negative concentrations are assumed to be p-type unless the N.TYPE or P.TYPE parameters are used.

ATHENA.1D	Specifies that the doping file is a Athena 1D export file. This parameter acts in a similar way to the SSuprem3 parameter.
ATHENA	Reads 2D doping information from Athena standard structure file (SSF) or PISCES-II format files. The PISCES-II format is an obsolete file format. Doping information obtained from this file will be added to each point of the current Atlas mesh. If points in the Atlas mesh do not coincide with points in the Athena mesh, doping for Atlas mesh points will be interpolated from Athena doping information. If this profile type is used, the INFILE parameter must also be specified.

Note: The X.STRETCH function available in previous versions of Atlas has been replaced by similar more powerful functions in DevEdit. This feature should no longer be used in Atlas.

FILE	This is an alias for **INFILE**.
IN.FILE	This is a synonym for **INFILE**. The alias for this parameter is **FILE**.
INFILE	Specifies the name of the appropriate input file. The synonym for this parameter is **IN.FILE**.
MASTER	Specifies that the INFILE is written in the Silvaco standard structure file (SSF) format. This file format is the default output format of Athena and SSuprem3. This parameter is typically combined with the SSUPREM3, ATHENA 1D, or ATHENA parameters. If neither of these are used the default is SSuprem3.
N.COLUMN	Specifies which column of an 2D ASCII table corresponds to the net donor concentration when 2D.ASCII is specified.
P.COLUMN	Specifies which column of an 2D ASCII table corresponds to the net acceptor concentration when 2D.ASCII is specified.
SUPREM3	Specifies the INFILE was produced by SSuprem3 in standard structure file (SSF) format or binary or an ASCII export format. If this profile type is used, an input file name, a dopant, and boundary parameters must be specified. When SSuprem3 produces an output file, the doping profiles are stored by dopant. Therefore, a dopant parameter should be specified in order to import the correct doping profile into Atlas. If a specific dopant is not specified the total donors and acceptor concentrations are loaded.
TMA.SUPREM3	Specifies that the file specified by INFILE is in TMA SUPREM3 binary format. The alias for this parameter is **1D.PROC**.

Note: Files containing 1D doping profiles can be loaded into Blaze, Blaze3D, Device 3D, or S-Pisces. Files containing 2D doping profiles can only be loaded into S-Pisces.

X.COLUMN	Specifies which column of an 2D ASCII table corresponds to the X coordinate value when 2D.ASCII is specified.
Y.COLUMN	Specifies which column of an 2D ASCII table corresponds to the Y coordinate value when 2D.ASCII is specified.

1D Profile Modifications

These parameters are used to modify the concentrations in 1D profiles.

C.MULT	Acts as a multiplier in 1D ASCII dopant profiles.
DOP.OFFSET	Subtracts a background doping value from the Athena or SSuprem3 doping. The alias for this parameter is **N.OFFSET**.

Dopant Type Specification Parameters

These parameters give information about the dopant species or type to be used in the specified profile. Different profile types require different profile specifications.

ACTIVE	Specifies that for the dopant specified the active concentration as opposed to the chemical concentration is added. This is true by default. Files from Athena or SSuprem3 contain both active and chemical concentrations for each dopant.
ALUMINUM	Specifies that aluminum dopant information be extracted from an imported file.
ANTIMONY	Specifies that antimony dopant information be extracted from an imported file.
ARSENIC	Specifies that arsenic dopant information be extracted from an imported file.
BORON	Specifies that boron dopant information be extracted from an imported file.
C.MULT	Acts as a multiplier in 1D ASCII dopant profiles.
CHEMICAL	Specifies that the chemical concentration (as opposed to the active concentration) will be read from the imported file. This is generally not advisable.
DOP.OFFSET	Subtracts a background doping value from the Athena or SSuprem3 doping. The alias for this parameter is **N.OFFSET**.
E.LEVEL	Sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge. For donors, it is relative to the valence band edge.
INDIUM	Specifies that indium dopant information be extracted from an imported file.
METAL	Specifies that the **DOPING** statement will define metal atomic concentration used in calculation of electrode quenching.
NET	Specifies that net doping information be extracted from an imported file. This is usually not advisable. It is better to use several DOPING statements to extract data dopant by dopant from a file.
N.OFFSET	This is an alias for **DOP.OFFSET**.
N.TYPE, **N-TYPE**, **DONOR**	Specifies an n-type or donor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.
OX.CHARGE	Specifies a fixed oxide charge profile. Oxide charge can only be placed in any insulator region. The N.TYPE/P.TYPE parameters are not used hence a negative concentration implies a negative charge.
P.TYPE, **P-TYPE**, **ACCEPTOR**	Specifies a p-type or acceptor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.

PHOSPHORUS	Specifies that phosphorus dopant information be extracted from an imported file.
RESISTI	This can be used to specify resistivity (in units of Ohm.cm) as an alternative to using the CONC parameter. Tabulated values are used to convert from the value of RESISTI to doping Concentration (in units of cm^{-3}). The tables are related to the ARORA mobility model at 300K and the result depends on whether the dopant has been specified as a donor or an acceptor. At high doping levels, the resistivity can also depend on the particular dopant species. But this is not taken into account in this model.
SPECIES1	Stipulates that the doping profile is to apply to generic ion species 1.
SPECIES2	Stipulates that the doping profile is to apply to generic ion species 2.
SPECIES3	Stipulates that the doping profile is to apply to generic ion species 3.
TRAP	Specifies that the dopant concentration is to be treated as a trap state density.
X.COMP	Specifies a profile of composition fraction x as defined in Appendix B "Material Systems". This profile can be used to change the composition fraction of cations in ternary and quaternary materials over a spatial distribution.
Y.COMP	Specifies a profile of composition fraction y as defined in Appendix B "Material Systems". This profile can be used to change the composition fraction of anions in ternary and quaternary materials over a spatial distribution.

Vertical Distribution Parameters

CHARACTERISTIC	Specifies the principal characteristic length of the implant. For Gaussians, the characteristic length is equal to the square root of two times the standard deviation. If this parameter is left unspecified, the principal characteristic can be computed from the values of the • Polarity Parameters • Boundary Parameters • Concentration and Junction parameters The alias for this parameter is **Y.CHAR**.
CONCENTRATION	Specifies the peak concentration when a Gaussian profile is used. If this parameter is not specified, peak concentration may be computed from the values of the polarity, boundary, DOSE, or RESISTI, CHARACTERISTIC concentrations. When a uniform profile is specified, the CONCENTRATION parameter sets the value of the uniform doping level. Concentrations must be positive. The alias for this parameter is **N.PEAK**.
DOSE	Specifies the total dose for a Gaussian profile.

JUNCTION	Specifies the location of a p-n junction within the silicon region of a Gaussian profile. When JUNCTION is specified, the characteristic length is computed by examining the doping at a point halfway between the end of the constant box and the given depth. The JUNCTION location is evaluated considering all previous DOPING statements only. This means that in some cases the order of **DOPING** statements is important.
N.PEAK	This is an alias for **CONCENTRATION**.
PEAK	Specifies the depth location of the peak doping in a Gaussian profile.
Y.CHAR	This is an alias for **CHARACTERISTIC**. See Equation 22-3.
Y.JUNCTI	This is an alias for **JUNCTION**. See Equation 22-3.

$$N(Y) = \mathrm{PEAK} \cdot \exp\left[-\left(\frac{Y}{\mathrm{Y.CHAR}}\right)^2\right] \qquad 22\text{-}3$$

Location Parameters

DIRECTION	Specifies the axis along which a one-dimensional profile is directed in a two-dimensional device (x or y). DIR=Y will typically be used for implanted profiles.
REGION	Specifies the region number where doping is to be added.
START	Specifies the depth in the Y direction where the profile should start.

Lateral Extent Parameters

These parameters must be specified when a 1D doping profile type is used (MASTER, GAUSSIAN, ASCII, ERFC, or UNIFORM). These boundary parameters set the doping boundaries before applying lateral spreading. This is equivalent to setting implant mask edges.

WIDTH	Specifies the extent of the profile in the X direction. Specifying WIDTH is equivalent to specifying X.MAX such that X.MAX=X.MIN+WIDTH.
R.MIN, **R.MAX**, **A.MIN**, and **A.MAX**	For an Atlas 3D device created using the MESH CYLINDRICAL option, these parameters restrict the radial and angular positions respectively of the analytical doping profile. The principal direction of the analytical doping profile is the Z direction in this case.
X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MAX**, and **Z.MIN**	Specify the x, y and z bounds of a rectangular shaped region or box in the device. The dopant profile within this box will be constant with a density equal to the value specified by the CONC parameter. Outside this box the profile decreases from the peak, CONC, with distance, from the box along the principal axes. The relationship between the concentration, outside the box, to distance will depend upon the profile type as specified by the GAUSSIAN, MASTER, ATHENA, ATLAS, and UNIFORM parameters.
X.LEFT, **X.MIN**	Specifies the left boundary of a vertical 1D profile.

X.RIGHT, **X.MAX**	Specifies the right boundary of a vertical 1D profile.
Y.BOTTOM, **Y.MAX**	Specifies the bottom boundary of a horizontal 1D profile.
Y.TOP, **Y.MIN**	Specifies the top boundary of a horizontal 1D profile.
Z.BACK, **Z.MIN**	Specifies the back boundary of a z directed 1D or 2D profile.
Z.FRONT, **Z.MAX**	Specifies the front boundary of a z directed 1D or 2D profile.

Lateral Distribution Parameters

These parameters specify how a vertical 1D profile is extended outside the box defined by the lateral extent parameters.

BACKDOPE	Specifies the value to which the doping profile specified from the 1D profile will roll-off to outside its lateral and vertical extents. If this value is not specified, then the last doping value in the ASCII file is used as the background doping level, regardless of windowing in the Y direction. If NOXROLLOFF is used, then BACKDOPE will be ignored for the X direction. If NOYROLLOFF is used, then BACKDOPE will be ignored for the Y direction. If NOROLLOFF is used, then BACKDOPE will be completely ignored.
ERFC.LATERAL	Specifies that the complementary error function will be used to calculate the lateral falloff of doping level in the X direction. If you set the X.DIR flag, then this flag will apply to the Y direction instead. If you set ERFC for the analytical doping profile, then ERFC.LATERAL will be automatically enabled by default. If you set GAUSSIAN or UNIFORM, then ERFC.LATERAL will be disabled by default and the falloff will follow a Gaussian profile. The aliases for this parameter are **X.ERFC** and **XERFC.LAT**.
LAT.CHAR	Specifies the characteristic length of the lateral profile. If this parameter is not specified, the characteristic length is defined by: $CL = RL \times OCL$ 22-4 where: • CL is the lateral characteristic length in the X direction. • RL is the value of RATIO.LATERAL. • OCL is the characteristic length of the original profile in the Y direction. The alias for this parameter is **X.CHAR**.
NOXROLLOFF	Causes the doping level to abruptly change to zero outside the x-limits.
NOYROLLOFF	Causes the doping level to abruptly change to zero outside the y-limits.
NOROLLOFF	This is the same as setting both NOXROLLOFF and NOYROLLOFF.
NOZROLLOFF	Causes the doping levels to abruptly change to zero outside the z-limits.
RATIO.LATERAL	This is the ratio of characteristic lengths in the X and Y directions.

SLICE.LAT	Specifies the point at which the doping is examined to compute the characteristic length of a Gaussian profile after JUNCTION has been specified. The default for this parameter is a point halfway between the end of the constant box and the given depth.
X.CHAR	This is an alias for **LAT.CHAR**.
X.ERFC	This is an alias for **ERFC.LAT**.
XERFC.LAT	This is an alias for **ERFC.LAT**.
XY.RATIO	This is an alias for **RATIO.LATERAL**.
ZLAT.CHAR	Specifies the characteristic length of the lateral profile in the Z direction. See also **LAT.CHAR**.
ZERFC.LAT	Specifies that the complementary error function will be used to calculate the lateral falloff of doping level in the Z direction in Atlas3D. If you specify ERFC as the analytical doping profile, then ZERFC.LAT will be enabled by default. The parameters ZLAT.CHAR and ZRATIO.LAT control the degree of lateral rolloff in the Z direction.
ZRATIO.LAT	This is used analogously to RATIO.LATERAL but applies to lateral spreading in the Z direction. See also **LAT.CHAR**.
ZSLICE.LAT	This is similar to SLICE.LAT but applies to profiles in the Z direction.

Trap Parameters

E.LEVEL	Sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge, for donors it is relative to the valence band edge.
DEGEN.FAC	Specifies the degeneracy factor of the trap level used to calculate the density.
SIGN	Specifies the capture cross section of the trap for electrons.
SIGP	Specifies the capture cross section of the trap for holes.
TAT.TRAP	Causes the trap level to be used in the ITAT/RTAT model. The trap must be located in an insulator or wide bandgap semiconductor quantum barrier.
TAUN	Specifies the lifetime of electrons in the trap level.
TAUP	Specifies the lifetime of holes in the trap level.

Note: See Section 22.66 “TRAP” for more information on each of these parameters

Angled Distribution Parameters

X1, **Y1**, **X2**, **Y2**	Specify the X and Y coordinates of the ends of the line segment describing the location of the specified angled profile.
X.DIR and **Y.DIR**	Specify the direction in which the angled profile is extended.

In Device 3D, (see Chapter 7 "3D Device Simulator" for more information about this simulator), dopants can be added along angled segments in the XY plane. The start and ending coordinates of the line segment are defined by the X1, Y1, X2, and Y2 parameters. You can then specify whether a 2D profile is extended from the line segment in either the X or the Y direction. To specify it, set the X.DIR or Y.DIR parameter. You can then specify a 2D doping profile in the same DOPING statement. The profile can be an analytic, SUPREM, ASCII, or SUPREM4.

The dopants are placed relative to the defined line segment, according to the setting of X.DIR or Y.DIR. If X.DIR is specified, then the effective Y coordinate of the profile is the device Z coordinate and the effective X coordinate of the profile is the distance in the X direction from the center of the line segment. No dopants are added if the device Y coordinate is outside of the Y coordinates of the line segment. If Y.DIR is specified, then the effective Y coordinate of the profile is the device Z coordinate and the effective X coordinate of the profile is the distance in the Y direction from the of the line segment. No dopants are added if the device X coordinate is outside the X coordinates of the line segment.

Analytical Doping Definition Example

This example describes a 1.0μm n-channel MOSFET using Gaussian source and drain profiles. The lateral extent of the source is given by X.RIGHT=2. This corresponds to the mask edge for the implant. Sub-diffusion is determined by an error function based on the RATIO.LAT and JUNCTION parameters. For both source and drain, the n+ doping is added to the uniform p-well concentration to ensure a junction depth of 0.3μm.

```
DOP UNIF CONC=1E16 P.TYPE

DOP GAUSS CONC=9E19 N.TYPE X.RIGHT=2 JUNC=0.3 RATIO.LAT=0.6
ERFC.LAT

DOP GAUSS CONC=9E19 N.TYPE X.LEFT=3 JUNC=0.3 RATIO.LAT=0.6
ERFC.LAT
```

1D Athena or SSuprem3 Interface Example

This example reads a 1D Athena bipolar profile and adds it to a uniform substrate concentration. The base and emitter doping are loaded from the same file by specifying the impurity required for each area (boron in the base and arsenic in the emitter).

The DOPOFF parameter is used to subtract the substrate arsenic dopant out of the 1D profile that is loaded since this dopant was already specified in the substrate doping line.

Versions of SSuprem3 later than 5.0 use standard structure files as default when saving data. These can be loaded in Atlas with the syntax below by replacing ATHENA.1D with SSUPREM3.

```
#   SUBSTRATE

DOP REGION=1 UNIF CONC=1E16 N.TYPE

#   BASE

 DOP REGION=1 MASTER ATHENA.1D BORON RATIO.LAT=0.7 INF=bipolar.exp

 #   EMITTER

 DOP REGION=1 MASTER ATHENA.1D ARSENIC RATIO.LAT=0.6 \

 INF=bipolar.exp X.LEFT=12.0 X.RIGHT=13.0 DOPOFF=1e16
```

Athena Doping Interface Example

This example demonstrates how to use an SSF format Athena file to interpolate doping onto a Atlas grid and save the doping information for subsequent regrid operations.This is an alternative to the Athena/Atlas interface, which is described in Section 2.6.1 “Interface From Athena”.

```
DOPING ATHENA MASTER INFILE=NMOS.DOP OUTFILE=NMOS.DOP

REGRID DOPING ABS LOG RATIO=4 OUTFILE=NMESH1.STR

DOPFILE=NMOS.DOP
```

3D Doping Definition Example

The following example illustrates the formation of a Gaussian highly doped n-type area in a three-dimensional structure.

```
DOPING GAUSS N.TYPE CONC=1e20 PEAK=0.0 CHAR=0.2 X.LEFT=0.5 \

      X.RIGHT=1.0 Z.LEFT=0.5 Z.RIGHT=1.0
```

For a cylindrical structure you can use the following syntax. In this case, it is for complementary error function doping.

```
doping erfc start=0.0 char=0.005 ratio.lat=0.1 conc=1.0e20 n.type
r.min=0.01 a.min=45.0 a.max=225.0
```

3D Doping From ASCII 1D File

You can read a 1D doping profile from an ASCII file and apply it to the entire or part of a 3D device. The X.DIR, Y.DIR, or Z.DIR parameters specify the direction of the profile as applied in the 3D device. The starting position of the doping profile will be set by the relevant X.MIN, Y.MIN, or Z.MIN parameter. The ending position of the doping profile will be the minimum of the relevant X.MIN, Y.MIN, or Z.MIN plus the spatial extent of the doping profile in the ASCII file, or the X.MAX, Y.MAX, Z.MAX parameter or a physical boundary of the device. You can specify a general parallelogram in the plane perpendicular to the direction of doping direction. To specify it, use the appropriate combination of X1, X2, Y1, Y2, Z1, Z2, X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN, Z.MAX. For example, we consider the Y.DIR parameter specified. We can then specify a parallelogram in the XZ plane as shown in Figure 22-2.

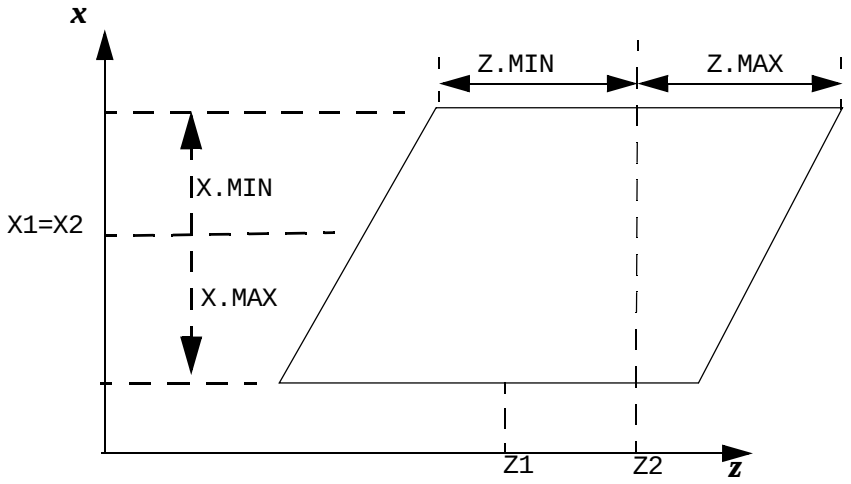


Figure 22-2: Parallelogram Geometry Example

In the figure, two sides are parallel to the Z axis. Z1 and Z2 specify the Z coordinate of the midpoints of these two sides. Z.MIN and Z.MAX together specify the lengths of the sides, and Z.MIN will be negative and equal to -Z.MAX. (In fact, the positions are evaluated as Z1+Z.MIN and Z1+Z.MAX for one side and Z2+Z.MIN and Z2+Z.MAX for the other. Therefore, having Z1 and Z2 and so on refer to the side midpoint is just a convenient convention.)

You must specify X1 and X2 as having the same value, and specify X.MIN and X.MAX the side lengths the same way as for Z.MIN and Z.MAX.

Setting Z1 and Z2 to the same value will result in a rectangle. By then changing X1 and X2 so they are not equal will result in a parallelogram with sides parallel to the X axis. If a different direction of doping variation is defined, then a parallelogram can be set up in the plane perpendicular to it in analogously to the above example.

Outside the specified parallelogram the doping level will be either the value of the last point in the ASCII file, or the value specified by the BACKDOPE parameter. A smooth transition of the doping from the value inside the parallelogram to the value outside can be applied elsewhere using the LAT.CHAR or RATIO.LAT parameters.

X1, **Y1**, **X2**, **Y2**, **Z1**, **Z2**	By convention, specify the midpoints of the sides making up the parallelogram.
X.MIN, **X.MAX**	Determine the X coordinate of the start and end points of the placement of the doping profile read in from an ASCII file (if you specify X.DIR). Otherwise, they specify the lateral extent of the parallelogram sides defining the location of the doping in a plane perpendicular to the specified direction of the doping variation. In this case, they are added to X1 and to X2 to determine the coordinates of the ends of sides.
Y.MIN, **Y.MAX**, **Z.MIN**, **Z.MAX**	These are the same as X.MIN and X.MAX except they apply to the y and Z directions respectively.

Example

```
doping ascii inf=ydop.dat y.dir n.type x1=6.0 x2=4.0 x.min=-2
x.max=2

z.min=3.0 z.max=7.0 y.min=2.0 y.max=10.0 lat.char=0.005
backdope=0.0
```

This will apply the doping profile specified in ydop.dat as n-type doping in the Y direction between the positions y=2 microns and y=10 microns (unless the length range in ydop.dat is less than this distance). In the XZ plane, it will be applied to a parallelogram with sides parallel to the X direction of length 4 microns. The Z coordinates of these sides are 3.0 microns and 7.0 microns. The doping will transition from the value inside the parallelogram to 0.0 outside on a length scale of 0.005 microns.

3D Doping From 2D Athena Master Files

In a 2D Athena master file, the doping profile is defined over a box in two dimensions. This doping is put over 2D-sections normal to the plane of a parallelogram to be defined in the doping statement. This parallelogram must lie in one of the XY, YZ, and XZ planes and have two edges parallel to one of the coordinate axes.

As the mesh in the Athena master file doesn't necessarily coincide with the three-dimensional mesh in Atlas, an interpolation routine is required to import the doping in Atlas. You can then choose between a linear and a logarithmic interpolation algorithm.

Here's a list of the parameters that are used for this kind of doping.

ASPECT.RATIO	Specifies the aspect ratio of the doping will be preserved when loaded into the doping parallelogram.
DOP.XMAX	Specifies the maximum X coordinate of the doping in the Athena file to be loaded into the doping parallelogram.
DOP.XMIN	Specifies the minimum X coordinate of the doping in the Athena file to be loaded into the doping parallelogram.
DOP.YMAX	Specifies the maximum Y coordinate of the doping in the Athena file to be loaded into the doping parallelogram.
DOP.YMIN	Specifies the minimum Y coordinate of the doping in the Athena file to be loaded into the doping parallelogram.
INT.LIN	Specifies that a linear interpolation is to be used to import doping from Athena into Atlas.
INT.LOG	Specifies that the Athena doping is imported in Atlas by using a logarithmic interpolation algorithm.
INT.OPTM	Enables an optimized interpolation routine, which attempts to reduce CPU time due to the doping interpolation.
LAT.CHAR	Defines the characteristic length, CL, of the lateral spreading (in the y>y2 and y<y1 planes). If RATIO.LAT is used instead, then the characteristic length is assumed to come from this parameter by the height of the parallelogram (i.e., CL=RL X OCL, OCL being the height of the parallelogram (y2-y1 in this case). If both LAT.CHAR and RATIO.LAT aren't specified, then no lateral spreading is done.
XY	Specifies that the parallelogram containing doping in Atlas3D is in the xy plane.
YX	This is a synonym for **XY**.
YZ	Specifies that a parallelogram containing doping in Atlas3D is in the YZ plane.
ZY	This is a synonym for **YZ**.
XZ	Specifies that a parallelogram containing doping in Atlas3D is in the XZ plane.
ZX	This is a synonym for **ZX**.
X1, **Y1**, **X2**, **Y2**, **Z1**, **Z2**	Specify an average segment in one of the three coordinate planes defining the orientation of the parallelogram.

X.DIR, **Y.DIR**, **Z.DIR**	Specify that a parallelogram containing doping in Atlas3D has two edges in the X direction, Y direction, and Z direction.
X.FLIP	Specifies that the orientation of the Athena file X axis will be flipped.
X.MIN, **X.MAX**	Define the minimum and maximum lateral extent of a parallelogram along the X direction lying in the xy or zx plane, starting from its average segment (they must be specified with X1 and X2). If a parallelogram is defined in the YZ plane, X.MIN can then be used to specify the initial coordinate to start the doping along the X direction.
X.SCALE	Specifies that the Athena file doping will be scaled in the X direction.
Y.FLIP	Specifies that the orientation of the Athena file Y axis will be flipped.
Y.MIN, **Y.MAX**	Define the minimum and maximum lateral extent of a parallelogram along the Y direction lying in the XY or YZ plane, starting from its average segment (they must be specified with Y1 and Y2). If a parallelogram is defined in the zx plane, Y.MIN can then be used to specify the initial coordinate to start the doping along the Y direction.
Y.SCALE	Specifies that the Athena file doping will be scaled in the Y direction.
Z.MIN, **Z.MAX**	Define the minimum and maximum lateral extent of a parallelogram along the Z direction lying in the yz or zx plane, starting from its average segment (they must be specified with Z1 and Z2). If a parallelogram is defined in the xy plane, Z.MIN can then be used to specify the initial coordinate to start the doping along the Z direction.
Z.SCALE	Specifies that the Athena file doping will be scaled in the Z direction.

The following three cases, which correspond to the parallelograms in the xy, yz and zx planes, describe how the doping from the Athena master file is put into the Atlas structure .

First Case: Parallelogram In The XY Plane

SUB-CASE A: Parallelogram along the X direction.

Sections of the Atlas structure in zx planes are considered, which intersect for y1<y<y2, the segment (x_a, x_b) inside the parallelogram. Each of these sections defines a box where the 2D doping is imported from Athena. Particularly, all the coordinates in Athena are scaled and translated so that the edge (x_c, x_d) of the Athena box containing the doping coincide with the segment (x_a, x_b) of the Atlas box. This is done for all the sections inside the parallelogram. See Figure 22-3.

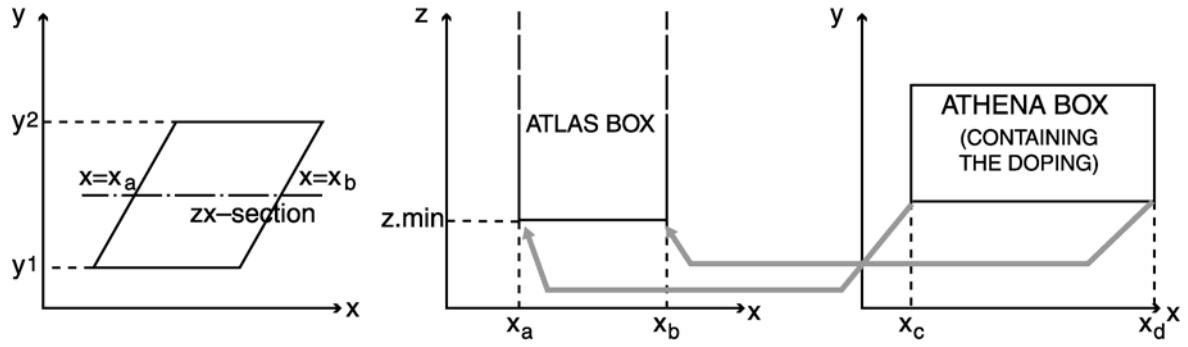


Figure 22-3: Parallelogram in the XY plane in the X direction and doping from the Athena2D master file

Examples:

```
doping athena master inf=athena.str boron xy \
x1=2.0 x2=3.0 y1=1.5 y2=4.0 x.dir \
x.min=-2.0 x.max=1.0 z.min=0.6 lat.char=0.05 int.log int.optm
```

For the Z coordinate, `z.min` is used to specify the minimum value where to start putting the doping into the zx-sections (which properly defines the segment (x_a, x_b) in this plane).

In addition, a lateral spreading can be partially accomplished by extending the parallelogram into the planes, y>y2 and y<y1, where the doping is spread out according to a Gaussian law.

SUB-CASE B: Parallelogram along the Y direction

Sections of the Atlas structure are considered in the YZ planes. A transformation of coordinates in Athena is accomplished in order to place the Athena segment (y_c, y_d) into the Atlas one (y_a, y_b). See Figure 22-4.

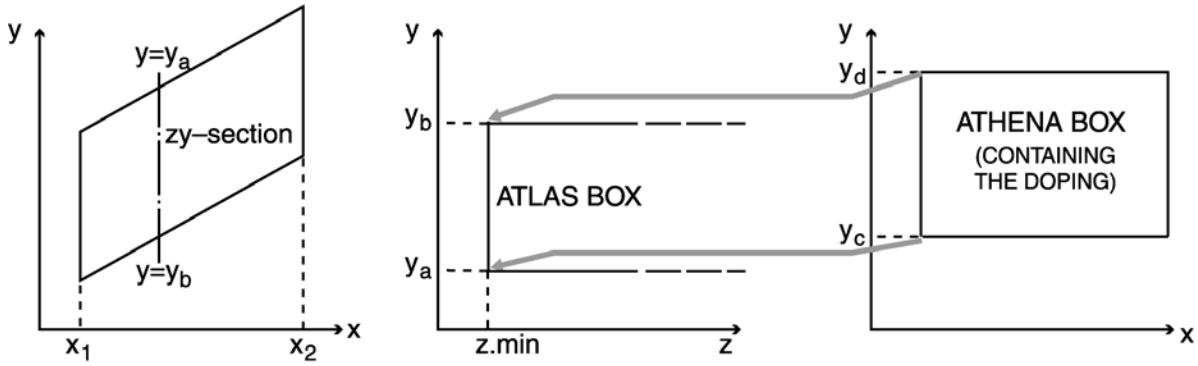


Figure 22-4: Parallelogram in the XY plane in the Y direction and doping from the Athena2D master file

Examples:

```
doping athena master inf=athena.str boron xy \
x1=2.0 x2=3.0 y1=1.5 y2=4.0 y.dir \
y.min=-2.0 y.max=1.0 z.min=0.6 lat.char=0.05 int.lin int.optm
```

Second Case: YZ Plane

The YZ plane containing the parallelogram is defined by the YZ parameter (or ZY) in the DOPING statement.

If parallelograms along the Z direction are defined (specify Z.DIR in doping statement), the doping from Athena is put over sections in the zx planes. If the parallelograms along the Y direction are defined (using Y.DIR in DOPING statement), the doping from Athena is put over sections in the xy planes.

The minimum X coordinate in Atlas to start adding doping can be specified by X.MIN in DOPING statement.

Examples:

```
doping athena master inf=athena.str boron zy z.dir \
z1=0.3 z2=0.6 y1=0.35 y2=0.25 z.min=-0.2 z.max=0.6 \
x.min=0.2 ratio.lat=0.05


doping athena master inf=athena.str boron zy y.dir \
z1=0.3 z2=0.6 y1=0.35 y2=0.25 y.min=-0.2 y.max=0.6 \
x.min=0.2 ratio.lat=0.05
```

Third Case: ZX Plane

The ZX plane containing the parallelogram is defined by the ZX parameter (or XZ) in DOPING statement. If parallelograms along the Z direction are defined (using Z.DIR in the DOPING statement), the doping from Athena is placed over sections in the YZ planes.

If the parallelograms along the X direction are defined (using X.DIR in DOPING statement), the doping from Athena is placed over the sections in the XY planes.

The minimum Y coordinate in Atlas to start adding doping can be specified by Y.MIN in doping statement.

Examples:

```
doping athena master inf=athena.str boron int.lin \
xz z.dir z1=0.45 z2=0.55 x1=0.2 x2=0.7 x.min=-0.4 \
x.max=0.2 y.min=0.0 lat.char=0.02
doping athena master inf=athena.str boron int.lin \
xz x.dir z1=0.4 z2=0.8 x1=0.2 x2=0.7 z.min=-0.4 \
z.max=0.2 y.min=0.5 ratio.lat=0.5
```

22.14 DOSEXTRACT

DOSEXTRACT specifies the grain boundary and interface trap density of states as a function of energy [156] are to be extracted from IV and CV files and saved to log files.

Syntax

```
DOSEXTRACT [<parameters>]
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
CVFILE	Character		
DONOR	Logical	False	
EG300	Real		eV
EI	Real		eV
EMAX	Real		eV
GRAIN.SIZE	Real		μm
GRAINFILE	Character		
INTFILE	Character		
IVFILE	Character		
LENGTH	Real		μm
MIN.DENSITY	Real	1	cm^{-2}
MU	Real		cm^2/Vs
NUM.DIVISIONS	Real	25	
NUM.GRAINS	Real		
OUTFILE	Character		
TEMPERATURE	Real	300	K
TOX	Real		μm
TSI	Real		μm
VDRAIN	Real		V
WIDTH	Real		μm

Description

ACCEPTOR	Specifies that acceptor traps are to be extracted.
CVFILE	Specifies the file name which contains the capacitance/gate voltage data.
DONOR	Specifies that donor traps are to be extracted.
EG300	Specifies the value of bandgap of the material. EMAX will be calculated from this value if it is specified along with EI.
EI	Specifies the intrinsic energy level of the material.
EMAX	Specifies the maximum energy seperation from the intrinsic energy level (EI) that should be calculated.
GRAIN.SIZE	Specifies the size of the grains.
GRAINFILE	Specifies the file name where the grain boundary DOS distribution, as a function of energy, will be stored.
INTFILE	Specifies the file name where the interface DOS distribution, as a function of energy, will be stored.
IVFILE	Specifies the file name that contains the drain current/gate voltage data.
LENGTH	Specifies the x-direction length under the gate.
MIN.DENSITY	Specifies the minimum trap density to be used for the grain boundary traps.
MU	Specifies the majority carrier mobility.
NUM.DIVISIONS	Specifies the number of divisions used from the oxide-silicon interface to the substrate.
NUM.GRAINS	Specifies the number of grains in the device.
OUTFILE	Specifies the file name where the grain boundary and interface DOS distribution, as a function of energy, will be stored.
TEMPERATURE	Specifies the temperature used in the DOS extraction.
TOX	Specifies the oxide thickness.
TSI	Specifies the silicon thickness.
VDRAIN	Specifies the drain voltage used when the data in CVFILE and IVFILE were created.
WIDTH	Specifies the width of the device.

Example

```
dosextract  tox=0.1  tsi=0.05  width=25  length=5  vd=0.1  mu=50
num.grains=4 ei=0.54

eg=1.08 cvfile=cv.dat ivfile=iv.dat donor grainfile=grain.dat

intfile=int.dat
```

22.15 ELECTRODE

ELECTRODE specifies the locations and names of electrodes in a previously defined mesh.

Syntax

```
ELECTRODE NAME=<en> [NUMBER=<n>] [SUBSTRATE] <pos> <reg>
```

Parameter	Type	Default	Units
A.MAX	Real		degrees
A.MIN	Real		degrees
BOTTOM	Logical	False	
FLOATING	Logical	False	
HELMHOLTZ	Logical	True	
IX.HIGH	Integer	right side of structure	
IX.LOW	Integer	left side of structure	
IX.MAX	Integer	right side of structure	
IX.MIN	Integer	left side of structure	
IY.MAX	Integer	bottom side of structure	
IY.MIN	Integer	top side of structure	
IY.HIGH	Integer	bottom of structure	
IY.LOW	Integer	top of structure	
IZ.MAX	Integer		
IZ.MIN	Integer		
IZ.HIGH	Integer		
IZ.LOW	Integer		
LEFT	Logical	False	
LENGTH	Real	length of structure	μm
MATERIAL	Character	Contact	
MODIFY	Logical	False	
NAME	Character		
NUMBER	Integer	defined #(electrodes)+ 1	
R.MAX	Real		μm
R.MIN	Real		μm
RIGHT	Logical	False	

Parameter	Type	Default	Units
SUBSTRATE	Logical	False	
THERMAL	Logical	False	
TOP	Logical	False	
X.MAX	Real	right side of structure	μm
X.MIN	Real	left side of structure	μm
Y.MAX	Real	Y.MIN	μm
Y.MIN	Real	top of the structure	μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

FLOATING	Marks an enclosed electrode as a floating semiconductor region. See Section 3.5.10 "Floating Semiconductor Regions".
HELMHOLTZ	If set to False, this excludes the corresponding region from the solution of the Helmholtz equation. This is commonly used to prevent Atlas from solving Helmholtz equation inside an electrode that overlaps with an otherwise rectangular grid defining the solution domain.
MATERIAL	Specifies a material for the electrode (see Table B-1). This material will be displayed in TonyPlot. The electrode material can also be used to define the electrode thermal characteristics (thermal conductivity) and optical characteristics (complex index of refraction). Setting the material here does not apply any electrical property such as workfunction to the terminal. All electrical properties of electrodes are set on the CONTACT statement.
MODIFY	To mark an electrode read from a Standard Structure File as a floating semiconductor region specify both MODIFY and FLOATING flags. See Section 3.5.10 "Floating Semiconductor Regions".

NAME	Specifies an electrode name. The electrode name can be referenced by other Atlas statements to modify characteristics of the specified electrode. For reference by the **CONTACT** or **THERMCONTACT** statements any valid character string can be used and properly cross-referenced. But when setting voltages, currents and charge from the **SOLVE** statement certain electrode names are recognized in a simplified syntax. By prepending the electrode name with "V" for voltage, "I" for current and "Q" for charge, you can directly and conveniently set the electrode bias, current or charge respectively. For example: `SOLVE VGATE=1.0` can be used to assign 1 volt bias to the electrode named "GATE". In such a manner, the following list of names can be used to set voltage, current or charge: GATE, FGATE, CGATE, NGATE, PGATE, and VGG. The following list of names can be used to assign only voltage or current: DRAIN, SOURCE, BULK, SUBSTRATE, EMITTER, COLLECTOR, BASE, ANODE, CATHODE, WELL, NWELL, PWELL, CHANNEL, GROUND, NSOURCE, PSOURCE, NDRAIN, PDRAIN, VDD, VSS, VEE, and VBBVCC.
NUMBER	Specifies an electrode number from 1 to 50. Electrode numbers may be specified in any order. If NUMBER is not specified, electrodes will be automatically numbered in sequential order. This parameter cannot re-number electrodes already defined in Atlas or other programs.
pos	This is one of the position parameters described below.
reg	This is a set of the region parameters described on the next page.
SUBSTRATE	Places the specified electrode at the bottom of the device and names the electrode, substrate.
THERMAL	Specifies that the electrode is treated as a boundary condition for heatflow simulation using Giga.

Position Parameters

BOTTOM or **SUBSTRATE**	Specifies that the electrode is positioned along the bottom of the device.
LEFT	Specifies that the electrode starts at the left-hand edge of the device. The electrode will be positioned from left to right along the top of the device.
RIGHT	Specifies that the electrode starts at the right-hand edge of the device. The electrode will be positioned from right to left along the top of the device.
TOP	Specifies that the electrode is positioned along the top of the device.

Region Parameters

Device coordinates may be used to add regions to both rectangular and irregular meshes. In either case, boundaries must be specified with the A.MAX, A.MIN, R.MAX, R.MIN, X.MAX, X.MIN, Y.MAX, Y.MIN, Z.MAX, and Z.MIN parameters.

LENGTH	Specifies the length of the electrode in the X direction. It is not necessary to specify X.MIN, X.MAX, and LENGTH. If two of these parameters are specified, the value of the third parameter will be calculated.
A.MAX	Specifies the maximum angle of a 3D cylindrical electrode.
A.MIN	Specifies the minimum angle of a 3D cylindrical electrode.
R.MAX	Specifies the maximum radius of a 3D cylindrical electrode.
R.MIN	Specifies the minimum radius of a 3D cylindrical electrode.
X.MAX	Specifies the maximum x-boundary of the electrode.
X.MIN	Specifies the minimum x-boundary of the electrode.
Y.MAX	Specifies the maximum y-boundary of the electrode.
Y.MIN	Specifies the minimum y-boundary of the electrode.
Z.MIN	Specifies the minimum z-boundary of the electrode.
Z.MAX	Specifies the maximum z-boundary of the electrode.

Note: If an electrode has been shortened to fit the current mesh, a warning message will be generated by Atlas. Electrode placement can only occur at previously defined mesh nodes.

Grid Indices

As an alternative to the region parameters, you can use grid indices to define a region only when the mesh is rectangular although these parameters are not recommended. To define a region with a rectangular mesh:

1. Use the **X.MESH** and **Y.MESH** statements to specify grid indices.
2. Use the IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters to specify x and y values.

IX.HIGH	Specifies the maximum x-value of the grid index. The alias for this parameter is **IX.MAX**.
IX.LOW	Specifies the minimum x-value of the grid index. The alias for this parameter is **IX.MIN**
IY.HIGH	Specifies the maximum y-value of the grid index. The alias for this parameter is **IY.MAX**.
IY.LOW	Specifies the minimum y-value of the grid index. The alias for this parameter is **IX.MIN**. Nodes, which have x and y grid indices, between IX.LOW and IX.HIGH and between IY.LOW and IY.HIGH are designated electrode nodes. Normally, horizontal planar electrodes will be used. In this case, IY.LOW equals IY.HIGH.

IZ.HIGH	Specifies the maximum z-value of the grid index. The alias for this parameter is **IZ.MAX**.
IZ.LOW	Specifies the minimum z-value of the grid index. The alias for this parameter is **IX.MIN**.
IX.MAX, **IX.MIN**, **IY.MIN**, **IZ.MAX**, **IZ.MIN**	These are aliases for **IX.HIGH**, **IX.LOW**, **IY.HIGH**, **IY.LOW**, **IZ.HIGH**, and **IZ.LOW**.

MOS Electrode Definition Example

This example defines electrodes for a typical MOS structure.

```
ELEC X.MIN=0.5 LENGTH=0.25 NAME=gate

ELEC LENGTH=0.25 Y.MIN=0 LEFT NAME=source

ELEC LENGTH=0.25 Y.MIN=0 RIGHT NAME=drain

ELEC SUBSTRATE
```

3D Electrode Definition Example

The following example illustrates electrode definition for a 3D structure.

```
ELECTRODE NAME=ANODE X.MIN=0.5 X.MAX=1.0 \

          Z.MIN=0.5 Z.MAX=1.0
```

Note: In Atlas, it is preferred to refer to **ELECTRODE** by name rather than number. Some functions, however, may require the electrode number. The syntax, **MODELS** PRINT, can be used to echo electrode numbers to the run-time output.

22.16 ELIMINATE

ELIMINATE terminates mesh points along lines in a rectangular grid defined within Atlas in order to reduce the local mesh density.

Syntax

```
ELIMINATE X.DIRECTION|Y.DIRECTION [<boundary>]
```

Parameter	Type	Default	Units
COLUMNS	Logical	False	
IX.LOW	Integer		
IX.HIGH	Integer		
IY.LOW	Integer		
IY.HIGH	Integer		
ROWS	Logical	False	
X.DIRECTION	Logical	False	
X.MIN	Real		μm
X.MAX	Real		μm
Y.DIRECTION	Logical	False	
Y.MIN	Real		μm
Y.MAX	Real		μm

Description

The **ELIMINATE** statement is used to remove points along every other line within the chosen range. Successive eliminations of the same range remove points along every fourth line. For horizontal elimination, the vertical bounds should be decreased by one at each re-elimination of the same region. For vertical elimination, the horizontal bounds should be decreased by one at each re-elimination of the same region.

ROWS	This is an alias for **X.DIRECTION**.
COLUMNS	This is an alias for **Y.DIRECTION**.
X.DIRECTION	Eliminates points along horizontal lines.
Y.DIRECTION	Eliminates points along vertical lines.

Boundary Parameters

X.MIN, **X.MAX**, **Y.MIN**, and **Y.MAX**	Specify the location of the boundaries of an area in coordinates, where the elimination is applied.

The following are provided for backward compatibility only. Their use is not recommended.

IX.HIGH	Specifies the mesh line number high boundary in the X direction.
IX.LOW	Specifies the mesh line number low boundary in the X direction.
IY.HIGH	Specifies the mesh line number high boundary in the Y direction.
IY.LOW	Specifies the mesh line number low boundary in the Y direction.

Substrate Mesh Reduction Example

This example removes vertical points between the depth of 10μm and 20μm.

```
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=8
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=7
```

Note: In some cases, applications of the **ELIMINATE** statement can cause internal inconsistencies in the mesh. When this occurs, an error message will appear, warning you that there are triangles that are not associated with any region.

Note: The **ELIMINATE** statement only works on meshes defined using Atlas syntax. You can eliminate mesh points on arbitrary meshes in DevEdit

22.17 EXTRACT

EXTRACT statements are used to measure parameters from both log and solution files.

Note: These commands are executed by DeckBuild. This statement is documented in the DeckBuild User's Manual.

Terminal Current Extraction Example

By default, **EXTRACT** works on the currently open log file. For example, to extract peak drain current from a run immediately after solution, type:

```
LOG OUTF=myfile.log

SOLVE .........

EXTRACT NAME="peak Id" max(i."drain")
```

Extraction Example from Previously Generated Results

To extract the same data from a previously run simulation, use the INIT parameter.

```
EXTRACT INIT INFILE="myfile.log"

XTRACT NAME="peak Id" max(i."drain")
```

Solution Quantities Extraction Example

To use **EXTRACT** with solution files, use the INIT parameter. To find the integrated number of electrons in a 1D slice at X=1.0, use:

```
SAVE OUTF=mysolve.str    or    SOLVE ...... MASTER OUTF=mysolve.str

     EXTRACT INIT INFILE="mysolve.str"

     EXTRACT NAME="integrated e-" area from curve(depth,n.conc \

             material="Silicon" mat.occno=1 x.val=1.0)
```

Note: **EXTRACT** commands are generally case sensitive.

22.18 EYE.DIAGRAM

EYE.DIAGRAM specifies that an eye diagram should be generated from the specified log file. An eye diagram is created by dividing up transient data in periods of fixed size and then overlaying it.

Syntax

```
EYE.DIAGRAM INFILE OUTFILE PERIOD + [OPTIONAL PARAMETERS]
```

Parameter	Type	Default	Units
INFILE	Character		
OUTFILE	Character		
PERIOD	Real		s
T.START	Real		s
T.STOP	Real		s

Description

INFILE	Specifies the input log file. This should contain data from a transient simulation.
OUTFILE	Specifies the file output file for the eye diagram.
PERIOD	Specifies the window period.
T.START	Specifies the inital time value to be used. The default valueis the first time point in the input log file.
T.STOP	Secifies the final time value to be used. The default value is the last time point in the input log file.

Examples

```
EYE.DIAGRAM INFILE=laser.log OUTFILE=eye.log PERIOD=2e-10
T.START=1.5e-9
```

In this example, laser.log is a log file from the transient simulation of a buried heterostructure laser when it is biased using a pseudo-random bit sequence. The data values from time=1.5ns onwards are used to create the eye diagram with a data period of 0.2ns. The results are then saved to the log file, eye.log. Figure 22-5 shows the eye diagram in TonyPlot.

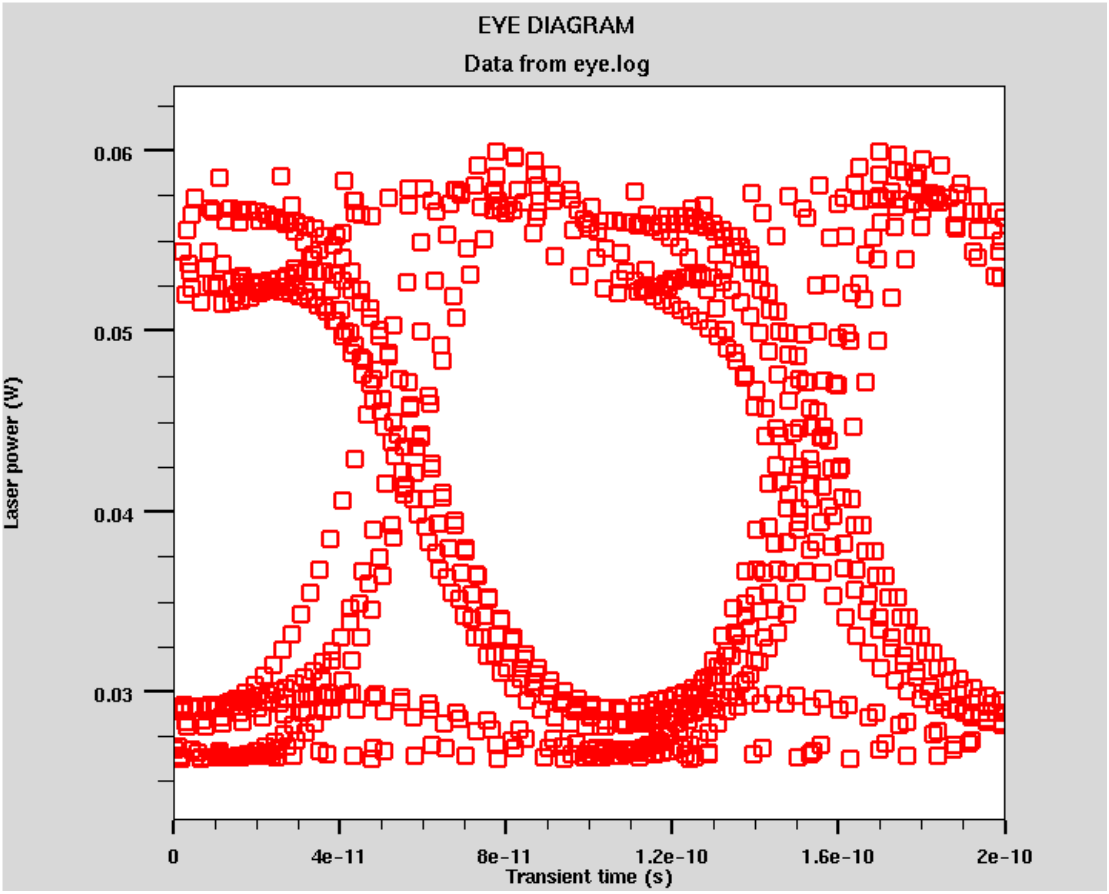


Figure 22-5: Eye diagram for BH Laser diode with a psuedo-random bit sequence input

22.19 FDX.MESH, FDY.MESH

FD<n>**.MESH** specifies the location of grid lines along the <n>-axis in a rectangular mesh used in FDTD simulation. The syntax is equivalent for X and Y directions.

Syntax

```
FDX.MESH NODE=<n> LOCATION=<n>

FDX.MESH SPACING=<n> LOCATION=<n>

FDY.MESH NODE=<n> LOCATION=<n>

FDY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE	Specifies the mesh line index. These mesh lines are assigned consecutively.
LOCATION	Specifies the location of the grid line.
SPACING	Specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

FDTD Mesh Example

This syntax defines a mesh of 33×33 covering the area bounded by (0.3,0.0) to (2.4,1.0).

```
FDX.M n=1 l=0.3

FDX.M n=33 l=2.4

FDY.M n=1 l=0.0

FDY.M n=33 l=1.0
```

Note: The mesh defined in these statements for the FDTD Solution is entirely separate from the electrical device simulation mesh defined on the **MESH** statement. The FDTD Mesh is defined in the coordinate system relative to the direction of beam propagation (see Figure 11-1).

Setting Locally Fine Grids Example

This example shows how to space grid lines closely around a topological feature such as a junction at y=0.85 microns.

```
LY.MESH  LOC=0.0  SPAC=0.2

LY.MESH  LOC=0.85 SPAC=0.01

LY.MESH  LOC=2    SPAC=0.35
```

22.20 FOURIER

FOURIER enables you to do Fourier transformations.

Syntax

```
FOURIER INFILE OUTFILE + [ OPTIONAL PARAMETERS ]
```

Parameter	Type	Default	Units
COMPLEX.VALUES	Logical	False	
FUNDAMENTAL	Real		Hz
INFILE	Character		
INTERPOLATE	Logical	False	
MAX.HARMONIC	Real		Hz
NO.DC	Logical	False	
NUM.SAMPLES	Integer	64	
OUTFILE	Character		
T.START	Real		s
T.STOP	Real		s

Description

The FOURIER statement performs a Fast Fourier Transform on log file data.

COMPLEX.VALUES	Specifies that the real and imaginary components are saved to file as well as the magnitude and phase values The synonym for this parameter is REAL.VALUES.
FUNDAMENTAL	Specifies the fundamental frequency. If the fundamental frequency is specified, T.STOP is set to T.START+1/FUNDAMENTAL. If this is not specified, the fundamental frequency is set to (T.STOP-T.START)/NUM.SAMPLES.
INFILE	Specifies the input log file. This should contain data from a transient simulation.
INTERPOLATE	Specifies that the input data should be linearly interpolated such that data at uniform time steps are created. Interpolation of data can introduce addition (inaccurate) harmonic values into the Fourier transform. INTERPOLATE must be used if the log file contains non-uniform time steps.
MAX.HARMONIC	Specifies the maximum harmonic frequency that the Fourier transform should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.
NO.DC	Specifies that the DC component will not be saved to the log file.

NUM.SAMPLES	Specifies the number of discrete samples. This should be an integer power of 2 (i.e., 2^n where n is a positive integer). The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case the number of samples is set to the nearest integer power of 2 which will generate this frequency.
OUTFILE	Specifies the output file for the Fourier transform data.
T.START	Specifies the start of time data to be used for the Fourier transform. The default value is the first time point in the input log file.
T.STOP	Specifies the end of time data to be used for the Fourier transform. The default value is the last time point in the input log file.

Example 1

In this example, the transient data previously written to log file, hemt1.log, is transformed from the time domain to the frequency domain. The fundamental frequency is set to 0.5 GHz, and harmonic frequencies up to 16 GHz are calculated. Since the data in hemt1.log has non-uniform time steps, the INTERPOLATE flag must be enabled. The complex values as well as the magnitude and phase values are stored in fftout1.log.

```
FOURIER INFILE=hemt1.log FUNDAMENTAL=5e8 MAX.HARMONIC=1.6E10 \
        OUTFILE=fftout1.log INTERPOLATE COMPLEX.VALUES
```

Example 2

In this example, the log file values between 31.25 ps and 2ns are transformed into the frequency domain. The fundamental frequency is automatically determined from the time period set by T.START and T.STOP. The data values from this time period are interpolated into 64 samples, giving a maximum harmonic frequency of 15.5 GHz. The magnitude and phase values are then stored in fftout2.log.

```
FOURIER     INFILE=hemt1.log     T.START=3.125e-11     T.STOP=2e-9
NUM.SAMPLES=64 \
            OUTFILE=fftout2.log INTERPOLATE
```

22.21 GO

GO quits and restarts Atlas and also defines certain global parameters for Atlas execution

Note: This command is executed by DeckBuild. This statement is documented in the DeckBuild User's Manual.

Example starting a given Atlas Version

To start a given version of Atlas the syntax is set by the `simflags` argument. To start version 5.12.0.R, type:

```
go atlas simflags="-V 5.12.0.R"
```

Parallel Atlas Example

To define the number of processors to be used in parallel Atlas, use the `P` flag. For example, to start parallel Atlas use four processors. For example:

```
go atlas simflags="-V 5.12.0.R -P 4"
```

22.22 IMPACT

IMPACT specifies and set parameters for impact ionization models.

Syntax

```
IMPACT <model>
```

Parameter	Type	Default	Units
A.NT	Real	0.588	
A.PT	Real	0.588	
AE0001	Real	1.76×10^{8}	cm^{-1}
AE1120	Real	3.30×10^{7}	cm^{-1}
AH0001	Real	3.41×10^{8}	cm^{-1}
AH1120	Real	2.50×10^{7}	cm^{-1}
AN1	Real	7.03×10^{5}	cm^{-1}
AN2	Real	7.03×10^{5}	cm^{-1}
AN0.VALD	Real	4.3383	
AN1.VALD	Real	-2.42×10^{-12}	
AN2.VALD	Real	4.1233	
ANGLE	Real	0.0	Degrees
ANISO	Logical	False	
ANOKUTO	Real	0.426	V
ANLACKNER	Real	1.316×10^{6}	cm
AP0.VALD	Real	2.376	
AP1	Real	6.71×10^{5}	cm^{-1}
AP2	Real	1.682×10^{6}	cm^{-1}
AP1.VALD	Real	0.01033	
AP2.VALD	Real	1.0	
APOKUTO	Real	0.243	V
APLACKNER	Real	1.818×10^{6}	cm
B.NT	Real	0.248	
B.PT	Real	0.248	

Parameter	Type	Default	Units
BE0001	Real	2.10×10^{7}	V/cm
BE1120	Real	1.70×10^{7}	V/cm
BETAN	Real	1.0	
BETAP	Real	1.0	
BH0001	Real	2.96×10^{7}	V/cm
BH1120	Real	1.690×10^{7}	V/cm
BN1	Real	1.231×10^{6}	V/cm
BN2	Real	1.231×10^{6}	V/cm
BN1.VALD	Real	0.0	
BN0.VALD	Real	0.235	
BNLACKNER	Real	1.474×10^{6}	V/cm
BNOKUTO	Real	4.81×10^{5}	V/cm
BP1	Real	1.231×10^{6}	V/cm
BP2	Real	2.036×10^{6}	V/cm
BP0.VALD	Real	0.17714	
BP1.VALD	Real	-0.002178	
BPOKUTO	Real	6.53×10^{5}	V/cm
BPLACKNER	Real	2.036×10^{6}	V/cm
C0	Real	2.5×10^{-10}	
CHIA	Real	3.0×10^{5}	
CHIB	Real	5.0×10^{4}	
CHI.HOLES	Real	4.6×10^{4}	
CN2	Real	0.0	$cm^{-1}K^{-1}$
CN0.VALD	Real	1.6831×10^{4}	
CN1.VALD	Real	4.3796	
CN2.VALD	Real	1.0	
CN3.VALD	Real	0.13005	

Parameter	Type	Default	Units
CNOKUTO	Real	3.05×10^{-4}	K
CP2	Real	0.0	$cm^{-1}K^{-1}$
CP0.VALD	Real	0.0	
CP1.VALD	Real	0.00947	
CP2.VALD	Real	2.4929	
CP3.VALD	Real	0.0	
CPOKUTO	Real	5.35×10^{-4}	K
CROWELL	Logical	False	
CSUB.N	Real	2.0×10^{14}	
CSUB.P	Real	4.0×10^{12}	
DN0.VALD	Real	1.233735×10^{6}	
DN1.VALD	Real	1.2039×10^{3}	
DN2.VALD	Real	0.56703	
DNOKUTO	Real	6.86×10^{-4}	K
DP0.VALD	Real	1.4043×10^{6}	
DP1.VALD	Real	2.9744×10^{3}	
DP2.VALD	Real	1.4829	
DPOKUTO	Real	5.67×10^{-4}	K
DEVICE	Character		
DIRECTED	Logical	False	
E.DIR	Logical	True	
E.SIDE	Logical	False	
E.VECTOR	Logical	False	
ECN.II	Real	1.231×10^{6}	
ECP.II	Real	2.036×10^{6}	
EGRAN	Real	4.0×10^{5}	V/cm
ENERGY.STEP	Real	0.0025	

Parameter	Type	Default	Units
ETH.N	Real	1.8	eV
ETH.P	Real	3.5	eV
EXN.II	Real	1.0	
EXP.II	Real	1.0	
F.EDIIN	Character		
F.EDIIP	Character		
GRADQFL	Logical	False	
GRANT	Logical	False	
IINOFF	Character		
IIPOFF	Character		
JNX.MIN	Real	0.0	A/cm^2
JNY.MIN	Real	0.0	A/cm^2
JNZ.MIN	Real	0.0	A/cm^2
JPX.MIN	Real	0.0	A/cm^2
JPY.MIN	Real	0.0	A/cm^2
JPZ.MIN	Real	0.0	A/cm^2
ICRIT	Real	4.0×10^{-3}	A/cm^2
INFINITY	Real	0.001	
LACKNER	Logical	False	
LAMBDAE	Real	6.2×10^{-7}	cm
LAMBDAH	Real	3.8×10^{-7}	cm
LAN300	Real	6.2×10^{-7}	cm
LAP300	Real	3.8×10^{-7}	cm
LENGTH.REL	Logical	False	
LREL.EL	Real	3.35×10^{-2}	µm
LREL.HO	Real	2.68×10^{-2}	µm
M.ANT	Real	1.0	
M.APT	Real	1.0	

Parameter	Type	Default	Units
M.BNT	Real	1.0	
M.BPT	Real	1.0	
MATERIAL	Character		
NAME	Character		
NEW	Logical	False	
NISEDELTA	Real	1.5	
N.CONCANNON	Logical	False	
N.CHEN	Logical	False	
N.ION.1	Real	0.0	$cm^{-1}K^{-1}$
N.ION.2	Real	0.0	$cm^{-1}K^{-2}$
N.IONIZA	Real	7.03×10^{5}	cm^{-1}
N.LACKHW	Real	0.063	eV
N.MCCLINTOCK	Logical	False	
N.OGUZMAN	Logical	False	
N.TUT	Logical	False	
N.UPSILON	Logical	True/False (in silicon)/(otherwise)	
N.VANHW	Real	0.063	eV
NDELOKUTO	Real	2.0	
NGAMOKUTO	Real	1.0	
NISELAMBDA	Real	1.0	
OKUTO	Logical	False	
OLD	Logical	False	
OP.PH.EN	Real	0.063	eV
OPPHE	Real	0.063	eV
P.CHEN	Logical	False	
P.CONCANNON	Logical	False	
P.ION.1	Real	0.0	$cm^{-1}K^{-1}$
P.ION.2	Real	0.0	$cm^{-1}K^{-2}$

Parameter	Type	Default	Units
P.IONIZA	Real	1.682×10^{6}	cm^{-1}
P.LACKHW	Real	0.063	eV
P.MCCLINTOCK	Logical	False	
P.OGUZMAN	Logical	False	
P.TUT	Logical	False	
P.UPSILON	Logical	True/False (in silicon)/(otherwise)	
P.VANHW	Real	0.063	eV
PDELOKUTO	Real	2.0	
PGAMOKUTO	Real	1.0	
PISEDELTA	Real	1.5	
PISELAMBDA	Real	1.0	
REGION	Integer		
SIC4H0001	Logical	False	
SIC4H1120	Logical	False	
SELB	Logical	False	
STRUCTURE	Character		
TAUSN	Real	0.4×10^{-12}	s
TAUSP	Real	0.4×10^{-12}	s
TBL.EXTRAP	Logical	True	
TOYABE	Logical	False	
VAL.AN0	Real	4.3383	
VAL.AN1	Real	-2.42×10^{-12}	
VAL.AN2	Real	4.1233	
VAL.AP0	Real	2.376	
VAL.AP1	Real	0.01033	
VAL.AP2	Real	1.0	
VAL.BN0	Real	0.235	
VAL.BN1	Real	0.0	

Parameter	Type	Default	Units
VAL.CN0	Real	1.6831×10^{4}	
VAL.CN1	Real	4.3796	
VAL.CN2	Real	1.0	
VAL.CN3	Real	0.13005	
VAL.DN0	Real	1.233735×10^{6}	
VAL.DN1	Real	1.2039×10^{3}	
VAL.DN2	Real	0.56703	
VAL.BP0	Real	0.17714	
VAL.BP1	Real	-0.002178	
VAL.CP0	Real	0.0	
VAL.CP1	Real	0.00947	
VAL.CP2	Real	2.4924	
VAL.CP3	Real	0.0	
VAL.DP0	Real	1.4043×10^{6}	
VAL.DP1	Real	2.9744×10^{3}	
VAL.DP2	Real	1.4829	
VALDINOCI	Logical	False	
VANOVERS	Logical	False	
ZAPPA	Logical	False	
ZAP.AN	Real	1.9	eV
ZAP.BN	Real	41.7	angstroms
ZAP.CN	Real	46.0	meV
ZAP.DN	Real	46.0	meV
ZAP.EN	Real	46.0	meV
ZAP.AP	Real	1.4	eV
ZAP.BP	Real	41.3	angstroms
ZAP.CP	Real	36.0	meV
ZAP.DP	Real	36.0	meV
ZAP.EP	Real	36.0	meV

Description

The impact ionization model for continuity equations allows the accurate prediction of avalanche breakdown for many devices. Since impact ionization is a two-carrier process, the following statement must be specified after setting impact ionization models.

```
METHOD CARRIERS=2
```

Model Selection Flags

ANISO	Specifies the 4hSiC Model Equations 5-132 through 6-142.
CROWELL	Specifies the Crowell and Sze formulae [68].
GRADQFL	Enables Gradient of Quasi-Fermi levels as drivers.
GRANT	Selects Grant's impact ionization model [106] (see Equations 3-412 through 3-448).
LACKNER	Enables the Lackner model [169].
SELB	Selects the impact ionization model described by Selberherr [286].
OKUTO	Enables the Okuto-Crowell model [226].
N.CHEN **P.CHEN**	Enable the tabular ionization rates [52].
N.CONCANNON **P.CONCANNON**	Set the Concannon substrate current model.
N.MCCLINTOCK **P.MCCLINTOCK**	Enables the tabular ionization rates [205].
N.OGUZMAN **N.OGUZMAN**	Enables the tabular ionization rates [228].
N.TUT **P.TUT**	Enables the tabular ionization rates [315].
N.UPSILON	Enables non-linear dependence of effective field on electron temperature if you use the `-ISE` flag.
P.UPSILON	Enables non-linear dependence of effective field on hole temperature if you use the `-ISE` flag.
SIC4H0001	Specifies that the y axis concides with the 0001 crystal orientation for the anisotropic impact ionization model.
SIC4H0001	Specifies that the y axis concides with the 1120 crystal orientation for the anisotropic impact ionization model.
TOYABE	This is a synonym for **SELB**.

VALDINOCI	Enables the Valdinoci impact ionization model. See Equations 3-399, 3-432, and 3-440. The alias for this parmeter is II.VALDI.
VANOVERS	Enables the VanOverstraeten de Man model [325].
ZAPPA	Enables Zappa's model for ionization rates in InP.

Note: If no model selection flag is set, the model parameters from Grant [106] are used. See "Grant's Impact Ionization Model" on page 245.

Okuto-Crowell Model Parameters

ANOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
APOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.
BNOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
BPOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.
CNOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
CPOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.
DNOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
DPOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.
NGAMOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
PGAMOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.
NDELOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-460, Table 3-101.
PDELOKUTO	See Section 3.6.4 "Impact Ionization Models", Equation 3-461, Table 3-101.

van Overstraeten de Man Model Parameters

N.VANHW	See Section 3.6.4 "Impact Ionization Models", Equation 3-430, Table 3-97.
P.VANHW	See Section 3.6.4 "Impact Ionization Models", Equation 3-431, Table 3-97.

Note: You must also use the Selberherr model parameters with this model.

Lackner Model Parameters

ANLACKNER	See Section 3.6.4 "Impact Ionization Models", Equation 3-463, Table 3-102.
APLACKNER	See Section 3.6.4 "Impact Ionization Models", Equation 3-464, Table 3-102.
BNLACKNER	See Section 3.6.4 "Impact Ionization Models", Equation 3-463, Table 3-102.

ANLACKNER	See Section 3.6.4 “Impact Ionization Models”, Equation 3-463, Table 3-102.
BPLACKNER	See Section 3.6.4 “Impact Ionization Models”, Equation 3-464, Table 3-102.
N.LACKHW	See Section 3.6.4 “Impact Ionization Models”, Equation 3-465, Table 3-102.
P.LACKHW	See Section 3.6.4 “Impact Ionization Models”, Equation 3-465, Table 3-102.

Non-linear Hydrodynamic Model

To enable this model, use the command line flag `-ISE`.

N.UPSILON	See Section 3.6.4 “Impact Ionization Models”, Equation 3-478, Table 3-106.
P.UPSILON	See Section 3.6.4 “Impact Ionization Models”, Equation 3-479, Table 3-106.
NISELAMBDA	See Section 3.6.4 “Impact Ionization Models”, Equation 3-478, Table 3-106.
PISELAMBDA	See Section 3.6.4 “Impact Ionization Models”, Equation 3-479, Table 3-106.
NISEDELTA	See Section 3.6.4 “Impact Ionization Models”, Equation 3-478, Table 3-106.
PISEDELTA	See Section 3.6.4 “Impact Ionization Models”, Equation 3-479, Table 3-106.

Electric Field Model Selection Flags

The default electric field to use for ionization rate calculations is calculated as the modulus of the electric field over a triangle. In Atlas 3D, this is combined with the z-component of the field at each corner of the prism to give an overall field modulus at each node. This corresponds to Equation 3-412 for a triangle.

ANGLE	Specifies the direction along which field magnitude will be calculated for use in calculating ionization rate when you specify `DIRECTED`.
DIRECTED	Specifies that the field for ionization rate calculations will be calculated along a user-specified direction. That direction is specified by the `ANGLE` parameter.
E.DIR	Specifies that the impact ionization rate will be calculated as a function of the electric field in the direction of the current (see Equation 3-414). Its alias is **NEW**.
E.SIDE	Specifies that the impact ionization rate will be calculated as a function of the electric field along the side of the triangle (see Equation 3-413). This model is only supported in Atlas2D. Its alias is **OLD**.
E.VECTOR	Specifies that the vector electric field will be used in the calculated of the impact ionization rate (see Equation 3-412).
OLD	This is the model that corresponds to Equation 3-413 and is only supported in Atlas 2D.
NEW	This is the model that corresponds to Equation 3-414, where the component of the field in the direction of the current is used in the ionisation rate calculation. It is implemented in both Atlas 2D and Atlas 3D. If the dot product of E and J is negative, then the field component is taken as 0. Consequently, impact ionisation may only occur when a current is dominated by the drift term.

Model Localization Parameters

DEVICE	Specifies the device in MixedMode simulation to which the statement should apply. The synonym for this parameter is **STRUCTURE**.
MATERIAL	Specifies what material from Table B-1 the statement should apply. If a material is specified then all regions defined as being composed of that material will be affected.
NAME	Specifies what region that the **IMPACT** statement should apply. Note that the name must match the name specified in the NAME parameter of the **REGION** statement or the region number.
REGION	Specifies that index of the region to which the impact parameters apply.
STRUCTURE	This is a synonym for **DEVICE**.

Valdinoci Model Parameters

AN0.VALD	This is an alias for **VAL.AN0**.
AN1.VALD	This is an alias for **VAL.AN1**.
AN2.VALD	This is an alias for **VAL.AN2**.
AP0.VALD	This is an alias for **VAL.AP0**.
AP1.VALD	This is an alias for **VAL.AP1**.
AP2.VALD	This is an alias for **VAL.AP2**.
BN0.VALD	This is an alias for **VAL.BN0**.
BN1.VALD	This is an alias for **VAL.BN1**.
BP0.VALD	This is an alias for **VAL.BP0**.
BP1.VALD	This is an alias for **VAL.BP1**.
CN0.VALD	This is an alias for **VAL.CN0**.
CN1.VALD	This is an alias for **VAL.CN1**.
CN2.VALD	This is an alias for **VAL.CN2**.
CN3.VALD	This is an alias for **VAL.CN3**.
CP0.VALD	This is an alias for **VAL.CP0**.
CP1.VALD	This is an alias for **VAL.CP1**.
CP2.VALD	This is an alias for **VAL.CP2**.
CP3.VALD	This is an alias for **VAL.CP3**.
DN0.VALD	This is an alias for **VAL.DN0**.
DN1.VALD	This is an alias for **VAL.DN1**.

DN2.VALD	This is an alias for **VAL.DN2**.
DP0.VALD	This is an alias for **VAL.DP0**.
DP1.VALD	This is an alias for **VAL.DP1**.
DP2.VALD	This is an alias for **VAL.DP2**.
VAL.AN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN0.VALD**.
VAL.AN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN1.VALD**.
VAL.AN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN2.VALD**.
VAL.BN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-434. The alias for this parameter is **BN0.VALD**.
VAL.BN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-434. The alias for this parameter is **BN1.VALD**.
VAL.CN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN0.VALD**.
VAL.CN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN1.VALD**.
VAL.CN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN2.VALD**.
VAL.CN3	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN3.VALD**.
VAL.DN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN0.VALD**.
VAL.DN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN1.VALD**.
VAL.DN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN2.VALD**.
VAL.AP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437.
VAL.AP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437. The alias for this parameter is **AP1.VALD**.
VAL.AP2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437. The alias for this parameter is **AP2.VALD**.
VAL.BP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-438. The alias for this parameter is **BP0.VALD**.

VAL.BP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-438. The alias for this parameter is **BP1.VALD**.
VAL.CP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP0.VALD**.
VAL.CP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP1.VALD**.
VAL.CP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP2.VALD**.
VAL.CP3	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP3.VALD**.
VAL.DP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP0.VALD**.
VAL.DP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP1.VALD**.
VAL.DP2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP2.VALD**.

Zappa Model Parameters

ZAP.AN	Specifies the value of a temperature dependent ionization parameter in Equation 3-441.
ZAP.BN	Specifies the value of a temperature dependent ionization parameter in Equation 3-442.
ZAP.CN	Specifies the value of a temperature dependent ionization parameter in Equation 3-442.
ZAP.DN	Specifies the value of a temperature dependent ionization parameter in Equation 3-443.
ZAP.EN	Specifies the value of a temperature dependent ionization parameter in Equation 3-443.
ZAP.AP	Specifies the value of a temperature dependent ionization parameter in Equation 3-444.
ZAP.BP	Specifies the value of a temperature dependent ionization parameter in Equation 3-445.
ZAP.CP	Specifies the value of a temperature dependent ionization parameter in Equation 3-445.
ZAP.DP	Specifies the value of a temperature dependent ionization parameter in Equation 3-446.
ZAP.EP	Specifies the value of a temperature dependent ionization parameter in Equation 3-446.

Crowell Model Parameters

LAMBDAE	Specifies the mean free path for electrons. The alias for this parameter is **LAN300**.
LAMBDAH	Specifies the mean free path for holes. The alias for this parameter is **LAP300**.
LAN300	This is an alias for **LAMBDAE**.
LAP300	This is an alias for **LAMBDAH**.
OP.PH.EN	This is an alias for **OPPHE**.
OPPHE	Specifies the optical phonon energy. The alias for this parameter is **OP.PH.EN**.

Selberrherr Model Parameters

AN1, **AN2**, **BN1**, **BN2**, **EGRAN**	Specify the basic set of parameters for Selberherr's impact ionization model. Index 1 (**AP1**, **AP1**, AN1, and BN1) corresponds to field values greater than EGRAN, and index 2 (**AP2**, **BP2**, AN2, and BN2) corresponds to field values less than EGRAN. The aliases for these parameters are **N.IONIZA**, **P.IONIZA**, **ECN.II**, and **ECP.II**.
BETAN	This is for electrons and **BETAP** is for holes correspond to coefficients for the power of ECRIT/E. The aliases for these parameters are **EXN.II** and **EXP.II**.
EXN.II	This is an alias for **BETAN**.
EXP.II	This is an alias for **BETAP**.

Temperature Dependence Parameters

A.NT	Specifies the value of the temperature-dependent parameter in Equation 3-418.
A.PT	Specifies the value of the temperature-dependent parameter in Equation 3-419.
B.NT	Specifies the value of the temperature-dependent parameter in Equation 3-420.
B.PT	Specifies the value of the temperature-dependent parameter in Equation 3-421.
M.ANT	Specifies the value of the temperature-dependent parameter in Equation 3-418.
M.APT	Specifies the value of the temperature-dependent parameter in Equation 3-419.
M.BNT	Specifies the value of the temperature-dependent parameter in Equation 3-420.
M.BPT	Specifies the value of the temperature-dependent parameter in Equation 3-421.
CN2, **CP2**, **DN2**, and **DP2**	These are specifiable coefficients in the temperature dependent models described in Equations 3-399 and 3-420. The aliases for these parameters are **N.ION.1**, **P.ION.1**, **N.ION.2**, and **P.ION.2**.
N.ION.1, **P.ION.1**, **N.ION.2**, and **P.ION.2**	These are the aliases for **CN2**, **CP2**, DN2, and DP2.

Parameters for use with Energy Balance

F.EDIIN	Specifies the name of the file containing a C-Interpreter function describing the values of the parameters in Equation 3-416 as a function of electron temperature.
F.EDIIP	Specifies the name of the file containing a C-Interpreter function describing the values of the parameters in Equation 3-417 as a function of hole temperature.
LENGTH.REL	Specifies the use of energy relaxation length for the impact ionization model with the energy balance model. If LENGTH.REL is specified, TAUSN and TAUSP cannot be specified and have any affect.
LREL.EL	Specifies an energy relaxation length for electrons if LENGTH.REL is specified.
LREL.HO	Specifies an energy relaxation length for holes if LENGTH.REL is specified.
TAUSN	Specifies the relaxation time for electrons in the temperature dependent impact model.
TAUSP	Specifies the relaxation time for holes in the temperature dependent impact model.

Note: When energy balance simulations are run, the Toyabe impact ionization model is used. This model is used regardless of the SELB or CROWELL settings. See "Toyabe Impact Ionization Model" on page 249 for more information about this model.

Concannon Model Parameters

CSUB.N	This is an empirical tuning factor used in Concannon's Substrate Current Model (see Equation 3-472) for electrons.
CSUB.P	This is an empirical tuning factor used in Concannon's Substrate Current Model (see Equation 3-473) for holes.
ETH.N	Specifies the ionization threshold energy for electrons used in Concannon's Substrate Current Model (see Equation 3-472).
ETH.P	Specifies the ionization threshold energy for holes used in Concannon's Substrate Current Model (see Equation 3-473).
C0	Specifies the electron distribution weight factor used in Concannon's Substrate Current Model (see Equation 3-476).
CHIA	Specifies the electron distribution function constant used in Concannon's Substrate Current Model (see Equation 3-476).
CHIB	Specifies the electron distribution function constant used in Concannon's Substrate Current Model (see Equation 3-476).

CHI.HOLES	Specifies the hole distribution function constant used in Concannon's Substrate Current Model (see Equation 3-477).
ENERGY.STEP	Specifies the energy step for numeric integration used in Concannon's Substrate Current Model
INFINITY	Specifies the limit for the highest energy in numeric integration used in Concannon's Substrate Current Model.

Anisotropic Impact Ionization Parameters

AE0001	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
AE1120	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
AH0001	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
AH1120	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
BE0001	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
BE1120	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
BH0001	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.
BH1120	Anisotropic impact ionization parameter for Equations 5-132 through 6-142.

Wide Bandgap Semiconductor Parameters

JNX.MIN	Specifies minimum electron current in x-direction to be used in calculating impact generation rate for `E.SIDE` model.
JNY.MIN	Specifies minimum electron current in y-direction to be used in calculating impact generation rate for `E.SIDE` model.
JNZ.MIN	Specifies minimum electron current in z-direction to be used in calculating impact generation rate for `E.SIDE` model.
JPX.MIN	Specifies minimum hole current in x-direction to be used in calculating impact generation rate for `E.SIDE` model.
JPY.MIN	Specifies minimum hole current in y-direction to be used in calculating impact generation rate for `E.SIDE` model.
JPZ.MIN	Specifies minimum hole current in z-direction to be used in calculating impact generation rate for `E.SIDE` model.

Tabular Selberherr Parameters

IINOFF	Specifies the name of a file containing an ASCII table of the electron ionization rates as a funciton of electric field.
IIPOFF	Specifies the name of a file containing an ASCII table of the hole ionization rates as a function of electric field.
TBL.EXTRAP	Specifies whether extrapolation is applied to tabular ionization rates. By default, this is `true`.

Selberrherr Model Example

This example shows an **IMPACT** statement which specifies all parameters used by the model selected by the SELB parameter. In this case, only parameters for holes are field dependent. The AP1 and BP1 parameters correspond to parameters at field values more than EGRAN. The AP2 and BP2 parameters correspond to field values less than EGRAN. Coefficients for electrons should be repeated.

```
IMPACT SELB AN1=7.03E5 AN2=7.03E5 BN1=1.231E6 \
       BN2=1.231E6 AP1=6.71E5 AP2=1.58E6 BP1=1.693E6 \
       BP2=2.036E6 BETAN=1 BETAP=1 EGRAN=4.0E5
```

22.23 INTDEFECTS

INTDEFECTS activates the band gap interface defect model and sets the parameter values. This model can be used when thin-film transistor simulations are performed using the TFT product.

Syntax

```
INTDEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
AMPHOTERIC	Logical	False	
CONTINUOUS	Logical	True	
DEVICE	Character		
DFILE	Characater		
EGA	Real	0.4	eV
EGD	Real	0.4	eV
EP.AMPHOTERIC	Real	1.27	eV
EU.AMPHOTERIC	Real	0.2	eV
EV0.AMPHOTERIC	Real	0.056	eV
F.INTDEFECTS	Character		
F.TFTACC	Character		
F.TFTDON	Character		
FILE.AMPHOTERIC	Character		
FILEX.AMPHOTERIC	Character		
FILEY.AMPHOTERIC	Character		
FILEZ.AMPHOTERIC	Character		
HCONC.AMPHOTERIC	Real	5×10^{10}	cm^{-2}
ID	Real		
INTMATERIAL	Character		
INTNAME	Character		
INTNUMBER	Character		
INTREGION	Character		
MODIFY	Logical	False	

Parameter	Type	Default	Units
NGA	Real	5.0×10^{10}	cm^{-2}/eV
NGD	Real	1.5×10^{11}	cm^{-2}/eV
NSISI.AMPHOTERIC	Real	2×10^{12}	cm^{-2}
NTA	Real	1.12×10^{14}	cm^{-2}/eV
NTD	Real	4.0×10^{13}	cm^{-2}/eV
NUM.AMPHOTERIC	Real	20	
NUMBER	Real	All	
NUMA	Real	12	
NUMD	Real	12	
NV0.AMPHOTERIC	Real	NV300	cm^{-2}
R1MATERIAL	Character		
R1NAME	Character		
R1NUMBER	Integer		
R1REGION	Character		
R2MATERIAL	Character		
R2NAME	Character		
R2NUMBER	Integer		
R2REGION	Character		
S.I	Logical	True	
S.M	Logical	False	
S.S	Logical	False	
S.X	Logical	False	
SIGGAE	Real	1.0×10^{-16}	cm^2
SIGGAH	Real	1.0×10^{-14}	cm^2
SIGGDE	Real	1.0×10^{-14}	cm^2
SIGGDH	Real	1.0×10^{-16}	cm^2
SIGMA.AMPHOTERIC	Real	0.19	eV

Parameter	Type	Default	Units
SIGN0.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGNP.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGP0.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGPN.AMPHOTERIC	Real	1.0×10^{-16}	cm^2
SIGTAE	Real	1.0×10^{-16}	cm^2
SIGTAH	Real	1.0×10^{-14}	cm^2
SIGTDE	Real	1.0×10^{-14}	cm^2
SIGTDH	Real	1.0×10^{-16}	cm^2
STRUCTURE	Character		
TFILE	Character		
T0.AMPHOTERIC	Real	300	K
WGA	Real	0.1	eV
WGD	Real	0.1	eV
WTA	Real	0.025	eV
WTD	Real	0.05	ev
X.MIN	Real	left of structure	μm
X.MAX	Real	right of structure	μm
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MIN	Real	back of structure	μm
Z.MAX	Real	front of structure	μm

Description

The **INTDEFECTS** statement is used to describe the density of defect states in the band gap at semiconductor interfaces. You can specify up to four distributions, two for donor-like states

and two for acceptor-like states. Each type of state may contain one exponential (tail) distribution and one Gaussian distribution.

AFILE	Specifies the file name where the acceptor state density distribution, as a function of energy, will be stored. You can examine this file by using TonyPlot.
AMPHOTERIC	Specifies the amphoteric defect model will be used.
CONTINUOUS	Specifies that the continuous defect integral model will be used.
DEVICE	Specifies which device the statement applies in mixed mode simulation. The synonym for this parameter is **STRUCTURE**.
DFILE	Specifies the file name where the donor state density distribution, as a function of energy, will be stored. You can examine this file by using TonyPlot.
EGA	Specifies the energy that corresponds to the Gaussian distribution peak for acceptor-like states. This energy is measured from the conduction band edge.
EGD	Specifies the energy that corresponds to the Gaussian distribution peak for donor-like states. This energy is measured from the valence band edge.
EP.AMPHOTERIC	Specifies the most probable potential defect energy.
EU.AMPHOTERIC	Specifies the defect electron correlation energy.
EV0.AMPHOTERIC	Specifies the characteristic energy.
F.INTDEFECTS	Specifies the name of a file containing a C-Interpreter function, describing the **INTDEFECTS** statement parameters as a function of position.
F.TFTACC	Specifies the name of a file containing a C-Interpreter function, describing the distribution of acceptor state densities as a function of energy.
F.TFTDON	Specifies the name of a file containing a C-Interpreter function, describing the distribution of donor state densities as a function of energy.
FILE.AMPHOTERIC	Specifies the file name where the dangling bond density of states distribution, as a function of energy, will be stored. If you specify FILEX.AMPHOTERIC, FILEY.AMPHOTERIC, and FILEZ.AMPHOTERIC, then the file will be saved at the coordinate closest to the one specified by FILEX.AMPHOTERIC, FILEY.AMPHOTERIC, and FILEZ.AMPHOTERIC.
FILEX.AMPHOTERIC	Specifies the X coordinate used in FILE.AMPHOTERIC.
FILEY.AMPHOTERIC	Specifies the Y coordinate used in FILE.AMPHOTERIC.
FILEZ.AMPHOTERIC	Specifies the Z coordinate used in FILE.AMPHOTERIC.
HCONC.AMPHOTERIC	Specifies the density of hydrogen.
ID	Specifies a reference number for use with the CIgetNodalVoid C-Interpreter callback function.
INTMATERIAL	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, INTMATERIAL="oxide/silicon".

INTNAME	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, INTNAME="channel/substrate".
INTNUMBER	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, INTNUMBER="2/3".
INTREGION	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, INTREGION="channel/substrate".
MODIFY	Specifies that a previously defined INTDEFECTS statement will be modified. MODIFY must be specified along with the ID parameter and only applies to CONTINUOUS interface defects. The previously defined INTDEFECTS statement with the same ID number will be modified with the new values. Note: NUMA and NUMD cannot be modified.
NGA	Specifies the total density of acceptor-like states in a Gaussian distribution.
NGD	Specifies the total density of donor-like states in a Gaussian distribution.
NSISI.AMPHOTERIC	Specifies the density of Si-Si bonds.
NTA	Specifies the density of acceptor-like states in the tail distribution at the conduction band edge.
NTD	Specifies the density of donor-like states in the tail distribution at the valence band edge.
NUM.AMPHOTERIC	Specifies the number of energy levels used in the DOS integration for amphoteric defects.
NV0.AMPHOTERIC	Specifies the density of states of the valance-band tail exponential region extrapolated to the valence-band edge. If this parameter is not specified, then the valence band density of states (NV300 on the **MATERIAL** statement) will be used instead.
NUMBER or **REGION**	Specifies the region index to which the **DEFECTS** statement applies.
NUMA	Specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.
NUMD	Specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.
R1MATERIAL **R2MATERIAL**	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, R1MATERIAL=oxide R2MATERIAL=silicon.

R1NAME **R2NAME**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, `R1NAME=channel R2NAME=substrate`.
R1NUMBER **R2NUMBER**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, `R1NUMBER=2 R2NUMBER=3`.
R1REGION **R2REGION**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, `R1REGION=channel R2REGION=substrate`.
S.I	Specifies that the **INTDEFECTS** statement should apply to semiconductor-insulator interfaces.
S.M	Specifies that the **INTDEFECTS** statement should apply to semiconductor-metal interfaces.
S.S	Specifies that the **INTDEFECTS** statement should apply to semiconductor-semiconductor interfaces.
S.X	Specifies that the **INTDEFECTS** statement should apply to all semiconductor-insulator interfaces, including those to the outside domain.
SIGGAE	Specifies the capture cross-section for electrons in a Gaussian distribution of acceptor-like states.
SIGGAH	Specifies the capture cross-section for holes in a Gaussian distribution of acceptor-like states.
SIGGDE	Specifies the capture cross-section for electrons in a Gaussian distribution of donor-like states.
SIGGDH	Specifies the capture cross-section for holes in a Gaussian distribution of donor-like states.
SIGMA.AMPHOTERIC	Specifies the defect pool width.
SIGN0.AMPHOTERIC	Specifies the electron capture cross-section for neutral defects.
SIGNP.AMPHOTERIC	Specifies the electron capture cross-section for positive defects.
SIGP0.AMPHOTERIC	Specifies the hole capture cross-section for neutral defects.
SIGPN.AMPHOTERIC	Specifies the hole capture cross-section for negative defects.
SIGTAE	Specifies the capture cross-section for electrons in a tail distribution of acceptor-like states.
SIGTAH	Specifies the capture cross-section for holes in a tail distribution of acceptor-like states.
SIGTDE	Specifies the capture cross-section for electrons in a tail distribution of donor-like states.

SIGTDH	Specifies the capture cross-section for holes in a tail distribution of donor-like states.
STRUCTURE	This is a synonym for **DEVICE**.
TFILE	Specifies the file name where the acceptor and donor state density distributions, as a function of energy referenced from E_v, will be stored. You can examine this file by using TonyPlot.
T0.AMPHOTERIC	Specifies the freeze-in temperature.
WGA	Specifies the characteristic decay energy for a Gaussian distribution of acceptor-like states.
WGD	Specifies the characteristic decay energy for a Gaussian distribution of donor-like states.
WTA	Specifies the characteristic decay energy for the tail distribution of acceptor-like states.
WTD	Specifies the characteristic decay energy for the tail distribution of donor-like states.
X.MIN	Specifies the left boundary of a box, where an interface must exist, where defects are to be applied.
X.MAX	Specifies the right boundary of a box, where an interface must exist, where defects are to be applied.
Y.MIN	Specifies the top boundary of a box, where an interface must exist, where defects are to be applied.
Y.MAX	Specifies the bottom boundary of a box, where an interface must exist, where defects are to be applied.
Z.MIN	Specifies the back boundary of a box, where an interface must exist, where defects are to be applied.
Z.MAX	Specifies the front boundary of a box, where an interface must exist, where defects are to be applied.

22.24 INTERFACE

INTERFACE specifies interface parameters at semiconductor/insulator boundaries. All parameters apply only at the boundary nodes except where stated.

Syntax

```
INTERFACE [<params>]
```

Parameter	Type	Default	Units
A.MAX	Real	360.0	Degrees
A.MIN	Real	0.0	Degrees
ABSORPTION	Real	0.0	
ADF.TABLE	Character		
AR.ABSORB	Real	0.0	cm^{-1}
AR.MATERIAL	Character		
AR.INDEX	Real	1	
AR.THICK	Real	0	μm
BESONOS	Logical	False	
CHARGE	Real	0.0	cm^{-2}
COATING	Integer	1	
CONSTANT	Logical	False	
CR	Real	1.0	
CT	Real	1.0	
DANIELSSON	Logical	False	
DEVICE	Character		
DD.TUNNEL	Real	0.1	
DIFFUSIVE	Logical	False	
DISPERSION	Real	0.0	
DNLS.ACC	Logical	False	
DNLS.DON	Logical	False	
DNLS.CN	Real	0.0	cm^3s^{-1}
DNLS.CP	Real	0.0	cm^3s^{-1}
DNLS.NT	Real	0.0	cm^{-3}
DNLS.ET	Real	0.0	eV

Parameter	Type	Default	Units
DY.TUNNEL	Real	0.1	
DYNASONOS	Logical	True	
ELLIPSE	Logical	False	
F.QF	Character		
GAUSS	Logical	False	
I.I	Logical	False	
ID	Real		
IITHERM	Logical	False	
INT.RESIST	Real	0.0	Ωcm^2
INTMATERIAL	Character		
INTNAME	Character		
INTNUMBER	Character		
INTREGION	Character		
LAM1	Real	0.0	μm
LAM2	Real	0.0	μm
LAMBERT	Logical	False	
LAYER	Integer	1	
LORENTZ	Logical	False	
MATERIAL	Character		
MODIFY	Logical	False	
N.I	Logical	False	
NEUTRALIZE	Logical	False	
NLAM	Integer	0	
NOM.ANGLE	Real	0.0	Degrees
NOS.ANGLE	Real	0.0	Degrees
NOM.DIST	Real	1.0	Microns
NOS.DIST	Real	1.0	Microns
NR	Real	2.0	
NT	Real	3.0	

Parameter	Type	Default	Units
OPTICAL	Logical	False	
OUTADF	Character		
OUT.HR	Character		
OUT.HT	Character		
QF	Real	0.0	cm^{-2}
Q.THERMIONIC	Real	1.0	
P1.X	Real		μm
P2.X	Real		μm
P1.Y	Real		μm
P2.Y	Real		μm
PELTIER	Real	0.0	W/cm^{-2}
PGG.NSIGMA	Real	5.12	
R.MAX	Real	Device radius	μm
R.MIN	Real	0.0	μm
R1MATERIAL	Character		
R1NAME	Character		
R1NUMBER	Integer		
R1REGION	Character		
R2MATERIAL	Character		
R2NAME	Character		
R2NUMBER	Integer		
R2REGION	Character		
REFLECT	Real		
REGION	Integer		
S.C	Logical	False	
S.I	Logical	True	
S.M	Logical	False	
S.N	Real	0.0	cm/s
S.P	Real	0.0	cm/s

Parameter	Type	Default	Units
S.S	Logical	False	
S.X	Logical	False	
SAVE.ELEC	Logical	True	
SAVE.HOLE	Logical	True	
SAVE.PJUMP	Character		
SCATTERED	Integer	11	
SIGMA	Real	20.0	nm
SITHERM	Logical	False	
SEMIMINOR	Real	0.31	
SPECULAR	Real	0.0	
SPJ.DE	Real	100	V/cm
STRUCTURE	Character		
S.WELL.N	Real	0.0	cm/s
S.WELL.P	Real	0.0	cm/s
TCR.TABLE	Character		
TCT.TABLE	Character		
THERMIONIC	Logical	False	
TRIANGLE	Logical	False	
TUNNEL	Logical	False	
X.MAX	Real	right hand side of structure	μm
X.MIN	Real	left hand side of structure	μm
Y.MAX	Real	bottom of structure	μm
Y.MIN	Real	top of structure	μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

The **INTERFACE** statement consists of a set of boundary condition parameters for the interface and a set of parameter to localize the effect of these parameters.

Boundary Condition Parameters

ABSORPTION	Specifies the fraction of the incident power lost to absorption in diffusive reflection.
ADF.TABLE	The name of the file containing a table describing the angular distribution function. The first entry is the number of samples. The following rows contain two numbers each. The first number is the angle in degrees, and the second number is the value of the ADF.
AR.ABSORB	Specifies the absorption coefficient of a anti-refractive coating layer. Default value is 0.0 cm^{-1}.
AR.INDEX	Specifies the real component refractive index for the anti-reflective coating model in Luminous. See Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method" for more information about the anti-reflective coating model.
AR.MATERIAL	Specifies the material type for an anti-reflective layer. The alias for this parameter is **MATERIAL**.
AR.THICK	Specifies the thickness of an anti-reflective coating layer for the reflection model in Luminous. This layer should generally not exist in the device mesh structure. See Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method" for more information about the anti-reflective coating model.
BESONOS	If DYNASONOS is enabled, this additionally enables the BESONOS model. This allows the use of a Band-Engineered tunnel insulator stack.
CHARGE	Specifies interface charge density (cm^{-2}) applied between two materials. The additional parameters (S.I, S.S, and S.X) allow this value to be applied at semiconductor-insulator, semiconductor-semiconductor, and semiconductor-domain edges respectively. A value of 1e10 cm^{-2} represents 1e10 electronic charges per cm^{-2} at the interface. A positive value will introduce a positive charge value and a negative value will introduce a negative charge value.
COATING	Specifies the number of a multilayer anti-reflective coating (ARC) referred to in the **INTERFACE** statement. If the COATING parameter is not set, the first coating is assumed. You must specify coatings in order (i.e., you must define COATING=2 before you define COATING=3). Different coatings should not overlap. If this occurs, the later coating is assumed in the overlapping part. For more information about ARC, see Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method".
CONSTANT	Specifies a constant angular distribution function as described in Equation 11-42.
CR	Specifies a user tuning parameter of the reflection haze function (see Equation 11-41).
CT	Specifies a user tuning parameter of the transmission haze function (see Equation 11-40).

DANIELSSON	Enables the interface band-to-band tunnelling model for Type-II heterojunctions.
DD.TUNNEL	Controls the numerical integration in Equation 6-54. The integral in considered converged when the argument of the outer integral is less than the value of DD.TUNNEL.
NEUTRALIZE	Specifies that the polarization charge will be removed from the interface.

Note: The numerical integral starts at the peak of the barrier where the argument should be equal to 1.0. Reducing this value increases simulation time but may improve accuracy.

DIFFUSIVE	Enables diffusive reflection and transmission at an interface.
DISPERSION	Specifies the spread of diffusive reflection.
DNLS.ACC	Treats the interface traps in the DANIELSSON model as acceptor-like.
DNLS.DON	Treats the interface traps in the DANIELSSON model as donor-like.
DNLS.CN	Electron capture coefficient for DANIELSSON model.
DNLS.CP	Hole capture coefficient for DANIELSSON model.
DNLS.NT	Interface trap density for DANIELSSON model.
DNLS.ET	Interface trap level relative to equilibrium Fermi level for Danielsson model.
DY.TUNNEL	Controls the step size in the numerical integration in Equation 6-54. The value of DY.TUNNEL specifies the size of the energy step, dEx. The energy step is chosen as the product of the value for DY.TUNNEL and the energy change over the first triangle adjacent to the edge at the location in question. Reducing this value increases simulation time but may improve accuracy.
DYNASONOS	Enables the DYNASONOS model unless exlicitly cleared by using ^DYNASONOS. In this case, the SONOS model will be enabled.
ELLIPSE	Specifies a elliptical angular distribution function as described in Equation 11-47.
F.QF	Specifies the name of a file containing a C-Interpreter function describing the density of the interface fixed charge as a function of position.
GAUSS	Enables Gaussian distribution of diffusive reflection (see Equation 11-9).
ID	Specifies a reference number for use with the CIgetNodalVoid C-Interpreter callback function.
I.I	Specifies that the application of interface models specified in this **INTERFACE** statement should include insulator-insulator interfaces.
IITHERM	This sets up the interface between two insulators as being one that can model thermionic emission if they are both subsequently changed to wide bandgap semiconductors using the SEMICONDUCTOR parameter of the **MATERIAL** statement. To do this, use the THERMIONIC and I.I parameters on the **INTERFACE** statement.

INT.RESIST	Specifies the value of a non-zero interface resistance between a conductor region and a semiconductor region.
LAYER	Describes the order of layers in a coating. If LAYER is not specified, the first (top) layer is assumed. You must specify the layers in order (i.e., you must define LAYER=2 before you define LAYER=3). Note that you only need to specify the coordinates of the interface for the first layer of each coating.
LAMBERT	Specifies a Lambertian angular distribution function as described in Equation 11-46.
LAM1 **LAM2**	Specify the beginning and ending wavelengths in microns for outputting haze functions. See **OUT.HT** and **OUT.HR**.
LORENTZ	Enables Lorentzian distribution of diffusive reflection (see Equation 11-10).

Note: When specifying INT.RESIST, you should also specify S.C on the **INTERFACE** statement.

MATERIAL	Specifies material name for the layer of a coating defined by the statement. Setting the material of a coating in this way enables you to apply any refractive index model supported by **MATERIAL** statement in Atlas. See Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method" for more information about anti-reflective coatings. The alias for this parameter is **AR.MATERIAL**L.
MODIFY	Specifies that a previously defined INTERFACE statement will be modified. The previous values of fixed interface charge at the specified interface will be discarded and replace with the specified value or value from a C-Interpreter file.
N.I	Enables either the SONOS or DYNASONOS model.
NLAM	Specifies the number of samples to output haze functions. See **OUT.HT** and **OUT.HR**..
NR	Specifies a user tuning parameter of the reflection haze function (see Equation 11-41).
NT	Specifies a user tuning parameter of the transmission haze function (see Equation 11-40).
OPTICAL	Specifies the optical properties of the interface defined in the statement are modeled using Transfer Matrix Method. See Section 11.3 "Matrix Method" for more information.
OUTADF	Specifies the file name for outputting the angular distribution function.
OUT.HR **OUT.HT**	Specify file names for outputting reflective and transmissive haze functions for display in TonyPlot. See **NLAM**, **LAM1**, and **LAM2**.
PELTIER	Interfacial Peltier heat power term used in Thermal 3D (see Chapter 18 "Thermal 3D: Thermal Packaging Simulator")
PGG.NSIGMA	Sets the upper limit, as a multiple of the Gaussian width, for the calculation of the probability of thermionic emission over a barrier between two organic materials.

REFLECT	Specifies the reflection coefficient for ray tracing in Luminous. The value should be between 0.0 abd 1.0 inclusive. You should also specify OPTICAL and values for P1.X, P1.Y, and P2.Y.
QF	Specifies fixed oxide charge density (cm^{-2}) applied at a semiconductor to insulator interface. A value of 1e10 cm^{-2} represents 1e10 electronic charges per cm^{-2} at the interface. A positive value will introduce a positive charge value and a negative value will introduce a negative charge value.
S.C	Specifies that the application of interface models specified in this **INTERFACE** statement should include semiconductor-conductor interfaces.
S.I	Specifies that the application of interface models specified in this **INTERFACE** statement should include semiconductor-insulator interfaces.
S.M	Specifies that the application of interface models in this **INTERFACE** statement should include semiconductor-metlal interfaces.
S.N	Specifies the electron surface recombination velocity.
S.P	Specifies the hole surface recombination velocity.
S.S	Specifies that the application of interface models specified in this **INTERFACE** statement should include semiconductor-semiconductor interfaces.
S.X	Specifies that the application of interface models specified in this **INTERFACE** statement should include any interfaces including interfaces with the outside domain.
SAVE.ELEC	Saves electron thermionic jump probability to the SAVE.PJUMP files.
SAVE.HOLE	Saves hole thermionic jump probability to the SAVE.PJUMP files.
SAVE.PJUMP	The root of a set of filenames that are used to store the probability of thermionic emission over the barrier between pairs of regions.
SCATTERED	Specifies the number of rays scattering event for diffusive reflections.
SIGMA	Characterizes the mean feature size for rough interfaces as used in Equations 10-40 and 11-41.
SITHERM	This sets up the interface between a semiconductor and an insulator as being one which can model thermionic emission if the insulator is subsequently changed to wide bandgap semiconductors using the SEMICONDUCTOR parameter of the **MATERIAL** statement. Additionally, any hot carrier current generated at the interface is injected into the carrier continuity equations in the wide bandgap semiconductor. The usual algorithm for propagating the hot carrier current by drift only is inactive.
SEMIMINOR	Specifies the semiminor axis of the elliptical angular distribution function described in Equation 11-47.
SPECULAR	Specifies the fraction of the incident power that experiences specular reflection.
SPJ.DE	The minimum electric field between points for data stored in the SAVE.PJUMP files.
S.WELL.N	Specifies the electron surface recombination velocity at quantum well edges for the CAPT.SRH model.

S.WELL.P	Specifies the hole surface recombination velocity at quantum well edges for the CAPT.SRH model.
TCR.TABLEE **TCT.TABLE**	Specify the names of files containing tables describing the wavelength dependent haze function parameters CR and CT. The first entry of the tables is the number of samples. The following rows contain pairs, where the first number is the wavelength in microns and the second number is the value for CR or CT.
THERMIONIC	Specifies that carrier transport across an interface is modeled by thermionic emission. This model will only be applied at semiconductor/semiconductor boundaries. See Section 6.2.2 "The Thermionic Emission and Field Emission Transport Model" for more information on thermionic emission.
TRIANGLE	Specifies a triangular angular distribution function as described in Equation 11-43.
TUNNEL	Specifies that the carrier transport across the interface will account for thermionic field emission. When TUNNEL is specified, THERMIONIC should also be specified. See Section 6.2.2 "The Thermionic Emission and Field Emission Transport Model" for more information on thermionic emission.

Note: To use the **INTERFACE** statement to specify thermionic or tunneling interfaces, place the **INTERFACE** statement immediately after all **MESH**, **REGION**, **ELECTRODE**, and **DOPING** statements and before any **CONTACT**, **MODELS**, **MATERIAL**, **MOBILITY**, or **IMPACT** statements.

Position Parameters

A.MAX	Specifies the maximum angle for the application of the statement. This is used with 2D circular or 3D cylindrical structures.
A.MIN	Specifies the minimum angle for the application of the statement. This is used with 2D circular or 3D cylindrical structures.
INTMATERIAL	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, INTMATERIAL="oxide/silicon".
INTNAME	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, INTNAME="channel/substrate".
INTNUMBER	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, INTNUMBER="2/3".
INTREGION	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, INTREGION="channel/substrate".

NOM.ANGLE	Sets a maximum tunneling angle in DYNASONOS model.
NOS.ANGLE	Sets a maximum tunneling angle in DYNASONOS model.
NOM.DIST	Sets a maximum tunneling distance in DYNASONOS model.
NOS.DIST	Sets a maximum tunneling distance in DYNASONOS model.
R1MATERIAL **R2MATERIAL**	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, `R1MATERIAL=oxide R2MATERIAL=silicon`.
R1NAME and **R2NAME**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, `R1NAME=channel R2NAME=substrate`.
R1NUMBER and **R2NUMBER**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, `R1NUMBER=2 R2NUMBER=3`.
R1REGION and **R2REGION**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, `R1REGION=channel R2REGION=substrate`.
R.MAX	Specifies the maximum radius for the application of the statement. This is used with 2D circular or 3D cylindrical structures.
R.MIN	Specifies the minimum radius for the application of the statement. This is used with 2D circular or 3D cylindrical structures.
X.MIN, **X.MAX**, **Y.MIN**, and **Y.MAX**	Define a bounding box. Any semiconductor/insulator interfaces found within this region are charged. If there is only one interface in a device, a non-planar surface may be defined using a box which contains the whole device.
X.MIN	Specifies the left X coordinate of the bounding box.
X.MAX	Specifies the right X coordinate of the bounding box.
Y.MIN	Specifies the bottom Y coordinate of the bounding box.
Y.MAX	Specifies the top Y coordinate of the bounding box.

Z.MIN	Specifies the front Z coordinate of the bounding box. It is used in 3D modules only.
Z.MAX	Specifies the back Z coordinate of the bounding box. It is used in 3D modules only.
P1.X, **P1.Y**, **P2.X**, and **P2.Y**	Define a bounding box. Within this box must lie the interface that is to be represented as the anti-reflective coating. See Section 11.9 "Anti-Reflective (AR) Coatings for Ray Tracing and Matrix Method" for more information about the anti-reflective coating model.

Note: For anti-reflective coatings in Luminous 3D, use X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN, and Z.MAX to define the bounding box.

P1.X	Specifies the left X coordinate of the bounding box.
P2.X	Specifies the right X coordinate of the bounding box.
P1.Y	Specifies the bottom Y coordinate of the bounding box.
P2.Y	Specifies the top Y coordinate of the bounding box.
DEVICE	Specifies which device in a MixedMode simulation applies to the statement. The synonym for this parameter is **STRUCTURE**.
REGION	Specifies which region number applies to the statement.
STRUCTURE	This is a synonym for **DEVICE**.

MOS Example

This example defines an interface with both fixed charge and recombination velocities.

```
INTERFACE X.MIN=-4  X.MAX=4 Y.MIN=-0.5 Y.MAX=4 \
          QF=1E10 S.N=1E4 S.P=1E4
```

SOI Example

To define different fixed charge on the front and back interfaces of an SOI transistor, you need two **INTERFACE** statements.

In the syntax below, the first statement will apply 5.10^{10} cm^{-2} charge to any silicon/oxide interface above Y=0.01µm. The second statement applied a higher charge to any interface below Y=0.01µm. Note that charges are only applied at the material interfaces so the Y coordinate needs only to be somewhere within the silicon film.

```
INTERFACE Y.MAX-0.01 QF=5e10

INTERFACE Y.MIN=0.01 QF=2e11
```

Interface Charge for III-V Devices

By default, the **INTERFACE** statement is applied to semiconductor-insulator interfaces. Interface charge can, however, be added at the interfaces between two semiconductor regions or at the edges of semiconductor regions.

The CHARGE parameter defines the interface charge value in cm^{-2}. The S.I, S.S, and S.X parameters control whether the charge is placed between semiconductor-insulator regions, semiconductor-semiconductor regions, or at the semiconductor domain edges. You can control the location of the added charge by using the position parameters.

2D Circular and 3D Cylindrical Limits Example

```
INTERFACE    S.S    A.MIN=0.0    A.MAX=180.0    R.MIN=0.1    R.MAX=0.2
CHARGE=1.0e12
```

This restricts the application of interface charge to being between 0 and 180° and between a radius of 0.1 um and 0.2 um. The angular and radial limits can also be used with the QF, S.N, S.P, and THERMIONIC parameters.

22.25 INTTRAP

INTTRAP activates interface defect traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
INTTRAP <type> E.LEVEL=<r> DENSITY=<r> <capture parameters>
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
DEGEN.FAC	Real	1	
DENSITY	Real		cm^{-2}
DEPTH	Real	5.0e-3	μm
DEVICE	Character		
DONOR	Logical	False	
E.LEVEL	Real		eV
E0N	Real		s^{-1}
E0P	Real		s^{-1}
F.E0N	Real		s^{-1}
F.E0P	Real		s^{-1}
F.DENSITY	Character		
F.MSCRATES	Character		
GRA.3.DET	Logical	False	
GRA.3.STO	Logical	False	
GRA.4.DET	Logical	False	
GRA.4.STO	Logical	False	
GRA.5.DET	Logical	False	
GRA.5.STO	Logical	False	
GRA.EA	Real	1.0	eV
GRA.EA.SD	Real	0.025	eV
GRA.EB.SD	Real	0.025	eV
GRA.EB.ELEC	Real	0.054	eV
GRA.EB.HOLE	Real	0.054	eV

Parameter	Type	Default	Units
GRA.EC.ELEC	Real	0.02	eV
GRA.EC.HOLE	Real	0.02	eV
GRA.EC.SD	Real	0.025	eV
GRA.ED	Real	0.1	eV
GRA.ED.SD	Real	0.1, GRA.ED	eV
GRA.ET1	Real	-0.6	eV
GRA.ET2	Real	0.2	eV
GRA.ET4	Real	0.25	eV
GRA.ET1.SD	Real	0.025	eV
GRA.ET2.SD	Real	0.025	eV
GRA.ET4.SD	Real	0.025	eV
GRA.FC	Real	1.0×10^{7}	V/cm
GRA.FC.SD	Real	0.1, GRA.FC	V/cm
GRA.GAMMA	Real	0.0	[Q] cm
GRA.GAMMA.SD	Real	0.0	[Q] cm
GRA.NU	Real	1.0×10^{13}	Hz
GRA.NU.SD	Real	0.1, GRA.NU	Hz
GRA.SAMPLES	Integer	10	
GRA.SIGN	Real	3.0×10^{-14}	cm2
GRA.SIGN.SD	Real	0.0	eV
GRA.SIGP	Real	3.0×10^{-14}	cm2
GRA.SIGP.SD	Real	0.0	eV
HEIMAN	Logical	False	
HPOINTS	Real	10	
I.C	Logical	False	
I.I	Logical	False	
I.M	Logical	False	
INTMATERIAL	Character		
INTNAME	Character		

Parameter	Type	Default	Units
INTNUMBER	Character		
INTREGION	Character		
MIDGAP	Logical	False	
MSC.NSTATES	Integer	0	
MSC1.CHARGE	Integer	0	
MSC1.ELEC	Integer	0	
MSC1.ELEVEL	Real	0	eV
MSC1.ET.CB	Logical	False	
MSC1.ET.VB	Logical	False	
MSC1.FT0	Real	0	
MSC1.H1	Integer	0	
MSC1.H2	Integer	0	
MSC1.HOLE	Integer	0	
MSC1.SP1	Integer	0	
MSC1.SP2	Integer	0	
MSC1.SP3	Integer	0	
MSC2.CHARGE	Integer	0	
MSC2.ELEC	Integer	0	
MSC2.ELEVEL	Real	0	eV
MSC2.ET.CB	Logical	False	
MSC2.ET.VB	Logical	False	
MSC2.FT0	Real	0	
MSC2.HOLE	Integer	0	
MSC2.H1	Integer	0	
MSC2.H2	Integer	0	
MSC2.SP1	Integer	0	
MSC2.SP2	Integer	0	
MSC2.SP3	Integer	0	
MSC3.CHARGE	Integer	0	

Parameter	Type	Default	Units
MSC3.ELEC	Integer	0	
MSC3.ELEVEL	Real	0	eV
MSC3.ET.CB	Logical	False	
MSC3.ET.VB	Logical	False	
MSC3.FT0	Real	0	
MSC3.H1	Integer	0	
MSC3.H2	Integer	0	
MSC3.HOLE	Integer	0	
MSC3.SP1	Integer	0	
MSC3.SP2	Integer	0	
MSC3.SP3	Integer	0	
MSC4.CHARGE	Integer	0	
MSC4.ELEC	Integer	0	
MSC4.FT0	Real	0	
MSC4.H1	Integer	0	
MSC4.H2	Integer	0	
MSC4.HOLE	Integer	0	
MSC4.SP1	Integer	0	
MSC4.SP2	Integer	0	
MSC4.SP3	Integer	0	
MSC5.CHARGE	Integer	0	
MSC5.ELEC	Integer	0	
MSC4.ELEVEL	Real	0	eV
MSC4.ET.CB	Logical	False	
MSC4.ET.VB	Logical	False	
MSC5.ELEVEL	Real	0	eV
MSC5.ET.CB	Logical	False	
MSC5.ET.VB	Logical	False	
MSC5.FT0	Real	0	

Parameter	Type	Default	Units
MSC5.H1	Integer	0	
MSC5.H2	Integer	0	
MSC5.HOLE	Integer	0	
MSC5.SP1	Integer	0	
MSC5.SP2	Integer	0	
MSC5.SP3	Integer	0	
MSCTRAP	Logical	False	
QWELL	Logical	False	
R1MATERIAL	Character		
R1NAME	Character		
R1NUMBER	Integer		
R1REGION	Character		
R2MATERIAL	Character		
R2NAME	Character		
R2NUMBER	Integer		
R2REGION	Character		
REGION	Integer		
S.C	Logical	False	
S.I	Logical	True	
S.M	Logical	False	
S.S	Logical	False	
S.X	Logical	False	
SIGN	Real		cm2
SIGP	Real		cm2
SR.EMIT	Real	0.5	eV
SR.HEIMAN	Logical	False	
SR.TRAP	Real	0.5	eV
STRUCTURE	Character		
TAUN	Real		s

Parameter	Type	Default	Units
TAUP	Real		s
X.MIN	Real	left of structure	μm
X.MAX	Real	right of structure	μm
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MIN	Real	back of structure	μm
Z.MAX	Real	front of structure	μm

Description

DEVICE	Specifies which device the statement applies to in MixedMode simulation. The synonym for this parameter is STRUCTURE.
DONOR	Specifies a donor-type trap level.
ACCEPTOR	Specifies an acceptor-type trap level.
DEGEN.FAC	Specifies the degeneracy factor of the trap level used to calculate the density.
DENSITY	Sets the maximum density of states of the trap level.
DEPTH	The depth that the uniform interface trap density penetrates into the insulator. This requires the HEIMAN model.
E.LEVEL	Sets the energy of the discrete trap level. It is equal to the energy distance between conductance band and trap level for acceptor trap, and to energy distance between trap level and valence band for donor trap.
E0N	Specifies the trap emission rate for elections.
E0P	Specifies the trap emission rate for holes.
F.E0N	Specifies the name of a file containing a C-Interpreter function describing the trap emission rate for electrons as a function of time.
F.E0P	Specifies the name of a file containing a C-Interpreter function describing the trap emission rate for holes as a function of time.
F.DENSITY	Specifies the name of a file containing a C-Interpreter function describing the density of donor/acceptor interface traps as a function of position.
HEIMAN	Enables the hysteresis interface traps model.
HPOINTS	Number of uniformly spaced points inside the insulator for which the interface trap density is calculated. This requires the HEIMAN model.
I.C	Specifies that the INTTRAP statement should apply to Insulator-Conductor interfaces.

I.I	Specifies that the **INTTRAP** statement should apply to Insulator-Insulator interfaces.
I.M	Specifies that the **INTTRAP** statement should apply to Insulator-Electrode interfaces.
INTMATERIAL	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, INTMATERIAL="oxide/silicon".
INTNAME	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, INTNAME="channel/substrate".
INTNUMBER	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, INTNUMBER="2/3".
INTREGION	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, INTREGION="channel/substrate".
MIDGAP	Specifies that the energy of the trap be set to the middle of the bandgap.
QWELL	Specifies that the INTTRAP will only be used at an interface with a QWELL region. It is only active in the Capture-Escape model.
R1MATERIAL **R2MATERIAL**	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, R1MATERIAL=oxide R2MATERIAL=silicon.
R1NAME **R2NAME**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, R1NAME=channel R2NAME=substrate.
R1NUMBER **R2NUMBER**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, R1NUMBER=2 R2NUMBER=3.
R1REGION **R2REGION**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, R1REGION=channel R2REGION=substrate.
S.C	Specifies that the **INTTRAP** statement should apply to semiconductor-conductor interfaces.
S.I	Specifies that the **INTTRAP** statement should apply to semiconductor-insulator interfaces.
S.M	Specifies that the **INTTRAP** statement should apply to semiconductor-metal interfaces.

S.S	Specifies that the **INTTRAP** statement should apply to semiconductor-semiconductor interfaces.
S.X	Specifies that the **INTTRAP** statement should apply to all semiconductor-insulator interfaces, including those to the outside domain.
SR.EMIT	Activation energy for temperature dependent emission coefficients in SR.HEIMAN model.
SR.HEIMAN	Enables the structural relaxation enhancement of the Heiman model.
SR.TRAP	Activation energy for temperature dependent capture coefficients in SR.HEIMAN model.
STRUCTURE	This is a synonym for **DEVICE**.
X.MIN	Specifies the left boundary of a box, where an interface must exist and where traps are to be applied.
X.MAX	Specifies the right boundary of a box, where an interface must exist and where traps are to be applied.
Y.MIN	Specifies the top boundary of a box, where an interface must exist and where traps are to be applied.
Y.MAX	Specifies the bottom boundary of a box, where an interface must exist and where traps are to be applied.
Z.MIN	Specifies the back boundary of a box, where an interface must exist and where traps are to be applied.
Z.MAX	Specifies the front boundary of a box, where an interface must exist and where traps are to be applied.

2-State NBTI Degradation Models Parameters

GRA.3.DET	Enables the deterministic 3-state model.
GRA.3.STO	Enables the stochastic 3-state model.
GRA.4.DET	Enables the deterministic 4-state model.
GRA.4.STO	Enables the stochastic 4-state model.
GRA.5.DET	Enables the deterministic 5-state model.
GRA.5.STO	Enables the stochastic 5-state model.
GRA.EA	Activation energy for process of relaxation back to precursor state from neutral switching state.
GRA.EA.SD	Variation parameter for GRA.EA.
GRA.EB.SD	Variation parameter for GRA.EB.ELEC and GRA.EB.HOLE.
GRA.EB.ELEC	Activation energy for MPFAT process of electron emission by the precursor.

GRA.EB.HOLE	Activation energy for MPFAT process of electron emission by the precursor.
GRA.EC.ELEC	Activation energy for the switching trap electron rates.
GRA.EC.HOLE	Activation energy for the switching trap hole rates.
GRA.EC.SD	Variation parameter for GRA.EC.ELEC and GRA.EC.HOLE.
GRA.ED	Activation energy for the dangling bond depassivation rate.
GRA.ET1	Energy level of precursor state relative to valence band.
GRA.ET2	Energy level of switching trap relative to valence band.
GRA.ET4	Energy level of P_b dangling bond state relative to valence band.
GRA.ET1.SD	Variation parameter for GRA.ET1.
GRA.ET2.SD	Variation parameter for GRA.ET2.
GRA.ET4.SD	Variation parameter for GRA.ET4.
GRA.FC	Field scaling factor for MPFAT tunneling process.
GRA.FC.SD	Variation parameter for GRA.FC.
GRA.GAMMA	Coefficient for field dependent thermal activation energy in dangling bond depassivation rates.
GRA.GAMMA.SD	Variation parameter for GRA.GAMMA.
GRA.ED.SD	Standard deviation parameter for GRA.ED.
GRA.NU	Attempt frequency for some processes.
GRA.NU.SD	Variation parameter for GRA.NU.
GRA.SAMPLES	Number of sample traps per interface point for stochastic models.
GRA.SIGN	Electron capture cross-section of trap (alias for **SIGN**).
GRA.SIGN.SD	Variation parameter for GRA.SIGN.
GRA.SIGP	Hole capture cross-section of trap (alias for **SIGP**).
GRA.SIGP.SD	Variation parameter for GRA.SIGP.

Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN	Specifies the capture cross section of the trap for electrons.
SIGP	Specifies the capture cross section of the trap for holes.
TAUN	Specifies the lifetime of electrons in the trap level.
TAUP	Specifies the lifetime of holes in the trap level.

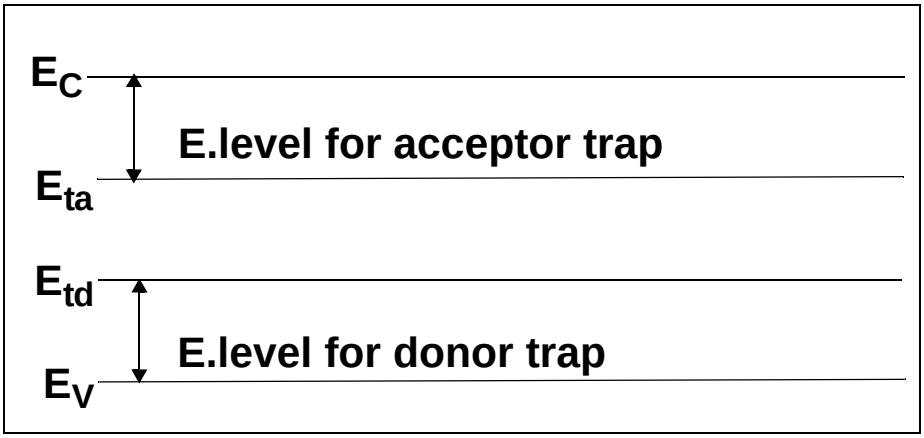


Figure 22-6: Acceptor and Donor Interface Trap Energy Levels

Multistate Trap Model

F.MSCRATES	Specifies the file containing the C-Interpreter function for transition rates in multistate trap model.
MSC.NSTATES	Specifies the number of internal states in the multistate trap model.
MSC1.CHARGE	Effective Charge of state 1 of Multistate model.
MSC1.ELEC	Number of trapped electrons in state 1 of Multistate model.
MSC1.ELEVEL	Energy level associated with state 1 of Multistate model.
MSC1.ET.CB	If true, then MSC1.ELEVEL is relative to conduction band edge.
MSC1.ET.VB	If true, then MSC1.ELEVEL is relative to valence band edge.
MSC1.FT0	Initial concentration of state 1 of Multistate model.
MSC1.HOLE	Number of trapped holes in state 1 of Multistate model.
MSC1.H1	Alias for **MSC1.SP1** and used for atomic hydrogen.
MSC1.H2	Alias for **MSC1.SP2** and used for molecular hydrogen.
MSC1.SP1	Number of trapped species 1 ions in state 1 of Multistate model.
MSC1.SP2	Number of trapped species 2 ions in state 1 of Multistate model.
MSC1.SP3	Number of trapped species 3 ions in state 1 of Multistate model.
MSC2.FT0	Initial concentration of state 2 of Multistate model.
MSC2.CHARGE	Effective Charge of state 2 of Multistate model.
MSC2.ELEVEL	Energy level associated with state 2 of Multistate model.
MSC2.ELEC	Number of trapped electrons in state 2 of Multistate model.
MSC2.ET.CB	If true, then MSC2.ELEVEL is relative to conduction band edge.

MSC2.ET.VB	If true, then MSC2.ELEVEL is relative to valence band edge.
MSC2.H1	Alias for **MSC2.SP1** and used for atomic hydrogen.
MSC2.H2	Alias for **MSC2.SP2** and used for molecular hydrogen.
MSC2.HOLE	Number of trapped holes in state 2 of Multistate model.
MSC2.SP1	Number of trapped species 1 ions in state 2 of Multistate model.
MSC2.SP2	Number of trapped species 2 ions in state 2 of Multistate model.
MSC2.SP3	Number of trapped species 3 ions in state 2 of Multistate model.
MSC3.CHARGE	Effective Charge of state 3 of Multistate model.
MSC3.ELEC	Number of trapped electrons in state 3 of Multistate model.
MSC3.FT0	Initial concentration of state 3 of Multistate model.
MSC3.H1	Alias for **MSC3.SP1** and used for atomic hydrogen.
MSC3.H2	Alias for **MSC3.SP2** and used for molecular hydrogen.
MSC3.HOLE	Number of trapped holes in state 3 of Multistate model.
MSC3.SP1	Number of trapped species 1 ions in state 3 of Multistate model.
MSC3.SP2	Number of trapped species 2 ions in state 3 of Multistate model.
MSC3.SP3	Number of trapped species 3 ions in state 3 of Multistate model.
MSC3.ELEVEL	Energy level associated with state 3 of Multistate model.
MSC3.ET.CB	If true, then MSC3.ELEVEL is relative to conduction band edge.
MSC3.ET.VB	If true, then MSC3.ELEVEL is relative to valence band edge.
MSC4.CHARGE	Effective Charge of state 4 of Multistate model.
MSC4.ELEC	Number of trapped electrons in state 4 of Multistate model.
MSC4.FT0	Initial concentration of state 4 of Multistate model.
MSC4.H1	Alias for **MSC4.SP1** and used for atomic hydrogen.
MSC4.H2	Alias for **MSC4.SP2** and used for molecular hydrogen.
MSC4.HOLE	Number of trapped holes in state 4 of Multistate model.
MSC4.SP1	Number of trapped species 1 ions in state 4 of Multistate model.
MSC4.SP2	Number of trapped species 2 ions in state 4 of Multistate model.
MSC4.SP3	Number of trapped species 3 ions in state 4 of Multistate model.
MSC4.ELEVEL	Energy level associated with state 4 of Multistate model.
MSC4.ET.CB	If true, then MSC4.ELEVEL is relative to conduction band edge.

MSC4.ET.VB	If true, then MSC4.ELEVEL is relative to valence band edge.
MSC5.FT0	Initial concentration of state 5 of Multistate model.
MSC5.CHARGE	Effective Charge of state 5 of Multistate model.
MSC5.ELEC	Number of trapped electrons in state 5 of Multistate model.
MSC5.ELEVEL	Energy level associated with state 5 of Multistate model.
MSC5.ET.CB	If true, then MSC5.ELEVEL is relative to conduction band edge.
MSC5.ET.VB	If true, then MSC5.ELEVEL is relative to valence band edge.
MSC5.H1	Alias for **MSC5.SP1** and used for atomic hydrogen.
MSC5.H2	Alias for **MSC5.SP2** and used for molecular hydrogen.
MSC5.HOLE	Number of trapped holes in state 5 of Multistate model.
MSC5.SP1	Number of trapped species 1 ions in state 5 of Multistate model.
MSC5.SP2	Number of trapped species 2 ions in state 5 of Multistate model.
MSC5.SP3	Number of trapped species 3 ions in state 5 of Multistate model.
MSCTRAP	Enables the multistate trap model.

Multiple Interface Trap States Example

The following example sets three discrete interface trap levels within the silicon bandgap. These trap levels will capture carriers, which slows down the switching speed of any device. In this example, the capture cross sections are used to define the properties of each trap.

```
inttrap e.level=0.49 acceptor density=2.e10 degen=12 \
sign=2.84e-15 sigp=2.84e-14
inttrap e.level=0.41 acceptor density=1.e10 degen=12 \
sign=7.24e-16 sigp=7.24e-15
inttrap e.level=0.32 donor density=1.e10 degen=1 \
sign=1.00e-16 sigp=1.00e-17
```

Note: For semiconductor bulk trap levels, see Section 22.66 “TRAP”.

22.26 LASER

LASER defines physical models and model parameters for laser and Vertical Cavity Surface-Emitting Lasers (VCSEL) simulation. For more information about VCSEL, see Chapter 10 "VCSEL Simulator".

Syntax

LASER<parameters>

Parameter	Type	Default	Units
ABS.FCARRIER	Logical	False	
ABSORPTION	Logical	False	
ACLOWFREQ	Real	1	Hz
ATRAP	Real	0.5	
CAVITY.DEPTH	Real	1e16	
CAVITY.END	Real	0	
CAVITY.LENGTH	Real	100.0	cm
CAVITY.START	Real	0	
COUPLED	Logical	False	
DBR1.START	Real	0.0	μm
DBR1.FINAL	Real	0.0	μm
DBR2.START	Real	0.0	μm
DBR2.FINAL	Real	0.0	μm
DEVICE	Character		
EFFMODE	Integer	1	
EFINAL	Real	0.0	eV
EINIT	Real	0.0	eV
ESEP(DELTAE)	Real	0.0	eV
ETRANS	Logical	False	
F.MIRROR	Character		
FAR.NX	Integer	100	
FAR.NY	Integer	100	
FAR.SPANX	Real	180	
FAR.SPANY	Real	180	

FCARRIER	Logical	False	
FRONTMIRROR.LOC	Real	-999	μm
GAINMOD	Integer	-999	
HELM.GEOM	Character	2DXY	
HELM.WHICH	Character	LR or SR (cylind)	
HELM.PEC	Logical	True	
HELM.PMC	Logical	False	
HELM.TE	Logical	True	
HELM.TM	Logical	True	
HELM.SYM	Logical	True	
HELM.ASYM	Logical	True	
HYBRID	Logical	False	
HYBRID.SPECRES	Real	0.0001	eV
INDEX.BOTTOM	Real	1.0	
INDEX.MODEL	Integer	0	
INDEX.TOP	Real	1.0	
ITMAX	Integer	30	
LMODES	Logical	False	
LOSS.DIFFRACTION	Logical	False	
LOSSES	Real	0	
MAX.LMODES	Integer	20	
LX.MAX	Real	999	μm
LX.MIN	Real	-999	μm
LY.MAX	Real	999	μm
LY.MIN	Real	-999	μm
MATERIAL	Character		
MAXCH	Real	2.5	
MAXTRAPS	Integer	2	
MIRROR	Real	90.0	%
MODAL.REFLECTANCE	Logical	False	

MULTISAVE	Logical	True	
NAME	Character		
NB.ITEREIG	Integer	4	
NEAR.NX	Integer	100	
NEAR.NY	Integer	100	
NEFF	Real	3.57	
NEFF.DIR	Character	Y	
NMOD.NEWTON	Integer	50	
NMODE	Integer	1	
NX	Integer		
NSPEC	Integer	100	
NY	Integer		
OMEGA	Real	2.16×10^{15}	Hz
OMEGA.FINAL	Real	-999	1/s
OMEGA.INIT	Real	-999	1/s
ORBIT.NUM (AZIMUTH.NUM)	Integer	0	
ORB.MIN, ORB.MAX (AZI.MIN, AZI.MAX)	Integer	0	
PHOTON.ENERGY	Real	0	eV
PMINCONC	Real	1.0e-20	1/cm
PROJ	Logical	False	
PRT.EVAL	Logical	False	
PRT.ITEREIG	Logical	False	
REFLECT	Logical	False	
RF	Real	90.0	%
RF.FILE	Character		
REGION	Integer		
RR	Real	90.0	%
RR.FILE	Character		
S.HELM	Logical	False	

SIN	Real	100000	cm^2
SPEC.NAME	Character		
SPECSAVE	Integer	1	
SPONTANEOUS	Integer	1	
START	Logical	False	
STIM.HEAT	Logical	False	
STRUCTURE	Character		
TAUSS	Real	0.05	
TIMERATE	Logical	True	
TOLER	Real	0.01	
TRANS_ENERGY	Real		eV
TRAP	Logical	False	
V.HELM	Logical	False	
VCSEL.CHECK	Integer	1	
VCSEL.INCIDENCE	Integer	False	
XMIN	Real	min. X	μm
XMAX	Real	max. X	μm
YMIN	Real	min. Y	μm
YMAX	Real	max. Y	μm

Description

Localization Parameters

CAVITY.END	Larger of the y-coordinates bounding the laser cavity, excluding the DBR mirror.
CAVITY.START	Smaller of the y-coordinates bounding the laser cavity, excluding the DBR mirror.
DEVICE	Specifies which device the **LASER** statement should apply to in MixedMode simulation. The synonym for this parameter is **STRUCTURE**.
FRONTMIRROR.LOC	Specifies the location of the external mirror facing the bottom facet in the HYBRID model.
LX.MAX	Sets the upper x boundary of solution domain for vector Helmholtz solver.
LX.MIN	Sets the lower x boundary of solution domain for vector Helmholtz solver.

LY.MAX	Sets the upper y boundary of solution domain for vector Helmholtz solver.
LY.MIN	Sets the lower y boundary of solution domain for vector Helmholtz solver.
MATERIAL	Specifies which material from the table in Appendix B "Material Systems" that the **LASER** statement should apply. If a material is specified, then all regions defined as being composed of that material will be affected.
NAME	Specifies which region the **LASER** statement should apply. Note that the name must match the name specified in the NAME parameter of the **REGION** statement or the region number.
REGION	Specifies the region number to which these parameters apply. If there is more than one semiconductor region, specification of different parameters for each region is allowed. If **LASER** is not specified, all regions in the structure are changed.
STRUCTURE	This is a synonym for **DEVICE**.

Laser Model Parameters

ABS.FCARRIER	Enables the free carrier absorption model
ABSORPTION	Enables the absorption loss in photon rate equations.
ACLOWFREQ	Normalizes small signal AC response with respect to the response at this frequency.
ATRAP	Specifies the photon rate equation cutback ratio (see Section 9.5.3 "Numerical Parameters").
DBR1.START **DBR1.FINAL** **DBR2.START** **DBR2.FINAL**	These parameters specify the locations of the front and rear DBR mirrors for a VCSEL device (See Figure 22-7). The reflectivity of front and rear DBR mirrors will be calculated from the optical intensity profile.
CAVITY.DEPTH	The dimension of the cavity along the direction normal to the calculation plane in Atlas. By default, this is set to a large number to neglect the effects of optical confinement along the normal.
CAVITY.LENGTH	Specifies the cavity length in the longitudinal direction (in μm).
COUPLED	Specifies that the solution process for the photon rate equation is fully coupled to the Jacobian of the drift-diffusion equations.
EFFMODE	In case of effective index (1.5D) solver, this sets the number of modes in vertical direction.
EINIT and **EFINAL**	Specify the lower and upper photon energies. Laser will calculate multiple longitudinal photon rates within this range. Using wide ranges can slow down simulation.

ESEP(DELTAE)	Specifies the photon energy separation. If this isn't specified, Laser will automatically calculate the number of longitudinal modes based on the cavity length and the energy range.
ETRANS	Enables the selection of a specific transverse mode. The selected transverse mode will be the mode with the closest energy to TRANS_ENERGY.
F.MIRROR	Specifies the name of a file containing a C-Interpreter function that defines the front and rear mirror reflectivities as a function of wavelength.
FAR.NX and **FAR.NY**	Describe the number of samples to output for the far-field pattern in the X and Y directions. For more information about the far-field pattern, see Section 9.5.5 "Generation of Near-Field and Far-Field Patterns".
FAR.SPANX and **FAR.SPANY**	Specifies the range of angles, where the farfield pattern is to be calculated: [-FAR.SPANX/2; FAR.SPANX/2] and [-FAR.SPANY/2; FAR.SPANY/2]
FCARRIER	Enables the free carrier loss model in Laser.
GAINMOD	Sets the number of the gain model.
HELM.GEOM	Sets dimenionality and direction of optical eigenmode solver. Possible choices are 2DXY, 1DX,1DY,15DX, and 15DY.
HELM.WHICH	Determines which part of the eigenvalue spectrum has to be found. "LR" - eigenvalues with largest real part. "SR" - smallest real part. "SI" - smallest imaginary part.
HELM.PEC	Perfect electric conductor boundary condition for Helmholtz solver.
HELM.PMC	Perfect electric conductor boundary condition for Helmholtz solver.
HELM.TE and **HELM.TM**	In scalar Helmholtz solver, it allows or blocks a solution for TE, TM transverse modes.
HELM.SYM and **HELM.ASYM**	When REFLECT parameter is specified, it allows or blocks a solution for aymmetric and antisymmetric transverse modes.
HYBRID	Activates the effective frequency model for a cavity with non-uniform index of refraction along the mode propagation direction Y.

HYBRID.SPECRES	Sets the resolution for each spectral peak in the transmission coefficient spectrum. This resolution is is used only around each modal frequency (up to the number of modes specified). For the rest of the frequencies, the resolution is controlled by NSPEC, E.INIT, and E.FINAL parameters
INDEX.BOTTOM	Specifies the refractive index of the medium below the structure. The default value is 1.0.
INDEX.MODEL	Specifies whether the simple refractive index model (INDEX.MODEL=0) or the more complex gain dependent refractive index (INDEX.MODEL=1) is used.
INDEX.TOP	Specifies the refractive index of the medium above the structure. The default value is 1.0.

Note: When using INDEX.MODEL=1, a complex value Eigenvalue solver is used. This requires a refined X direction Laser mesh and a refined Y direction Laser mesh to ensure the accuracy of the solution. When the bulk refractive index model is used, only a refined Y direction Laser mesh is required for the Eigenvalue solver.

ITMAX	Specifies the maximum number of iterations allowed for Laser simulation at each bias point.
LMODES	Specifies that multiple longitudinal models are to be accounted for during laser simulation.
LOSS.DIFFRACTION	Specifies that diffraction is taken into account. If left unspecified (or set to false) modal reflection is only done for the zeroth Fourier component of the transverse profile, which only accounts for reflection at normal incidence.
LOSSES	Specifies the total losses in Equation 9-28.
MAX.LMODES	The maximum number of longitudinal to search for each transverse mode.
MAXCH	Specifies the maximum allowed relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.
MAXTRAPS	Specifies the maximum number of cutbacks for the photon rate equation (see Section 9.5.3 “Numerical Parameters”).
MIRROR	Specifies the percentage facet reflectivity for the mirror loss in Laser. 100% reflectivity is equivalent to no mirror loss. Both facets are assumed to have this value of reflectivity.
MODAL.REFLECTANCE	Includes the frequency dependent Fresnel reflection and the DBR reflectance individually for each mode in calculating the escape rate and modal loss in the photon rate equations.
MULTISAVE	Specifies the whether to save the transient laser spectrum as one file or multiple files.

NB.ITEREIG	Sets the ratio of basis size for iterative Helmholtz eigensolver to the number of requested transverse modes, which is given by NMODE parameter.
NEAR.NX NEAR.NY	Describe the number of samples to output for the near-field pattern in the X and Y directions. For more information about the near-field pattern, see Section 9.5.5 "Generation of Near-Field and Far-Field Patterns".
NEFF	Specifies the effective refractive index in Equation 9-23.
NEFF.DIR	Selects the direction along which to apply the effective index. Choose "Y" (default) as a first choice. The effective index is then calculated for each y on the Laser mesh by integrating the refractive index with the transverse modes along x direction.
NMOD.NEWTON	Sets the number of lasing modes, which enter Jacobian for Newton iterations. It can be smaller than the actual number of modes. It is decreased automatically, if it is larger than the actual number of modes.
NMODE	Specifies the number of transverse modes simulated.
NSPEC	Specifies the number of sampling points between EINIT and EFINAL in the reflectivity test. The default value is NSPEC=100.
NX	Defines the number of mesh divisions for the laser mesh in the X direction.
NY	Defines the number of mesh divisions for the laser mesh in the Y direction.
PHOTON.ENERGY	Specifies the energy of photons to be used in Equation 9-1.
PMINCONC	Sets the minimum photon concentration.
PROJ	Extrapolates that photon rates for the initial guess during bias ramp.
PRT.EVAL	Prints the eigenvalues (propagation constants or eigen frequencies) of the Helmholtz equation on the screen.
PRT.ITEREIG	Specifies that iterative Helmholtz eigensolver prints information about its internal parameters and convergence.
OMEGA	Specifies the lasing frequency to be used in Equation 9-1. If model 2 is used for simulation, then this parameter will estimate the lasing frequency. If that's the case, use the PHOTON.ENERGY parameter to specify photon energy instead.
OMEGA.FINAL	Sets the upper boundary for frequencies of multiple longitudinal mode.
OMEGA.INIT	Sets the lower boundary for frequencies of miltiple longitudinal mode.
ORBIT.NUM	In cylindrical Helmholtz solver, this sets orbital number.
ORB.MIN, ORB.MAX	In cylindrical Helmholtz solver, this sets the range of orbital numbers.

REFLECT	Specifies that only half of the Laser structure is to be simulated. The axis of symmetry in this case is at x=0. Specify the laser mesh so that the minimum X coordinate is zero.
RF **RR**	Specify the front and rear facet mirror reflectivities of Fabry-Perot type lasers. If these parameters aren't specified, then they will be calculated from the average mirror loss parameter (MIRROR.LOSS parameter in the **LASER** or **MODELS** statements). If MIRROR.LOSS is used, then RF and RR shouldn't be specified.

RF.FILE
RR.FILE
	Specify the front and rear facet mirror reflectivity files. If RF.FILE is specified, the rear reflectivity will be read from the specified file. A linear interpolation algorithm will calculate value of the reflectivity from the tabulated values with the bounds given by the first and last wavelength value. The wavelenght values in the file must be ascending order. The file format is as follows:
	`number of samples`
	`wavelength (in microns) reflectivity (in %)`

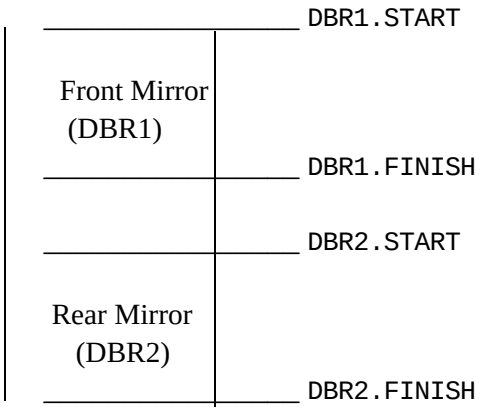


Figure 22-7: DBR Location Parameters

The cavity length used in the mirror loss calculation is given by DBR2.START - DBR1.FINISH. If both DBR mirrors are not defined, then the mirror loss will set to 0.

SIN	Specifies an initial photon density in the fundamental lasing mode. This value provides an initial guess for subsequent iterations. This parameter is used only when the single frequency model has been selected.
SPEC.NAME	Specifies the name of a spectrum file, which Laser will produce for each bias point, if the LMODES parameter has been specified.
SPECSAVE	The spectrum file will be saved after every LAS.SPECSAVE laser solution step.

SPONTANEOUS	Enables the Spontaneous Recombination Model (see Sections 3.9.1 "The General Radiative Recombination Model" and 3.9.2 "The Default Radiative Recombination Model").
STIM.HEAT	Includes stimulated recombination into GR source for heat equation.
START	Specifies that laser simulation starts now.
TAUSS	Specifies the relaxation parameter to be used for the photon rate equation. See Section 9.5.3 "Numerical Parameters" for more information.
TIMERATE	Specifies that the time dependent photon rate equation will be used in a transient laser simulation.
TOLER	Specifies the desired accuracy in photon areas.
TRANS_ENERGY	Specifies the energy for selective a single transverse mode (see also **ETRANS**).
TRAP	Specifies the photon rate equation cutback scheme (see Section 9.5.3 "Numerical Parameters").
V.HELM or **S.HELM**	Specifies that vector or scalar Helmholtz solver will be used for solution for transverse optical modes.
VCSEL.CHECK	Enables reflectivity test simulation of the VCSEL structure.
VCSEL.INCIDENCE	Specifies the direction of light incident on the structure. VCSEL.INCIDENCE=1 is the light incident from the top. VCSEL.INCIDENCE=0 is the light incident from the bottom. VCSEL.INCIDENCE=2 or >2 means both directions of light incidence are considered. By default, light is incident from the top of the structure.
XMIN	Defines the minimum X coordinate for the laser mesh. See also **LX.MESH**.
XMAX	Defines the maximum X coordinate for the laser mesh. See also **LX.MESH**.
YMIN	Defines the minimum Y coordinate for the laser mesh. See also **LY.MESH**.
YMAX	Defines the maximum Y coordinate for the laser mesh. See also **LY.MESH**.

22.27 LED

LED saves LED output characteristics after each subsequent solution step.

Syntax

```
LED <parameters>
```

Parameter	Type	Default	Units
AMBIEN.IMAG	Real	0.0	
AMBIEN.REAL	Real	1.0	
ANGLE.OUTPUT	Real	30.0	degrees
ANGPOWER	Character		
DIPOLE	Logical	False	
EDGE_ONLY	Logical	False	
EDGE_REFLECT	Logical	False	
EMIN	Real		eV
EMAX	Real		eV
ESAMPLE	Logical	False	
HORIZONTAL	Real	2/3	
INCOHERENT	Logical	False	
INCOH.TOP	Logical	False	
L.WAVE	Real	0.8	μm
LMAX	Real		μm
LMIN	Real		μm
MIN.POWER	Real	1×10^{-4}	
MIR.BOTTOM	Logical	False	
MIR.TOP	Logical	False	
NSAMP	Integer	100	
NUMRAYS	Integer	180	
ONEFILE	Logical	False	
POLAR	Real	0.0	
REFLECTS	Integer	0	
OUT.USPEC	Character		

Parameter	Type	Default	Units
SIDE	Logical	False	
SMOOTHNESS	Real	8.0	
SOURCE.TERM	Logical	False	
SOURCE_TERM	Logical	False	
SPECT.ANGLE	Character		
SPECTRUM	Character		
SURFACE_ONLY	Logical	False	
TEMPER	Real	300.0	
TR.MATRIX	Logical	False	
USER.SPECT	Character		
X	Real	0.0	μm
XMAX	Real	0.0	μm
XMIN	Real	0.0	μm
XNUM	Integer	0	
Y	Real	0.0	μm
YMAX	Real	0.0	μm
Y.MAX	Real	0.0	μm
YMIN	Real	0.0	μm
Y.MIN	Real	0.0	μm
YNUM	Integer	0	
ZMAX	Real	0.0	μm
Z.MAX	Real	0.0	μm
ZMIN	Real	0.0	μm
Z.MIN	Real	0.0	μm

Description

AMBIEN.IMAG	Specifies the imaginary index of refraction for the ambient outside the device domain for LED reverse ray tracing.
AMBIEN.REAL	Specifies the real index of refraction for the ambient outside the device domain for LED reverse ray tracing.
ANGLE.OUTPUT	Specifies the angular interval in degrees where spectrum versus angle are written to the file specified by the SPECT.ANGLE parameter.
ANGPOWER	Enables the reverse ray-tracing algorithm (see Section 12.6 "Reverse Ray-Tracing") for analysis of output coupling of light from the structure of a Light Emitting Diode. ANGPOWER specifies the name of the output file for the angular power density vs. output angle dependence.
DIPOLE	Specifies a particular angular distribution of the internal radiating field that corresponds to a preferred in-plane orientation of dipoles often relevant to OLED devices (see Section 12.6 "Reverse Ray-Tracing").
EDGE_ONLY	Specifies that for LED reverse ray tracing only the rays exiting from the edges are considered. Rays exiting at the top and bottom surfaces are not shown.
EDGE_REFLECT	Specifies the rays in reverse ray tracing reaching the edges of the device are totally reflected.
EMAX and **EMIN**	Specify the energy range for saving a spectrum file.
ESAMPLE	Specifies that samples are taken evenly in energy unlike even sampling in wavelength.
HORIZONTAL	Specifies the proportion of horizontally oriented dipoles for the dipole emission model.
INCOHERENT	Specifies that an incoherent material in LED/OLED structure is used for the purpose of combined transfer matrix method/reverse ray tracing (TTM+RRT) optical analysis.
INCOH.TOP	When specified with the SOURCE.TERM parameter, it allows a combination of transfer matrix method in thin coherent layers and ray tracing in thick incoherent layers for LED reverse ray tracing.
LMAX and **LMIN**	Specify the range of wavelength for saving spectrum files.
L.WAVE	Specifies the wavelength for reverse ray-tracing (see Section 12.6 "Reverse Ray-Tracing").
MIN.POWER	Specifies the minimum relative power of a ray (see Section 12.6 "Reverse Ray-Tracing"). The ray is not traced after its power falls below MIN.POWER value. This is useful to limit the number of rays traced. The default value is MIN.POWER= 1e-4.
MIR.TOP	Specifies that the top surface of the device be treated as an ideal mirror in reverse ray-tracing (see Section 12.6 "Reverse Ray-Tracing").

MIR.BOTTOM	Specifies that the bottom surface of the device be treated as an ideal mirror in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
NSAMP	Specifies the number of samples to use for a spectrum plot.
NUMRAYS	Specifies the number of rays starting from the origin in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”). The default is 180. Acceptable range is 36-3600.
ONEFILE	Specifies that successive files (output at each solution) will overwrite previous specified by SPECTRUM, ANGPOWER or SPECT.ANGLE or all three. By default, unique output files are written by incrementing the specified name.
POLAR	Specifies polarization of the emitted photons in degrees while linearly polarized light is assumed (see Section 12.6 “Reverse Ray-Tracing”). Parallel (TM-mode, POLAR=90.0) and perpendicular (TE-mode, POLAR=0.0) polarizations result in significantly different output coupling values. Use POLAR=45.0 if there is no preferred direction of polarization of emitted photons (unpolarized light emission).
REFLECTS	Specifies a number of reflections to be traced for each ray in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”). The default value is REFLECTS=0. The maximum allowed value is REFLECTS=10. Setting the number of reflections to 3 or 4 is often a good choice.
SIDE	Specifies that the rays reaching the sides of the device are terminated there and do not contribute to the total light output (see Section 12.6 “Reverse Ray-Tracing”).
SOURCE.TERM	Indicates that dipole source terms are combined with transfer matrix calculation of light output for LEDs. The alias is **SOURCE_TERM**.
SPECT.ANGLE	Specifies the output file name where LED spectrum is written as a function of emission angle.
SPECTRUM	Specifies the name of a file for saving spectrum plots.
SURFACE_ONLY	Specifies that for LED reverse ray tracing only the rays exiting from the top and bottom surfaces are considered. Rays exiting at the edges are not shown.
TEMPER	This is the temperature (needed for using appropriate refractive indexes of the materials in reverse ray-tracing). The default setting of 300 K will be used if TEMPER is unspecified.
TR.MATRIX	Specifies that the transfer matrix method should be used to handle thin film LEDs.
OUT.USPEC	Specifies the TonyPlot compatible file name associated with the USER.SPECT parameter.
X	This is the X coordinate of light origin for a single point reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
XMAX	This is the maximum X coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).

XMIN	This is the minimum X coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
XNUM	Specifies the number of points along X axis within a rectuangular area in multiple origin reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
Y	This is the Y coordinate of light origin for a single point reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
YMAX	This is the maximum Y coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
YMIN	This is the minimum Y coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
YNUM	Specifies the number of points along Y axis within a rectuangular area in multiple origin reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).

22.28 LENS

The LENS statement is used to specify lenslet characteristics used for light propagation analysis in Luminous.

Syntax

```
LENS <parameters>
```

Parameter	Type	Default	Units
ANGLE	Real	54.74	degrees
APEX	Real		μm
AR.INDEX	Real	1.0	
AR.THICKNESS	Real	0.0	
ASPHERIC	Logical	False	
COMPOSITE	Logical	False	
DENSITY	Real	0.0	per sq cm
ELLIPSE	Logical	False	
EXTINCT	Real	0.0	
F.LENS	Character		
GIZA	Logical	False	
HEIGHT	Real		μm
INDEX	Real	1.0	
K	Real	0.0	
MATERIAL	Character		
MEAN	Real	0.1	μm
N	Real	1.0	
N.FEATURES	Integer	0	
NEGATIVE	Logical	False	
NSAMP	Integer		
PC.DX	Real	0.0	μm
PC.DZ	Real	0.0	μm
PLANE	Real		
PYRAMID	Logical	False	
RADIUS	Real		μm

Parameter	Type	Default	Units
RANDOM	Logical	False	
SAG.OUT	Character		
SAG.ITS	Integer		
SEED	Integer	-10	
SIGMA	Real	0.1	μm
SPHERIC	Logical	False	
TWODEE	Logical	False	
USER	Logical	False	
WIDTH	Real		μm
X.LOC	Real		μm
X.MAX	Real		μm
X.MEAN	Real	0.1	μm
X.MIN	Real		μm
X.SAGS	Character		
X.SEMI	Real		μm
X.SIGMA	Real	0.1	μm
X.SOUT	Character		
Y.LOC	Real		μm
Y.SEMI	Real		μm
Y.SIGMA	Real	0.1	μm
Z.LOC	Character		
Z.MAX	Real		μm
Z.MEAN	Real	0.1	μm
Z.MIN	Real		μm
Z.SAGS	Character		
Z.SEMI	Real		μm
Z.SIGMA	Real	0.1	μm
Z.SOUT	Character		

Description

ANGLE	Specifies the elevation angke for all 4 sides of the pyramid(s) enabled by the GIZA flag.
APEX	Specifies the height of a pyramid lenslet.
AR.INDEX	Specifies the anti-reflective coating layer index of refraction for ray tracing.
AR.THICKNESS	Specifies the anti-reflective coating layer thickness in microns for ray tracing.
ASPHERIC	Specifies that the lenslet specification describes an aspheric lens.
COMPOSITE	Specifies that the lenslet specification describes a composite lens.
DENSITY	Specifies the density of primitives to be randomly placed when the parameter RANDOM is specified along with one of the primitive flags (e.g., SPHERE, GIZA, or CYLINDER).
ELLIPSE	Specifies that the lenslet specification describes an elliptical lens.
EXTINCT	Specifies the extinction coefficient of the lenslet.
F.LENS	Specifies the file name of a C-Interpreter function for user-definable lenslets.
HEIGHT	Specifies the height of a composite lens.
INDEX	Specifies the real part of the index of refraction of the lenslet.
GIZA	Specifies an equillateral pyramid with an elevation angle specified by the ANGLE parameter.
K	This is an alias for **EXTINCT**.
MATERIAL	Specifies the material name used to determine the complex index of refraction versus wavelength for the lens.
MEAN	Specifies the mean value of the random step size in the X and Z directions in microns for RANDOM lenslets.
N	This is an alias for **INDEX**.
N.FEATURES	Specifies the total number of primitive lens features to add when the RANDOM parameter is specified along with one of the flags associated with one of the primitives (e.g., SPHERE, GIZA, or CYLINDER).
NEGATIVE	Specifies that for spherical and elliptical lenses the concave surface is to be used rather than the convex side.
NSAMP	Specifies the number of samples for outputting aspheric lens data.
PC.DX **PC.DZ**	Specify the periodicity in x and z directions for lenslet representation of a photonic crystal.
PLANE	Specifies the Y location of the planar background of the lenslet.
PYRAMID	Specifies that the lenslet specification describes a Pyramid lens.
RADIUS	Specifies the radius of a spherical lenslet.

RANDOM	Specifies that the lenslet is a random textured surface lenslet.
SAG.OUT	Specifies the file name for aspheric lens output.
SAG.ITS	Specifies the number of iterations to be used for the Levenberg-Marquardt non-linear least squares algorithm for aspheric lenslet specification.
SEED	Specifies the random number generator seed value for random textured surface lenslets.
SIGMA	Specifies the standard deviation of the Gaussian distributed random step size in the X, Y, and Z directions for RANDOM lenslets in microns.
SPHERIC	Specifies that the lenslet specification describes a spheric lens.
TWODEE	Specifies that there is no variation in the lens in the z direction (i.e., the z coordinate is set to 0 in the analytic formula describing the lens surface).
USER	Specifies that the lenslet will use the C-Interpreter function specified by the F.LENS parameter for a user-definable lenslet.
WIDTH	Specifies the periphery width of a composite lenslet.
X.LOC **Y.LOC** **Z.LOC**	Specify the center coordinates of a spheric or pyramid lenslet.
X.MEAN	Specifies the mean value of the random step size in the X direction in microns for RANDOM lenslets.
X.MIN **X.MAX** **Z.MIN** **Z.MAX**	Specify the perimeter of a composite or pyramid lenslet.
X.SAGS **Z.SAGS**	Specify the file names of the input sag data for an aspheric lenslet.
X.SEMI **Y.SEMI** **Z.SEMI**	Specify the semimajor axes of a elliptical lenslet.
X.SIGMA	Specifies the standard deviation of the Gaussian distributed random step size in the X direction for RANDOM lenslets in microns.
X.SOUT **Z.SOUT**	Specify the output file names for the X and Z aspheric lenslet.
Y.SIGMA	Specifies the standard deviation of the Gaussian distributed random step size in the Y direction for RANDOM lenslets in microns.
Z.MEAN	Specifies the mean value of the random step size in the Z direction in microns for RANDOM lenslets.
Z.SIGMA	Specifies the standard deviation of the Gaussian distributed random step size in the Z direction for RANDOM lenslets in microns.

22.29 LOAD

LOAD loads previous solutions from files as initial guesses to other bias points.

Syntax

```
LOAD [ASCII|MASTER] [NO.CHECK] <files>
```

Parameter	Type	Default	Units
ASCII	Logical	False	
IN.FILE	Character		
INFILE	Character		
IN1FILE	Character		
IN2FILE	Character		
LAS.SPECTRUM	Logical	False	
MASTER	Logical	False	
NO.CHECK	Logical	False	
TWOD	Logical	False	

Description

ASCII	Specifies that any original PISCES format files read or written by this statement will be in an ASCII rather than in a binary format.
LAS.SPECTRUM	Loads the spectrum file (specified by the IN.FILE parameter) into Laser.

Note: The number of longitudinal and transverse modes in the spectrum file must be the same as created by the **LASER** statement.

MASTER	Specifies that any files read by this statement will be in a standard structure file rather than the original PISCES format. Specify this parameter if you are using TonyPlot to plot simulation results.
NO.CHECK	Prevents checking material parameter differences between loaded binary files and the values set in the current input file.
TWOD	Allows loading of a 2D solution into a 3D structure. Note that the values from the 2D solution are loaded uniformly in the Z direction.

File Parameters

The **LOAD** statement requires that one of the following file parameter syntax be used.

```
LOAD INFILE=<filename>
```

or

```
LOAD IN1FILE=<filename> IN2FILE=<filename>
```

IN.FILE	This is a synonym for **INFILE**.
INFILE	Specifies a single input filename for solution data. Use this parameter when you want to load only one solution, which is the most common case. The synonym for this parameter is **IN.FILE**.
IN1FILE	Specifies a filename for present solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the **SOLVE** statement).
IN2FILE	Specifies an input filename for previous solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the **SOLVE** statement). The solution specified by this parameter is the first to be overwritten when new solutions are obtained.

Simple Save and Load Examples

This example saves and loads the master format solution file, **SOL.STR**.

```
SAVE OUTF=SOL.STR.

....

LOAD INFILE=SOL.STR MASTER
```

As before but using the SOLVE syntax.

```
SOLVE OUTF=SOL.STR MASTER

..

LOAD INF=SOL.STR MASTER
```

When the save and load operations are not done within the same Atlas run see the note below.

Binary Format Example

Saving and loading using the binary format. This is quicker but these files cannot be plotted in TonyPlot.

```
SOLVE OUTF=SOLVE_TMP

 ..

LOAD INF=SOLVE_TMP
```

Note: The function to calculate the difference between two files is now inside TonyPlot. It has been discontinued from the **LOAD** statement

Note: The **LOAD** statement loads only the saved solution quantities into Atlas. The mesh, electrodes, doping, regions, contact settings, material parameters, models, and numerical methods must all be specified in advance of any **LOAD** statement. See "Re-initializing Atlas at a Given Bias Point" on page 99.

22.30 LOG

LOG allows all terminal characteristics of a run to be saved to a file. Any DC, transient, or AC data generated by **SOLVE** statements after the LOG statement is saved. Any parameters specified by the **PROBE** statement are also stored in the logfile. If a log file is already open, the open log file is closed and a new log file is opened.

Syntax

```
LOG [OUTFILE=<filename>] [MASTER] [acparams]
```

Parameter	Type	Default	Units
ABCD.PARAM	Logical	False	
APPEND	Logical	False	
CLOSE	Logical	False	
CSVFILE	Character		
DEVDEG	Logical	False	
FILE	Character		
GAINS	Logical	False	
H.PARAM	Logical	False	
IMPEDANCE	Real	50	Ω
INPORT	Character		
IN2PORT	Character		
J.DISP	Logical	False	
J.ELECTRON	Logical	False	
J.HEI	Logical	False	
J.HHI	Logical	False	
J.HOLE	Logical	False	
J.TUN	Logical	False	
LCOMMON	Real	0	H
LGROUND	Real	0	H
LIN	Real	0	H
LOUT	Real	0	H
MASTER	Logical	True	
NO.TRAP	Logical	False	
NOISE	Logical	False	

Parameter	Type	Default	Units
NOISE.ALL	Logical	False	
NOISE.I	Logical	False	
NOISE.IV	Logical	False	
NOISE.I.ALL	Logical	False	
NOISE.V	Logical	False	
NOISE.VI	Logical	False	
NOISE.V.ALL	Logical	False	
OFF	Logical	False	
OLD	Logical	False	
OUTPORT	Character		
OUT2PORT	Character		
OUT.FILE	Character		
OUTFILE	Character		
RCOMMON	Real	0	Ω
RGROUND	Real	0	Ω
RIN	Real	0	Ω
ROUT	Real	0	Ω
S.PARAM	Logical	False	
SIM.TIME	Logical	False	s
SONOS.CURR	Logical	False	
WIDTH	Real	1	μm
Y.PARAM	Logical	False	
Z.PARAM	Logical	False	

File Output Parameters

APPEND	Specifies that the output I-V information should be appended to an existing log file. Make sure that the existing log files contain the same type of data (e.g., DC, AC, transient) as the subsequent SOLVE statements.
CLOSE	This is an alias for OFF.
CSVFILE	Specifies the file name that will be used to store DC, AC, or transient IV information in Comma Seperated Value (CSV) format.

DEVDEG	When used with Device degradation models, it causes the total interface charges to be written to a LOG file.
FILE	This is an alias for **OUTFILE**.
J.DISP	Specifies that displacement currents are written to the log file.
J.ELECTRON	Specifies that electron currents are to be written into the log file.
J.HEI	Specifies that hot electron currents are written to the logfile.
J.HHI	Specifies that hot hole currents are written to the logfile.
J.HOLE	Specifies that hole currents are to be written into the log file.
J.TUN	Specifies that tunneling currents are to be written into the log file.
MASTER	Specifies that AC data and I-V information will be saved in a standard structure file format. This is the default format.
NO.TRAP	Specifies that no data will be written to the file for DC bias cut back steps.
OFF	Specifies that any currently open log file will be closed and log file output is discontinued. The alias for this parameter is **CLOSE**.
OLD	Specifies that AC data and IV information will be saved in the original PISCES-II file format. A synonym for this parameter is PISCES.
OUT.FILE	This is a synonym for **OUTFILE**.
OUTFILE	Specifies the log file that will be used to store DC, AC, or transient I-V information. The synonym for this parameter is **OUT.FILE**.
SONOS.CURR	Causes the output of SONOS Tunneling Insulator Current and SONOS Blocking Insulator Current from the DYNASONOS model.

Note: The older ACFILE syntax is not supported and should not be used. AC results are stored in the file specified by OUTFILE as long as the first **SOLVE** statement after the **LOG** statement contains AC analysis.

RF Analysis Parameters

If S.PARAM, H.PARAM, Z.PARAM, GAINS, or ABCD.PARAM is specified, the capacitance and conductance data will be converted into the requested set of AC parameters. See Appendix C "RF and Small Signal AC Parameter Extraction" for more information.

S.PARAM	Elects s parameter analysis. For S-parameter analysis, you can also choose to set any of the parasitic element parameters.
H.PARAM	Selects H parameter analysis.
Y.PARAM	Selects Y parameter analysis
Z.PARAM	Selects Z parameter analysis.
ABCD.PARAM	Selects ABCD parameter analysis.

GAINS	Selects the calculation of several types of gains used in RF analysis [252]. These are the stability factor, unilateral power gain (GUmax), maximum unilateral transducer power gain (GTmax), maximum available power gain (Gma), and the maximum stable power gain (Gms). The magnitude of H21 is also calculated.
IMPEDANCE	Specifies the matching impedance for S-parameter calculation.
INPORT	Specifies the electrode name for the primary input port used when performing any AC parameter calculations.
IN2PORT	Specifies the electrode name of the secondary input port.
OUTPORT	Specifies the electrode name for the output ports used when performing any AC parameter calculations.
OUT2PORT	Specifies the electrode n of the secondary output port.

NOISE Parameters

To perform noise analysis on a one-port device, define the INPORT. To perform noise analysis on a two-port device, define the INPORT and the OUTPORT.

NOISE	Selects F_{min}, Z_O, and g_n (the two-port noise figures of merit, 2pNFOM) for output.
NOISE.ALL	Selects all noise results for output. These are 2pNFOM, total and individual noise voltage sources, and total and individual noise current sources.
NOISE.I	Selects the 2pNFOM and the correlation of the total noise current sources.
NOISE.I.ALL	Selects the 2pNFOM, the correlation of the total noise current sources, and the correlation of the noise current sources from the individual noise mechanisms (GR, II, electron and hole diffusion; electron and hole flicker).
NOISE.V	Selects the 2pNFOM and the correlation of the total noise voltage sources.
NOISE.VI **NOISE.IV**	Selects the 2pNFOM and the correlations of both the total noise voltage sources and the total noise current sources.
NOISE.V.ALL	Selects the 2pNFOM, the correlation of the total noise voltage sources, and the correlation of the noise voltage sources from the individual noise mechanisms (GR, II, electron and hole diffusion; electron and hole flicker).

Note: The 2pNFOM have no meaning for a one-port device and therefore don't output. If NOISE is the only parameter defined for a one-port device, then the correlation of the total noise voltage sources outputs (rather than outputting nothing).

Parasitic Element Parameters

For RF parameter extraction, you can also set any of the parasitic element parameters. By setting the parasitic element parameters, you can apply lumped parasitic resistances or inductances to the terminal of the two-port device during the RF parameter extraction. These parameters will not affect the capacitance or conductance matrices calculated by Atlas.

RIN	Specifies the lumped parasitic resistance on the input to the two port device for s-parameter extraction. The value of RIN is in Ohms.
ROUT	Specifies the lumped parasitic resistance on the output to the two port device for s-parameter extraction. The value of ROUT is in Ohms.
RCOMMON	This is an alias for **RGROUND**.
RGROUND	Specifies the lumped parasitic resistance on the ground or common side of the two port device for s-parameter extraction. The value of RGROUND is in Ohms.
LIN	Specifies the lumped parasitic inductance on the input to the two port device for s-parameter extraction. The value of LIN is in Henrys.
LOUT	Specifies the lumped parasitic inductance on the output to the two port device for s-parameter extraction. The value of LOUT is in Henrys.
LCOMMON	This is an alias for **LGROUND**.
LGROUND	Specifies the lumped parasitic inductance on the ground or common side of the two port device for s-parameter extraction. The value of LGROUND is in Henrys.
SIM.TIME	Saves the time taken for a bias point into the log file and measured in seconds. Note that if multiple jobs are using the same CPU that this method may not be a true reflection of processor speed.
WIDTH	Specified an output width (in Z direction) to apply during the s-parameter calculation. Note that this parameter affects only the derived RF parameters and not currents, capacitances or conductances. The WIDTH parameter of the **MESH** statement can be used to scale these. Using both these WIDTH parameters will lead to a multiplication of the two widths for the RF parameters.

Simple Logfile Definition Example

This example saves all I-V data in file, myfile.log.

```
LOG OUTF=myfile.log
```

Results should be plotted using TonyPlot.

RF Analysis Example

To generate s-parameters, assume the input is gate/source and the output is drain/source. A width of 100 microns is also defined along with 100ohm resistance on the input. For example:

```
LOG OUTF=mysparams.log S.PARAM INPORT=gate OUTPORT=drain \
    IN2PORT=source OUT2PORT=source WIDTH=100 RIN=100
```

Transient or AC Logfile Example

The contents of LOG files varies for different types of simulations (e.g., DC, transient, AC). The content is set by the first **SOLVE** statement after the LOG statement. Therefore, the following syntax is required.

```
SOLVE VDRAIN=0.5

  LOG OUTF=myfile.log

  SOLVE VDRAIN=3.0 RAMPTIME=1e-9 DT=1e-11 TSTOP=1e-7
```

Correct transient parameters would have not been stored, if the **LOG** statement had been placed before the first **SOLVE** statement, which is DC.

22.31 LX.MESH, LY.MESH

L<n>.MESH specifies the location of grid lines along the <n>-axis in a rectangular mesh used in Laser simulation. The syntax is equivalent for X and Y directions.

Syntax

```
LX.MESH NODE=<n> LOCATION=<n>

LX.MESH SPACING=<n> LOCATION=<n>

LY.MESH NODE=<n> LOCATION=<n>

LY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE	Specifies the mesh line index. These mesh lines are assigned consecutively.
LOCATION	Specifies the location of the grid line.
SPACING	Specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

LASER Mesh Example

This syntax defines a mesh of 33×33 covering the area bounded by (0.3,0.0) to (2.4,1.0).

```
LX.M n=1 l=0.3

LX.M n=33 l=2.4

LY.M n=1 l=0.0

LY.M n=33 l=1.0
```

Note: The mesh defined in these statements for the Laser Helmholtz Solver is entirely separate from the electrical device simulation mesh defined on the MESH statement.

Setting Locally Fine Grids Example

This example shows how to space grid lines closely around a topological feature such as a junction at y=0.85 microns.

```
LY.MESH  LOC=0.0  SPAC=0.2

LY.MESH  LOC=0.85 SPAC=0.01

LY.MESH  LOC=2    SPAC=0.35
```

22.32 MATERIAL

MATERIAL associates physical parameters with materials in the mesh. The parameter default values for standard semiconductors are shown in Appendix B "Material Systems".

Syntax

```
MATERIAL <localization>  <material_definition>
```

Parameter	Type	Default	Units
A.DEFPOT	Real	2.1	eV
A.LANGEVIN	Real	1.0	
A.2D.LANGEVIN	Real	0.75	
A.PASSLER	Real	3.18×10^{-4}	eV/K
A.SINGLET	Real	1.8	nm
A.TRAPCOULOMBIC	Real	1.0	
A1.RAJ	Real	3.231×10^{2}	cm^{-1}
A2.RAJ	Real	7.237×10^{3}	cm^{-1}
AD.RAJ	Real	1.056×10^{6}	cm^{-1}
A1	Real	0.0	
A2	Real	0.0	
A3	Real	0.0	
A4	Real	0.0	
A5	Real	0.0	
A6	Real	0.0	
AADACHI	Real	see Appendix B	
ABSORPTION.SAT	Real	1.0×10^{7}	
AC	Real	0.0	eV
ADDPOLARON	Logical	False	
ADDSTATE	Logical	False	
AFFINITY	Real	see Appendix B	eV
ALATTICE	Real	0.0	A
ALIGN	Real	use **AFFINITY**	
ALPHA.2D.LANGEVIN	Real	0.037	

Parameter	Type	Default	Units
ALPHAA	Real	0.0	
ALPHAR	Real	4.0	
AN	Real	1.0	
AN2	Real	7.03×10^{5}	cm^{-1}
AN0.VALD	Real	4.3383	
AN1.VALD	Real	-2.42×10^{-12}	
AN2.VALD	Real	4.1233	
AP0.VALD	Real	2.376	
AP1.VALD	Real	0.01033	
AP2.VALD	Real	1.0	
AP	Real	1.0	
AP2	Real	1.682×10^{6}	cm^{-1}
ARADIUS.EXCITON	Real	0	nm
ARICHN	Real	see Appendix B	$A/cm^2/K^2$
ARICHP	Real	see Appendix B	$A/cm^2/K^2$
ASTR	Real	0.0	
ASYMMETRY	Real	0.5	
A.TC.A	Real	0.0	cm K /W
A.TC.B	Real	0.0	cm /W
A.TC.C	Real	0.0	cm / W /K
A.TC.CONST	Real	0.0	(W/cm/K)
A.TC.D	Real	0.0	K
A.TC.E	Real	0.0	W/cm
A.TC.NPOW	Real	0.0	-
AUG.CNL	Real	2.2×10^{-31}	cm^6/s
AUG.CPL	Real	9.2×10^{-32}	cm^6/s
AUG.CHI	Real	1.66×10^{-80}	cm^6/s
AUGN	Real	see Appendix B	cm^6/s

Parameter	Type	Default	Units
AUGP	Real	see Appendix B	cm^6/s
AUGKN	Num	0.0	cm^3
AUGKP	Num	0.0	cm^3
AV	Real	0.0	eV
B.TRAPCOULOMBIC	Real	1.0	
B.DEFPOT	Real	-2.33	eV
BADACHI	Real	see Appendix B	
BBB	Real	0.0	eV
BB.A	Real	4.0×10^{14}	$eV^{-2}*s^{-1}*cm^{-1}$
BB.B	Real	1.97×10^{7}	V/cm
BB.GAMMA	Real	2.5	
BBT.ALPHA	Real	0	
BETAN	Real	1.0	
BETAP	Real	1.0	
BETA.RAJ	Real	7.021×10^{4}	eV/K
BGN.C	Real	0.5	
BGN.E	Real	9.0×10^{-3} (Slotboom) 6.92×10^{-3} (Klaassen) 6.84×10^{-3} (Bennett) 1.87×10^{-2} (del Alamo)	eV
BGN.E.ACC	Real	See Table 3-8.	eV
BGN.E.DON	Real	See Table 3-8.	eV

Parameter	Type	Default	Units
BGN.N	Real	1.0×10^{17} (Slotboom) 1.3×10^{17} (Klaassen) 3.162×10^{18} (Bennett) 7.0×10^{17} (del Alamo)	cm^{-3}
BGN.N.ACC	Real	See Table 3-8.	cm^{-3}
BGN.N.DON	Real	See Table 3-8.	cm^{-3}
BGN.LIND.ANC	Real	See Table 3-7.	eV
BGN.LIND.ANV	Real	See Table 3-7.	eV
BGN.LIND.APC	Real	See Table 3-7.	eV
BGN.LIND.APV	Real	See Table 3-7.	eV
BGN.LIND.BNC	Real	See Table 3-7.	eV
BGN.LIND.BNV	Real	See Table 3-7.	eV
BGN.LIND.BPC	Real	See Table 3-7.	eV
BGN.LIND.BPV	Real	See Table 3-7.	eV
BGN.SHNK.EPS	Real	11.7	
BGN.SHNK.GE	Real	12	
BGN.SHNK.GH	Real	4	
BGN.SHNK.ME	Real	0.321	
BGN.SHNK.MH	Real	0.346	
BN	Real	1.0	
BN2	Real	1.231×10^{6}	V/cm
BN1.VALD	Real	0.0	
BN0.VALD	Real	0.235	
BP	Real	1.0	
BP2	Real	2.036×10^{6}	V/cm
BP0.VALD	Real	0.17714	
BP1.VALD	Real	-0.002178	

Parameter	Type	Default	Units
BQP.NGAMMA	Real	1.2	
BQP.NALPHA	Real	0.5	
BQP.PGAMMA	Real	1.0	
BQP.PALPHA	Real	0.5	
BSTR	Real	0.0	
BTE.AC_PHONON	Logical	True	
BTE.CB.ACOUST	Real	5.0	eV
BTE.CB.II_EG	Real	1.2	eV
BTE.CB.II_SCALE	Real	5.0	cm^2 eV s^{-1}
BTE.CB.INVSL	Real	3.04e7	cm
BTE.CB.NONPAR	Real	0.35	eV
BTE.CB.OPCC	Real	5.5×10^8	eV/cm
BTE.CB.OPHW	Real	0.052	eV
BTE.CB.ZION	Real	0.13	
BTE.DENSITY	Real	2329.0	kg/m^3
BTE.GEMODEL	Real	0	
BTE.IMPACT	Logical	False	
BTE.IONIZED	Logical	True	
BTE.OP_PHONON	Logical	True	
BTE.SOUND	Real	9000.0	m/s
BTE.VB.ACOUST	Real	3.68	eV
BTE.VB.II_EG	Real	1.7	eV
BTE.VB.II_SCALE	Real	5.0	cm^2 eV s^{-1}
BTE.VB.INVSL	Real	5.5e7	cm
BTE.VB.NONPAR	Real	0.2	eV
BTE.VB.OPCC	Real	4.45×10^8	eV/cm
BTE.VB.OPHW	Real	0.065	eV
BTE.VB.ZION	Real	0.11	
C.DEFPOT	Real	-4.75	eV

Parameter	Type	Default	Units
C.DIRECT	Real	0.0	cm^3/s
C1.KERR	Real	1.8×10^{-24}	$cm^{3\,(1+P1.KERR)}s^{-1}$
C2.KERR	Real	6.0×10^{-24}	$cm^{3\,(1+P2.KERR)}s^{-1}$
C3.KERR	Real	3.0×10^{-27}	$cm^{3\,(1+P3.KERR)}s^{-1}$
C1.RAJ	Real	5.5	
C2.RAJ	Real	4.0	
C1.RICHTER	Real	2.5×10^{-31}	cm^6s^{-1}
C2.RICHTER	Real	8.5×10^{-32}	cm^6s^{-1}
C3.RICHTER	Real	3.0×10^{-29}	$cm^{3\,(1+P3.RICHTER)}s^{-1}$
C11	Real	0.0	GPa
C12	Real	0.0	GPa
C13	Real	0.0	GPa
C33	Real	0.0	GPa
CAPT.AUGERN	Real	10^{-30}	cm^6s^{-1}
CAPT.AUGERP	Real	10^{-30}	cm^6s^{-1}
CAPT.MUN0	Real	0	cm^2/V_s
CAPT.MUP0	Real	0	cm^2/V_s
CAPT.SRH.TAUN	Real	10^{-3}	s
CAPT.SRH.TAUP	Real	10^{-3}	s
CDL.COUPLING	Real	1.0	$s^{-1}cm^{-3}$
CDL.ETR1	Real	0.0	eV
CDL.ETR2	Real	0.0	eV
CDL.TN1	Real	1.0	s
CDL.TN2	Real	1.0	s
CDL.TP1	Real	1.0	s
CDL.TP2	Real	1.0	s
CHI.EG.TDEP	Real	0.5	
CON.BGN	Real	0.5	

Parameter	Type	Default	Units
CN	Real	0.0	
CN2	Real	0.0	$cm^{-1}K^{-1}$
CN0.VALD	Real	1.6831×10^{4}	
CN1.VALD	Real	4.3796	
CN2.VALD	Real	1.0	
CN3.VALD	Real	0.13005	
COPT	Real	0.0	cm^{3}/s
CP	Real	0.0	
CP2	Real	0.0	$cm^{-1}K^{-1}$
CP0.VALD	Real	0.0	
CP1.VALD	Real	0.00947	
CP2.VALD	Real	2.4929	
CP3.VALD	Real	0.0	
CRELMAX.N	Real	0.5	
CRELMAX.P	Real	0.5	
CRELMIN.N	Real	1e-10	
CRELMIN.P	Real	1e-10	
CSTR	Real	0.0	
D.DEFPOT	Real	1.1	eV
D.TUNNEL	Real	10^{-6}	cm
D0.H1	Real	1.75×10^{-4}	$cm^{2}s^{-1}$
D0.H2	Real	1.75×10^{-4}	$cm^{2}s^{-1}$
D1	Real	0.0	eV
D2	Real	0.0	eV
D3	Real	0.0	eV
D4	Real	0.0	eV
DAA.LANGEVIN	Real	1	
DADACHI	Real	see Appendix B	eV

Parameter	Type	Default	Units
DAD.LANGEVIN	Real	1	
DEC.DEG	Real	see Table 6-15	
DEC.C1	Real	0.0	eV
DEC.C2	Real	0.0	eV
DEC.C3	Real	0.0	eV
DEC.ISO	Real	0.0	eV
DEC.D2	Real	0.0	eV
DEC.D4	Real	0.0	eV
DEV.HH	Real	0.0	eV
DEV.LH	Real	0.0	eV
DEV.ISO	Real	0.0	eV
DEV.SO	Real	0.0	eV
DEGENERACY	Integer	2	
DELTA1	Real	0.0	eV
DELTA2	Real	0.0	eV
DELTA3	Real	0.0	eV
DEVICE	Character		
DGN.GAMMA	Real	3.6	
DGP.GAMMA	Real	3.6	
DINDEXDT	Real	0.0	1/K
DIST.SBT	Real	10^{-6}	cm
DKCQ.EXCITON	Real	1.1×10^{-8}	s^{-1}
DKNRS.EXCITON	Real	0	s^{-1}
DKSS.EXCITON	Real	3×10^{-8}	cm^3s^{-1}
DKTP.EXCITON	Real	0	s^{-1}
DLDS.EXCITON	Real	0	μm
DN2	Real	0.0	$cm^{-1}K^{-1}$
DN0.VALD	Real	1.233735×10^{6}	

Parameter	Type	Default	Units
DN1.VALD	Real	1.2039×10^{3}	
DN2.VALD	Real	0.56703	
DOPANT	Logical	False	
DOPE.SPECT	Character		
DP2	Real	0.0	$cm^{-1}K^{-1}$
DP0.VALD	Real	1.4043×10^{6}	
DP1.VALD	Real	2.9744×10^{3}	
DP2.VALD	Real	1.4829	
DPHEFF.EXCITON	Real	1	
DRST.EXCITON	Real	0.25	
DTAUS.EXCITON	Real	5×10^{-9}	s
DTAUT.EXCITON	Real	0	s
DRHODT	Real	0.0	μΩ·cm/k
DSTR	Real	0.0	
E.CONDUC	Real	0.0	Ω/cm
E.FULL.ANISO	Logical	False	E.FULL.ANISO
E31	Real	0.0	cm^{-2}
E33	Real	0.0	cm^{-2}
EA.CUBIC	Real	See Table 6-31.	eV
EA.H1	Real	0.1685	$cm^{2}s^{-1}$
EA.H2	Real	0.1685	$cm^{2}s^{-1}$
EA.HEXAGONAL	Real	See Table 6-31.	eV
EAB	Real	0.045	eV
ECN.II	Real	1.231×10^{6}	
ECP.II	Real	2.036×10^{6}	
ED.CUBIC	Real	See Table 6-31.	eV
ED.HEXAGONAL	Real	See Table 6-31.	eV
EDB	Real	0.044	eV

Parameter	Type	Default	Units
EG1.RAJ	Real	1.1557	eV
EG2.RAJ	Real	2.5	eV
EG3.RAJ	Real	3.2	eV
EG300	Real	see Appendix B	eV
EG1300	Real	0.0	eV
EG2300	Real	0.0	eV
EG12BOW	Real	0.0	eV
EG1ALPH	Real	0.0	eV/K
EG2ALPH	Real	0.0	eV/K
EG1BETA	Real	0.0	K
EG2BETA	Real	0.0	K
EGALPHA	Real	see Appendix B	eV/K
EGBETA	Real	see Appendix B	K
ELEC.MP.POWER	Integer	3	
ELEC.MP.SIGMA	Real	10^{-14}	cm^2
ELEC.MP.THRESH	Real	1.0	eV
ELEC.SP.POWER	Integer	2	
ELEC.MP.SIGMA	Real	10^{-16}	cm^2
ELEC.SP.THRESH	Real	1.75	eV
EMISS.EFFI	Real	1.0	
EMISS.LAMB	Real	0.5	microns
EMISS.NX	Integer	1	
EMISS.NY	Integer	1	
EMISS.WIDE	Real	1.0	microns
EMISSION.FACTOR	Real	1.0	
EMIT.FILE	Character		
EMIT.RATE	Real	0.0	
EN	Real	0.0	
EP	Real	0.0	

Parameter	Type	Default	Units
EP.MBULK	Real	see Table 3-138	eV
EP1.RAJ	Real	1.827×10^{-2}	eV
EP2.RAJ	Real	5.773×10^{-2}	eV
EPS11	Real	0.0	
EPS12	Real	0.0	
EPS13	Real	0.0	
EPS22	Real	0.0	
EPS23	Real	0.0	
EPS33	Real	0.0	
EPS.XX	Real	0.0	
EPS.YY	Real	0.0	
EPS.ZZ	Real	0.0	
ESTR	Real	0.0	
ETRAP	Real	0.0	eV
EPSINF	Real		
EX.AMPLITUDE	Real	0.0	
EX.COUPLING	Real	0.0	eV
EX.ECEN	Real	0.0	eV
EX.EPHONON	Real	0.0	eV
EX.HOMOBROADENING	Real	0.0	eV
EX.HOPPING	Real	0.0	eV
EX.INHOMOBROADENING	Real	0.0	eV
EX.MAXCLOUDSIZE	Integer	0.0	
EX.MAXPHONONS	Integer	0.0	
EXCITON.GAP	Real	0.0	eV
EXCITON.DIPOLE1	Real	0.0	Angstorm
EXCITON.DIPOLE2	Real	0.0	Angstorm
EXN.II	Real	1.0	
EXP.II	Real	1.0	

Parameter	Type	Default	Units
F.ALPHAA	Character		
F.BANDCOMP	Character		
F.BBT	Character		
F.BGN	Character		
F.CAPT.MUN	Character		
F.CAPT.MUP	Character		
F.CBDOSFN	Character		
F.CONMUN	Character		
F.CONMUP	Character		
F.COPT	Character		
F.DSANNIHILATION	Character		
F.EPSILON	Character		
F.ENMUN	Character		
F.ENMUP	Character		
F.PDN	Character		
F.PDP	Character		
F.FERRO	Character		
F.GAUN	Character		
F.GAUP	Character		
F.INDEX	Character		
F.MAX	Real	0.0	V/cm
F.MIN	Real	0.0	V/cm
F.MNSNDIFF	Character		
F.MNSPDIFF	Character		
F.MNSNFLICKER	Character		
F.MNSPFLICKER	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.NUMBER	Integer	0	

Parameter	Type	Default	Units
F.RECOMB	Character		
F.SRH	Character		
F.TAUN	Character		
F.TAUP	Character		
F.TAURN	Character		
F.TAURP	Character		
F.TCAP	Character		
F.TCOND	Character		
F.VBDOSFN	Character		
F.VSATN	Character		
F.VSATP	Character		
FB.MBULK	Real	see Table 3-138	
FC.RN	Real	3.0×10^{-18}	cm^{-2}
FC.RP	Real	7.0×10^{-18}	cm^{-2}
FERRO.EC	Real	0.0	V/cm
FERRO.EPSF	Real	1.0	
FERRO.PS	Real	0.0	C/cm^2
FERRO.PR	Real	0.0	C/cm^2
FSTR	Real	0.0	
G.2D.LANGEVIN	Real	1	
GAIN.SAT	Real	1.0×10^{7}	
GAIN0	Real	2000.0	cm^{-1}
GAIN00	Real	-200.0	cm^{-1}
GAIN1N	Real	0	cm^2
GAIN1P	Real	0	cm^2
GAIN1MIN	Real	3.0×10^{-16}	cm^2
GAIN2NP	Real	0	cm^5
GAMMA	Real		

Parameter	Type	Default	Units
GAMMA.RAJ	Real	1108.0	K
GAUSS.TABLE	Logical	False	
GCB	Real	2.0	
GN1	Real	3.0×10^{-16}	cm^{-2}
GN2	Real	4.0×10^{-15}	cm^{-2}
G3MAX.N	Real	100.0	
G3MAX.P	Real	100.0	
G.SURF	Real	1.0	cm^{2}
GDEC.GAUSS	Real	3×(SIGC.GAUSS+GSIGC.GAUSS)	eV
G1DEC.GAUSS	Real	3×(SIGC.GAUSS+G1SIGC.GAUSS)	eV
G2DEC.GAUSS	Real	3×(SIGC.GAUSS+G2SIGC.GAUSS)	eV
GDEV.GAUSS	Real	3×(SIGV.GAUSS+GSIGV.GAUSS)	eV
G1DEV.GAUSS	Real	3×(SIGV.GAUSS+G1SIGV.GAUSS)	eV
G2DEV.GAUSS	Real	3×(SIGV.GAUSS+G2SIGV.GAUSS)	eV
GNTC.GAUSS	Real	0.1×NTC.GAUSS	cm^{-3}
G1NTC.GAUSS	Real	0.1×NTC.GAUSS	cm^{-3}
G2NTC.GAUSS	Real	0.1×NTC.GAUSS	cm^{-3}
GNTV.GAUSS	Real	0.1×NTV.GAUSS	cm^{-3}
G1NTV.GAUSS	Real	0.1×NTV.GAUSS	
G2NTV.GAUSS	Real	0.1×NTV.GAUSS	
GS.AMPLITUDE	Real	0.0	eV
GS.ECEN	Real	0.0	eV
GS.HOMOBROADENING	Real	0.0	eV
GS.INHOMOBROADENING	Real	0.0	eV
GS.MAXCLOUDSIZE	Integer	0.0	
GS.MAXPHONONS	Integer	0.0	
GSIGC.GAUSS	Real	SIGC.GAUSS	eV
G1SIGC.GAUSS	Real	SIGC.GAUSS	eV

Parameter	Type	Default	Units
G2SIGC.GAUSS	Real	SIGC.GAUSS	eV
GSIGV.GAUSS	Real	SIGV.GAUSS	eV
G1SIGV.GAUSS	Real	SIGV.GAUSS	eV
G2SIGV.GAUSS	Real	SIGV.GAUSS	eV
GONZALEZ.C0	Real	See Table 3-23	
GONZALEZ.C1	Real	See Table 3-23	
GONZALEZ.C2	Real	See Table 3-23	
GONZALEZ.C3	Real	See Table 3-23	
GONZALEZ.CEEBOW	Real	See Table 3-23	
GONZALEZ.TAUBOW	Real	See Table 3-23	s
GONZALEZ.TAUN0	Real	See Table 3-23	s
GONZALEZ.TAUN1	Real	See Table 3-23	s
GONZALEZ.TAUP0	Real	See Table 3-23	s
GVB	Real	4.0	
H1TOH2RATE	Real	1.0E-2	$cm^{-3}s^{-1}$
H2TOH1RATE	Real	100.0	s^{-1}
H1SRV	Real	0.0	cm/s
H2SRV	Real	0.0	cm/s
HC.A	Real	See Appendix B	J/Kcm^3
HC.B	Real	See Appendix B	J/K^2cm^3
HC.BETA	Real		
HC.C	Real	See Appendix B	J/K^3cm^3
HC.C1	Real		J/K/kg
HC.C300	Real		J/K/kg
HC.D	Real	See Appendix B	JK/cm^3
HC.RHO	Real		g/cm^3
HOOGEN	Real	0.0	
HNS.AE	Real	6.7×10^{-32}	cm^6/s

Parameter	Type	Default	Units
HNS.AH	Real	7.2×10^{-32}	cm^6/s
HNS.BE	Real	2.45×10^{-31}	cm^6/s
HNS.BH	Real	4.5×10^{-33}	cm^6/s
HNS.CE	Real	-2.2×10^{-32}	cm^6/s
HNS.CH	Real	2.63×10^{-32}	cm^6/s
HNS.HE	Real	3.4667	
HNS.HH	Real	8.25688	
HNS.N0E	Real	1.0×10^{18}	cm^{-3}
HNS.N0H	Real	1.0×10^{18}	cm^{-3}
HNU1	Real	1.0	eV
HNU2	Real	1.0	eV
HOLE.MP.POWER	Integer	3	
HOLE.MP.SIGMA	Real	10^{-16}	cm^2
HOLE.MP.THRESH	Real	1.0	eV
HOLE.SP.POWER	Integer	2	
HOLE.SP.SIGMA	Real	10^{-18}	cm^2
HOLE.SP.THRESH	Real	1.75	eV
HOOGEP	Real	0.0	
HOPN.BETA	Real	1.5	
HOPN.GAMMA	Real	5×10^{7}	cm^{-1}
HOPN.V0	Real	1×10^{11}	Hz
HOPP.BETA	Real	1.5	
HOPP.GAMMA	Real	5×10^{7}	cm^{-1}
HOPP.V0	Real	1×10^{11}	Hz
HUANG.RHYS	Real	3.5	
IF.CHAR	Real	0.0	microns
IG.ELINR	Real	6.16×10^{-6}	cm

Parameter	Type	Default	Units
IG.HLINR	Real	6.16×10^{-6}	cm
IG.ELINF	Real	9.2×10^{-7}	cm
IG.HLINF	Real	9.2×10^{-7}	cm
IINOFF	Character		
IIPOFF	Character		
IMAG.INDEX	Real	See Appendix B	
INCOHERENT	Logical	False	
INDEX.FILE	Character		
INDX.IMAG	Character		
INDX.REAL	Character		
INSULATOR	Logical	False	
J.ELECT	Real	0.0	A/cm^2
J.MAGNET	Real	0.0	V/cm^2
JTAT.M2	Real	10^{-20}	V^2cm^3
KRISC.EXCITON	Real	0	s^{-1}
K.SINGLET	Real		
KAUGCN	Real	1.83×10^{-31}	cm^6/s
KAUGCP	Real	2.78×10^{-31}	cm^6/s
KAUGDN	Real	1.18	
KAUGDP	Real	0.72	
KD.LID	Real	5×10^{-15}	cm^3s^{-1}
KDD.EXCITON	Real	0	s^{-1}
KEE.EXCITON	Real	0	s^{-1}
KH.LID	Real	5×10^{-18}	cm^3s^{-1}
KISC.EXCITON	Real	0	s^{-1}
KNRS.EXCITON	Real	0	s^{-1}
KNRT.EXCITON	Real	0	s^{-1}

Parameter	Type	Default	Units
KSP.EXCITON	Real	0	$cm^{-3}s^{-1}$
KSRHCN	Real	3.0×10^{-13}	cm^3
KSRHCP	Real	11.76×10^{-13}	cm^3
KSRHGN	Real	1.77	
KSRHGP	Real	0.57	
KSRHTN	Real	2.5×10^{-3}	s
KSRHTP	Real	2.5×10^{-3}	s
KSS.EXCITON	Real	0	$cm^{-3}s^{-1}$
KST.EXCITON	Real	0	$cm^{-3}s^{-1}$
KTP.EXCITON	Real	0	$cm^{-3}s^{-1}$
KTT.EXCITON	Real	0	$cm^{-3}s^{-1}$
L.2D.LANGEVIN	Real	1.6	nm
L1SELL	Real	See Appendix B	µm
L2SELL	Real	See Appendix B	µm
LAM1	Real	0	µm
LAM2	Real	0	µm
LAMDAE	Real	6.2×10^{-7}	cm
LAMDAH	Real	3.8×10^{-7}	cm
LAMHN	Real	9.2×10^{-7}	cm
LAMHP	Real	9.2×10^{-7}	cm
LAMRN	Real	6.16×10^{-6}	cm
LAMRP	Real	6.16×10^{-6}	cm
LAN300	Real	6.2×10^{-7}	cm
LAP300	Real	3.8×10^{-7}	cm
LDS.EXCITON	Real	0.01	µm
LDT.EXCITON	Real	0.632	µm
LOG.COLOR	Character		

Parameter	Type	Default	Units
LT.TAUN	Real	0.0	
LT.TAUP	Real	0.0	
LUTT1	Real	0.0	
LUTT2	Real	0.0	
LUTT3	Real	0.0	
M.DSN	Real	See Appendix B	
M.DSP	Real	See Appendix B	
M.VTHN	Real		
M.VTHP	Real		
MATERIAL	Character		
MAG.LOSS	Real	0.0	mho/cm
MBULKSQ	Real		eV
MC	Real	0.0	
ME.TUNNEL	Real		
MH.TUNNEL	Real		
ME.SBT	Real		
MH.SBT	Real		
MHH	Real	0.49	
MHHZ	Real		
MINIMA	Real	6	
MIX.NAME	Character		
ML	Real	0.916	
MLH	Real	0.16	
MLHZ	Real		
MSO	Real	0.23	
MSOZ	Real		
MSTAR	Real	0.0	
MTT	Real	0.0	
MT1	Real	0.191	

Parameter	Type	Default	Units
MT2	Real	0.191	
MUN	Real	See Appendix B	
MUNOFF	Character		
MUPOFF	Character		
MUP	Real	See Appendix B	
MV	Real	0.0	
MZZ	Real	0.0	
N.ANISO	Logical	False	
N.ALPHA	Real	0.0	degrees
N.BETA	Real	0.0	degrees
N.GAMMA	Real	0.0	degrees
N.ION.1	Real	0.0	$cm^{-1}K^{-1}$
N.ION.2	Real	0.0	$cm^{-1}K^{-2}$
N.IONIZA	Real	7.03×10^{5}	cm^{-1}
N.SHIFT	Real	0.0	eV
N.XX	Real	1.0	
N.YY	Real	1.0	
N.ZZ	Real	1.0	
N0.BGN	Real	1.0×10^{17}	cm^{-3}
NAME	Character		
NC.F	Real	1.5	
NC300	Real	See Appendix B	cm^{-3}
NDX.ADACHI	Logical	False	
NDX.SELLMEIER	Logical	False	
NE.RICHTER	Real	3.3×10^{17}	cm^{-3}
NH.RICHTER	Real	7.0×10^{17}	cm^{-3}
NHNU	Integer	0	
NI.MIN	Real	0.0	cm^{-3}

Parameter	Type	Default	Units
NK.EV	Logical	False	
NK.NM	Logical	False	
NK.SHIFT	Real	0.0	eV
NLAM	Integer	0	
N.SCH.GAMMA	Real	1.0	s
N.SCH.MAX	Real	3.0×10^{-6}	s
N.SCH.MIN	Real	0.0	s
N.SCH.NREF	Real	1.0×10^{16}	cm^{-3}
NSRHN	Real	See Appendix B	cm^{-3}
NSRHP	Real	See Appendix B	cm^{-3}
NTC.GAUSS	Real	NTV.GAUSS/1.0e21	cm^{-3}
NTV.GAUSS	Real	NTC.GAUSS/1.0e21	cm^{-3}
NTRANSPARENT	Real	2.0×10^{18}	cm^{-3}
NUE.EXTR	Real		
NUH.EXTR	Real		
NV.F	Real	1.5	
NV300	Real	See Appendix B	cm^{-3}
OP.PH.EN	Real	0.063	eV
OPPHE	Real	0.063	eV
OPTAU.ORIENT	Real		
OPTAU.LUM	Real		um
OUT.DSPEC	Character		
OUT.INDEX	Character		
OUT.USPEC	Character		
OXCH.ONLY	Logical	False	
P1.KERR	Real	0.65	
P2.KERR	Real	0.65	
P3.KERR	Real	0.8	

Parameter	Type	Default	Units
P1.RICHTER	Real	0.66	
P2.RICHTER	Real	0.63	
P3.RICHTER	Real	0.92	
P.ION.1	Real	0.0	$cm^{-1}K^{-1}$
P.ION.2	Real	0.0	$cm^{-1}K^{-1}$
P.IONIZA	Real	1.682×10^{6}	cm^{-1}
P.PASSLER	Real	2.33	
P.SCH.GAMMA	Real	1.0	s
P.SCH.MAX	Real	3.0×10^{-6}	s
P.SCH.MIN	Real	0.0	s
P.SCH.NREF	Real	1.0×10^{16}	cm^{-3}
PAASCH.GFI	Integer	0	
PCM.AEA	Real	3.2	eV
PCM.AP	Real	1.0	
PCM.ARHO	Real		μΩcm
PCM.ATAU	Real		S
PCM.ATC	Real		K
PCM.CEA	Real	3.7	eV
PCM.CP	Real	1.0	
PCM.CRHO	Real		μΩcm
PCM.CTAU	Real		S
PCM.CTC	Real		K
PCM.LATHEAT	Real	0.0	J
PDA.N	Real	0.2	
PDA.P	Real	0.2	
PDEXP.N	Real	-2.5	
PDEXP.P	Real	-2.5	
PERM.ANISO	Real	-999.0	
PERMEABILITY	Real	1.0	

Parameter	Type	Default	Units
PERMITTIVITY	Real	See Appendix B	
PHEFF.EXCITON	Real	1	
PHONON.ENERGY	Real	0.068	eV
PIP.ACC	Real	2.0	
PIP.ET	Real	1.0	eV
PIP.NT	Real	0.0	cm^2
PIP.OMEGA	Real	0.07	eV
POWER	Real	0.0	W
POWERHI	Real	0.0	W
POWERLO	Real	0.0	W
PROFILE.GAUSS	Character		
PSP	Real	0.0	cm^{-2}
QE.EXCITON	Real	0.0	
QR.EXCITON	Real	0.0	
R0.EXCITON	Real	0	nm
REAL.INDEX	Real	See Appendix B	
REGION		All regions	
RESISTIVITY	Real		μΩ·cm
RST.EXCITON	Real	0.25	
RST.TT.EXCITON	Real	RST.EXCITON	
S.BINDING	Real	0.1	eV
S0SELL	Real	See Appendix B	
S1SELL	Real	See Appendix B	
S2SELL	Real	See Appendix B	
SIG.LID	Real	1.0×10^{-8}	$cm^{-3}s^{-1}$
SIGC.GAUSS	Real	SIGV.GAUSS/0.1	eV
SIGSN.EXCITON	0	cm^{-2}	
SIGSP.EXCITON	0	cm^{-2}	
SIGV.GAUSS	Real	SIGC.GAUSS/0.1	eV

Parameter	Type	Default	Units
SINGLET	Logical	False	
SEMICONDUC	Logical	False	
SO.DELTA	Real	see Table 3-138	eV
SOPRA	Character		
SPECIES1.AF	Real	1.0e15	Hz
SPECIES1.EA	Real	0.25	eV
SPECIES1.HOP	Real	1.0e-6	cm
SPECIES2.AF	Real	1.0e15	Hz
SPECIES2.EA	Real	0.25	eV
SPECIES2.HOP	Real	1.0e-6	cm
SPECIES3.AF	Real	1.0e15	Hz
SPECIES3.EA	Real	0.25	eV
SPECIES3.HOP	Real	1.0e-6	cm
STABLE	Integer		
STATEID	Integer	0.0	
STRUCTURE	Character		
STOKES	Real	0.0	eV
T.PASSLER	Real	406.0	
TAA.CN	Real	1.0×10^{-12}	cm^3/s
TAA.CP	Real	1.0×10^{-12}	cm^3/s
TANI.CONST	Logical	False	
TANI.POWER	Logical	False	
TANI.POLYNOM	Logical	False	
TANI.RECIP	Logical	false	
TAUMOB.EL	Real	0.25×10^{-12}	s
TAUMOB.HO	Real	0.25×10^{-12}	s
TAUN0	Real	See Appendix B	s
TAUP0	Real	See Appendix B	s
TAUREL.EL	Real	0.4×10^{-12}	s

Parameter	Type	Default	Units
TAUREL.HO	Real	0.4×10^{-12}	s
TAUS.EXCITON	Real	1×10^{-9}	s
TAUT.EXCITON	Real	1×10^{-9}	s
TC.A	Real	See Appendix B	$\frac{cm\cdot K}{W}$
TC.B	Real	See Appendix B	$\frac{cm}{W}$
TC.C	Real	See Appendix B	$\frac{cm}{W\cdot K}$
TC.CONST	Real		W/cm·K
TC.NPOW	Real		
TC.D	Real		K
TC.E	Real		W/cm
TC.FULL.ANISO	Logical	False	
TCOEFF.N	Real	2.55	
TCOEFF.P	Real	2.55	
TCON.CONST	Logical		
TCON.POWER	Logical		
TCON.POLYNOM	Logical		
TCON.RECIPRO	Logical		
TE.MODES	Logical	False	
TLU.A	Real	0	eV
TLU.C	Real	0	eV
TLU.E0	Real	0	eV
TLU.EC	Real	0	eV
TLU.EG	Real	0	eV
TLU.EPS	Real	1.0	
TMUN	Real		
TMUP	Real		
TP.RAMPTIME	Real		s

Parameter	Type	Default	Units
TRE.T1	Real		s
TRE.T2	Real		s
TRE.T3	Real		s
TRE.W1	Real		eV
TRE.W2	Real		eV
TRE.W3	Real		eV
TRH.T1	Real		s
TRH.T2	Real		s
TRH.T3	Real		s
TRH.W1	Real		eV
TRH.W2	Real		eV
TRH.W3	Real		eV
TRIPLET	Logical	False	
U.DEFPOT	Real	10.5	eV
UBGN.B	Real	3.1×10^{12}	
UBGN.C	Real	-3.9×10^{-5}	
USER.DEFAULT	Character		
USER.GROUP	Character	SEMICONDUCTOR	
USER.SPECT	Character		
V0.BGN	Real	9.0×10^{-3}	eV
VAL.AN0	Real	4.3383	
VAL.AN1	Real	-2.42×10^{-12}	
VAL.AN2	Real	4.1233	
VAL.AP0	Real	2.376	
VAL.AP1	Real	0.01033	
VAL.AP2	Real	1.0	
VAL.BN0	Real	0.235	
VAL.BN1	Real	0.0	
VAL.CN0	Real	1.6831×10^{4}	

Parameter	Type	Default	Units
VAL.CN1	Real	4.3796	
VAL.CN2	Real	1.0	
VAL.CN3	Real	0.13005	
VAL.DN0	Real	1.233735×10^{6}	
VAL.DN1	Real	1.2039e3	
VAL.DN2	Real	0.56703	
VAL.BP0	Real	0.17714	
VAL.BP1	Real	-0.002178	
VAL.CP0	Real	0.0	
VAL.CP1	Real	0.00947	
VAL.CP2	Real	2.4924	
VAL.CP3	Real	0.0	
VAL.DP0	Real	1.4043×10^{6}	
VAL.DP1	Real	2.9744×10^{3}	
VAL.DP2	Real	1.4829	
VSAT	Real		cm/s
VSATN	Real		cm/s
VSATP	Real		cm/s
WELL.DELTA	Real	see Table 3-138	eV
WELL.EPS	Real	0	eV
WELL.GAMMA0	Real	2×10^{-3}	eV
WELL.TAUIN	Real	3.3×10^{-13}	s
WELL.TAUN	Real	1e-12	sec
WELL.TAUP	Real	1e-12	sec
X.X	Real	1.0	
X.Y	Real	0.0	
X.Z	Real	0.0	
XDIR.ANISO	Logical	False	
Y.X	Real	0.0	

Parameter	Type	Default	Units
Y.Y	Real	1.0	
Y.Z	Real	0.0	
YDIR.ANISO	Logical	False	
ZDIR.ANISO	Logical	True	
Z.SCHENK	Real	1.0	
Z.X	Real	0.0	
Z.Y	Real	0.0	
Z.Z	Real	1.0	
ZAMDMER.Z0	Real	7.0×10^{-7}	cm

Description

The **MATERIAL** statement is used set basic material parameters related to band structure and parameters for certain mobility, recombination or carrier statistics models. Parameters for temperature dependence are noted in a separate section below.

Localization of Material Parameters

DEVICE	Specifies which device the **MATERIAL** statement should apply to in MixedMode simulation. The synonym for this parameter is **STRUCTURE**.
MATERIAL	Specifies which material from the table in Appendix B "Material Systems" that the **MATERIAL** statement should apply. If a material is specified, then all regions defined as being composed of that material will be affected.

Note: You can specify the following logical parameters to indicate the material instead of assigning the MATERIAL parameter: SILICON, GAAS, POLYSILI, GERMANIU, SIC, SEMICOND, SIGE, ALGAAS, A-SILICO, DIAMOND, HGCDTE, INAS, INGAAS, INP, S.OXIDE, ZNSE, ZNTE, ALINAS, GAASP, INGAP and MINASP.

NAME	Specifies which region the **MATERIAL** statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.
REGION	Specifies the region number to which these parameters apply. If there is more than one semiconductor region, specification of different parameters for each region is allowed. If REGION is not specified, all regions in the structure are changed.
STRUCTURE	This is a synonym for **DEVICE**.

Band Structure Parameters

A.DEFPOT, **B.DEFPOT**, **C.DEFPOT**, **D.DEFPOT**, and **U.DEFPOT**	Specify the deformation potentials used in calculating the effects of strains on the bandgap of silicon (see Section 3.6.12 "Stress Effects on Bandgap in Si").
AFFINITY	Specifies the electron affinity.
ALIGN	Specifies the fraction of the bandgap difference that is applied to the conduction band edge, relative to the minimum bandgap material in the device. Note that specifying this parameter overrides any electron affinity specification. See Section 6.1.2 "Alignment" for information on setting the band alignment.
ARICHN	Specifies the effective Richardson constant for electrons.
ARICHP	Specifies the effective Richardson constant for holes.
CHI.EG.TDEP	This is a ratio that specifies what fraction of the change in bandgap due to temperature change ascribed to the electron affinity. CHI.EG.TDEP is not active when ALIGN is specified.
CRELMAX.N and **CRELMAX.P**	Specifies the maximum normalized electron and hole densities used to calculate g_3 for Gaussian band structures.
CRELMIN.N and **CRELMIN.P**	Specifies the minimum normalized electron and hole densities used to calculate g_3 for Gaussian band structures.
D.TUNNEL	Specifies the maximum tunneling distance for the universal Schottky tunneling model (see Section 3.5.2 "Schottky Contacts"). The alias for this parameter is **DIST.SBT**.
DEC.DEG	Overrides the default values of dE_c/dE_v given in Table 6-15.
DEC.C1	This is a conduction band shift for 1st pair of electron valleys.
DEC.C2	This is a conduction band shift for 2nd pair of electron valleys.
DEC.C3	This is a conduction band shift for 3rd pair of electron valleys.
DEC.D2	This is a conduction band shift for $\Delta 2$ electron valleys if DEC.C1 is not set.
DEC.D4	This is a conduction band shift for $\Delta 4$ electron valleys if DEC.C2 is not set.
DEV.ISO	This is a conduction band shift for isotropic electron band, when NUM.DIRECT=1.
DEV.HH	This is a valence band shift for heavy holes.
DEV.LH	This is a valence band shift for light holes.

DEV.ISO	This is a valence band shift for isotropic hole band, when NUM.BAND=1.
DEV.SO	This is a valence band shift for split-off holes.
DIST.SBT	This is an alias for **D.TUNNEL**.
EG300	Specifies energy gap at 300K (see Equation 3-38). All semiconductor materials in Atlas must have a defined EG300.
EG1300, **EG2300**, **EG12BOW**, **EG1ALPH**, **EG2ALPH**, **EG1BETA**, and **EG2BETA**	Specify parameters of the General Ternary bandgap model described in Equations 3-42 through 3-44.
EPS11, **EPS12**, **EPS13**, **EPS22**, **EPS23**, and **EPS33**	Specify the strain tensor used in the calculation of the strain effects on bandgap of silicon (see Section 3.6.12 “Stress Effects on Bandgap in Si”).
EPS.XX	XX component of dielectric permittivity tensor used by vector Helmholtz solver.
EPS.YY	YY component of dielectric permittivity tensor used by vector Helmholtz solver.
EPS.ZZ	ZZ component of dielectric permittivity tensor, used by vector Helmholtz solver.
F.BANDCOMP	Specifies the name of a file containing a C-Interpreter function that defines temperature and composition dependent band parameter models.
F.CBDOSFN	Specifies the C-Interpreter file for the Conduction band effective density of states as a function of Lattice/Electron temperature. This function is called cbdosfn.
F.TCAP	Specifies the name of a file containing a C-Interpreter function that defines the lattice thermal capacity as a function of the lattice temperature, position, doping and fraction composition.
F.TCOND	Specifies the name of a file containing a C-Interpreter function that defines the lattice thermal conductivity as a function of the lattice temperature, position, doping and fraction composition.
F.EPSILON	Specifies the name of a file containing a C-Interpreter function that defines temperature and composition dependent static dielectric constant models.
F.FERRO	Specifies the name of a file containing a C-Interpreter function that defines dielectric permittivity as a function of electric field and position (x,y). You need a license to use this parameter.

F.VBDOSFN	Specifies the C-Interpreter file for the Valence band effective density of states as a function of Lattice/Hole temperature. This function is called vbdosfn.
G3MAX.N **G3MAX.P**	Specifies the maximum value of g_3 for Gaussian band structures.
GDEC.GAUSS **G1DEC.GAUSS** **G2DEC.GAUSS** **GDEV.GAUSS** **G1DEV.GAUSS** **G2DEV.GAUSS**	Specifies the shift between the center of the host Gaussian and the guest Gaussian. The positive values move towards the center of the bandgap.
GNTC.GAUSS **G1NTC.GAUSS** **G2NTC.GAUSS** **GNTV.GAUSS** **G1NTV.GAUSS** **G2NTV.GAUSS**	Specifies the density of states in the guest Gaussian.
GSIGC.GAUSS **G1SIGC.GAUSS** **G2SIGC.GAUSS** **GSIGV.GAUSS** **G1SIGV.GAUSS** **G2SIGV.GAUSS**	Specifies the standard deviation of the guest Gaussian.
GAUSS.TABLE	If this flag is true Gaussian band structure calculations will directly calculate n(E) (rather than using an interpolation table). This is very slow and should be avoided.
M.DSN	Specifies the electron density of states effective mass. When specified, it is in Equation 3-31 to calculate electron density of states.
M.DSP	Specifies the hole density of states effective mass. When specified it is in Equation 3-32 to calculate hole density of states.
M.VTHN	Specifies the electron effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Equation 6-52).
M.VTHP	Specifies the hole effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Equation 6-53).
ME.SBT and **MH.SBT**	These are aliases for **ME.TUNNEL** and **MH.TUNNEL**.

ME.TUNNEL and **MH.TUNNEL**	Specify the electron and hole effective masses for tunneling used in the universal Schottky tunneling model (see Section 3.5.2 “Schottky Contacts”). The aliases for these parameters are ME.SBT and MH.SBT.
NC.F	Specifies the conduction band density of states temperature (see Equation 3-31).
NV.F	Specifies the valence band density of states temperature (see Equation 3-32).
NV300	Specifies the conduction band density at 300K. (see Equation 3-31).
NI.MIN	Specifies the minimum allowable value of the intrinsic carrier density.
NTC.GAUSS and **NTV.GAUSS**	Specifies the total density of states of a conduction and valence band Gaussian DOS.
NV300	Specifies valence band density at 300K (see Equation 3-32).
PAASCH.GFI	Specifies an analytical model for the Gauss-Fermi integral [230] should be used. PAASCH.GFI=1 uses the analytic expressions, and PAASCH.GFI=2 uses the polynomial fit.
PERMITTIVITY	Specifies relative dielectric permittivity of the material. All materials in an Atlas structure must have a defined permittivity.
PROFILE.GAUSS	Specifies the file name to store the DOS/carrier density profile of the Gaussian band structure (this file is generated on a **SOLVE** INIT).
SIGC.GAUSS and **SIGC.GAUSS**	Specifies the standard deviation of a conduction and valence band Gaussian DOS.

Boltzmann Transport Equation Solver

BTE.AC_PHONON	Enables acoustic phonon scattering in Boltzmann equation solver.
BTE.CB.ACOUST	The acoustic phonon coupling energy in conduction band.
BTE.CB.II_EG	The effective bandgap for the electron impact ionization scattering model.
BTE.CB.II_SCALE	Overall scale factor for electron impact ionization scattering model.
BTE.CB.INVSL	The inverse scattering length for the electron impact ionization scattering model.
BTE.CB.NONPAR	The conduction band non-parabolicity factor for E(k).
BTE.CB.OPCC	The optical phonon coupling coefficient in conduction band.

BTE.CB.OPHW	The energy of optical phonons in conduction band.
BTE.CB.ZION	Fitting parameter for conduction band ionized impurity scattering.
BTE.DENSITY	The mass density for the materials.
BTE.GEMODEL	The band structure model selector in the Boltzmann equation solver.
BTE.IMPACT	Enables impact ionization scattering in Boltzmann equation solver.
BTE.IONIZED	Enables ionized impurity scattering in Boltzmann equation solver.
BTE.OP_PHONON	Enables optical phonon scattering in Boltzmann equation solver.
BTE.SOUND	The speed of longitudinal acoustic waves.
BTE.VB.ACOUST	The acoustic phonon coupling energy in valence band.
BTE.VB.II_EG	The effective bandgap for the hole impact ionization scattering model.
BTE.VB.II_SCALE	Overall scale factor for hole impact ionisation scattering model.
BTE.VB.INVSL	The inverse scattering length for the hole impact ionization scattering model.
BTE.VB.NONPAR	The valence band non-parabolicity factor for E(k).
BTE.VB.OPCC	The optical phonon coupling coefficient in valence band.
BTE.VB.OPHW	The energy of optical phonons in valence band.
BTE.VB.ZION	Fitting parameter for valence band ionized impurity scattering.
ELEC.MP.POWER	The power law exponent for Keldysh model for multiple electron degradation processes.
ELEC.MP.SIGMA	The cross-section for Keldysh model for multiple electron degradation processes.
ELEC.MP.THRESH	The energy threshold for Keldysh model for multiple electron degradation processes.
ELEC.SP.POWER	The power law exponent for Keldysh model for single electron degradation processes.
ELEC.MP.SIGMA	The cross-section for Keldysh model for single electron degradation processes.
ELEC.SP.THRESH	The energy threshold for Keldysh model for single electron degradation processes.
HOLE.MP.POWER	The power law exponent for Keldysh model for multiple hole degradation processes.
HOLE.MP.SIGMA	The cross-section for Keldysh model for multiple hole degradation processes.

HOLE.MP.THRESH	The energy threshold for Keldysh model for multiple hole degradation processes.
HOLE.SP.POWER	The power law exponent for Keldysh model for single hole degradation processes.
HOLE.SP.SIGMA	The cross-section for Keldysh model for single hole degradation processes.
HOLE.SP.THRESH	The energy threshold for Keldysh model for single hole degradation processes.

BQP Parameters

BQP.NGAMMA and **BQP.PGAMMA**	These parameters allow you to set the γ parameter of the BQP model for electrons and holes respectively.
BQP.NALPHA and **BQP.PALPHA**	These parameters allow you to set the α parameter of the BQP model for electrons and holes respectively.

Mobility Model Parameters

F.CONMUN	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition and doping dependent electron mobility models.
F.CONMUP	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition and doping dependent hole mobility models.
F.ENMUN	Specifies a C-Interpreter file for the electron mobility as a function of Electron Temperature and perpendicular field as well as other choice variables. The function itself is named endepmun and only applies to Atlas 2D.
F.ENMUP	Specifies a C-Interpreter file for the hole mobility as a function of Hole Temperature and perpendicular field as well as other choice variables. The function itself is named endepmup and only applies to Atlas 2D.
F.MUNSAT	Specifies the name of a file containing a C-Interpreter function for the specification of parallel field dependent electron mobility model for velocity saturation.
F.MUPSAT	Specifies the name of a file containing a C-Interpreter function for the specification of parallel field dependent hole mobility model for velocity saturation.
F.VSATN	Specifies the name of a file containing a C-Interpreter function for the specification of temperature and composition dependent electron saturation velocity models.
F.VSATP	Specifies the name of a file containing a C-Interpreter function for the specification of temperature and composition dependent hole saturation velocity models.

G.SURF	Specifies a factor by which mobility is reduced at the semiconductor surface. This is a simple but not accurate alternative to the transverse field dependent or surface mobility models set on the **MODELS** statement.
MUN	Specifies low-field electron mobility. This parameter is only used if no concentration dependent mobility model is specified.
MUP	Specifies low-field hole mobility.This parameter is only used if no concentration dependent mobility model is specified.
VSAT	Specifies the saturation velocity for the electric field dependent mobility.
VSATN	Specifies the saturation velocity for electrons.
VSATP	Specifies the saturation velocity for holes.

Recombination Model Parameters

ALPHA.2D.LANGEVIN	Specifies the prefactor for the Juska 2D Langevin recombination model. See Equations 16-64 and 16-65.
A.LANGEVIN	Specifies the prefactor for the bimolecular Langevin recombination model. See Equation 16-62.
A.TRAPCOULOMBIC **B.TRAPCOULOMBIC**	Specify the Poole-Frenkel thermal emission enhancement factor parameters. See Equations 3-109 and 3-110.
AN, **AP**, **BN**, **BP**, **CN**, **EN**, and **EP**	The parameter for the concentration dependent lifetime model (see Equations 3-338 and 3-351).
AUG.CNL **AUG.CPL** **AUG.CHI**	Specify the coefficients for the carrier and doping concentration dependent Auger rate coefficient models described in Equations 3-383 and 3-404.
AUGN	Specifies the Auger coefficient, *cn* (see Equation 3-386).
AUGP	Specifies the Auger coefficient, *cp* (see Equation 3-386).
AUGKN	The parameter of the narrow band-gap electron Auger recombination coefficient model.
AUGKP	The parameter of the narrow band-gap electron Auger recombination coefficient model.
BB.A, **BB.B**, and **BB.GAMMA**	Specify the band-to-band tunneling parameters (see Equation 3-480).
BBT.ALPHA	Specifies the scaling factor in the Hurkx band-to-band tunneling model (see Equation 3-488).
C1.KERR	Coefficient in Kerr auger recombination model.
C2.KERR	Coefficient in Kerr auger recombination model.
C3.KERR	Coefficient in Kerr auger recombination model.

C1.RICHTER	Coefficient in Richter auger recombination model.
C2.RICHTER	Coefficient in Richter auger recombination model.
C3.RICHTER	Coefficient in Richter auger recombination model.
C.DIRECT	This is an alias for **COPT**.

See Equations 3-348 through 3-363 to see how the following CDL parameters are used.

CDL.ETR1 and **CDL.ETR2**	There are the trap energy levels in the Coupled Defect Level model. They are relative to the intrinsic energy level and are in Electron Volts.
CDL.TN1, **CDL.TN2**, **CDL.TP1**, and **CDL.TP2**	These are the recombination lifetimes for the two energy levels in the Coupled Defect Level model.
CDL.COUPLING	Specifies the coupling constant in units of $cm^{-3}s^{-1}$ for the Coupled Defect Level model.
COPT	Specifies the optical recombination rate for the material. This parameter has no meaning unless MODELS OPTR has been specified (see Equation 3-382). The alias for this parameter is **C.DIRECT**.
DAA.LANGEVIN	Specifies the prefactor for the p-acceptor Langevin recombination model. See equation 16-113
DAD.LANGEVIN	Specifies the prefactor for the n-donor Langevin recombination model. See equation 16-113
EMIT.FILE	Specifies the file name for PL data.
EMIT.RATE	Specifies the spontaneous emission rate.
ETRAP	Specifies the trap energy for SRH recombination
F.BBT	Specifies the name of a file containing a C-Interpreter function for the specification of the electric field and carrier concentration dependent band-to-band generation rate.
F.COPT	Specifies the name of a file containing a C-Interpreter function for the specification of composition and temperature dependence of the radiative recombination rate.
F.GAUN	Specifies the name of a file containing a C-Interpreter function for the specification of composition and temperature dependence of the electron Auger coefficient.
F.GAUP	Specifies the name of a file containing a C-Interpreter function for the specification of composition and temperature dependence of the hole Auger coefficient.

F.RECOMB	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition, electron and hole concentration dependent recombination rate models.
F.SRH	Specifies the name of a file containing a C-Interpreter function, which is suitable for the specification of an SRH-like recombination rate. Available parameters are electron and hole concentrations, acceptor and donor doping concentrations, intrinsic carrier concentration, and lattice temperature.
F.TAUN	Specifies the name of a file containing a C-Interpreter function for the specification of position dependent electron lifetime models.
F.TAUP	Specifies the name of a file containing a C-Interpreter function for the specification of position dependent hole lifetime models.
F.TAURN	Specifies the name of a file containing a C-Interpreter function specifying the electron relaxation time as a function of electron energy.
F.TAURP	Specifies the name of a file containing a C-Interpreter function specifying the hole relaxation time as a function of hole energy.
G.2D.LANGEVIN	Specifies the prefactor for the Juska 2D Langevin recombination model. See Equations 16-64 and 16-65.
HNS.AE, **HNS.BE**, **HNS.CE**, **HNS.AH**,**HNS.BH**, and **HNS.CH**	These are the temperature dependence parameters for HNSAUG model (see Equations 3-407 and 3-408).
HNS.HE and **HNS.N0E**	These are the Electron concentration dependence parameters for HNSAUG model (see Equations 3-407 and 3-408).
HNS.HH and **HNS.N0H**	These are the Hole concentration dependence parameters for HNSAUG model (see Equations 3-407 and 3-408).
INCOHERENT	Specifies whether the layer region is incoherent.
JTAT.M2	Matrix element squared for TRAP.JTAT model.
L.2D.LANGEVIN	Specifies the prefactor for the Juska 2D Langevin recombination model. See Equation 16-67.
N.SCH.MIN **P.SCH.MIN** **N.SCH.MAX** **P.SCH.MAX** **N.SCH.GAMMA** **P.SCH.GAMMA** **N.SCH.NREF** **P.SCH.NREF**	These are used in the Scharfetter doping concentration dependent lifetime model. They are applied in Equations 3-342 and 3-355.

TCOEFF.N and **TCOEFF.P**	These are used in the alternative lifetime lattice temperature dependence model.
TAA.CN and **TAA.CP**	These are used in the carrier concentration dependent recombination lifetime model of Equations 3-352 and 3-365.
ZAMDMER.Z0	The nominal spatial separation of recombination centers in the ZAMDMER model of electron trap lifetime in LT-GaAs.

Impact Ionization Parameters

AN0.VALD	This is an alias for **VAL.AN0**.
AN1.VALD	This is an alias for **VAL.AN1**.
AN2.VALD	This is an alias for **VAL.AN2**.
AP0.VALD	This is an alias for **VAL.AP0**.
AP1.VALD	This is an alias for **VAL.AP1**.
AP2.VALD	This is an alias for **VAL.AP2**.
BETAN	This is for electrons and BETAP for holes correspond to coefficients for the power of ECRIT/E. The aliases for these parameters are **EXN.II** and **EXP.II**.
BN0.VALD	This is an alias for **VAL.BN0**.
BN1.VALD	This is an alias for **VAL.BN1**.
BP0.VALD	This is an alias for **VAL.BP0**.
BP1.VALD	This is an alias for **VAL.BP1**.
CN2, **CP2**, **DN2**, and **DP2**	These are specifiable coefficients in the temperature dependent models described in Equations 3-402 and 3-423. The aliases for these parameter are **N.ION.1**, **P.ION.1**, **N.ION.2**, and **P.ION.2**.
CN0.VALD	This is an alias for **VAL.CN0**.
CN1.VALD	This is an alias for **VAL.CN1**.
CN2.VALD	This is an alias for **VAL.CN2**.
CN3.VALD	This is an alias for **VAL.CN3**.
CP0.VALD	This is an alias for **VAL.CP0**.
CP1.VALD	This is an alias for **VAL.CP1**.
CP2.VALD	This is an alias for **VAL.CP2**.
CP3.VALD	This is an alias for **VAL.CP3**.
DN0.VALD	This is an alias for **VAL.DN0**.
DN1.VALD	This is an alias for **VAL.DN1**.

DN2.VALD	This is an alias for **VAL.DN2**.
DP0.VALD	This is an alias for **VAL.DP0**.
DP1.VALD	This is an alias for **VAL.DP1**.
DP2.VALD	This is an alias for **VAL.DP2**.
ECN.II	This is an alias for BN2.
ECP.II	This is an alias for BP2.
EXN.II	This is an alias for **BETAN**.
EXP.II	This is an alias for BETAP.
LAMDAE	Specifies the mean free path for electrons. The alias for this parameter is **LAN300**.
LAMDAH	Specifies the mean free path for holes. The alias for this parameter is **LAP300**.
LAN300	This is an alias for **LAMDAE**.
LAP300	This is an alias for **LAMDAH**.
NE.RICHTER	Density in Richter auger recombination model.
NH.RICHTER	Density in Richter auger recombination model.
N.ION.1 **P.ION.1** **N.ION.2** **P.ION.2**	These are the aliases for **CN2**, **CP2**, **DN2** and **DP2**.
N.IONIZA	This is an alias for AN2.
OP.PH.EN	This is an alias for **OPPHE**.
OPPHE	Specifies the optical phonon energy. The alias for this parameter is **OP.PH.EN**.
P1.KERR	Exponent in Kerr auger recombination model.
P2.KERR	Exponent in Kerr auger recombination model.
P3.KERR	Exponent in Kerr auger recombination model.
P1.RICHTER	Exponent in Richter auger recombination model.
P2.RICHTER	Exponent in Richter auger recombination model.
P3.RICHTER	Exponent in Richter auger recombination model.
P.IONIZA	This is an alias for AP2.
VAL.AN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN0.VALD**.

VAL.AN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN1.VALD**.
VAL.AN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-433. The alias for this parameter is **AN2.VALD**.
VAL.BN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-434. The alias for this parameter is **BN0.VALD**.
VAL.BN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-434. The alias for this parameter is **BN1.VALD**.
VAL.CN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN0.VALD**.
VAL.CN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN1.VALD**.
VAL.CN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN2.VALD**.
VAL.CN3	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-435. The alias for this parameter is **CN3.VALD**.
VAL.DN0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN0.VALD**.
VAL.DN1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN1.VALD**.
VAL.DN2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-436. The alias for this parameter is **DN2.VALD**.
VAL.AP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437.
VAL.AP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437. The alias for this parameter is **AP1.VALD**.
VAL.AP2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-437. The alias for this parameter is **AP2.VALD**.
VAL.BP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-438. The alias for this parameter is **BP0.VALD**.
VAL.BP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-438. The alias for this parameter is **BP1.VALD**.
VAL.CP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP0.VALD**.
VAL.CP1	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP1.VALD**.
VAL.CP2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP2.VALD**.

VAL.CP3	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-439. The alias for this parameter is **CP3.VALD**.
VAL.DP0	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP0.VALD**.
VAL.DP1	specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP1.VALD**.
VAL.DP2	Specifies the value of a temperature dependent impact ionization parameter in Equation 3-440. The alias for this parameter is **DP2.VALD**.

Klaassen Model Parameters

KSRHTN	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KSRHTP	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KSRHCN	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KSRHCP	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KSRHGN	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KSRHGP	Coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.
KAUGCN	Coefficient for Klaassen's concentration dependent Auger model.
KAUGCP	Coefficient for Klaassen's concentration dependent Auger model.
KAUGDN	Coefficient for Klaassen's concentration dependent Auger model.
KAUGDP	Coefficient for Klaassen's concentration dependent Auger model.
NSRHN	Specifies the SRH concentration parameter for electrons (see Equation 3-350).
NSRHP	Specifies the SRH concentration parameter for holes (see Equation 3-351).
TAUN0	Specifies SRH lifetime for electrons (see Equation 3-350).
TAUP0	Specifies SRH lifetime for holes (see Equation 3-351).

See also Equations 3-340 and 3-353 and Equations 3-375 and 3-389 for more information about Klaassen models.

Carrier Statistics Model Parameters

ASYMMETRY	Specifies the relative degree where band gap narrowing applies to the conduction band versus the valence band. The value of the ASYMMETRY parameter is multiplied by the total change in band gap due to band gap narrowing and that product is applied to the conduction band edge. For example, if the ASYMMETRY parameter has a value of 1.0, then the change in band gap due to band gap narrowing is applied only to the conduction band edge and the valence band edge remains unaffected.
BGN.E **BGN.N** **BGN.C**	Specify the parameters of the band gap narrowing model given in Equation 3-46. The aliases for these parameters are **V0.BGN**, **N0.BGN**, and **CON.BGN**. BGN.E and BGN.N are also used in the Bennett-Wilson and del Alamo Bandgap narrowing models (see Equations 3-51 and 3-52).
BGN.E.ACC **BGN.E.DON** **BGN.N.ACC** **BGN.N.ACC**	Parameters for the BGN and BGN2 models when used in 4H-SiC or 6H-SiC.
BGN.LIND.ANC **BGN.LIND.ANV** **BGN.LIND.APC** **BGN.LIND.ANV** **BGN.LIND.BNC** **BGN.LIND.BNV** **BGN.LIND.BPC** **BGN.LIND.BPV**	Parameters for the Lindefelt Bandgap narrowing model described in Section 3.2.13 “Lindefelt Bandgap Narrowing Model”.
BGN.SHNK.EPS	Relative dielectric permittivity for Schenk bandgap narrowing model.
BGN.SHNK.GE	Electron degeneracy factor for Schenk bandgap narrowing model.
BGN.SHNK.GH	Hole degeneracy factor for Schenk bandgap narrowing model.
BGN.SHNK.ME	Electron effective mass for Schenk bandgap narrowing model.
BGN.SHNK.MH	Hole effective mass for Schenk bandgap narrowing model.
CON.BGN	This is an alias for **BGN.C**.
EA.CUBIC	Acceptor ionization energy in SiC for cubic symmetry site.
EA.HEXAGONAL	Acceptor ionization energy in SiC for hexagonal symmetry site.
ED.CUBIC	Donor ionization energy in SiC for cubic symmetry site.
ED.HEXAGONAL	Donor ionization energy in SiC for hexagonal symetry site.
EAB	Specifies acceptor energy level (see Equation 3-74).
EDB	Specifies donor energy level (see Equation 3-73).
F.BGN	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition and doping dependent bandgap narrowing models.

GCB	Specifies the conduction-band degeneracy factor (see Equation 3-73).
GVB	Specifies the valence-band degeneracy factor (see Equation 3-74).
N0.BGN	This is an alias for **BGN.N**.
V0.BGN	This is an alias for **BGN.E**.

Energy Balance Parameters

GONZALEZ.C0 **GONZALEZ.C1** **GONZALEZ.C2** **GONZALEZ.C3** **GONZALEZ.CEEBOW**	Parameters for the Gonzalez Energy relaxation lifetime model.
GONZALEZ.TAUN0 **GONZALEZ.TAUN1** **GONZALEZ.TAUP0** **GONZALEZ.TAUBOW**	Time parameters for the Gonzalez Energy relaxation lifetime model.
TAUMOB.EL	Specifies the relaxation time for electrons in the temperature dependent mobility model (see Equation 3-162).
TAUMOB.HO	Specifies the relaxation time for holes in the temperature dependent mobility model (see Equation 3-163).
TAUREL.EL	Specifies the relaxation time for electrons in the energy balance model (see Equation 3-151).
TAUREL.HO	Specifies the relaxation time for holes in the energy balance model (see Equation 3-152).
TRE.T1, **TRE.T2**, **TRE.T3**, **TRE.W1**, **TRE.W2**, **TRE.W3**, **TRH.T1**, **TRH.T2**, **TRH.T3**, **TRH.W1**, **TRH.W2**, and **TRH.W3**	These are used in the temperature dependent energy relaxation time model based on table data from Laux-Fischetti Monte-Carlo simulation (see Table 3-21).

Hot Carrier Injection Parameters

IG.ELINR	Specifies the electron mean free path between redirecting collisions. The alias for this parameter is **LAMRN**.
IG.HLINR	Specifies the hole mean free path between redirecting collisions. The alias for this parameter is **LAMRP**.
IG.ELINF	Specifies the electron mean free path length for scattering by optical phonons. The alias for this parameter is **LAMHN**.
IG.HLINF	Specifies the hole mean free path length for scattering by optical phonons. The alias for this parameter is **LAMHP**.
LAMRN	This is an alias for **IG.ELINR**.

LAMRP	This is an alias for **IG.HLINR**.
LAMHN	This is an alias for **IG.ELINF**.
LAMHP	This is an alias for **IG.HLINF**.

Lattice Temperature Dependence Parameters

A.PASSLER	Specifies a parameter of the Passler temperature dependent bandgap model given in Equation 3-39.
EGALPHA	Specifies the alpha coefficient for temperature dependence of bandgap (see Equation 3-38).
EGBETA	Specifies the beta coefficient for temperature dependence of bandgap (see Equation 3-38).
F.PDN	Uses C-Interpreter function for phonon drag contribution to the electron thermopower.
F.PDP	Uses C-Interpreter function for phonon drag contribution to the hole thermopower.
HC.A, **HC.B**, **HC.C**, and **HC.D**	Specify the values of the four coefficient of the heat capacity equation (see Equation 8-6).
HC.BETA, **HC.C1**, **HC.C300**, and **HC.RHO**	Specify user overrides for the parameters of Equation 8-7.
LT.TAUN	Specifies the temperature dependence for electron lifetimes (see Equation 8-32).
LT.TAUP	Specifies the temperature dependence for hole lifetimes (see Equation 8-33).
P.PASSLER	Specifies a parameter of the Passler temperature dependent bandgap model given in Equation 3-39.
PCM.AEA	Specifies the activation energy for amorphization of a phase change material.
PCM.AP	Specifies the time exponent for amorphous change of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.ARHO	Specifies the amorphous resistivity of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.ATC	Specifies the critical temperature for amorphous change of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.CEA	Specifies the activation energy for crystallization of a phase change material.

PCM.CP	Specifies the time exponent for crystallization of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.CRHO	Specifies the crystalline resistivity of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.CTC	Specifies the critical temperature for crystallization of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.ATAU	Specifies the time constant for amorphous change of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.CTAU	Specifies the time constant for crystallization of a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PCM.LATHEAT	Specifies the latent of change from crystalline to amorphous state in a phase change material (see Section 8.2.9 “Phase Change Materials (PCMs)”).
PDA.N	Specifies the value of a parameter in the model of phonon drag contribution to the electron thermopower (see Equation 8-13).
PDA.P	Specifies the value of a parameter in the model of phonon drag contribution to the hole thermopower (see Equation 8-14).
PDEXP.N	Specifies the value of a parameter in the model of phonon drag contribution to the electron thermopower (see Equation 8-13).
PDEXP.P	Specifies the value of a parameter in the model of phonon drag contribution to the hole thermopower (see Equation 8-14).
POWER	Specifies the value of thermal power generated in a power source associated with a region in Thermal3D (see Chapter 18 “Thermal 3D: Thermal Packaging Simulator”).
POWERHI	Specifies a maximum power level associated with a region in the transient mode of Thermal3D (see Chapter 18 “Thermal 3D: Thermal Packaging Simulator”).
POWERLO	Specifies a minimum power level associated with a region in the transient mode of Thermal3D (see Chapter 18 “Thermal 3D: Thermal Packaging Simulator”).
T.PASSLER	Specifies a parameter of the Passler temperature dependent bandgap model given in Equation 3-39.
TC.A, **TC.B**, and **TC.C**	Specify the three thermal conductivity coefficients (see Table 8-1).
TC.CONST	Specifies the equilibrium value of thermal conductivity, $k(T0)$, in Table 8-1. The synonym for this parameter is TC.CONST. TC.0 is an alias for TC.CONST.
TC.D	Specifies the value of the parameter D in Table 8-1.

TC.E	Specifies the value of the parameter *E* in Table 8-1.
TCON.CONST	Specifies that thermal conductivity should be modeled as constant with respect to temperature. The value of the thermal conductivity is given by the value of the TC.C0 parameter.
TCON.POWER	Specifies that the temperature dependence of thermal conductivity should be modeled using Table 8-1.
TCON.POLYNOM	Specifies that the temperature dependence of thermal conductivity should be modeled using Table 8-1.
TCON.RECIPRO	Specifies that the temperature dependence of thermal conductivity should be modeled using Table 8-1.
TC.NPOW	Specifies the value of the coefficient of temperature dependence of thermal conductivity, *n*, in Table 8-1.
TMUN and **TMUP**	Specify the lattice temperature coefficients for the temperature dependence of electron lattice mobility, and of hole lattice mobility respectively.
TP.RAMPTIME	Specifies a transition time for the power level associated with a region in the transient mode of Thermal3D (see Chapter 18 "Thermal 3D: Thermal Packaging Simulator").
UBGN.B and **UBGN.C**	Define the numerical values of the fitting parameters for Equation 3-50.

Oxide Material Parameters

INSULATOR	Specifies that a semiconductor region is to be treated as an insulator.
OXCH.ONLY	Specifies that electron and hole concentrations are omitted from Poisson's Equation in oxides.
SEMICONDUC	Specifies that an oxide region is to be treated as a semiconductor.

Photogeneration Parameters

A1.RAJ, **A2.RAJ**, and **AD.RAJ**	Specify parameters for Equation 3-612.
AADACHI, **BADACHI**, and **DADACHI**	These are the parameters for Adachi's refractive index model. See Equation 3-689.
BETA.RAJ **GAMMA.RAJ**	Specify parameters in Equations 3-630 through 3-615.
C1.RAJ and **C2.RAJ**	Specify parameters of Equation 3-612.
E.CONDUC	Specifies the electric conductivity in all directions in mho/cm.
EG1.RAJ, **EG2.RAJ**, and **EG3.RAJ**	Specify parameters of Equations 3-630 through 3-615.

F.INDEX	Specifies the name of a file containing a C-Interpreter function for the specification of wavelength dependent complex index of refraction models.
J.ELECT	Specifies the electric current density in all directions in A/cm^2.
J.MAGNET	Specifies the equivalent magnetic current density in all directions in V/cm^2.
HNU1 and **HNU2**	These are used to specify the range of energies over which to plot the complex index of refraction. You also should specify NHNU and OUT.INDEX.
IMAG.INDEX	Specifies the imaginary portion of the refractive index of the semiconductor (see Equation 11-12). Wavelength dependent defaults exist for certain materials as documented in Appendix B "Material Systems".
INDEX.FILE	Specifies the filename of an ASCII file from which complex refractive indices for a material are read. See Section 11.8.2 "Setting A Wavelength Dependent Refractive Index" for more information.
INDX.IMAG	Specifies the name of a file analogous to **INDEX.FILE** described above with only two columns wavelength and imaginary index. See Section 11.8.2 "Setting A Wavelength Dependent Refractive Index" for more information.
INDX.REAL	Specifies the name of a file analogous to **INDEX.FILE** described above with only two columns wavelength and real index. See Section 11.8.2 "Setting A Wavelength Dependent Refractive Index" for more information.
LAM1 and **LAM2**	These are used to specify the range of wavelengths over which to plot the complex index of refraction. You also should specify NLAM and OUT.INDEX.
MAG.LOSS	Specifies the equivalent magnetic loss in all directions in Ω/cm.
NDX.ADACHI	Enables Adachi's refractive index model. See Equation 3-689.
N.ANISO	Specifies that the region will simulated in FDTD with an anisotropic index of refraction described by the parameters N.XX, N.YY, N.ZZ, N.ALPHA, N.BETA, and N.GAMMA.
N.SHIFT	Specifies the energy shift applied to the input real index spectrum in electron volts.
N.XX, **N.YY**, and **N.ZZ**	Specify the anisotropic indices of refraction along the principal crystalline axes.
N.ALPHA, **N.BETA**, and **N.GAMMA**	Specify the Euler angles describing the relative orientation between the crystal and the optical axis for anisotropic index of refraction in FDTD.
NDX.SELLMEIER	Enables Sellmeier's refractive index model. See Equation 3-688.
NHNU	Specifies the number of energy samples between HNU1 and HNU2 to plotting the complex index of refraction.
NK.EV	Specifies that wavelengths included in index files specified by INDEX.FILE, INDX.IMAG, or INDX.REAL should be interpreted as energies in units of electron volts.

NK.NM	Specifies that wavelengths included in index files specified by INDEX.FILE, INDX.IMAG, or INDX.REAL should be interpreted as having units of nanometers.
NK.SHIFT	Specifies the energy shift applied to the complex index of refraction in electron volts.
NLAM	Specifies the number of wavelength samples between LAM1 and LAM2 to plotting the complex index of refraction.
OUT.INDEX	Specifies the root file name for output of wavelength dependent samples of complex index of refraction suitable for display in TonyPlot (Section 11.8 "Defining Optical Properties of Materials").
PERMEABILITY	Specifies the relative permeability.
S0SELL, **S1SELL**, **S2SELL**, **L1SELL**, and **L2SELL**	These are parameters for Sellmeier's refractive index model. See Equation 3-688.
TLU.A, **TLU.C**, **TLU.E0**, **TLU.EC**, **TLU.EG**, and **TLU.EPS**	These are used to specify the parameters of the Tauc-Lorentz dielectric function model for complex index of refraction described in Section 3.10.3 "Tauc-Lorentz Dielectric Function with Optional Urbach Tail Model for Complex Index of Refraction"

Note: The index file must be ordered by increasing wavelength .

REAL.INDEX	Specifies the real portion of the refractive index of the semiconductor. Wavelength dependent defaults exist for certain materials as documented in Appendix B "Material Systems".

LASER Parameters

ABSORPTION.SAT	Specifies the absorption saturation intensity used in the non-linear absorption loss model given in Equation 9-39.
ALPHAA	Specifies the bulk absorption coefficient in Equation 9-17.
ALPHAR	Specifies the line width broadening factor in Equation 9-18.
EMISSION.FACTOR	Specifies a scale factor accounting the proportion of light directionally coupled into the lasing mode.
EPSINF	Specifies the high frequency relative dielectric permittivity ($\varepsilon\infty$) (see Equation 9-17). If this parameter is not specified, it will be set equal to the static dielectric permittivity of the material.
F.ALPHAA	Specifies the name of a file containing a C-Interpreter function for the bulk absorption coefficient.
FC.RN	This is the free-carrier loss model parameter in Equation 9-19.
FC.RP	This is the free-carrier loss model parameter in Equation .

GAIN.SAT	Specifies the non-linear gain saturation factor used in Equation 9-38.
GAIN0	Specifies the parameter in Sections 3.9.2 "The Default Radiative Recombination Model" and 3.9.3 "The Standard Gain Model".
GAIN00	Specifies the parameter in Section 3.9.4 "The Empirical Gain Model".
GAIN1N	Specifies the parameter in Section 3.9.4 "The Empirical Gain Model".
GAIN1P	Specifies the parameter in Section 3.9.4 "The Empirical Gain Model".
GAIN1MIN	Specifies the parameter in Section 3.9.4 "The Empirical Gain Model".
GAIN2NP	Specifies the parameter in Section 3.9.4 "The Empirical Gain Model".
GAMMA	Specifies the parameter in Sections 3.9.2 "The Default Radiative Recombination Model" and 3.9.3 "The Standard Gain Model". If this parameter is not specified, it will be calculated using Equation 3-622.
GN1	Specifies the parameter in Section 3.9.5 "Takayama's Gain Model".
GN2	Specifies the parameter in Section 3.9.5 "Takayama's Gain Model".
NTRANSPARENT	Specifies the parameter in Section 3.9.5 "Takayama's Gain Model".

NOISE Parameters

F.MNSNDIFF	Specifies the name of a file containing a C-Interpreter function for the microscopic noise source for electron diffusion noise.
F.MNSPDIFF	Specifies the name of a file containing a C-Interpreter function for the microscopic noise source for hole diffusion noise.
F.MNSNFLICKER	Specifies the name of a file containing a C-Interpreter function for the microscopic noise source for electron flicker noise.
F.MNSPFLICKER	Specifies the name of a file containing a C-Interpreter function for the microscopic noise source for hole flicker noise.
HOOGEN	Specifies the Hooge constant for electron flicker noise.
HOOGEP	Specifies the Hooge constant for hole flicker noise.

Organic Transport Parameters

HOPN.BETA	Specifies the percolation constant for electrons.
HOPN.GAMMA	Specifies 1/carrier localization radius for electrons.
HOPN.V0	Specifies the attempt-to-jump frequency for electrons.
HOPP.BETA	Specifies the percolation constant for holes.
HOPP.GAMMA	Specifies 1/carrier localization radius for holes.
HOPP.V0	Specifies the attempt-to-jump frequency for holes.

Exciton Material Parameters

A.SINGLET	Specifies the singlet electron hole separation distance used in Equation 16-63.
ADDPOLARON	Adds a new exciton-polaron to the material, which is associated with the name specified as **MIX.NAME** parameter on the same **MATERIAL** statement
ADDSTATE	Adds a state to the material.
ARADIUS.EXCITON	Specifies the radius of the host molecule.
DKCQ.EXCITON	Specifies the dopant concentration quenching rate.
DKNRS.EXCITON	Specifies the dopant non-radiative decay rate.
DKSS.EXCITON	Specifies the dopant bimolecular annihilation constant.
DKTP.EXCITON	Specifies the dopant triplet-polaron constant.
DLDS.EXCITON	Specifies the dopant singlet diffusion length.
DOPANT	Flags the added polaron as a dopant. This is necessary for proper assignment of Forster transfer between intrinsic and dopant materials
DOPE.SPECT	Specifies the file name of a second user defined spectrum file. See also **USER.SPECT**.
DPHEFF.EXCITON	Specifies the dopant photoluminescence quantum efficiency.
DRST.EXCITON	Specifies the fraction of dopant singlets formed.
DTAUS.EXCITON	Specifies the dopant singlet radiative decay lifetime.
DTAUT.EXCITON	Specifies the dopant triplet radiative decay lifetime.
EX.AMPLITUDE	Specifies the amplitude of the lineshape for the level at **EX.ECEN** relative to the exciton-polaron lineshape
EX.COUPLING	This is the coupling between excitons and phonons in eV.
EX.ECEN	Adds a level in the excited state manifold .
EX.EPHONON	This is the energy of a single phonon in eV.
EX.HOPPING	The nearest neighbor hopping matrix element for excitons in tight binding Holstein Hamiltonian.
EX.HOMOBROADENING	Homogeneous broadening of the excited state.
EX.INHOMOBROADENING	Inhomogeneous broadening of the excited state.
EX.MAXCLOUDSIZE	This is the maximum size of phonon clouds given as the number of sites from the exciton position to the edge of the cloud
EX.MAXPHONONS	This is the maximum total number of phonons in the cloud.
EXCITON.DIPOLE1	The first of the two components of the dipole vector of the Frenkel exciton.
EXCITON.DIPOLE2	The second of the two components of the dipole vector of theFrenkel exciton.

EXCITON.GAP	This is the band gap energy without the phonon renormalization.
F.DSANNIHILATION	Specifies the file name of a C-Interpreter function that models the annihilation of dopant singlet excitons.
GS.AMPLITUDE	Specifies the amplitude of the lineshape for the level at **GS.ECEN** relative to the exciton-polaron lineshape.
GS.ECEN	Adds a level in the electronic ground state manifold.
GS.HOMOBROADENING	Homogeneous broadening of the vibrational levels of the electronic ground state.
GS.INHOMOBROADENING	Inhomogeneous broadening of the vibrational levels of the electronic ground state.
GS.MAXCLOUDSIZE	This is the maximum size of phonon clouds in the electronic ground state.
GS.MAXPHONONS	This is the maximum number of phonons defining the vibrational levels of the electronic ground state (i.e., when no exciton exists).
KRISC.EXCITON	Specifies the triplet intersystem crossing constant.
K.SINGLET	Specifies the coulombic relation between **S.BINDING** and **A.SINGLET** as described in Equation 16-119.
KDD.EXCITON	Specifies the dipole-dipole exciton transfer rate.
KEE.EXCITON	Specifies the electron exchange exciton transfer rate.
KISC.EXCITON	Specifies the exciton intersystem crossing constant.
KNRS.EXCITON	Specifies the singlet non-radiative decay constant.
KNRT.EXCITON	Specifies the triplet non-radiative decay constant.
KSP.EXCITON	Specifies the singlet-polaron constant.
KSS.EXCITON	Specifies the singlet-singlet constant.
KST.EXCITON	Specifies the singlet-triplet constant.
KTP.EXCITON	Specifies the triplet-polaron constant.
KTT.EXCITON	Specifies the triplet-triplet constant.
LDS.EXCITON	Specifies the singlet diffusion length. The synonym for this parameter is LD.EXCITON.
LDT.EXCITON	Specifies the triplet diffusion length.
MIX.NAME	Names the polaron and acts as its unique identifier in a mixture of materials with polaron states. You must specify this parameter even when using single material. It is a unique ID for adding extra states via the **ADDSTATE** parameter.
OPTAU.ORIENT	Activates the Optical S-Matrix cavity effect. It indicates the singlet relaxation time should be scaled by the Optical S-Matrix DOS results. ORIENT is the orientation of the dipoles.

OPTAU.LUM	If the Optical S-Matrix cavity effect is activated, this gives at which wavelength the DOS is used.
OUT.DSPEC	Specifies the TonyPlot compatible file name associated with the **DOPE.SPECT** parameter.
OUT.USPEC	Specifies the TonyPlot compatible file name associated with the **USER.SPECT** parameter.
PHEFF.EXCITON	Specifies the photoluminescence quantum efficiency for the host material.
QE.EXCITON	Specifies the fraction of electron-hole photogeneration rate that goes to the generation of excitons.
QR.EXCITON	Specifies the quenching rate per unit metal density for electrode quenching.
R0.EXCITON	Specifies the Forster radius.
RST.EXCITON	Specifies the fraction of singlets formed during Langevin recombination.
RST.TT.EXCITON	Specifies the fraction of singlets formed during triplet-triplet annihilation.
S.BINDING	Specifies the singlet binding energy used in Equation 16-63.
SIGSN.EXCITON	Specifies the exciton-electron capture cross section.
SIGSP.EXCITON	Specifies the exciton-hole capture cross section.
SINGLET	Flags the state as a singlet.
STATEID	An identifier for a state, which allows it to be referred to or modified later in the input deck, once it has been defined in the **MATERIAL** statement once.
STOKES	Stokes shift due to reorganization energy of the environment, which adds to the Stokes shift due to the exciton-phonon coupling included in the **HOLSTEIN** model
TAUS.EXCITON	Specifies the singlet radiative decay lifetime. The synonym for this parameter is TAU.EXCITON.
TAUT.EXCITON	Specifies the triplet radiative decay lifetime.
TRIPLET	Flags the state as a triplet.
USER.SPECT	Specifies the file name of a user-defined spectrum file used during reverse raytrace. You can define a second user definable spectrum file using the DOPE.SPECT parameter.

Miscellaneous Material Parameters

A1, **A2**, **A3**, **A4**, **A5**, and **A6**	Specify valence band effective mass parameters.
AC, **AV**, and **BBB**	Specify cubic deformation potentials.
ALATTICE	Specifies the in plane lattice constant.

ASTR, **BSTR**, **CSTR**, **DSTR**, **ESTR**, and **FSTR**	These are user-definable parameters of the equations for heavy and light hole masses in Ishikawa's strain model (see Section 3.9.12 "Ishikawa's Strain Effects Model").
C11, **C12**, **C13**, and **C33**	Specify the elastic stiffness coeficients.
CAPT.AUGERN	Specifies the coefficient for electron initiated Auger recombination rate for quantum confined carriers.
CAPT.AUGERP	Specifies the coefficient for electron initiated Auger recombination rate for quantum confined carriers.
CAPT.MUN0	Optional constant electron mobility for inplane transport in WELL.INPLANE model.
CAPT.MUP0	Optional constant hole mobility for inplane transport in WELL.INPLANE model.
CAPT.SRH.TAUN	Electron lifetime for SRH model for quantum confined carriers.
CAPT.SRH.TAUP	Hole lifetime for SRH model for quantum confined carriers.
D0.H1	Diffusion coefficient for atomic hydrogen.
D0.H2	Diffusion coefficient for molecular hydrogen.
DEGENERACY	Specifies the spin degeneracy.
DELTA1, **DELTA2**, and **DELTA3**	Specify the valence band energy splits.
DGN.GAMMA and **DGP.GAMMA**	Specify the electron and hole tuning parameters for density gradient modeling.
DINDEXDT	Specifies the temperature coefficient of the index of refraction.
DRHODT	Specifies the temperature coefficient of conductor resistivity.
D1, **D2**, **D3**, and **D4**	Specify shear deformation potentials.
E31, **E33**, and **PSP**	Specify piezo-electric constants.
EA.H1	Activation energy for atomic hydrogen diffusion
EA.H2	Activation energy for molecular hydrogen diffusion.
EMISS.EFFI	Specifies the number of photons emitted per photon absorbed for frequency conversion materials (see Section 11.11 "Frequency Conversion Materials (2D Only)").
EMISS.LAMB	Specifies the emission wavelength of the frequency conversion material described in (see Section 11.11 "Frequency Conversion Materials (2D Only)").
EMISS.NANG	Specifies the number of angles evenly spaced around a circle of 360° in which emission centers emit light.

EMISS.NX	Specifies the number of emission centers in the x direction in a frequency conversion material.
EMISS.NY	Specifies the number of emission centers in the y direction in a frequency conversion material.
EMISS.WIDE	Specifies the width of each ray emitted in a frequency conversion material.
EP.MBULK	This is an energy parameter used to calculate the momentum matrix element (see Equation 3-625).
F.CAPT.MUN	C-Interpreter file for electron mobility for inplane transport in WELL.INPLANE model.
F.CAPT.MUP	C-Interpreter file for hole mobility for inplane transport in WELL.INPLANE model.
F.MAX and **F.MIN**	Specify the electric field range for outputting certain models for plotting in TonyPlot.
F.NUMBER	Specifies the number of field samples used to output certain field dependent models for plotting in TonyPlot.
FB.MBULK	This is a correction factor used to calculate the momentum matrix element (see Equation 3-625).
H1TOH2RATE	Hydrogen dimerisation rate.
H2TOH1RATE	Molecular hydrogen dissociation rate.
H1SRV	Atomic hydrogen surface recombination velocity at device exterior.
H2SRV	Molecular hydrogen surface recombination velocity at device exterior.
HUANG.RHYS	This is a parameter of the Schenk trap assisted tunneling model.
IF.CHAR	Specifies the Gaussian characteristic distance for spreading polarization charges away from the interface.
IINOFF and **IIPOFF**	Specify file names for outputting electron and hole ionization rate versus electric field for plotting in TonyPlot. **Note:** You should also specify F.MIN, F.MAX, and NUM.F
LUTT1, **LUTT2**, and **LUTT3**	Specify the luttinger parameters.
M.VTHN and **M.VTHP**	Specify the electron and hole effective masses used for calculation of thermal velocity.
MBULKSQ	This is the momentum matrix element.
MC	Specifies the conduction band effective mass.

MINIMA	Specifies the number of equavilent minima in the conduction band energy.
MHH	Specifies the heavy hole effecitive mass.
MHHZ	Specifies the heavy hole (z) effective mass.
ML	Specifies the conduction band longitudinal effective mass.
MLH	Specifies the light hole effective mass.
MLHZ	Specifies the light hole (z) effective mass.
MSO	Specifies the split off (crystal lattice) effective mass.
MSOZ	Specifies the split off hole (z) effective mass.
MSTAR	Specifies the conduction band effective mass dispersion used in Equation 3-625.
MTT	Specifies the transverse effective mass (wurtzite).
MT1 and **MT2**	Specify the transverse effective masses (zincblende).
MUNOFF and **MUPOFF**	Specify file names for writing electron and hole mobility field characteristics for plotting in TonyPlot.
MV	Specifies the valence band effective mass.
MZZ	Specifies the effective mass along the crystal axis (wurtzite).
PHONON.ENERGY	This is a parameter of the Schenk trap assisted tunneling model.
PIP.OMEGA	This is the phonon energy in eV for the Pipinys model.
PIP.ACC	This is the electron-phonon interaction constant for the Pipinys model.
PIP.ET	This is the trap depth in eV for the Pipinys model.
PIP.NT	This is the occupied interface state density for the Pipinys model.
PSP	Specifies the spontanteous polarization.
RESISTIVITY	Specifies the resistivity of conductor regions.
SO.DELTA	Specifies the spin orbital splitting energy in a quantum well. The alias for this parameter is WELL.DELTA.
SOPRA	Identifies the name of a file from the SOPRA database. See Section B.13.1 “SOPRA Database”
SPECIES1.AF	Attempt frequency for ionic species 1 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES1.EA	Activation energy for ionic species 1 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).

SPECIES1.HOP	Hopping distance for ionic species 1 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES2.AF	Attempt frequency for ionic species 2 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES2.EA	Activation energy for ionic species 2 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES2.HOP	Hopping distance for ionic species 2 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES3.AF	Attempt frequency for ionic species 3 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES3.EA	Activation energy for ionic species 3 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
SPECIES3.HOP	Hopping distance for ionic species 3 diffusion coefficient (see Section 3.15 “Generic Ion Transport Model”).
STABLE	Specifies the selected strain table for the Ishikawa strain effects model. See Section 3.9.12 “Ishikawa's Strain Effects Model”.
TE.MODES	Specifies whether TE or TM modes will be used for calculation of asymmetry factors for the LI model.
USER.DEFAULT	Specifies which material the user-defined material should use for its default parameters.
USER.GROUP	Specifies the material group for the user-defined material. `USER.GROUP` can be either `SEMICONDUCTOR`, `INSULATOR`, or `CONDUCTOR`.
WELL.EPS	Specifies the high frequency permittivity used in calculating gain and radiative recombination in a quantum well.
WELL.GAMMA0	Specifies the Lorentzian gain broadening factor from Section 3.9.11 “Lorentzian Gain Broadening”.
WELL.TAUIN	Specifies the Lorentzian gain broadening factor from Section 3.9.11 “Lorentzian Gain Broadening”.
WELL.TAUN	Specifies the quantum well capture escape model electron capture time.
WELL.TAUP	Specifies the quantum well capture escape model hole capture time.
Z.SCHENK	This is a parameter of the Schenk trap assisted tunneling model.

Conductor Parameters

DRHODT	Specifies the temperature coefficient of resistivity in µW·cm/K.
RESISTIVITY	Specifies the material resisitivity in µWcm.

Defect Generation Material Parameters

KD.LID	Specifies the light induced SiHD creation constant. See Equation 15-52.
KH.LID	Specifies the disassociation constant of a hydrogen atom from a SiHD defect. See Equation 15-52.
SIG.LID	Specifies the capture coefficient of free carriers by dangling bonds. See Equation 15-52.

Material Coefficient Definition Examples

Numbered region

This example specifies SRH lifetimes and concentration independent low-field mobilities for region number 2. All other parameters use default values and parameters in other regions are unaffected.

```
MATERIAL TAUN0=5.0E-6 TAUP0=5.0E-6 MUN=3000 MUP=500 REGION=2
```

All regions

This example defines carrier lifetimes and the refractive index for all semiconductor regions.

```
MATERIAL TAUP0=2.E-6 TAUN0=2.E-6 REAL.INDEX=3.7 \
         IMAG.INDEX=1.0E-2
```

Named Material

This shows the definition of bandgap for all InGaAs regions in the structure:

```
MATERIAL MATERIAL=InGaAs EG300=2.8
```

All materials are divided into three classes: semiconductors, insulators and conductors. See Appendix B “Material Systems” for more information about the parameters required for each material class.

Note: You can use the **MODELS** PRINT command to echo back default material parameters or MATERIAL parameter settings to the run-time output.

22.33 MEASURE

MEASURE extracts selected electrical data from the solution.

Note: This statement is almost obsolete. Its functions have been replaced by the **EXTRACT** , **OUTPUT** , or **PROBE** statement.

Syntax

```
MEASURE <dt> [<boundary>] [OUTFILE=<filename>]
```

Parameter	Type	Default	Units
CONTACT	Integer		
E.CRIT	Real1E-8		
ELECTRON	Logical	False	
HOLE	Logical	False	
IONIZINT	Logical	False	
LRATIO	Real	1.0	
METAL.CH	Logical	False	
N.CURRENT	Logical	False	
N.LAYER	Real	15	
N.LINES	Integer	50	
N.RESIST	Logical	False	
NET.CARR	Logical	False	
NET.CHAR	Logical	False	
OUTFILE	Character		
P.CURRENT	Logical	False	
P.RESIST	Logical	False	
REGIONS	Integer	All regions	
SUBSTR	Character		
U.AUGER	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
U.TOTAL	Logical	False	
X.MIN	Real	Left of device	μm

Parameter	Type	Default	Units
X.MAX	Real	Right of device	μm
Y.MIN	Real	Top of device	μm
Y.MAX	Real	Bottom of device	μm

Description

dt	This is used to specify the type of information to be measured.
boundary	Specifies which nodes will be measured.
OUTFILE	Specifies a filename where simulation results and bias information will be written.

Data Type Parameters

Net carrier concentration, charge concentration, electron concentration, or hole concentration can be integrated over a section of a device. The charge on part of an electrode can be calculated, just like the current through that part. This is useful for capacitance studies in conjunction with the difference mode of the LOAD statement. The resistance of a structure cross-section, such as a diffused line, can also be calculated.

E.CRIT	Specifies the critical electric field used to calculate integration integrals.
ELECTRON	Extracts integrated electron concentration.
HOLE	Extracts integrated hole concentration.
IONIZINT	Enables the calculation of ionization integrals. Other integral ionization parameters will be ignored unless IONIZINT is specified.
LRATIO	Specifies the ratio between electric field lines used in ionization integral calculation. The value of this parameter should be set from 0.5 to 1.5.
METAL.CH	Extracts integrated charge on a contact.
N.CURRENT	Extracts n-current through an electrode.
N.LINES	Specifies the number of ionization integrals.
N.RESIST	Extracts n-resistance of a cross-section.
NET.CARR	Extracts integrated carrier concentration.
NET.CHAR	Extracts integrated net charge.
N.LAYER	Controls the distance from the contact where electric field lines start.
P.CURRENT	Extracts p-current through an electrode.
P.RESIST	Extracts p-resistance of a cross-section.

SUBSTR	Selects the substrate electrode for electric field lines. You don't need to specify this parameter if a substrate electrode has been defined in the **ELECTRODE** statement.
U.AUGER	Specifies that the integrated Auger recombination rate is to be extracted.
U.RADIATIVE	Specifies that the integrated radiative recombination rate is to be extracted.
U.SRH	Specifies that the integrated SRH recombination rate is to be extracted.
U.TOTAL	Specifies that the integrated total recombination rate is to be extracted.

Boundary Parameters

Boundary parameters: X.MIN, X.MAX, Y.MIN, and Y.MAX define a bounding box. Only nodes falling within this bounding box are included in the integration. The default bounds are the boundaries of the entire device.

CONTACT	Specifies the contact number. For electrode quantities (current and metal charge), a contact must be selected. Only nodes falling within the bounds and belonging to the contact are included in the integration. When IONIZINT is specified, this is the electrode used to start electric field lines.
REGIONS	Specifies a particular set of regions. If REGIONS is specified, only nodes within the specified bounds that are part of a particular set of regions will be integrated.
X.MAX	Specifies the X coordinate of the right edge of the bounding box.
X.MIN	Specifies the X coordinate of the left edge of the bounding box.
Y.MAX	Specifies the Y coordinate of the top of the bounding box.
Y.MIN	Specifies the Y coordinate of the bottom of the bounding box.

Resistance Example

This example extracts the resistance of a p-type line diffused into a lightly doped n-substrate. Since the p-conductivity of the substrate is negligible, the integration bounds can include the whole device.

```
MEASURE P.RESIST
```

Gate Charge Example

In this example, the charge on the lower surface of a gate electrode is integrated. There is 0.05 μm of gate oxide on the surface, which is located at y=0.

```
MEASURE METAL.CH CONT=1 X.MIN=-2.0 X.MAX=2.0 \
Y.MAX=-0.0499 Y.MIN=-0.0501
```

Ionization Integral Example

This example shows how to extract the maximum ionization integral in the device.

```
MEASURE IONIZINT CONTACT=3 SUSTR=4 N.LINES=200 \
NLAYERS=15 LRATIO=1.1
```

This syntax was the original implementation of ionization integrals in early Atlas versions. It has been superseded. See Sections 22.58 “SOLVE” and 22.43 “OUTPUT” for the recommended approach to extract ionization integrals.

22.34 MESH

MESH generates a mesh or reads a previously generated mesh.

Syntax

```
MESH <prev>|<new> [<output>]
```

Parameter	Type	Default	Units
ATHENA	Logical	False	
AUTO	Logical	False	
CONDUCTOR	Logical	False	
CYLINDRICAL	Logical	False	
DATAFILE.ISE	Character		
DIAG.FLIP	Logical	True	
ELEC.BOT	Logical	False	
GRIDFILE.ISE	Character		
FLIP.POS	Real	Midpoint	μm
FLIP.Y	Logical	False	
IN.FILE	Character		
INFILE	Character		
MASTER.IN	Logical	True	
MASTER.OUT	Logical	True	
MAX.ANGLE	Real	360	Angles
MINOBTUSE	Logical	False	
NX	Integer		
NY	Integer		
NZ	Integer		
OUT.FILE	Character		
OUTFILE	Character		
PERIODIC	Logical	False	
PISCES.IN	Logical	False	
RECTANGULAR	Logical	True	
SCALE	Integer	1	
SCALE.X	Integer	1	

Parameter	Type	Default	Units
SCALE.Y	Integer	1	
SCALE.Z	Integer	1	
SMOOTH.KEY	Integer		
SPACE.MULT	Real	1.0	
THREE.D	Logical	False	
TIF	Logical	False	
VERT.FLIP	Logical	False	
WIDTH	Real	1.0	µm
X.EPI	Logical	False	
Z.EPI	Logical	False	

Description

prev	This is a set of parameters that allows you to read a previously generated mesh type.
new	This is a set of parameters that allow you to initiate the generation of a rectangular mesh.
output	This is a set of the parameters for saving the mesh.

Mesh File Parameters

CONDUCTOR	Interprets metal regions loaded in with the INFILE parameter as conductors .
CYLINDRICAL	Specifies that the mesh being read in contains cylindrical symmetry. Since this information is not saved in the mesh file, the CYLINDRICAL parameter must be specified each time a structure with cylindrical symmetry is loaded.
DATAFILE.ISE	Specifies the name of the data file for ISE formatted structures.
FLIP.Y	Reverses the sign of the Y coordinate.
GRIDFILE.ISE	Specifies the name of the grid file for ISE formatted structures.
IN.FILE	This is a synonym for INFILE.
INFILE	Specifies the name of a previously generated mesh that has been saved to disk. The synonym for this parameter is IN.FILE.
MASTER.IN	Specifies a filename to read mesh and doping information in the Silvaco Standard Structure File (SSF) Format. This parameter is used to read Athena or DevEdit structure files. Typically, these files contain all REGION, ELECTRODE, and DOPING information. Although Atlas allows you to modify the structure using these statements, this parameter is true by default and is the only file format supported by Silvaco.

PISCES.IN	Indicates that the mesh file is in the old PISCES-II format. This is not recommended or supported by Silvaco.
SCALE	Specifies a scale factor by which all X, Y, and Z coordinates are multiplied.
SCALE.X	Specifies a scale factor by which all X coordinates are multiplied.
SCALE.Y	Specifies a scale factor by which all X and Y coordinates are multiplied.
SCALE.Z	Specifies a scale factor by which all Z coordinates are multiplied.
SPACE.MULT	This is a scale factor that is applied to all specified grid spacings. This parameter can be used to produce a coarse mesh and thereby reduce the simulation time.
ATHENA	Reads mesh and doping data generated by the Athena PISCES-II format file. This parameter and file format is obsolete.

Mesh Parameters

AUTO	Specifies that mesh lines will be generated automatically from **REGION** statements. See "Specifying the Mesh" on page 600 for more information on how to specify mesh lines.
CYLINDRICAL	Specifies that the mesh contains cylindrical symetry. The exact meaning also depends on the state of the THREE.D parameter. If THREE.D is not set, the simulation will assume that a 2D mesh in X and Y coordinates is rotated by 360° about the Y axis. In this case, do not define mesh locations with negative X coordinates. Also, note that if such a structure is saved and re-loaded in subsequent simulations, the state of the CYLINDRICAL flag is lost and should be specified in each successive input deck.
DIAG.FLIP	Flips the diagonals in a square mesh about the center of the grid. If the parameter is negated, using DIAG.FLIP is specified, all diagonals will be in the same direction.
FLIP.POS	Works with DIAG.FLIP. If FLIP.POS is specified, the change of mesh diagonal direction occurs at the nearest mesh X coordinate to the value specified by FLIP.POS. If it is not specified, then the default value used is the average X coordinate of the device.
MAX.ANGLE	This can be used to specify the maximum angle of a 2D circular or 3D cylindical mesh, in the case where there is only one A.MESH statement. Otherwise, it is ignored. MAXANGLE is the alias. Its units are degrees.
MINOBTUSE	Minimizes the generation of obtuse triangles during the 3D cylindrical meshing (see Section 2.6.9 "Specifying 3D Cylindrical Structures").
NX	Specifies the number of nodes in the X direction.
NY	Specifies the number of nodes in the Y direction.
NZ	Specifies the number of nodes in the Z direction, used in Device 3D only.

PERIODIC	Specifies the left and right edges of the device will be modeled as periodic (i.e., the solution along the right edge wraps around to the left edge and vice-versa).
RECTANGULAR	Initiates the generation of a rectangular mesh.
THREE.D	Starts Atlas3D.
TIF	Specifies that structure file is in TIF format.
VERT.FLIP	Flips the direction of mesh triangle diagonals. In other words, the diagonals of the mesh are mirrored in the vertical direction. This also works in the XY plane for 3D structures generated within Atlas.
WIDTH	Specifies a scale factor to represent the un-simulated dimension for 2D simulations. This scale factor is applied to all run time and log file outputs.
X.EPI	Specifies that for calculation of uniaxial polarization charge in GaN/InGaN/AlGaN, we assume layers are stacked along the X axis.
Z.EPI	Specifies that for calculation of uniaxial polarization charge in GaN/InGaN/AlGaN, we assume layers are stacked along the Z axis.

Output Parameters

OUT.FILE	This is a synonym for **OUTFILE**.
OUTFILE	Specifies the output filename to which the mesh is written. The synonym for this parameter is **OUT.FILE**.
MASTER.OUT	Specifies the format of the output file. This parameter is `true` by default so the output file will conform to the Silvaco Standard Structure File Format and can be plotted in TonyPlot.
SMOOTH.KEY	Specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows: • Triangle smoothing. All region boundaries remain fixed. • Triangle smoothing. Only material boundaries are maintained. • Node averaging. • Improved triangle smoothing method. This method uses diagonal flipping to reduce the number of obtuse triangles. • Triangle smoothing by flipping diagonals according to electric field. Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSuprem 4. Option 4 is similar to Option 1 but Option 4 usually creates less obtuse triangles.

Mesh Definition Example

This example initiates a rectangular mesh and stores the mesh in file, MESH1.STR.

```
MESH RECTANGULAR NX=40 NY=17 OUTF=MESH1.STR
```

Athena Interface Example

This syntax reads in a mesh from Athena or DevEdit:

```
MESH INFILE=NMOS.STR
```

When the **auto-** interface feature is used in DeckBuild, the program will automatically insert the **MESH** statement to load the result of previous programs into Atlas.

Note: See Sections 2.6.1 “Interface From Athena” and 2.6.2 “Interface From DevEdit”, or the on-line examples for details of the interfaces from Athena or DevEdit to Atlas.

22.35 METHOD

METHOD sets the numerical methods to be used to solve the equations and parameters associated with the these algorithms.

Syntax

```
METHOD <gp> <mdp>
```

Parameter	Type	Default	Units
2NDORDER	Logical	True	
AC.FILL.LEVEL	Real	1	
AC.FILL.RATIO	Real	0.5	
AC.GMRES	Real	False	
AC.PRECONDITIONER	Real	0	
AC.ZIP.BICGST	Logical	False	0
ACONTINU	Real	0.5	
ALT.SCHRO	Logical	False	
ATRAP	Real	0.5	
AUTONR	Logical	False	
BICGST	Logical	False	
BLOCK	Logical	False	
BLOCK.TRAN	Logical	False	
BQP.ALTEB	Logical	False	
BQP.NEWTON	Logical	False	
BQP.NMIX	Real	1	
BQP.NOFERMI	logical	false	
BQP.PMIX	Real	1	
BQPX.TOL	Real	2.5×10^{-7}	
BQPR.TOL	Real	1.0×10^{-26} (2D) 1.0×10^{-18} (3D)	
BR.DAMPING	Logical	False	
BRD.DELTA	Real	0.0001	
BRD.KAPPA	Real	0.0	
BRD.MAX.ITER	Integer	5	

Parameter	Type	Default	Units
BRD.OFF	Logical	False	
BRD.SKIP.ITER	Logical	False	
BTE.STABILITY	Real	1×10^{-5}	
CAPT.ALT	Logical	False	
CARRIERS	Real	2	
C.ITLIMIT	Integer	500	
C.STABIL	Real	$1.0x10^{-10}$	
C.RESID	Real	$1.0x10^{-8}$	
CLIM.DD	Real	4.5×10^{13}	cm^{-3}
CLIM.EB	Real	0	cm^{-3}
CLIM.LAT	Real	10^{5}	cm^{-3}
CLIMIT	Real	10000	
CONT.RHS	Logical	True	
CONTINV	Logical	True	
CR.TOLER	Real	5.0×10^{-18}	
CUR.PROJ	Logical	False	
CX.TOLER	Real	1.0×10^{-5}	
DIRECT	Logical	False	
DG.INIT	logical	false	
DG.N.INIT	logical	false	
DG.P.INIT	logical	false	
DT.MAX	Real	1.0×10^{10}	s
DT.MIN	Real	1.0×10^{-25}	s
DTUPDATE	Real	10^{-9}	K
DVLIMIT	Real	0.1	
DVMAX	Real	2.5	V
EF.TOL	Real	10^{-12}	
ELECTRONS	Logical	True	

Parameter	Type	Default	Units
EM.DAMPING	Logical	False	
EMD.MAX.ITER	Integer	5	
EMD.OFF	Logical	False	
EMD.SKIP.ITER	Integer	5	
ETR.TOLE	Real	100	
ETX.TOLE	Real		
EXTRAPOLATE	Logical	False	
FAIL.QUIT	Logical	False	
FAIL.SAFE	Logical	True	
FGCNTRL	Logical	True	
FILL.LEVEL	Real	0	
FILL.RATIO	Real	0.5	
FIX.QF	Logical	False	
FLUX.JOULE	Logical	True	
GANSAT.DERIV	Logical	False	
GCARR.ITLIMIT	Real	1	
GCON.ITLIMIT	Real	0	
GEI.METH	Integer	0	
GMRES	Logical	False	
GUM.INIT	Integer	15	
GUMITS	Integer	100	
GUMMEL	Logical	False	
GUMMEL.NEWTON	Logical	False	
HALFIMPLICIT	Logical	False	
HCIR.TOL	Real	$5.0x10^{-11}$	
HCIX.TOL	Real	$5.0x10^{-4}$	
HCTE.THERM	Logical	False	
HOLES	Logical	True	
ICCG	Logical	False	

Parameter	Type	Default	Units
ITMIN	Integer	1	
IR.TOL	Real	5.0×10^{-15}	
ITAT.EXTENT	logical	False	
ITLIMIT	Integer	25	
IX.TOL	Real	2.0×10^{-5}	
LAS.PROJ	Logical	False	
LS.DAMPING	Logical	False	
LSD.ALPHA	Real	0.0001	
LSD.MAX.ITER	Integer	5	
LSD.OFF	Logical	False	
LSD.SKIP.ITER	Integer	5	
LTR.TOLE	Real	100	
LTX.TOLE	Real		
LTE2STEP	Logical	False	
LU1CRI	Real	3.0×10^{-3}	
LU2CRI	Real	3.0×10^{-2}	
L2NORM	Logical	True	
MAG2DLSQ	Logical	False	
MAG3DINT.N	Logical	False	
MAG3DINT.P	Logical	False	
MAXTRAPS	Integer	4	
MAX.TEMP	Real	2000	K
MEINR	Logical	True	
MIN.TEMP	Real	120	K
NBLOCKIT	Integer	15	
NEG.CONC	Logical	False	
NEW.EI	Logical	False	
NEWTON	Logical	True	
NITGUMM	Integer	5	

Parameter	Type	Default	Units
NOGRADMSTAR	Logical	False	
NT0	Integer	4	
NT1	Integer	10	
NT2	Integer	25	
NT3	Integer	100	
NRCRITER	Real	0.1	
NO.POISSON	Logical	False	
PC.FILL_LEVEL	Real	0	
PC.SCALE	Real	1	
PCM	Logical	False	
PR.TOLER	Real	1.0×10^{-26}	
PRECONDIT	Real	0	
PRINT	Logical	False	
PX.TOLER	Real	1.0×10^{-5}	
RAT.FGCNTRL	Real	1.0	
RATIO.TIME	Real	0.2	
REL.STOP	Logical	False	
RHSNORM	Logical	False	
RXNORM	Logical	True	
QMAX.FGCNTRL	Real	100.0	Q
QUASI	Logical	True	
RG.MSCTRAP	Logical	True	
SEMIIPLICIT	Logical	False	
SCHUR	Logical	True	
SINGLEPOISSON	Logical	False	
SP.NUMITER	Integer	30	
SP.NUMOSC	Integer	30	
SP.STAOSC	Integer	3	
SP.MINDIF	Real	1.0e-3	

Parameter	Type	Default	Units
SPECIES.INS	Logical	False	
SPECIES.MAXX	Real	1000.0	
SPECIES.MAT	Character		
SPECIES.REG	Real		
SPEEDS	Logical	False	
STACK	Integer	4	
TAUTO	Logical	True	
TCR.TOL	Real	100	
TCX.TOL	Real		
TLR.TOL	Real	100	
TLX.TOL	Real		
TMIN.FACT	Real	0.4	
TOL.LTEMP	Real	0.001	
TOL.RELAX	Real	1	
TOL.TIME	Real	5.0×10^{-3}	
TRAP	Logical	True	
TSTEP.INCR	Real	2.0	
TTNMA	Real	60000.0	K
TTPMA	Real	60000.0	K
TUN.WKB	Logical	False	
V.TOL	Real	1.0×10^{-6}	
VSATMOD.INC	Real	0.01	
XANDRNORM	Logical	False	
XNORM	Logical	False	
WEAK	Real	300	
WELL.ERROR	Real	1e-8	
WELL.ITERMAX	Integer	20	

Parameter	Type	Default	Units
WELL.PROJ	Logical	True	
WELL.SELFCON	Logical	False	
ZIP.BICGST	Logical	False	

Description

The METHOD statement is used to set the numerical methods for subsequent solutions. All structure and model definitions should precede the METHOD statement and all biasing conditions should follow it. Parameters in the METHOD statement are used to set the solution technique, specify options for each technique and tolerances for convergence.

Solution Method Parameters

AC.FILL.LEVEL	Specifies the fill level for the ILK preconditioner (AC.PRECONDITIONER=0) used by the AC.ZIP.BICST iterative solver in 3D AC simulations.
AC.FILL.RATIO	Specifies the fill ratio for the ILUP preconditioner (AC.PRECONDITIONER=2) used by the AC.ZIP.BICGST iterative solver in 3D AC simulations.
AC.GMRES	Specifies that the GMRES iterative solver will be used for 3D AC simulations.
AC.PRECONDITIONER	Specifies the preconditioner used by the AC.ZIP.BICST and AC.GMRES solvers for 3D AC simulators.
AC.ZIP.BICGST	Specifies that the ZIP.BICGST iterative solver will be used for 3D AC simulators.
ALT.SCHRO	Specifies that the alternative Schrodinger solver should be used.
BLOCK	Specifies that the block Newton solution method will be used as a possible solution method in subsequent solve statements until otherwise specified. The Block method only has meaning when either lattice heating or energy balance is included in the simulation. For isothermal drift diffusion simulations, BLOCK is equivalent to NEWTON.
BLOCK.TRAN	Specifies that the block Newton solution method will be used in subsequent transient solutions.
BICGST	Switches from the default ILUCGS iterative solver to the BICGST iterative solver for 3D simulations.
BQP.NEWTON	Enables a full newton method for either BQP.N or BQP.P models.
CAPT.ALT	Enables an alternative equation for the Capture-Escape rate model (Equation 1)
DIRECT	Specifies that a direct linear solver will be used to solve the linear problem during 3D simulation. By default, the ILUCGS iterative solver is used for 3D problems.
FILL.LEVEL	Specifies the fill level for the ILK preconditioner (PRECONDIT=1) used by the ZIP.BIGST iterative solver.

FILL.RATIO	Specifies the fill ratio for the ILUP preconditioner (PRECONDIT=2) used by the ZIP.BIGST iterative solver.
GEI.METH	Selects solution method for GEIGER model.
GMRES	Switches from the default ILUCGS iterative solver to the GMRES iterative solver for 3D simulations.
GUMMEL	Specifies the Gummel method will be used as a solution method in subsequent **SOLVE** statements until otherwise specified. If other methods (BLOCK or NEWTON) are specified in the same **METHOD** statement, each solution method will be applied in succession until convergence is obtained. The order that the solution methods will be applied is GUMMEL then BLOCK then NEWTON. If no solution methods are specified NEWTON is applied by default.
GUMMEL.NEWTON	Specifies that a NEWTON solution will always be performed after a GUMMEL solution, even if GUMMEL has converged for all equations.
HALFIMPLICIT	Specifies that a semi-implicit scheme will be used for transient solutions in 3D. In most cases this method is significantly less time consuming than the default TR-BDF method.
MAG2DLSQ	Enables least squares algorithm for the galvanic transport model.
MAG3DINT.N	Enables an implementation of Galvanic transport for electrons in Magnetic 3D. See Section 3.11 “Carrier Transport in an Applied Magnetic Field”.
MAG3DINT.P	Enables an implementation of Galvanic transport for holes in Magnetic 3D. See Section 3.11 “Carrier Transport in an Applied Magnetic Field”.
MEINR	Specifies the Meinerzhagens Method, whereby carrier temperature equations will be coupled with the associated carrier continuity equation and used during GUMMEL iterations.
NEWTON	Specifies that Newton’s method will be used as the solution method in subsequent SOLVE statements unless specified otherwise. Certain models and boundary conditions settings require that the Newton’s method is used. If no solution methods are specified, NEWTON is applied by default.
NEW.EI	Enables an alternative eigenvalue solver that obtains eigenvectors using the reverse iteration method.
PRECONDIT	Specifies the preconditioner used by the ZIP.BIGST iterative solver. PRECONDIT=0 specifies the ILU preconditioner (default). 1 specifies ILUK and 2 specifies ILUP.
REL.STOP	Specifies that relative stopping criteria will be used for the GMRES iterative solver.

SPEEDS	Specifies that the SPEEDS direct linear solve will be used.
TUN.WKB	Chooses the WKB approximation as the Schrodinger solution method for the SIS.EL and SIS.HO methods.
ZIP.BICGST	Specifies that the ZIP library version of the BICGST iterative solver will be used for 3D solution.

Note: Details on the different solution methods can be found in Chapter 21 "Numerical Techniques".

Equation Solver Parameters

CARRIERS	Specifies the number of carrier continuity equations that will be solved Valid values are 0, 1 and 2. CARRIERS=0 implies that only Poisson's Equation will be solved. CARRIERS=1 implies that only one carrier solution will be obtained. When this is specified, also specify either HOLES or ELECTRONS. CARRIERS=2 implies that solutions will be obtained for both electrons and holes.
ELECTRONS	Specifies that only electrons will be simulated for single carrier simulation.
HOLES	Specifies that only holes will be simulated for single carrier simulations.

Solution Tolerance Parameters

The default convergence criteria used in Atlas consists of a combination of relative and absolute values. The program will converge if either criterion is met. This is particularly useful when low-carrier concentrations would not converge using just relative criteria.

Current convergence criteria are also used. Terminal currents are monitored at each iteration and overall convergence is allowed if currents converge along with absolute potential error.

BQPX.TOL	This is the convergence criterion for the update vector in the BQP model.
BQPR.TOL	This is the convergence criterion for the Right hand side (residual) in the BQP model.
CR.TOLER	Specifies an absolute tolerance for the continuity equation.
CX.TOLER or **C.TOL**	This is the relative tolerance for the continuity equation. The XNORM parameter uses the CX.TOL and PX.TOL parameters to calculate convergence criteria.
EF.TOL	Specifies the Schrodinger solver tolerance when NEW.EI is specified.
ETR.TOLE	This is an alias for **TCR.TOL**.
ETX.TOLE	This is an alias for **TCX.TOL**.
HCIR.TOL	This is the absolute current convergence criteria for energy transport models.
HCIX.TOL	This is the relative current convergence criteria for energy transport models.
IR.TOL	Specifies absolute current convergence criteria.
IX.TOL	Specifies relative current convergence criteria.

LTR.TOLE	This is an alias for **TLR.TOL**.
LTX.TOLE	This is an alias for **TLX.TOL**.
PR.TOLER	Specifies an absolute tolerance for the Poisson Equation.
PX.TOLER	This is the relative tolerance for the potential equation. The XNORM parameter uses the CX.TOL and PX.TOL parameters to calculate convergence criteria.
RHSNORM	Specifies that only absolute errors will be used to determine convergence. If RHSNORM is selected Poisson error are measured in C/μm and the continuity error is measured in A/μm.
RXNORM	Specifies that both relative and absolute convergence criteria will be used in the solution method. This is the equivalent of specifying both XNORM and RHSNORM. This is the default and it is not recommended to change this.
TCR.TOL	Specifies the absolute (RHSNORM) tolerance for convergence of the carrier temperature equations. The alias for this parameter is **ETR.TOLE**.
TCX.TOL	Specifies the relative (XNORM) tolerance for convergence of the carrier temperature equations. The alias for this parameter is **ETX.TOLE**.
TOL.TIME	Specifies maximum local truncation error for transient simulations.
TOL.LTEMP	Specifies the temperature convergence tolerance in block iterations using the lattice heat equation.
TOL.RELAX	Specifies a relaxation factor for all six Poisson, continuity, and current convergence parameters (PX.TOL, CX.TOL, PR.TOL, CR.TOL, IX.TOL, and IR.TOL).
TLR.TOL	Specifies the relative (XNORM) tolerance for convergence of the lattice temperature equation. The alias for this parameter is **LTR.TOLE**.
TLX.TOL	Specifies the relative (XNORM) tolerance for convergence of the lattice temperature equation. The alias for this parameter is **LTX.TOLE**.
WEAK	Specifies the multiplication factor for weaker convergence tolerances applied when current convergence is obtained.
XANDRNORM	Specifies that the bias point will only be considered to be converged if both the X and RHS convergence norms are be satisfied.
XNORM	Specifies that only the relative errors will be used to determine convergence for the drift-diffusion equations. If XNORM is used Poisson updates are measured in kT/*q*. and carrier updates are measured relative to the local carrier concentration.

Note: Generally, the solution tolerances should not be changed. Convergence problems should be tackled by improving the mesh or checking the model and method combinations. Chapter 2: "Getting Started with Atlas" has useful hints.

General Parameters

ACONTINU	This is an alias for **ATRAP**.
ATRAP	Specifies the multiplication factor which reduces the electrode bias steps when a solution starts to diverge. This parameter has no effect unless the TRAP parameter is specified. The alias for this parameter for **ACONTINU**.
BQP.NMIX **BQP.PMIX**	Specify a mixing parameter with value between 0 and 1, which causes relaxation of fermi-dirac factor for electron and holes in the BQP model.
BTE.STABILITY	This can be used to mitigate the occurrence of oscillations in the solution of the Boltzmann transport equation.
CLIM.DD	This is analogous to CLIMIT except it is expressed in a dimensional value representing the minimum carrier concentration that can be resolved.
CLIM.EB	This can be treated as a regularization parameter for the case of very small carrier concentrations for energy balance simulation. It specifies the minimum value of carrier concentration for which the relaxation term in the energy balance equation will still be properly resolved. Carrier temperatures for points where the concentration is much less than CLIM.EB, will tend to the lattice temperature.
CLIM.LAT	Specifies the minimum resolvable carrier concentration for Joule heating calculation.
CLIMIT	Specifies a concentration normalization factor. See "Carrier Concentrations and CLIM.DD (CLIMIT)" on page 1043 for a complete description.
CONT.RHS	This is an alias for **TRAP**.
CONTINV	This is an alias for **TRAP**.
CUR.PROJ	Enables the use of projection method for initial guesses with current boundary conditions.
DTUPDATE	Specifies the change in local temperature during lattice heating simulations, which local composition and temperature dependent physical models (such as mobility and bandgap) are recalculated.
FAIL.QUIT	Requires that excessive trapping during continuation will exit the simulator.
FAIL.SAFE	Requires that premature exit from a static ramp due to compliance or excessive trapping during continuation will exit with the device in the last converged state.
FIX.QF	Fixes the quasi-Fermi potential of each non-solved for carrier to a single value, instead of picking a value based on local bias.
FGCNTRL	Enables the FGCG model stabilisation algorithm for addition of charge to floating gate.

FLUX.JOULE	If TRUE, then it uses a flux-like discretization for including Joule heating into the lattice temperature equation. If FALSE, then it uses a source-like discretization for including Joule heating into the lattice temperature equation. This applies when LAT.TEMP and JOULE.HEAT are enabled. It's TRUE by default. The source-like discretization is provided primarily for compatibility with Victory Device.
GANSAT.DERIV	Specifies that the full derivatives for the GANSAT.N and GANSAT.P mobility models will be used.
HCTE.THERM	Specifies that hot carrier simulation will use a thermodynamically correct Joule heating term. This term has the form $(\bar{J}_n \cdot \bar{J}_n)/(\mu_n N)$ rather than the default $(J_n \cdot E)$.
ITLIMIT **GITLIMIT**	Specifies the maximum number of allowed outer loops (Newton loops or Gummel continuity iterations).
ITMIN	Specifies a minimum number of iterations before checking for convergence.
LAS.PROJ	Extrapolates the photon rates for the initial guess during bias ramp.
MAXTRAPS	Specifies the number of times the trap procedure will be repeated in case of divergence. The value of MAXTRAPS may range from 1 to 10. The alias for this parameter is **STACK**.
MIN.TEMP **MAX.TEMP**	These are specified to control the absolute range of lattice temperatures allowed during Gummel loop iterations with lattice temperature. These parameters help insure that lattice temperatures converge during the outer loop iterations.
NBLOCKIT	Specifies the maximum number of BLOCK iterations. If METHOD BLOCK NEWTON is specified, the solver will switch to Newton's method after NBLOCKIT Block-Newton iterations.
NEG.CONC	This flag allows negative carrier concentrations.
NO.POISSON	This flag allows you to omit the solution of the Poisson's Equation (see Equation 3-1). This feature doesn't work with Gummel's Method (see Section 21.5.2 "Gummel Iteration").
NOGRADMSTAR	Specifies that the gradient of the log of the effective mass will be omitted from drift-diffusion and hydrodynamic/energy balance calculations. This may help convergence properties.
PC.FILL_LEVEL	Specifies the fill level to be used in the ZIP.BICGST iterative solver.
PC.SCALE	Specifies the scaling method to be used in the ZIP.BICGST iterative solver.
PCM	Sets certain models and parameter values for optimal simulation of phase change materials (see Table 8-12).
PRINT	Specifies that parameter information for the linear and non-linear solver will be displayed.
QMAX.FGCNTRL	Threshold charge parameter for FGCNTRL stabilization algorithm.

RAT.FGCNTRL	Step size ratio parameter for FGCNTRL stabilization algorithm.
RG.MSCTRAP	Clearing this flag decouples the multistate trap recombination rates from the corresponding carrier or ionic transport continuity equations.
SPECIES.INS	Restricts the generic ionic species to insulator regions only.
SPECIES.MAXX	Gives the maximum relative update of the species concentrations during Newton iteration.
SPECIES.MAT	Restricts the generic ionic species to the named material only.
SPECIES.REG	Restricts the generic ionic species to the numbered region only.
STACK	This is an alias for **MAXTRAPS**.
TMIN.FACT	Specifies the minimum electron or hole temperature allowable during non-linear iteration updates. TMIN.FACT is normalized to 300K.
TRAP	Specifies that if a solution process starts to diverge, the electrode bias steps taken from the initial approximation are reduced by the multiplication factor ATRAP. The aliases for this parameter is **CONT.RHS** and CONTINU.
TTNMA	Specifies the maximum allowable electron carrier temperature.
TTPMA	Specifies the maximum allowable hole carrier temperature.
VSATMOD.INC	Specifies that the derivatives of the negative differential mobility (MODEL FLDMOB EVSATMOD=1) will not be included into the Jacobian until the norm of Newton update for potential is less than the value specified by VSATMOD.INC. This is useful since the negative differential mobility model is highly nonlinear and causes numerical stability problems.

Gummel Parameters

DVLIMIT	Limits the maximum potential update for a single loop.
GCARR.ITLIMIT	Specifies the maximum number of iterations that the carrier continuity equations will solve when using GUMMEL.
GCON.ITLIMIT	Specifies the number of extra GUMMEL loop iterations that will be performed after all equations have converged.
GUM.INIT	Specifies the maximum number of Gummel iterations in order to obtain an initial approximation for successive Newton iterations. This parameter is used when METHOD GUMMEL NEWTON is specified
GUMITS	Specifies the maximum number of Gummel iterations.

LU1CRI and **LU2CRI**	Specifies amount of work per Poisson loop. The inner norm is required to either decrease by at least LU1CRI before returning, or reach a factor of LU2CRI below the projected Newton error, whichever is smaller. If the inner norm is exceeds the projected Newton error, the quadratic convergence is lost.
SINGLEPOISSON	Specifies that only a single Poisson iteration is to be performed per Gummel loop. In the default state, the continuity equation is only treated after the Poisson iteration has fully converged. This technique is useful where the carrier concentration and potential are strongly coupled but the initial guess is poor precluding the use of NEWTON.
NITGUMM, **NT0**, **NT1**, **NT2**, and **NT3**	Specify Gummel iteration control parameters described in Section 21.5.10 “Detailed Convergence Criteria”.

Newton Parameters

2NDORDER	Specifies that second-order discretization will be used when transient simulations are performed.
AUTONR	Implements an automated Newton-Richardson procedure, which attempts to reduce the number of LU decompositions per bias point. We strongly recommend that you use this parameter to increase the speed of NEWTON solutions. Iterations using AUTONR will appear annotated with an A in the run-time output. Often an extra iteration is added when using this parameters since the final iteration of any converged solution cannot be done using AUTONR.
DT.MAX	Specifies the maximum time-step for transient simulation.
DT.MIN	Specifies the minimum time-step for transient simulations.
DVMAX	Sets the maximum allowed potential update per Newton iteration. Large voltage steps are often required when simulating high voltage devices. If any simulation requires voltage steps of 10V or more, set DVMAX to 100,000. Reducing DVMAX may serve to damp oscillations in solutions in some cases leading to more robust behavior. Excessive reduction, however, in DVMAX is not recommended since the maximum voltage step allowed will be limited by DVMAX*ITLIMIT. The alias for this parameter is N.DVLIM.
EXTRAPOLATE	Specifies the use of second-order extrapolation to compute initial estimates for successive time-steps for transient simulations.
L2NORM	Specifies the use of L2 error norms rather than infinity norms when calculating time steps for transient simulations.
LTE2STEP	Use an alternative method for evaluating Local Truncation Error during a transient run. This method uses two adjacent time steps and is less prone to numerical noise than the default method.
NRCRITER	Specifies the ratio by which the norm from the previous Newton loop must decrease in order to be able to use the same Jacobian (LU decomposition) for the current Newton loop.
N.DVLIM	This is an alias for **DVMAX**.
RATIO.TIME	Specifies the minimum time step ratio allowed in transient simulations. If the calculated time step divided by the previous time step is less than RATIO.TIME. Atlas will cut back the transient solution instead of continuing on to the next time point.
TAUTO	Selects automatic, adaptive timesteps for transient simulations from local truncation error estimates. Automatic time-stepping is the default for second-order discretization, but is not allowed for first-order.
TSTEP.INCR	Specifies the maximum allowable ratio between the time step sizes of successive (increasing) time steps during transient simulation.
QUASI	Specifies a quasistatic approximation for transient simulations. This is useful in simulating transient simulations with long timescales where the device is in equilibrium at each timestep.

Quantum Parameters

BQP.NOFERMI	Forces the BQP model to only use its formulation derived using Maxwell Boltzmann statistics. If this is not set, then the BQP formulation used will be the same as the carrier statistics specified for the electrons or holes or both.
BQP.ALTEB	Enables an alternative iteration scheme that can be used when solving the BQP equation and the energy balance equation. This may produce a more robust convergence than for the default scheme. The scheme used with BQP.ALTEB set, however, will be slower.
DG.INIT	Enables an algorithm to initialize the Density Gradient method within the **SOLVE** INIT statement. This is not self-consistent with the solution of Poisson's equation, which makes it limited in usefulness. You can specify the carrier type to be initialized using either **DG.N.INIT** or **DG.P.INIT**. DG.INIT will enable both DG.N.INIT and DG.P.INIT.
ITAT.EXTENT	Extends the lower range of integration in ITAT models to be the trap level rather than the band edge. This gives better results for high field simulations.
SCHUR	Specifies that the Schur Compliment method is used for the quantum well rate equations for the quantum well capture escape model.
SP.NUMITER	Sets the maximum number of Schrodinger, NEGF, or DDMS-Poisson iterations, after which Atlas proceeds to the next bias point.
SP.NUMOSC	The number of previous Schrodinger-Poisson iterations, whose residual is compared to the residual of the current iteration in order to check whether the residual is oscillating.
SP.STAOSC	The number of Schrodinger-Poisson iteration, after which Atlas starts checking for oscillatory behavior of the residual.
SP.MINDIF	Minimum difference between logarithm of current and previous residuals of Schrodinger-Poisson below which the residual is regarded as oscillating and Atlas proceeds to the next bias point.
WELL.ERROR	Specifies the largest relative error of bound carrier density to regulate convergence for the quantum well capture escape model.
WELL.ITERMAX	Specifies the maximum number of Newton iterations for the coupled system for the quantum well capture escape model.
WELL.PROJ	Enables the projection method for the 2D densities for the quantum well capture escape model.
WELL.SELFCON	Enables charge self-consistency for the capture escape model by ensuring band edges as seen by the bulk carriers are modified to remove wells in the QWELL regions and bound carrier density is added to the Poisson equation.

Globally Convergent Scheme Parameters

BR.DAMPING	Turns on Bank-Rose damping.
BRD.OFF	Turns off Bank-Rose damping.
BRD.SKIP.ITER	Indicates how many Newton iterations should be performed before Bank-Rose damping is used.
BRD.MAX.ITER	Indicates the maximum number of Bank-Rose inner iterations that are performed before Atlas indicates a failure to find a suitable step.
BRD.DELTA	The minimum acceptable reduction in the residual, scaled to the full Newton step, is `BRD.DELTA*g(0)` (where `g(0)` is the residual at the current point).
BRD.KAPPA	The next step in a Bank-Rose inner iteration is `k = 1/(1 + K*g(0))` (where `g(0)` is the residual at the current point). This sets the initial value of `K`.
EM.DAMPING	Turns on explicit minimum damping.
EMD.OFF	Turns off explicit minimum damping.
EMD.SKIP.ITER	Indicates how many Newton iterations should be performed before explicit minimum damping is used.
EMD.MAX.ITER	Indicates the number of explicit minimum inner iterations used to bracket the minimum.
LS.DAMPING	Turns on line search damping.
LSD.OFF	Turns off line search damping.
LSD.SKIP.ITER	Indicates how many Newton iterations should be performed before line search damping is used.
LSD.MAX.ITER	Indicates the maximum number of line search inner iterations that are performed before Atlas indicates a failure to find a suitable step.
LSD.ALPHA	The minimum acceptable reduction in the residual, scaled to the full Newton step, is `-LSD.ALPHA*g'(0)` (where `g'(0)` is the gradient, in the direction of the Newton step, of the residual at the current point).

Numerical Method Definition Example

The default numerical method is the equivalent of:

```
METHOD NEWTON CARRIERS=2
```

For more complex problems including those involving floating regions the following is recommended

```
METHOD GUMMEL NEWTON GUM.INIT=5
```

When impact ionization is combined with floating regions as in SOI or guard ring breakdown simualation the above syntax can also be used. Quicker solutions, however, can be obtained using the following `SINGLEPOISSON` technique:

```
METHOD GUMMEL NEWTON GUM.INIT=5 SINGLE
```

TRAP Parameter Example

This example illustrates the trap feature (often used to capture knees of I-V curves for junction breakdown).

The first **SOLVE** statement solves for the initial, zero bias case. In the second **SOLVE** statement, we attempt to solve for V2=3 volts and V3=5 volts. If such a large bias change caused the solution algorithms to diverge for this bias point, the bias steps would be multiplied by ATRAP (0.5).

An intermediate point (V2=1.5 volts, V3=2.5 volts) would be attempted before trying to obtain V2=3 volts and V3=5 volts again. If the intermediate point can not be solved for either case, then the program will continue to reduce the bias step (the next would be V2=0.75 volts and V3=1.25 volts) up to MAXTRAPS times.

```
METHOD TRAP ATRAP=0.5

SOLVE INIT

SOLVE V2=3 V3=5 OUTFILE=OUTA
```

Transient Method Example

In this transient simulation example, second-order discretization is used (by default), but the required LTE (10^{-3}) is smaller than the default. Since the Jacobian is exact for the second part (BDF-2) of the composite timestep, there should be very few factorizations for the BDF-2 interval when AUTONR is specified.

```
METHOD NEWTON TOL.TIME=1E-3 AUTONR
```

Note: For recommendations on METHOD parameters for different simulations, see Chapter 2 "Getting Started with Atlas" or the on-line examples.

22.36 MOBILITY

MOBILITY allows specification of mobility model parameters.

Syntax

```
MOBILITY [NUMBER=<n>] [REGION=<n>] [MATERIAL=<name>]
         [NAME=<region_name>] <parameters>
```

Parameter	Type	Default	Units
A.BROOKS	Real	1.56×10^{21}	$(\text{cm V s})^{-1}$
A.PASV.N	Real	0.1	nm
A.PASV.P	Real	0.1	nm
ACCN.SF	Real	0.87	
ACCP.SF	Real	0.87	
ALBRCT.N	Logical	False	
ALBRCT.P	Logical	False	
AL1N.WATT	Real	-0.16	
AL1P.WATT	Real	-0.296	
AL2N.WATT	Real	-2.17	
AL2P.WATT	Real	-1.62	
AL3N.WATT	Real	1.07	
AL3P.WATT	Real	1.02	
ALP.PASV.N	Real		nm^{-1}
ALP.PASV.P	Real		nm^{-1}
ALPHAN	Real	0.0	
ALPHAN.ARORA	Real	-0.57	
ALPHAN.CAUG	Real	0.0	
ALPHAP	Real	0.0	
ALPHAP.ARORA	Real	-0.57	
ALPHAP.CAUG	Real	0.0	
ALPHAN.FLD	Real	2.4×10^{7}	cm/s
ALPHAP.FLD	Real	2.4×10^{7}	cm/s
ALPHA1N.KLA	Real	0.68	

Parameter	Type	Default	Units
ALPHA1P.KLA	Real	0.719	
ALPHAN.RCS	Real	-0.3	
ALPHAN.RPS	Real	-0.415	
ALPHAN.TAS	Real	2	
ALPHAP.RCS	Real	-0.3	
ALPHAP.RPS	Real	-0.415	
ALPHAP.TAS	Real	3.4	
ALPHN.CVT	Real	0.680	
ALPHP.CVT	Real	0.71	
ALN.CVT	Real	6.85×10^{-21}	
ALP.CVT	Real	7.82×10^{-21}	
ALPN.UM	Real	0.68	
ALPP.UM	Real	0.719	
ALTCVT.N	logical	false	
ALTCVT.P	logical	false	
ALT.N.ALPHA	Real	-0.5	
ALT.N.BETA	Real	0.0284	
ALT.N.DELTA	Real	1.6×10^{14}	cm^2/Vs
ALT.N.ETA	Real	1.6×10^{30}	cm^2/Vs
ALT.N.EXP1	Real	-11.6	
ALT.N.EXP2	Real	-2.74	
ALT.N.EXP3	Real	-12.5	
ALT.N.EXP4	Real	0.76	
ALT.N.KTEMP	Real	1.0	
ALT.N.MUBP1	Real	500.0	cm^2/Vs
ALT.N.MUBP2	Real	450.0	cm^2/Vs
ALT.N.MUMIN	Real	40.0	cm^2/Vs
ALT.N.NREF	Real	2.0×10^{17}	cm^{-3}

Parameter	Type	Default	Units
ALT.N.SPB	Real	1.0×10^{6}	cm/s
ALT.N.SPC	Real	1.077×10^{4}	$cm^{5/3}V^{-2/3}s^{-1}$
ALT.N.SPN0	Real	1.0	cm^{-3}
ALT.P.ALPHA	Real	-0.5	
ALT.P.BETA	Real	0.0284	
ALT.P.DELTA	Real	1.6×10^{14}	cm^2/Vs
ALT.P.ETA	Real	1.6×10^{30}	cm^2/Vs
ALT.P.EXP1	Real	-11.6	
ALT.P.EXP2	Real	-2.74	
ALT.P.EXP3	Real	-12.5	
ALT.P.EXP4	Real	0.76	
ALT.P.KTEMP	Real	1.0	
ALT.P.MUBP1	Real	500.0	cm^2/Vs
ALT.P.MUBP2	Real	450.0	cm^2/Vs
ALT.P.MUMIN	Real	40.0	cm^2/Vs
ALT.P.NREF	Real	2.0×10^{17}	cm^{-3}
ALT.P.SPB	Real	1.0×10^{6}	cm/s
ALT.P.SPC	Real	1.077×10^{4}	$cm^{5/3}V^{-2/3}s^{-1}$
ALT.P.SPN0	Real	1.0	cm^{-3}
ALT.SR.N	logical	true	
ALT.SR.P	logical	true	
ALT.SP.N	logical	true	
ALT.SP.P	logical	true	
AN.PLBRCT	Real		
AN.CCS	Real	4.61×10^{17}	cm^{-3}
AN.CVT	Real	2.58	
AN.IIS	Real	4.61×10^{17}	cm^{-3}
AP.IIS	Real	1.0×10^{17}	cm^{-3}

Parameter	Type	Default	Units
ANALYTIC.N	Logical	False	
ANALYTIC.P	Logical	False	
AP.CCS	Real	1.0×10^{17}	cm^{-3}
AP.CVT	Real	2.18	
ARORA.N	Logical	False	
ARORA.P	Logical	False	
ASN.YAMA	Real	1.54×10^{-5}	cm/V
ASP.YAMA	Real	5.35×10^{-5}	cm/V
B.BROOKS	Real	7.63×10^{19}	cm^{-3}
B1N.TAS	Real	1.75	
B1P.TAS	Real	1.5	
B2N.TAS	Real	-0.25	
B2P.TAS	Real	-0.3	
BETAN	Real	2.0	
BETAP	Real	1.0	
BETAN.ARORA	Real	-2.33	
BETAN.CAUG	Real	-2.3	
BETAN.CVT	Real	2.00	
BETAN.RCS	Real	2.1	
BETAN.RPS	Real	-0.505	
BETAN.TAS	Real	2	
BETAP.ARORA	Real	-2.33	
BETAP.CAUG	Real	-2.2	
BETAP.CVT	Real	2.00	
BETAP.RCS	Real	2.1	
BETAP.RPS	Real	-0.505	
BETAP.TAS	Real	1	
BN.CCS	Real	1.52×10^{15}	cm^{-3}

Parameter	Type	Default	Units
BP.CCS	Real	6.25×10^{14}	cm^{-3}
BN.CVT	Real	4.75×10^{7}	cm/(K·s)
BN.IIS	Real	1.52×10^{15}	cm^{-3}
BN.LSM	Real	4.75×10^{7}	$cm^2/(V*s)$
BP.CVT	Real	9.925×10^{6}	cm/(K·s)
BP.IIS	Real	6.25×10^{14}	cm^{-3}
BP.LSM	Real	9.925×10^{6}	$cm^2/(V*s)$
CA.KLA	Real	0.50	
CCSMOB.N	Logical	False	
CCSMOB.P	Logical	False	
CCUTOFF.N	Real	0.1	
CCUTOFF.P	Real	0.1	
CD.KLA	Real	0.21	
CHEN.N	Logical	False	
CHEN.P	Logical	False	
CMIN.PASV.N	Real	$1x10^{-10}$	
CMIN.PASV.P	Real	$1x10^{-10}$	
CONMOB.N	Logical	False	
CONMOB.P	Logical	False	
CVT.N	Logical	False	
CVT.P	Logical	False	
CN.ARORA	Real	1.432×10^{17}	cm^{-3}
CN.CVT	Real	1.74×10^{5}	
CN.LSM	Real	1.74×10^{5}	
CN.UCHIDA	Real	0.78125	$cm^2/Vs\mu m^6$
COULOMB.ACCT	Logical	True	
COULOMB.DONT	Logical	True	
COULOMB.DOP	Logical	True	

Parameter	Type	Default	Units
COULOMB.N	Logical	False	
COULOMB.P	Logical	False	
COULOMB.E0N	Real	2.0×10^{5}	V/cm
COULOMB.E0P	Real	2.0×10^{5}	V/cm
COULOMB.NCT	Real	10^{18}	cm^{-3}
COULOMB.NGAM	Real	2.0	
COULOMB.NKT	Real	1.0	
COULOMB.NMOB	Real	40.0	cm^2/Vs
COULOMB.NNU	Real	1.5	
COULOMB.NP1	Real	1.0	
COULOMB.NP2	Real	0.5	
COULOMB.NTD	Real	10^{18}	cm^{-3}
COULOMB.NTT	Real	10^{11}	cm^{-2}
COULOMB.PCT	Real	10^{18}	cm^{-3}
COULOMB.PGAM	Real	2.0	
COULOMB.PKT	Real	1.0	
COULOMB.PMOB	Real	40.0	cm^2/Vs
COULOMB.PNU	Real	1.5	
COULOMB.PP1	Real	1.0	
COULOMB.PP2	Real	0.5	
COULOMB.PTD	Real	10^{18}	cm^{-3}
COULOMB.PTT	Real	10^{11}	cm^{-2}
CP.ARORA	Real	2.67×10^{17}	cm^{-3}
CP.CVT	Real	8.842×10^{5}	
CP.LSM	Real	8.842×10^{5}	
CP.UCHIDA	Real	0.78125	$cm^2/Vs\mu m^6$
CRN.CVT	Real	9.68×10^{16}	cm^{-3}

Parameter	Type	Default	Units
CRN.LSM	Real	9.68×10^{16}	cm^{-3}
CRP.CVT	Real	2.23×10^{17}	cm^{-3}
CRP.LSM	Real	2.23×10^{17}	cm^{-3}
CSN.CVT	Real	3.43×10^{20}	cm^{-3}
CSN.LSM	Real	3.43×10^{20}	cm^{-3}
CSP.CVT	Real	6.10×10^{20}	cm^{-3}
CSP.LSM	Real	6.10×10^{20}	cm^{-3}
D.CONWELL	Real	1.04×10^{21}	$(cm\times Vs)^{-1}$
DELN.CVT	Real	5.82×10^{14}	V/s
DELP.CVT	Real	2.0546×10^{14}	V/s
DELTAN.CAUG	Real	0.73	
DELTAN.RCS	Real	0.035	
DELTAP.CAUG	Real	0.70	
DELTAP.RCS	Real	0.035	
DEVICE	Character		
DP.CVT	Real	1/3	
DN.CVT	Real	1/3	
DN.LSM	Real	3.58×10^{18}	$cm^2/(V*s)$
DN.TAS	Real	3.2×10^{-9}	
DP.LSM	Real	4.1×10^{15}	$cm^2/(V*s)$
DP.TAS	Real	2.35×10^{-9}	
EON	Real	4.0×10^{3}	V/cm
E1N.SHI	Real	6.3×10^{3}	V/cm
E2N.SHI	Real	0.77×10^{6}	V/cm
EOP	Real	4.0×10^{3}	V/cm
E1P.SHI	Real	8.0×10^{3}	V/cm

Parameter	Type	Default	Units
E2P.SHI	Real	3.9×10^{5}	V/cm
ECDM.N	Logical	False	
ECDM.P	Logical	False	
ECRITN	Real	4.0×10^{3}	V/cm
ECRITP	Real	4.0×10^{3}	V/cm
ECN.MV	Real	6.48×10^{4}	V/cm
ECP.MV	Real	1.87×10^{4}	V/cm
EGLEY.N	Logical	False	
EGLEY.P	Logical	False	
EGLEY.R	Real	2.79	
EN.CVT	Real	1.0	
EP.CVT	Real	1.0	
ESRN.TAS	Real	2.449×10^{7}	
ESRP.TAS	Real	1.0×10^{9}	
ETAN.CVT	Real	0.0767	
ETAP.CVT	Real	0.123	
ETAN	Real		
ETAN.WATT	Real	0.50	
ETAP.WATT	Real	0.33	
EVSATMOD	Real	0	
EX1N.SHI	Real	0.3	
EX1P.SHI	Real	2.9	
EXN1.ARO	Real	-0.57	
EXN1.LSM	Real	0.680	
EXN2.ARO	Real	-2.33	
EXN2.LSM	Real	2.0	
EXN3.ARO	Real	2.546	
EXN3.LSM	Real	2.5	

Parameter	Type	Default	Units
EXN4.LSM	Real	0.125	
EXN8.LSM	Real		
EXP1.ARO	Real	-0.57	
EXP1.LSM	Real	0.71	
EXP2.ARO	Real	-2.33	
EXP2.LSM	Real	2.0	
EXP3.ARO	Real	2.546	
EXP3.LSM	Real	2.2	
EXP4.LSM	Real	0.0317	
EXP8.LSM	Real		
EXP.WATT.N	Logical	False	
EXP.WATT.P	Logical	False	
F.CONWELL	Real	7.452×10^{13}	cm^{-2}
F.CONMUN	Character		
F.CONMUP	Character		
F.ENMUN	Character		
F.ENMUP	Character		
F.LOCALMUN	Character		
F.LOCALMUP	Character		
F.TOFIMUN	Character		
F.TOFIMUP	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.VSATN	Character		
F.VSATP	Character		
FBH.KLA	Real	3.828	
FCUTOFF.N	Real	2	
FCUTOFF.P	Real	2	
FCW.KLA	Real	2.459	

Parameter	Type	Default	Units
FELN.CVT	Real	1.0×10^{50}	$V^2/(cm\cdot s)$
FELP.CVT	Real	1.0×10^{50}	$V^2/(cm\cdot s)$
FLDMOB.N	Logical	False	
FLDMOB.P	Logical	False	
FLDMOB	Integer	0	
FMCT.N	Logical	False	
FMCT.P	Logical	False	
GAMMAN	Real	4.0	
GAMMAP	Real	1.0	
GAMMAN.ARORA	Real	2.546	
GAMMAP.ARORA	Real	2.546	
GAMMAN.CAUG	Real	-3.8	
GAMMAN.RCS	Real	0.4	
GAMMAP.CAUG	Real	-3.7	
GAMMAP.RCS	Real	0.4	
GAMN.CVT	Real	2.5	
GANSAT.N	Logical	False	
GANSAT.P	Logical	False	
GAMP.CVT	Real	2.2	
GN.YAMA	Real	8.8	
GP.YAMA	Real	1.6	
GSURFN	Real	1.0	
GSURFP	Real	1.0	
HOPMOB.N	Logical	False	
HOPMOB.P	Logical	False	
HVSATMOD	Real	0	
KAPPAN.CVT	Real	1.7	
KAPPAP.CVT	Real	0.9	
KLA.N	Logical	False	

Parameter	Type	Default	Units
KLA.P	Logical	False	
INVN.SF	Real	0.75	
INVP.SF	Real	0.75	
MATERIAL	Character		
ME.KLA	Real	1.0	
MH.KLA	Real	1.258	
MIN.NHANCE	Real	0.1	
MIN.PHANCE	Real	0.1	
MMNN.UM	Real	52.2	$cm^2/(V \cdot s)$
MMNP.UM	Real	44.9	$cm^2/(V \cdot s)$
MMXN.UM	Real	1417.0	$cm^2/(V \cdot s)$
MMXP.UM	Real	470.5	$cm^2/(V \cdot s)$
MOBMOD.N	Real	1	
MOBMOD.P	Real	1	
MOD.WATT.N	Logical	False	
MOD.WATT.P	Logical	False	
MREF1N.WATT	Real	481.0	$cm^2/(V \cdot s)$
MREF1P.WATT	Real	92.8	$cm^2/(V \cdot s)$
MREF2N.WATT	Real	591.0	$cm^2/(V \cdot s)$
MREF2P.WATT	Real	124.0	$cm^2/(V \cdot s)$
MREF3N.WATT	Real	1270.0	$cm^2/(V \cdot s)$
MREF3P.WATT	Real	534.0	$cm^2/(V \cdot s)$
MSTI.PHOS	Logical	False	
MTN.ALPHA	Real	0.68	
MTN.BETA	Real	2.0	
MTN.CR	Real	9.68×10^{16}	cm^{-3}
MTN.CS	Real	3.34×10^{20}	cm^{-3}

Parameter	Type	Default	Units
MTN.MAX	Real	1417	$cm^2/(Vs)$
MTN.MIN1	Real	52.2	$cm^2/(Vs)$
MTN.MIN2	Real	52.2	$cm^2/(Vs)$
MTN.MU1	Real	43.4	$cm^2/(Vs)$
MTN.PC	Real	0.0	cm^{-3}
MTP.ALPHA	Real	0.719	
MTP.BETA	Real	2.0	
MTP.CR	Real	2.23×10^{17}	cm^{-3}
MTP.CS	Real	6.1×10^{20}	cm^{-3}
MTP.MAX	Real	470.5	$cm^2/(Vs)$
MTP.MIN1	Real	44.9	$cm^2/(Vs)$
MTP.MIN2	Real	0.0	$cm^2/(Vs)$
MTP.MU1	Real	29.0	$cm^2/(Vs)$
MTP.PC	Real	9.23×10^{16}	cm^{-3}
MU0N.SHI	Real	1430.0	$cm^2/(V\cdot s)$
MU0P.SHI	Real	500.0	$cm^2/(V\cdot s)$
MU1N.ARORA	Real	88.0	$cm^2/(V\cdot s)$
MU1P.ARORA	Real	54.3	$cm^2/(V\cdot s)$
MU2N.ARORA	Real	1252.0	$cm^2/(V\cdot s)$
MU2P.ARORA	Real	407.0	$cm^2/(V\cdot s)$
MU1N.CAUG	Real	55.24	$cm^2/(V\cdot s)$
MU1P.CAUG	Real	49.7	$cm^2/(V\cdot s)$
MU2N.CAUG	Real	1429.23	$cm^2/(V\cdot s)$
MU2P.CAUG	Real	479.37	$cm^2/(V\cdot s)$
MU0N.CVT	Real	52.2	$cm^2/(V\cdot s)$
MU0P.CVT	Real	44.9	$cm^2/(V\cdot s)$

Parameter	Type	Default	Units
MU1N.CVT	Real	43.4	$cm^2/(V \cdot s)$
MU1P.CVT	Real	29.0	$cm^2/(V \cdot s)$
MUBN.TAS	Real	1150	
MUBP.TAS	Real	270	
MUDEG.N	Real	1.0	
MUDEG.P	Real	1.0	
MULN.YAMA	Real	1400.0	$cm^2/(V \cdot s)$
MULP.YAMA	Real	480.0	$cm^2/(V \cdot s)$
MUMAXN.CVT	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.CVT	Real	470.5	$cm^2/(V \cdot s)$
MUMAXN.KLA	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.KLA	Real	470.5.0	$cm^2/(V \cdot s)$
MUMINN.KLA	Real	52.2	$cm^2/(V \cdot s)$
MUMINP.KLA	Real	44.9	$cm^2/(V \cdot s)$
MUN	Real	1000	$cm^2/(V \cdot s)$
MUN.MAX	Real	1429.23	$cm^2/(V \cdot s)$
MUN.MIN	Real	55.24	$cm^2/(V \cdot s)$
MUN0.LSM	Real	52.2	$cm^2/(V \cdot s)$
MUN1.ARO	Real	88.0	$cm^2/(V \cdot s)$
MUN2.ARO	Real	1252.0	$cm^2/(V \cdot s)$
MUN1.LSM	Real	43.4	$cm^2/(V \cdot s)$
MUN2.LSM	Real	1417.0	$cm^2/(V \cdot s)$
MUP	Real	500	$cm^2/(V \cdot s)$
MUP.MAX	Real	479.37	$cm^2/(V \cdot s)$
MUP.MIN	Real	49.7	$cm^2/(V \cdot s)$
MUP0.LSM	Real	44.9	$cm^2/(V \cdot s)$

Parameter	Type	Default	Units
MUP1.ARO	Real	54.3	$cm^2/(V \cdot s)$
MUP1.LSM	Real	29.0	$cm^2/(V \cdot s)$
MUP2.ARO	Real	407.0	$cm^2/(V \cdot s)$
MUP2.LSM	Real	470.5	$cm^2/(V \cdot s)$
N.ANGLE	Real	0.0	Degrees
N.BETA0	Real	1.109 (CANALI) 0.6 (MEINR)	
N.BETAEXP	Real	0.66 (CANALI) 0.01 (MEINR)	
N.BROOKS	Logical	False	
N.CANALI	Logical	False	
N.CONWELL	Logical	False	
N.LCRIT	Real	1.0	cm
N.MASETTI	Logical	False	
N.MEINR	Logical	False	
N.SOTOODEH	Logical	False	
N2N.TAS	Real	1.1×10^{21}	
N2P.TAS	Real	1.4×10^{18}	
N1N.TAS	Real	2.0×10^{19}	
N1P.TAS	Real	8.4×10^{16}	
NAME	Character		
NCRITN.ARORA	Real	1.432×10^{17}	cm^{-3}
NCRITP.ARORA	Real	2.67×10^{17}	cm^{-3}
NCRITN.CAUG	Real	1.072×10^{17}	cm^{-3}
NCRITP.CAUG	Real	1.606×10^{17}	cm^{-3}
NEWCVT.N	Logical	False	
NEWCVT.P	Logical	False	
NHANCE	Logical	False	

Parameter	Type	Default	Units
NREF1N.KLA	Real	9.68×16	cm^{-3}
NREF1P.KLA	Real	2.23×17	cm^{-3}
NREFD.KLA	Real	4.0×20	cm^{-3}
NREFA.KLA	Real	7.2×20	cm^{-3}
NREFN	Real	1.072×10^{17}	cm^{-3}
NREFN.YAMA	Real	3.0×10^{16}	cm^{-3}
NREFP	Real	1.606×10^{17}	cm^{-3}
NREFP.YAMA	Real	4.0×10^{16}	cm^{-3}
NRFN.UM	Real	9.68×10^{16}	cm^{-3}
NRFP.UM	Real	2.23×10^{17}	cm^{-3}
NUN	Real	-2.3	
NUP	Real	-2.2	
OLDSURF.N	Logical	False	
OLDSURF.P	Logical	False	
OXLEFTN	Real		μm
OXLEFTP	Real		μm
OXRIGHTN	Real		μm
OXRIGHTP	Real		μm
OXBOTTOMN	Real		μm
OXBOTTOMP	Real		μm
OZGUR.N	Logical	False	
OZGUR.P	Logical	False	
P.ANGLE	Real	0.0	Degrees
P.BETA0	Real	1.213 (CANALI) 0.6 (MEINR)	
P.BETAEXP	Real	0.17 (CANALI) 0.01 (MEINR)	
P.BROOKS	Logical	False	
P.CANALI	Logical	False	

Parameter	Type	Default	Units
P.CONWELL	Logical	False	
P.LCRIT	Real	1.0	cm
P.MASETTI	Logical	False	
P.MEINR	Logical	False	
P.SOTOODEH	Logical	False	
P1N.SHI	Real	0.28	
P1P.SHI	Real	0.3	
P2N.SHI	Real	2.9	
P2P.SHI	Real	1.0	
P1N.TAS	Real	0.09	
P1P.TAS	Real	0.334	
P2N.TAS	Real	4.53×10^{-8}	
P2P.TAS	Real	3.14×10^{-7}	
PASVEER.N	Logical	False	
PASVEER.P	Logical	False	
PC.LSM	Real	9.23×10^{16}	cm^{-3}
PCN.CVT	Real	0.0	cm^{-3}
PCP.CVT	Real	0.23×10^{16}	cm^{-3}
PHANCE	Logical	False	
PFMOB.N	Logical	False	
PFMOB.P	Logical	False	
PRINT	Logical	False	
R1.KLA	Real	0.7643	
R2.KLA	Real	2.2999	
R3.KLA	Real	6.5502	
R4.KLA	Real	2.3670	
R5.KLA	Real	-0.8552	
R6.KLA	Real	0.6478	
RCS.N	Logical	False	

Parameter	Type	Default	Units
RCS.P	Logical	False	
RPS.N	Logical	False	
RPS.P	Logical	False	
RN.TAS	Real	2	
RP.TAS	Real	3	
REGION	Integer		
S1.KLA	Real	0.89233	
S2.KLA	Real	0.41372	
S3.KLA	Real	0.19778	
S4.KLA	Real	0.28227	
S5.KLA	Real	0.005978	
S6.KLA	Real	1.80618	
S7.KLA	Real	0.72169	
SCHWARZ.N	Logical	False	
SCHWARZ.P	Logical	False	
SHI.N	Logical	False	
SHI.P	Logical	False	
SIG.PASV.N	Logical	False	
SIG.PASV.P	Logical	False	
SN.YAMA	Real	350	
SP.YAMA	Real	81.0	
STRUCTURE	Character		
SURFMOB.N	Logical	False	
SURFMOB.P	Logical	False	
TASCH.N	Logical	False	
TASCH.P	Logical	False	
TAUN.CVT	Real	0.125	
TAUP.CVT	Real	0.0317	
TETN.UM	Real	-2.285	

Parameter	Type	Default	Units
TETP.UM	Real	-2.247	
THETAN.FLD	Real	0.8	
THETAP.FLD	Real	0.8	
THETAN.KLA	Real	2.285	
THETAN.SHI	Real	2.285	
THETAP.KLA	Real	2.247	
THETAP.SHI	Real	2.247	
TMUBN.TAS	Real	2.5	
TMUBP.TAS	Real	1.4	
TMUN	Real	1.5	
TMUP	Real	1.5	
TNOMN.FLD	Real	600.0	K
TNOMP.FLD	Real	600.0	K
TN.UCHIDA	Real	4.0	nm
TP.UCHIDA	Real	4.0	nm
ULN.YAMA	Real	4.9×10^{6}	cm/s
ULP.YAMA	Real	2.928×10^{6}	cm/s
VSATN	Real		cm/s
VSATP	Real		cm/s
VSAT.QFN	Logical	False	
VSAT.QFP	Logical	False	
VSAT.ZAKN	Logical	False	
VSAT.ZAKP	Logical	False	
VSN.YAMA	Real	1.036×10^{7}	cm/s
VSP.YAMA	Real	1.200×10^{7}	cm/s
VTHN.PFMOB	Real		cm/s
VTHP.PFMOB	Real		cm/s
UCHIDA.N	Logical	False	
UCHIDA.P	Logical	False	

Parameter	Type	Default	Units
XIN	Real	-3.8	
XIP	Real	-3.7	
XMINN.WATT	Real	-1.0×32	μm
XMAXN.WATT	Real	1.0×32	μm
YMAXN.WATT	Real	-1.0x32	μm
XMINP.WATT	Real	-1.0×32	μm
XMAXP.WATT	Real	1.0×32	μm
YMAXP.WATT	Real	-1.0×32	μm
YCHARN.WATT	Real	1.0×32	μm
YCHARP.WATT	Real	1.0×32	μm
Z11N.TAS	Real	0.0388	
Z11P.TAS	Real	0.039	
Z22N.TAS	Real	1.73×10^{-5}	
Z22P.TAS	Real	1.51×10^{-5}	

Description

MATERIAL	Specifies which material from Table B-1 should be applied to the MOBILITY statement. If a material is specified, then all regions defined as being composed of that material will be affected.

Note: You can specify the following logical parameters to indicate the material instead of assigning the MATERIAL parameter: SILICON, GAAS, POLYSILI, GERMANIU, SIC, SEMICOND, SIGE, ALGAAS, A-SILICO, DIAMOND, HGCDTE, INAS, INGAAS, INP, S.OXIDE, ZNSE, ZNTE, ALINAS, GAASP, INGAP and MINASP.

DEVICE	Specifies the device in MixedMode simulation (see Chapter 13 "MixedMode: Mixed Circuit and Device Simulator") to be applied to the MOBILITY statement. The synonym for this parameter is STRUCTURE.
NAME	Specifies the name of the region to be applied to the MOBILITY statement. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.
REGION	Specifies the number of the region to be applied to the MOBILITY statement.
PRINT	Acts the same as PRINT on the MODELS statement.
STRUCTURE	This is a synonym for DEVICE.

Mobility Model Flags

ALBRCT.N **ALBRCT.P**	Enable the Albrecht mobility model (see "The Albrecht Model" on page 518).
ALTCVT.N **ALTCVT.P**	Enables or Disables the Alternative CVT inversion layer mobility model.
ALT.SR.N **ALT.SR.P** **ALT.SP.N** **ALT.SP.P**	Enables or Disables the surface roughness term in the ALTCVT.N model. Enables or Disables the surface roughness term in the ALTCVT.P model. Enables or Disables the surface phonon term in the ALTCVT.N model. Enables or Disables the surface phonon term in the ALTCVT.P model.
ANALYTIC.N	Specifies that the analytic concentration dependent model is to be used for electrons (see Equation 3-207).
ANALYTIC.P	Specifies that the analytic concentration dependent model is to be used for holes (see Equation 3-208).
ARORA.N	Specifies that the Arora analytic concentration dependent model is to be used for electrons (see Equation 3-209).
ARORA.P	Specifies that the Arora analytic concentration dependent model is to be used for electrons (see Equation 3-210).
CHEN.N **CHEN.P**	When specified, enable a set of velocity overshoot model parameters corresponding to Chen et. al. [52] for electrons and holes to simulate electron overshoot in GaN.
CONMOB.N	Specifies that doping concentration dependent model to be used for electrons.
CONMOB.P	Specifies that a doping concentration dependent model is to be used for holes.
CCSMOB.N	Specifies that carrier-carrier scattering model is to be used for electrons (see Equations 3-213-3-217).
CCSMOB.P	Specifies that carrier-carrier scattering model is to be used for holes (see Equations 3-213-3-217).
CONMOB.N	Specifies that doping concentration dependent model to be used for electrons.
CONMOB.P	Specifies that a doping concentration dependent model is to be used for holes.
COULOMB.ACCT **COULOMB.DONT** **COULOMB.DOP**	COULOMB.ACCT Enables or Disables negative interface charge term in interface Coulomb scattering model. COULOMB.DONT Enables or Disables positive interface charge term in interface Coulomb scattering model. COULOMB.DOP Enables or Disables channel dopant term in interface Coulomb scattering model.
COULOMB.N	Enables or Disables the interface Coulomb scattering component of Lombardi, Darwish and Alternative CVT models.
COULOMB.P	Specifies that doping concentration dependent model to be used for electrons.
ECDM.N	Enables the correlated Gaussian disorder mobility model for electrons.

ECDM.P	Enables the correlated Gaussian disorder mobility model for holes.
EGLEY.N	Enables the low field strained silicon mobility model for electrons (see Section 3.6.13 “Low-Field Mobility in Strained Silicon”).
EGLEY.P	Enables the low field strained silicon mobility model for holes (see Section 3.6.13 “Low-Field Mobility in Strained Silicon”).
EVSATMOD	Specifies which parallel field dependent mobility model (see Equation 3-323 and Equation 6-67) should be used for electrons. The value of EVSATMOD should be assigned as follows: • 0 - Use the standard saturation model (Equation 3-323). You also can apply the option parameter, **MOBTEM.SIMPL** (see “Carrier Temperature Dependent Mobility” on page 209 for more information). • 1 - Use the negative differential velocity saturation model (Equation 6-67). • 2 - Use a simple velocity limiting model. In most cases, the default value of 0 should be used.
HVSATMOD	Specifies which parallel field dependent mobility model (see Equation 3-324 and Equation 6-68) should be used for holes. The value of HVSATMOD should be assigned as follows: • 0 - Use the standard saturation model (Equation 3-324). • 1 - Use the negative differential velocity saturation model (Equation 6-68). • 2 - Use a simple velocity limiting model. In most cases, the default value of 0 should be used.
EXP.WATT.N	Turns on exponertial modification to Watt’s mobility model for electrons (see Equation 3-321).
EXP.WATT.P	Turns on exponential modification to Watt’s mobility model for holes (see Equation 3-322).
FLDMOB	Specifies transverse field degradation for electrons as follows: 1. No transverse degradation. 2. Use the Watt or Tasch transverse field models depending on the settings of FIELDMOB1 and FIELDMOB2. 3. Use the Yamaguchi transverse field dependent model. 4. Use the CVT transverse field dependent model.
FLDMOB.N	Specifies a lateral electric field dependent model for electrons (see Equation 3-323).
FLDMOB.P	Specifies a lateral electric field dependent model for holes (see Equation 3-324).
FMCT.N	Turns on the Faramand Modified Caughey Thomas model for electrons (see Equation 6-159).

FMCT.P	Turns on the Faramand Modified Caughey Thomas model for holes (see Equation 6-160).
GANSAT.N	Turns on the Nitride Field Dependent model for electrons (see Equation 6-161).
GANSAT.P	Turns on the Nitride Field Dependent model for holes (see Equation 6-162).
KLA.N	Turns on Klaassen's mobility model for electrons (see Equations 3-225 through 3-250).
KLA.P	Turns on Klaassen's mobility model for holes (see Equations 3-225 through 3-250).
MOBMOD.N	Specifies transverse field degradation for electrons as follows: 1. No transverse degradation. 2. Use the Watt or Tasch transverse field models depending on the settings of FIELDMOB1 and FIELDMOB2. 3. Use the Yamaguchi transverse field dependent model. 4. Use the CVT transverse field dependent model.
MOBMOD.P	Specifies transverse field degradation for holes as follows: 1. No transverse degradation. 2. Use the Watt or Tasch transverse field models depending on the settings of FIELDMOB1 and FIELDMOB2. 3. Use the Yamaguchi transverse field dependent model. 4. Use the CVT transverse field dependent model.
MOD.WATT.N	Turns on modified Watt mobility model for electrons (see Equation 3-321).
MOD.WATT.P	Turns on modified Watt Mobility Model for holes (see Equation 3-322).
MSTI.PHOS	Selects the parameter set for Phosphorous doping in Masetti model. For more information about this parameter, see Tables 3-40, 3-41, and 3-42.
N.ANGLE	Specifies angle for application of electron mobility parameters in simulation of anisotropic mobility.
N.CANALI	Specifies that the Canali velocity saturation model is to be used for electrons (see Equations 3-315 through 3-334).
N.CONWELL	Specifies that the Conwell-Weisskopf carrier-carrier scattering model is to be used for electrons (see Equation 3-219).
N.MASETTI	Enables Masetti mobility model for electrons. For more information about this parameter, see Tables 3-40, 3-41, and 3-42.
N.MEINR	Specifies that the Meinerzhagen-Engl hydrodynamic scattering model is to be used for electrons (see Equation 3-343).
N.SOTOODEH **P.SOTOODEH**	Enables Sotoodeh's low-field mobility model for electrons or holes. See Section 6.4.5 "In(1-x)Ga(x)As(y)P(1-y) System".

NEWCVT.N	Specifies that a new surface mobility degradation calculation is used in the CVT mobility model for electrons in place of the original (see "Darwish CVT Model" on page 186).
NEWCVT.P	Specifies that a new surface mobility degradation calculation is used in the CVT mobility model for holes in place of the original (see "Darwish CVT Model" on page 186).
NHANCE	Specifies that low field mobility values will be modified by mobility enhancement values contained in the mesh structure file.
OZGUR.N **OZGUR.P**	When specified, enable a set of velocity overshoot model parameters corresponding to Ozgur et. al. [227] for electrons and holes to simulate electron overshoot in ZnO.
P.ANGLE	Specifies angle for application of hole mobility parameters in simulation of anisotropic mobility.
P.BROOKS	Specifies that the Brooks-Herring carrier-carrier scattering model is to be used for holes (see Equation 3-222).
P.CANALI	Specifies that the Canali velocity saturation model is to be used for holes (see Equations 3-315 through 3-334).
P.CONWELL	Specifies that the Conwell-Weisskopf carrier-carrier scattering model is to be used for holes (see Equation 3-219).
P.MASETTI	Enables Masetti mobility model for holes. For more information about this parameter, see Tables 3-40, 3-41 and 3-42.
P.MEINR	Specifies that the Meinerzhagen-Engl hydrodynamic scattering model is to be used for holes (see Equation 3-344).
PASVEER.N	Enables the generalized Gaussian disorder mobility model for electrons (see Equations 16-67 through 16-85).
PASVEER.P	Enables the generalized Gaussian disorder mobility model for holes (see Equations 16-67 through 16-85).
PHANCE	Specifies that low field hole mobilities will be modified by mobility enhancement factors contained in the mesh structure file.
RCS.N	Enables the remote coulomb scattering mobility model for electrons (see Equation 3-311).
RCS.P	Enables the remote coulomb scattering mobility model for holes (see Equation 3-312).
RPS.N	Enables the remote phonon scattering mobility model for electrons (see Equation 3-313).
RPS.P	Enables the remote phonon scattering mobility model for holes (see Equation 3-314).

SURFMOB.N	Invokes the effective field based surface mobility model for electrons (see Equation 3-317).
SURFMOB.P	Invokes the effective field based surface mobility model for holes (see Equation 3-318).
SCHWARZ.N	Specifies the use of transverse electric field-dependent mobility models for electrons. See **TFLDMB1** or **SCHWARZ** in the **MODELS** statement.
SCHWARZ.P	Specifies the use of transverse electric field-dependent mobility models for holes. See **TFLDMB1** or **SCHWARZ** in the **MODELS** statement.
SHI.N	Turns on Shirahata's Mobility Model for electrons (see Equation 3-309).
SHI.P	Turns on Shirahata's Mobility Model for holes (see Equation 3-310).
TASCH.N	Specifies a transverse electric field dependent mobility model for electrons based on Tasch [291,292] (see Equations 3-278 through 3-308).
TASCH.P	Specifies a transverse electric field dependent mobility model for holes based on Tasch [291,292] (see Equations 3-278 through 3-308).
VSAT.QFN	Uses the gradient of electron quasi-fermi level as the driver for the FLDMOB.N model.
VSAT.QFP	Uses the gradient of hole quasi-fermi level as the driver for the FLDMOB.P model.
VSAT.ZAKN	Uses the square root of the product of field and gradient of electron quasi-fermi level as the driver for the FLDMOB.N model.
VSAT.ZAKP	Uses the square root of the product of field and gradient of hole quasi-fermi level as the driver for the FLDMOB.P model.

Temperature Dependent Low Field Mobility Parameters

For information about these parameters, see Table 3-36.

Albrecht Model Parameters

For information about these parameters, see "The Albrecht Model" on page 518.

Alternative CVT model Transverse Field Dependent Model Parameters

For more information about these parameters, see Table 3-59, Table 3-60 and Table 3-61.

Caughey-Thomas Concentration Dependent Model Parameters

For more information about these parameters, see Table 3-38.

Arora Concentration Dependent Mobility Model Parameters

For more information about these parameters, see Table 3-39.

Brooks Model Parameters

For more information about these parameters, see Table 3-45.

Canali Model Parameters

For more information about these parameters, see Table 3-70.

Carrier-Carrier Scattering Model Parameters

For more information about these parameters, see Table 3-43.

Conwell Model Parameters

For more information about these parameters, see Table 3-44.

Masetti Model Parameters

For more information about these parameters, see Tables 3-40, 3-41, and 3-42.

Meinerzhagen-Engl Model Parameters

For more information about these parameters, see Table 3-73.

Parallel Field Dependent Model Parameters

For more information about these parameters, see Table 3-70, Tables 6-10, and 6-17.

Perpendicular Field Dependent Model Parameters

For more information about these parameters, see Table 3-67.

Yamaguchi Transverse Field Dependent Model Parameters

For more information about these parameters, see Table 3-62.

CVT Transverse Field Dependent Model Parameters

For more information about these parameters, see Table 3-54.

Darwish CVT Transverse Field Dependent Model Parameters

For more information about these parameters, see Table 3-55.

Interface Coulomb scattering mobility component parameters

For more information about these parameters, see Table 3-58

Watt Effective Transverse Field Depenedent Model Parameters

For more information about these parameters, see Table 3-68.

Tasch Mobility Model Parameters

For more information about these parameters, see Table 3-63.

Klaassen's Mobility Model Parameters

For more information about these parameters, see Tables 3-47 and 3-52.

The Modified Watt Mobility Model Parameters

For more information about these parameters, see Table 3-69.

Shirahata's Mobility Model Parameters

For more information about these parameters, see Table 3-64.

Uchida's Mobility Model Parameters

For more information about these parameters, see Table 3-53.

Remote Coulomb Scattering Mobility Model Parameters

For more information about these parameters, see Table 3-65.

Remote Phonon Scattering Mobility Model Parameters

For more information about these parameters, see Table 3-66.

Albrecht Mobility Model Parameters

For more information about these parameters, see Table 6-32.

Faramand Modified Caughey Thomas Mobility Model Parameters

For more information about these parameters, see Table 6-33.

Nitride Field Dependent Mobility Model Parameters

For more information about these parameters, see Table 6-36.

Poole-Frenkel Mobility Model Parameters

For more information about these parameters, Table 16-7.

Generalized Gaussian Disorder Mobility Model Parameters

For more information about these parameters, see Table 16-11.

SCHWARZ and TASCH Transverse Field Dependent Model Parameters

ACCN.SF	Specifies the accumulation saturation factor which describes the ratio of the electron concentration in the accumulation layer before and after bending of conductivity and valence bands for electron mobility.
ACCP.SF	Specifies the accumulation saturation factor which describes the ratio of the hole concentration in the accumulation layer before and after bending of conductivity and valence bands for hole mobility.
INVN.SF	Specifies the inversion saturation factor which describes the ratio of the electron concentration in the inversion layer before and after the bending of conductivity and valence bands for electron mobility.
INVP.SF	Specifies the inversion saturation factor which describes the ratio of the hole concentration in the inversion layer before and after the bending of conductivity and valence bands for hole mobility.
OXBOTTOMN	Specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for electron mobility.
OXBOTTOMP	Specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for hole mobility.
OXLEFTN	Specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for electron mobility.
OXLEFTP	Specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for hole mobility.
OXRIGHTN	Specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for electron mobility.
OXRIGHTP	Specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for hole mobility.

Miscellaneous Mobility Model Parameters

EGLEY.R	Specifies the ratio of unstrained light hole mobility model to the unstrained net mobility.
MIN.NHANCE	Specifies the minimum electron mobility after strain enhancement relative to the unstrained value (i.e., this is the minimum of the ratio of the strain enhanced mobility to the enhanced mobility).
MIN.PHANCE	Specifies the minimum hole mobility after strain enhancement relative to the unstrained value (i.e., this is the minimum of the ratio of the enhanced mobility to the enhanced mobility).

Interpreter Functions

F.CONMUN	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition and doping dependent electron mobility model.
F.CONMUP	Specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition and doping dependent hole mobility model.
F.ENMUN	Specifies a C-Interpreter file for the electron mobility as a function of Electron Temperature and perpendicular field as well as other choice variables. The function itself is named endepmun and only applies to Atlas 2D.
F.ENMUP	Specifies a C-Interpreter file for the hole mobility as a function of Hole Temperature and perpendicular field as well as other choice variables. The function itself is named endepmup and only applies to Atlas2d.
F.LOCALMUN	Specifies the name of a file containing a C-Interpreter function for the specification of electron mobility, which can depend explicitly on position.
F.LOCALMUP	Specifies the name of a file containing a C-Interpreter function for the specification of Hole mobility, which can depend explicitly on position.
F.TOFIMUN	Specifies the name of a file containing a C-Interpreter function for the specification of electron mobility, which can depend on both parallel and perpendicular electric field.
F.TOFIMUP	Specifies the name of a file containing a C-Interpreter function for the specification of hole mobility, which can depend on both parallel and perpendicular electric field.
F.MUNSAT	Specifies the name of a file containing a C-Interpreter function for the specification of parallel field dependent electron mobility model for velocity saturation.

F.MUPSAT	Specifies the name of a file containing a C-Interpreter function for the specification of parallel field dependent hole mobility model for velocity saturation.
F.VSATN	Specifies the name of a file containing a C-Interpreter function for the specification of temperature and composition dependent electron saturation velocity models.
F.VSATP	Specifies the name of a file containing a C-Interpreter function for the specification of temperature and composition dependent electron saturation velocity models.

Mobility Model Aliases

Aliases are available for certain parameters on the **MOBILITY** statement. Table 22-2 lists these aliases.

Table 22-2 Mobility Model Aliases

Parameter	Alias	Parameter	Alias
MU1N.CAUG	MUN.MIN	THETAN.KLA	TETN.UM*
MU1P.CAUG	MUP.MIN	THETAP.KLA	TETP.UM*
MU2N.CAUG	MUN.MAX	MUMAXN.KLA	MMXN.UM
MU2P.CAUG	MUP.MAX	MUMAXP.KLA	MMXP.UM
BETAN.CAUG	NUN	MUMINN.KLA	MMNN.UM
BETAP.CAUG	NUP	MUMINP.KLA	MMNP.UM
DELTAN.CAUG	ALPHAN	NREF1N.KLA	NRFN.UM
DELTAP.CAUG	ALPHAP	NREF1P.KLA	NRFP.UM
GAMMAN.CAUG	XIN	ALPHA1N.KLA	ALPN.UM
GAMMAP.CAUG	XIP	ALPHA1P.KLA	ALPP.UM
NCRITN.CAUG	NREFN	BN.CVT	BN.LSM
NCRITP.CAUG	NREFP	BP.CVT	BP.LSM
MU1N.ARORA	MUN1.ARO	CN.CVT	CN.LSM
MU1P.ARORA	MUP1.ARO	CP.CVT	CP.LSM
MU2N.ARORA	MUN2.ARO	TAUN.CVT	EXN4.LSM
MU2P.ARORA	MUP2.ARO	TAUP.CVT	EXP4.LSM
ALPHAN.ARORA	EXN1.ARO	MU0N.CVT	MUN0.LSM
ALPHAP.ARORA	EXP1.ARO	MU0P.CVT	MUP0.LSM
BETAN.ARORA	EXN2.ARO	MU1N.CVT	MUN1.LSM
BETAP.CAUG	EXP2.ARO	MU1P.CVT	MUP1.LSM

Table 22-2 Mobility Model Aliases			
Parameter	**Alias**	**Parameter**	**Alias**
GAMMAN.ARORA	EXN3.ARO	CRN.CVT	CRN.LSM
GAMMAP.ARORA	EXP3.ARO	CRP.CVT	CRP.LSM
NCRITN.ARORA	CN.ARORA	CSN.CVT	CSN.LSM
NCRITP.ARORA	CP.ARORA	CSP.CVT	CSP.LSM
AN.CCS	AN.IIS	ALPHAN.CVT	EXN1.LSM
AP.CCS	AP.IIS	ALPHAP.CVT	EXP1.LSM
BN.CCS	BN.IIS	BETAN.CVT	EXN2.LSM
BP.CCS	BP.IIS	BETAP.CVT	EXP2.LSM
P1N.SHI	EX1N.SHI	MUMAXN.CVT	MUN2.LSM
P1P.SHI	EX1P.SHI	MUMAXP.CVT	MUP2.LSM
ECRITN	EON	GAMN.CVT	EXN3.LSM
ECRITP	EOP	GAMP.CVT	EXP3.LSM
KN.CVT	EXN8.LSM	DELN.CVT	DN.LSM
KP.CVT	EXP8.LSM	DELP.CVT	DP.LSM
PCP.CVT	PC.LSM		

Note: These aliases are negated with respect to the primary parameter.

Modified Watt Model Example

The following example sets the Modified Watt Surface mobility model for MOSFETs. The **MOBILITY** statement is used to set the models and to specify the value of the depth of action of the Modified Watt model.

```
MODELS CONMOB FLDMOB SRH  MIN.SURF  PRINT
MOBILITY WATT.N MOD.WATT.N YMAXN.WATT=0.01
```

22.37 MODELS

MODELS specifies model flags to indicate the inclusion of various physical mechanisms, models, and other parameters such as the global temperature for the simulation.

Syntax

```
MODELS <mf> <gp> <mdp>
```

Parameter	Type	Default	Units
2DXY.SCHRO	Logical	False	
A.BBT.SCHENK	Real	8.977×10^{20}	$cm^2s^{-1}V^{-2}$
A.TEMP	Logical	False	
A0.TNLC	Real	0.0	degrees
ACC.SF	Real	0.87	
ADACHI	Logical	False	
ALBRCT	Logical	False	
ALL	Logical	True	
ALN1	Real	-0.160	
ALN2	Real	-2.17	
ALN3	Real	1.07	
ALP1	Real	-0.296	
ALP2	Real	-1.62	
ALP3	Real	1.02	
ALPHA.DOS	Logical	False	
ALT.SCHRO	Logical	False	
ANALYTIC	Logical	False	
AR.MU1N	Real	88	
AR.MU2N	Real	1252	
AR.MU1P	Real	54.3	
AR.MU2P	Real	407	
ARORA	Logical	False	
ASUB	Real	1.0	Å
AUGER	Logical	False	
AUGGEN	Logical	False	

Parameter	Type	Default	Units
AUTOBBT	Logical	False	
AUTOBBT2	Logical	False	
B.BBT.SCHENK	Real	2.14667×10^{7}	$eV^{(-3/2)}V\cdot cm^{-1}$
B.DORT	Real	1.0	
B.ELECTRONS	Real	2	
B.HOLES	Real	1	
BARLN	Real	2.59×10^{-4}	$(V\cdot cm)^{0.5}$
BARLP	Real	2.59×10^{-4}	$(V\cdot cm)^{0.5}$
BB.A	Real	4.0×10^{14}	$cm^{-1/2}V^{-5/2}s^{-1}$
BB.B	Real	1.97×10^{7}	V/cm
BB.GAMMA	Real	2.5	
BBT.A_KANE	Real	3.5×10^{21}	$eV^{1/2}$/cm-s-V^{2}
BBT.B_KANE	Real	2.25×10^{7}	V/cm-$eV^{3/2}$
BBT.ALPHA	Real	0	
BBT.DEHURKX	Logical	False	
BBT.DJHURKX	Logical	False	
BBT.FORWARD	Logical	False	
BBT.GAMMA	Real	2.5	
BBT.HURKX	Logical	False	
BBT.KANE	Logical	False	
BBT.KL	Logical	False	
BBT.NLDERIVS	Logical	False	
BBT.NONLOCAL	Logical	False	
BBT.SHURKX	Logical	False	
BBT.STD	Logical	False	
BBT1	Logical	False	
BBT2	Logical	False	
BBT3	Logical	False	

Parameter	Type	Default	Units
BC.QUANT	Logical	False	
BGN	Logical	False	
BEAM.LID	Logical	False	
BGN2	Logical	False	
BGN.ALAMO	Logical	False	
BGN.BENNETT	Logical	False	
BGN.KLAASSEN	Logical	False	
BGN.LIND	Logical	False	
BGN.SCHENK	Logical	False	
BGN.SLOTBOOM	Logical	False	
BH.FNONOS	Real	3.07	eV
BIPOLAR	Logical	False	
BIPOLAR2	Logical	False	
BOLTZMANN	Logical	True	
BOUND.TRAP	Logical	False	
BQP.N	Logical	False	
BQP.P	Logical	False	
BQP.NALPHA	Real	0.5	
BQP.NGAMMA	Real	1.2	
BQP.NEUMANN	Logical	True	
BQP.PALPHA	Real	0.5	
BQP.PGAMMA	Real	1.0	
BQP.QDIR	Integer	2	
BROOKS	Logical	False	
BTBT	Logical	False	
BTE.ALTDIS	Logical	False	
BTE.EQUIBCS	Logical	False	
BTE.HGEN	Logical	False	
BTE.PP.E	Logical	False	

Parameter	Type	Default	Units
BTE.PP.H	Logical	False	
BTE.VERBOSE	Logical	False	
BX	Real	0.0	Tesla
BY	Real	0.0	Tesla
BZ	Real	0.0	Tesla
C0	Real	2.5×10^{-10}	
CALC.FERMI	Logical	False	
CALC.PARTITION	Logical	False	
CALC.STRAIN	Logical	False	
CAPT.AUGER	Logical	False	
CAPT.SRH	Logical	False	
CAPT.TRAP	Logical	False	
CARRIERS	Real		
CAVITY.LENGTH	Real		μm
CCS.EA	Real	4.61×10^{17}	
CCS.EB	Real	1.52×10^{15}	
CCS.HA	Real	1.0×10^{17}	
CCS.HB	Real	6.25×10^{14}	
CCSMOB	Logical	False	
CCSNEW	Logical	False	
CCSSURF	Logical	False	
CDL	Logical	False	
CGATE.N	Real	5.0e4	
CGATE.P	Real	3.0e8	
CHI.HOLES	Real	4.6×10^{4}	
CHIA	Real	3×10^{5}	
CHIB	Real	5.0×10^{4}	
CHUANG	Logical	False	

Parameter	Type	Default	Units
CONCDL	Logical	False	
CONMOB	Logical	False	
CONSRH	Logical	False	
CONSRH.LAW	Logical	False	
CONWELL	Logical	False	
CUBIC35	Logical	False	See Table 6-12
CVT	Logical	False	
D.DORT	Real	2.5×10^{-6}	
DEC.C1	Real	0.0	eV
DEC.C2	Real	0.0	eV
DEC.C3	Real	0.0	eV
DEC.ISO	Real	0.0	eV
DEC.D2	Real	0.0	eV
DEC.D4	Real	0.0	eV
DEV.HH	Real	0.0	eV
DEV.LH	Real	0.0	eV
DEV.ISO	Real	0.0	eV
DEV.SO	Real	0.0	eV
DEVDEG.A	Logical	False	
DEVDEG.B	Logical	False	
DEVDEG.E	Logical	False	
DEVDEG.H	Logical	False	
DEVDEG.KN	Logical	False	
DEVDEG.PL	Logical	False	
DEVDEG.RD	Logical	False	
DEVICE	Character		
DG.N	Logical	False	
DG.P	Logical	False	
DGN.GAMMA	Real	3.6	

Parameter	Type	Default	Units
DGP.GAMMA	Real	3.6	
DGLOG	Logical	True	
DGROOT	Logical	False	
DRIFT.DIFF	Logical	True	
DT.CBET	Logical	False	
DT.CUR	Logical	False	
DT.METH	Real	1	
DT.VBET	Logical	False	
DT.VBHT	Logical	False	
DTPP	Logical	False	
E.BENDING	Logical	False	
E.TAUR.GONZALEZ	Logical	False	
E.TAUR.VAR	Logical	False	
E0_TCOEF	Real		
E0.TCOEF	Real		
ECRIT	Real	4000.0	V/cm
EF.TOL	Real	10×10^{-10}	
E.FGCGTUN	Logical	False	
EIGENS	Integer	15	
ELECTRONS	Logical	True	
ENERGY.STEP	Real	0.0025	eV
ERASE	Logical	False	
E.SC.FGCGTUN	Logical	False	
ESIZE.NEGF	Real	2001	
ET.MODEL	Logical	False	
ETA.FNONOS	Real	1.0	
ETAN	Real	0.50	
ETAP	Real	0.33	
ETA.NEGF	Real	0.0066	eV

Parameter	Type	Default	Units
EQUIL.NEGF	Logical	False	
EXCITONS	Logical	False	
EVSATMOD	Integer	0	
F.AE	Real	1.82×10^{-7}	$CV^{-2}s^{-1}$
F.BE	Real	1.90×10^{8}	V/cm
F.AH	Real	1.82×10^{-7}	$CV^{-2}s^{-1}$
F.BH	Real	1.90×10^{8}	V/cm
F.EFEMIS	Character		
F.HFEMIS	Character		
F.GENSINGLET	Character		
F.GENTRIPLET	Character		
F.KEXCITON	Character		
F.KSINGLET	Character		
F.KTRIPLET	Character		
F.KSN	Character		
F.KSP	Character		
F.PCM	Character		
F.PDN	Character		
F.PDP	Character		
F.TRAP.COULOMBIC	Character		
F.TRAP.TUNNEL	Character		
FAST.EIG	Logical	False	
FC.ACCURACY	Real	10^{-6}	
FERMIDIRAC	Logical	False	
FERRO	Logical	False	
FETIS	Logical	False	
FG.RATIO	Real		
FILE.LID	Character		
FIX.BQN	Logical	False	

Parameter	Type	Default	Units
FIX.BQP	Logical	False	
FIXED.FERMI	Logical	False	
FLDMOB	Logical	False	
FN.CUR	Logical	False	
FNHOLES	Logical	False	
FNHPP	Logical	False	
FNONOS	Logical	False	
FNORD	Logical	False	
FNPP	Logical	False	
FREQ.CONV	Logical	False	
GEIGER	Logical	False	
GEIGER.MINFIELD	Real	1.0	V/cm
G.EMPIRICAL	Logical	False	
G.STANDARD	Logical	False	
GAINMOD	Logical	0	
GANFET	Logical	False	
GATE1	Logical	False	
GATE2	Logical	False	
GENERATE.LID	Real	0	$cm^{-3}s^{-1}$
GIAR	Logical	False	
GR.HEAT	Logical	False	
HANSCHQM	Logical	False	
H.ATOM	Logical	False	
H.BENDING	Logical	False	
H.FGCGTUN	Logical	False	
H.MOLE	Logical	False	
H.SC.FGCGTUN	Logical	False	
H.TAUR.GONZALEZ	Logical	False	
H.TAUR.VAR	Logical	False	

Parameter	Type	Default	Units
HCTE	Logical	False	
HCTE.EL	Logical	False	
HCTE.HO	Logical	False	
HEAT.FULL	Logical	False	
HEAT.PETHOM	Logical	False	
HEI	Logical	False	
HGENFLD	Real	1.0	V/cm
HHI	Logical	False	
HNSAUG	Logical	False	
HOLE	Logical	False	
HOLSTEIN	Logical	False	
HOPMOB	Logical	False	
HVSATMOD	Integer	0	
HW.BBT.SCHENK	Real	0.0186	eV
IG.ELINR	Real	6.16×10^{-6}	cm
IG.HLINR	Real	6.16×10^{-6}	cm
IG.ELINF	Real	9.2×10^{-7}	cm
IG.HLINF	Real	9.2×10^{-7}	cm
IG.EBETA	Real	2.59×10^{-4}	$(V.cm)^{0.5}$
IG.HBETA	Real	2.59×10^{-4}	$(V.cm)^{0.5}$
IG.EETA	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
IG.HETA	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
IG.EEOX	Real	9.0×10^{4}	V/cm
IG.HEOX	Real	9.0×10^{4}	V/cm
IG.EB0	Real	3.2	V
IG.HB0	Real	4.0	V
IG.LRELE	Real	3.0×10-6	[Q=1] cm
IG.LRELH	Real	2.0×10-6	[Q=1] cm

Parameter	Type	Default	Units
II.NODE	Logical	False	
II.VALDI	Logical	False	
IMPACT	Logical	False	
INC.ION	Logical	False	
INCOMPLETE	Logical	False	
INFINITY	Real	0.001	
IONIZ	Logical	False	
INTERSUB.SPONT	Logical	False	
INV.SF	Real	0.75	
ISHIKAWA	Logical	False	
ITAT.HJ.EL	Logical	False	
ITAT.HJ.HO	Logical	False	
ITAT.PP.EL	Logical	False	
ITAT.PP.HO	Logical	False	
ITAT.SC.EL	Logical	False	
ITAT.SC.HO	Logical	False	
JOULE.HEAT	Logical	True	
JTAT	Logical	False	
JTAT.DENSITY	Real	0.0	cm^{-3}
JTAT.LEVEL	Real	0.0	eV
JTAT.M2	Real	10^{-20}	V^2cm^3
K.P	Logical	False	
KAUGCN	Real		
KAUGCP	Real		
KAUGDN	Real		
KAUGDP	Real		
KC.HYDROGEN	Logical	False	
KERR.AUG	Logical	False	
KLA	Logical	False	

Parameter	Type	Default	Units
KLAAUG	Logical	False	
KLASRH	Logical	False	
KOSTER	Logical	False	
KP.ADAPTIVE	Logical	False	
KP.CHIEMIN	Real	0.0	eV
KP.CHIEMAX	Real	0.0	eV
KP.CHIRES	Real	0.001	eV
KP.CNLDOS	Logical	False	
KP.COPYPARAMS	Logical	False	
KP.CV2	Logical	False	
KP.CV3	Logical	False	
KP.CV4	Logical	False	
KP.CV6	Logical	False	
KP.CVCOUPLED	Logical	False	
KP.DOS.CRANGE	Real	0.0	eV
KP.DOSTOL	Real	0.01	
KP.DOSRES	Real	0.001	eV
KP.EGAP	Real	0	eV
KP.FEM.HERMITE	Logical	False	
KP.FEM.INTERFACE	Logical	False	
KP.FEM.LAGRANGE	Logical	True	
KP.GAUSSIAN	Logical	False	
KP.INTERSUB	Logical	False	
KP.KMAX	Real	1.0	1/nm
KP.LORENTZ	Logical	False	
KP.NUMKPT	Integer	16	
KP.SOLVERTOL	Real	0.001	
KP.V2	Logical	False	
KP.V3	Logical	False	

Parameter	Type	Default	Units
KP.V4	Logical	False	
KP.V6	Logical	False	
KP.VBMAX	Real	0	eV
KP.VNLDOS	Logical	False	
KP.DOS.VRANGE	Real	0.0	eV
KSRHCN	Real		
KSRHCP	Real		
KSRHGN	Real		
KSRHGP	Real		
KSRHTN	Real		
KSRHTP	Real		
KSN	Integer	-1	
KSP	Integer	-1	
L.TEMP	Logical	False	
LAMHN	Real	9.2×10^{-7}	cm
LAMHP	Real	9.2×10^{-7}	cm
LAMRN	Real	6.16×10^{-6}	cm
LAMRP	Real	6.16×10^{-6}	cm
LAS.ABSOR_SAT	Logical	False	
LAS.ABSORPTION	Logical	False	
LAS.EFINAL	Real		eV
LAS.EINIT	Real		eV
LAS.ESEP	Real		eV
LAS.ETRANS	Logical	False	
LAS.FCARRIER	Logical	False	
LAS.GAIN_SAT	Logical	False	
LAS.ITMAX	Integer	30	
LAS.LOSSES	Real	0	cm^{-1}
LAS.MIRROR	Real	90	%

Parameter	Type	Default	Units
LAS.MULTISAVE	Logical	False	
LAS.OMEGA	Real	2.16×10^{15}	1/sec
LAS.NMODE	Integer	1	
LAS.NX	Real		
LAS.NY	Real		
LAS.REFLECT	Logical	False	
LAS.RF	Real		
LAS.RR	Real		
LAS.SIN	Real	1.0×10^{4}	cm^{-1}
LAS.SPECSAVE	Integer	1	
LAS.SPONTANEOUS	Real		
LAS.TIMERATE	Logical	True	
LAS.TOLER	Real	0.01	
LAS.TRANS_ENERGY	Real		
LAS.XMAX	Real		
LAS.XMIN	Real		
LAS.YMAX	Real		
LAS.YMIN	Real		
LAS_MAXCH	Real	2.5	
LASER	Logical	False	
LAT.TEMP	Logical	False	
LED	Logical	False	
LI	Logical	False	
LMODE	Logical	False	
LORENTZ	Logical	False	
LSMMOB	Logical	False	
MAGMIN.N	Real	0.0	
MAGMIN.P	Real	0.0	
MASS.TUNNEL	Real	0.25	

Parameter	Type	Default	Units
MASETTI	Logical	False	
MATERIAL	Character		
MIM.ITAT.EL	Logical	False	
MIM.ITAT.HO	Logical	False	
MIM.RTAT	Logical	False	
MIMSLICEPTS	Integer	51	
MIMTUN	Logical	False	
MIMTUN.BBT	Logical	False	
MIMTUN.EL	Logical	False	
MIMTUN.HO	Logical	False	
MIN.SURF	Logical	False	
MOB.INCOMPL	Logical	False	
MOBMOD	Integer		
MOBTEM.SIMPL	Logical	False	
MOS	Logical	False	
MOS2	Logical	False	
MREFN1	Real	481.0	cm^2/Vs
MREFN2	Real	591.0	cm^2/Vs
MREFN3	Real	1270.0	cm^2/Vs
MREFP1	Real	92.8	cm^2/Vs
MREFP2	Real	124.	cm^2/Vs
MREFP3	Real	534.	cm^2/Vs
N.ANISO	Logical	False	
MS.DISS	Integer	3	
N.CONCANNON	Logical	False	
N.DORT	Logical	False	
N.HOTSONOS	Logical	False	
N.KSI.VAR	Logical	False	

Parameter	Type	Default	Units
N.KSI_VAR	Logical	False	
N.NEGF_PL	Logical	False	
N.NEGF_PL1D	Logical	False	
N.QUANTUM	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
NACC.BETA	Real	8.45×10^{-8}	
NACC.DORT	Logical	False	
NACC.EC	Real	0.0	V/cm
NACC.N	Real	2.0×10^{19}	cm^{-3}
NACC.EXP	Real	2/3	
N1.GIAR	Real	1.0	
N2.GIAR	Real	1.0	
NE1.GIAR	Real	1.0	
NE2.GIAR	Real	1.0	
NE.TNLC	Real	1.0	
NO1.GIAR	Real	1.0	
NO2.GIAR	Real	1.0	
NO.TNLC	Real	1.0	
NEARFLG	Logical	False	
NEGF_CMS	Logical	False	
NEGF_MS	Logical	False	
NEGF_UMS	Logical	False	
NEP.BTE	Real	100	
NEW.EIG	Logical	False	
NEW.SCHRO	Logical	False	
NEWIMPACT	Logical	False	
NI.FERMI	Logical	False	
NINV.BETA	Real	5.92×10^{-8}	

Parameter	Type	Default	Units
NINV.DORT	Logical	False	
NINV.EC	Real	0.0	V/cm
NINV.EXP	Real	2/3	
NPRED.NEGF	Real	7	
NS.DISS	Integer	30	
NSUB.BTE	Real	4	
NT.FNONOS	Real	0.0	μm
NUM.BAND	Real		
NUM.DIRECT	Real		
NSPECIES	Integer	0	
NUMBER	Integer	0	
NUMCARR	Real		
OLDIMPACT	Logical	False	
OPTR	Logical	False	
ORTH.SCHRO	Real	10^{-3}	
OX.BOTTOM	Real	0.0	μm
OX.LEFT	Real		μm
OX.MARGIN	Real	0.0003	μm
OX.POISSON	Logical	False	
OX.RIGHT	Real		μm
OX.SCHRO	Logical	False	
P.CONCANNON	Logical	False	
P.DORT	Logical	False	
P.HOTSONOS	Logical	False	
P.KSI_VAR	Logical	False	
P.KSI.VAR	Logical	False	
P.NEGF_PL	Logical	False	
P.NEGF_PL1D	Logical	False	
P.SCHRODING	Logical	False	

Parameter	Type	Default	Units
P.TEMP	Logical	False	
PACC.BETA	Real	4.97×10^{-8}	
PACC.DORT	Logical	False	
PACC.EC	Real	0.0	V/cm
PACC.N	Real	3.6×10^{19}	cm^{-3}
PACC.EXP	Real	2/3	
PASSLER	Logical	False	
PATH.N	Real	3.4×10^{7}	microns
PATH.P	Real	2.38×10^{-7}	microns
PATRIN	Logical	False	
PCH.INS	Logical	False	
PCM	Logical	False	
PCM.LOG	Logical	True	
PEFF.N	Real	3.2	eV
PEFF.P	Real	4.8	eV
PF.NITRIDE	Logical	False	
PFMOB	Logical	False	
PFREQ.CONV	Logical	False	
PHONONDRAG	Logical	False	
PHOTON.ENERGY	Real		eV
PHUMOB	Logical	False	
PIC.AUG	Logical	False	
PIEZOELECTRIC	Logical	False	
PIEZO.SCALE	Real	1.0	
PINV.BETA	Real	6.1×10^{-8}	
PINV.DORT	Logical	False	
PINV.EC	Real	10^{5}	V/cm
PINV.EXP	Real	2/3	
POISSON	Logical	False	

Parameter	Type	Default	Units
POLAR.SCALE	Real	1.0	
POLARIZATION	Logical	False	
POST.SCHRO	Logical	False	
PRINT	Logical	False	
PRINT.REGION	Integer		
PRINT.MATERIAL	Character		
PRPMOB	Logical	False	
PROGRAM	Logical	False	
PRT.BAND	Logical	False	
PRT.COMP	Logical	False	
PRT.FLAGS	Logical	False	
PRT.IMPACT	Logical	False	
PRT.RECOM	Logical	False	
PRT.THERM	Logical	False	
PSP.SCALE	Real	1.0	
PT.HEAT	Logical	False	
QCRIT.NEGF	Real	3×10^{-3}	
QDOSTOL	Real	0.01	eV
QMINCONC	Real	1.0×10^{-10}	cm^{-3}
QSPEC.EMAX	Real	0	
QSPEC.EMIN	Real	0	
QSPEC.RES	Real	0.001	eV
QT.FILE	Charater		
QTNL.DERIVS	Logical	False	
QTNLSC.BBT	Logical	False	
QTNLSC.EL	Logical	False	
QTNLSC.HO	Logical	False	
QTREGION	Integer		
QTUNN	Logical	False	

Parameter	Type	Default	Units
QTUNN.BBT	Logical	False	
QTUNN.DIR	Real	0	
QTUNN.EL	Logical	False	
QTUNN.HO	Logical	False	
QTUNNSC	Logical	False	
QUANTUM	Logical	False	
QVERBOSITY	Integer	0	
QW.GEOM	Character	1DX	
QWELL	Logical	False	
QWNUM	Integer	-1	
QX.MAX	Real	RHS of device	μm
QX.MIN	Real	LHS of device	μm
QY.MAX	Real	Bottom of device	μm
QY.MIN	Real	Bottom of device	μm
R.ELEC	Real	1.1	
R.HOLE	Real	0.7	
RAJKANAN	Logical	False	
REGION	Integer	0	
RICHTER.AUG	Logical	False	
RTAT.MAJQFL	Logical	False	
RTAT.PP	Logical	False	
RTAT.SC	Logical	False	
S.DISSOC	Logical	False	
SBT	Logical	False	
SCHENK.BBT	Logical	False	
SCHENK.EL	Logical	False	
SCHENK.HO	Logical	False	
SCHENK	Logical	False	
SCHKSC.EL	Logical	False	

Parameter	Type	Default	Units
SCHKSC.HO	Logical	False	
SCHKSC	Logical	False	
SCHSRH	Logical	False	
SCHRO	Logical	False	
SDISS.TYPE	Integer	1	
SHAPEOX	Logical	False	
SHI	Logical	False	
SHIRAMOB	Logical	False	
SHJ.EL	Logical	False	
SHJ.HO	Logical	False	
SHJ.NLDERIVS	Logical	False	
SINGLET	Logical	False	
SIS.EL	Logical	False	
SIS.HO	Logical	False	
SIS.NLDERIVS	Logical	False	
SIS.REVERSE	Logical	False	
SIS.TAT	Logical	False	
SISTAT.TN	Real	1.0e-6	s
SISTAT.TP	Real	1.0e-6	s
SL.GEOM	Character	1DY	
SL.NUMBER	Integer		
SLATT	Logical	False	
SLATT.NUMBER	Integer		
SP.BAND	Logical	True	
SP.CHECK	Logical	False	
SP.DIR	Integer	0	
SP.EXCORR	Logical	False	
SP.FAST	Logical	False	
SP.GEOM	Character	1DX	

Parameter	Type	Default	Units
SP.SMOOTH	Logical	True	
SPEC.NAME	Character		
SPECIES1.A	Integer	0	
SPECIES2.A	Integer	0	
SPECIES3.A	Integer	0	
SPECIES1.Z	Integer	0	
SPECIES2.Z	Integer	0	
SPECIES3.Z	Integer	0	
SPONTANEOUS	Logical	False	
SRH	Logical	False	
SRH.EXPTEMP	Logical	False	
STRESS	Logical	False	
STRUCTURE	Character		
SUBSTRATE	Logical	False	
SURFMOB	Logical	False	
TAT.NLDEPTH	Real	0	eV
TAT.NONLOCAL	Logical	False	
TAT.LOCAL	Logical	False	
TAT.NLCURRS	Logical	False	
TAT.NLDERIVS	Logical	False	
TAT.POISSON	Logical	True	
TAT.RELEI	Logical	False	
TAT.SLICEPTS	Real	12	
TAUMOB	Logical	False	
TAUTEM	Logical	False	
TAYAMAYA	Logical	False	
TEMPERATURE	Real	300	K
TFLDMB1	Logical	False	
TFLDMB2	Logical	False	

Parameter	Type	Default	Units
THETA.N	Real	60.0	degrees
THETA.P	Real	60.0	degrees
TIME.EP	Real	0	s
TIME.LID	Real	0	s
TLU.INDEX	Logical	False	
TNLC	Logical	False	
TR.TNLC	Real	0.0	degrees
TRAP.AUGER	Logical	False	
TRAP.COULOMBIC	Logical	False	
TRAP.HURKX	Logical	False	
TRAP.LID	Logical	False	
TRAP.JTAT	Logical	False	
TRAP.TUNNEL	Logical	False	
TRIPLET	Logical	False	
TSQ.ZAIDMAN	Real	1.1	
TUNLN	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
TUNLP	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
UBGN	Logical	False	
UST	Logical	False	
UNSAT.FERRO	Logical	False	
VALDINOCI	Logical	False	
WELL.CAPT	Logical	False	
WELL.CNBS	Integer	1	
WELL.DEBUG	Logical	False	
WELL.DENERGY	Real	0.01	eV
WELL.FIELD	Logical	True	
WELL.GAIN	Real	1.0	
WELL.INPLANE	Logical	False	
WELL.MARGIN	Real	0.01	μm

Parameter	Type	Default	Units
WELL.NX	Integer	10	
WELL.NY	Integer	10	
WELL.NZ	Integer	10	
WELL.OVERLAP	Real		
WELL.SEPARATE	Logical	False	
WELL.VNBS	Integer	1	
WELL.WBTUNN	Logical	False	
WELL.WWSPONT	Logical	False	
WELL.WWTUNN	Logical	False	
WZ.KP	Logical	False	
YAMAGUCHI	Logical	False	
YAN	Logical	False	
ZAIDMAN	Logical	False	
ZAMDMER	Logical	False	
ZB.KP	Logical	False	
ZB.ONE	Logical	False	
ZB.TWO	Logical	False	

Description

mf	This is one or more of the model flags described on the next page.
gp	This is one or more of the general parameters described in the “General Parameters” on page 1408. These parameters are used to specify general information used during the simulation.
mdp	This is one or more of the model dependent parameters described in the “Model Dependent Parameters” on page 1411.

Degradation Model Flags

BTE.ALTDIS	Enables an alternative spatial discretisation for Boltzmann Transport equation.
BTE.PP.E	Enables the Boltzmann transport equation solver for electrons.
BTE.PP.H	Enables the Boltzmann transport equation solver for holes.
BTE.EQUIBCS	For spatial Boltzmann transport equation solver, selects equilibrium boundary conditions instead of homogeneous field solutions as boundary conditions.
BTE.HGEN	Specifies the Boltzmann transport equation is solved in homogeneous mode. \\

BTE.VERBOSE	This causes Boltzmann transport equation solution to be written to a TonyPlot file for every node in the solution domain.
NEP.BTE	Sets the number of major energy planes in Boltzmann transport equation solver.
NSUB.BTE	Sets the number of energy spacings between major energy planes.
DEVDEG.A	This causes the dangling bonds created by the DEVDEG.RD or DEVDEG.KN models to be treated as amphoteric.
DEVDEG.B	Activates the device degradation caused by both the hot electron and hot hole injection currents. HEI and HHI should also be specified in this case.
DEVDEG.E	Activates the device degradation caused by hot electron injection current. HEI should also be specified in this case.
DEVDEG.H	Activates the device degradation caused by hot hole injection current. HHI should also be specified in this case.
DEVDEG.KN	Enables the kinetic degradation model.
DEVDEG.PL	Enables the power law degradation model.
DEVDEG.RD	Enable the Reaction-Diffusion degradation model.
KC.HYDROGEN	Enables the hydrogen diffusion model in kinetic degradation model.
HGENFLD	The field used for the solution of the Boltzmann transport equation in homogeneous mode.

Mobility Model Flags

ALBRCT	Enables the Albrecht mobility model for electrons and holes (see Equation 6-158).
ANALYTIC	Specifies an analytic concentration dependent mobility model for silicon which includes temperature dependence (see Equations 3-207 and 3-208).
ARORA	Specifies an analytic concentration and temperature dependent model according to Arora (see Equations 3-209 and 3-210).
BROOKS	Enables the Brooks-Herring mobility model for both electrons and holes (Equation 3-222).
CCSMOB	Specifies carrier-carrier scattering model according to Dorkel and Leturq (see Equations 3-213-3-217).
CONMOB	Specifies that a concentration dependent mobility model be used for silicon and gallium arsenide. This model is a doping versus mobility table valid for 300K only.
CVT	Specifies that the CVT transverse field dependent mobility model is used for the simulation. The alias for this parameter is **LSMMOB**.
FLDMOB	Specifies a lateral electric field-dependent model (see Equation 3-323 and Equation 6-65). The EVSATMOD parameter may be used to define which field dependent equation is used.

HOPMOB	Specifies that the hopping mobility model will be used for electrons and holes (see Chapter 16 “Organic Display and Organic Solar: Organic Simulators”).
KLA	Specifies that the Klaassen Mobility Model (see Equations 3-225-3-250) will be used for electrons and holes.
LSMMOB	This is an alias for **CVT**.
MASETTI	Enables the Masetti mobility model for electrons and holes (see Equations 3-211 and 3-212).
MIN.SURF	Specifies that the WATT, TASCH, or SHI mobility models should only apply to minority carriers.
MOBMOD	Specifies mobility degradation by longitudinal electric field only (MOBMOD=1), or by both longitudinal and transverse electric fields (MOBMOD=2). When MOBMOD=2, specify the ACC.SF, INV.SF, OX.BOTTOM, OX.LEFT, and OX.RIGHT parameters. This parameter is only used with TFLDMB1 and TFLDMB2.
MOB.INCOMPL	Causes the low field mobility models having a dependence on doping level to depend on ionized dopant density rather than total dopant density. See “Incomplete Ionization and Doping Dependent Mobility” on page 176.
PF.NITRIDE	Enables a model for steady-state conduction in Silicon-Nitride that behaves like Poole-Frenkel emission at high fields.
PFMOB	Specifies that the Poole-Frenkel mobility model will be used for electrons and holes (see Chapter 16 “Organic Display and Organic Solar: Organic Simulators”).
PHUMOB	This is an alias for **KLA**.
PRPMOB	Enables the perpendicular field dependent mobility model as described in Equations 3-303 and 3-316.
SHIRAMOB	Specifies that the Shirahata Mobility Model (see Equation 3-310) will be used for electrons and holes.
SURFMOB **WATT**	Invokes the effective field based surface mobility model (see Equations 3-305 and 3-318). SURFMOB parameters are used in the calculation of surface mobility according to the Watt Model [342]. Do not specify this parameter unless S-Pisces is installed on your system.
TFLDMB1 **SCHWARZ**	Specifies the use of transverse electric field-dependent mobility models. The electron model is based on the Schwarz and Russe equations [281] and is implemented in [291]. The hole model, which is used only when MOBMOD=2, is based on the Watt and Plummer equations [343].
TFLDMB2 **TASCH**	Specifies a transverse electric field dependent mobility model for electrons and holes based on a semi-empirical equation [291, 292].
YAMAGUCHI	Specifies that the Yamaguchi transverse field dependent mobility model is used in the simulation.

Note: See Section 22.36 "MOBILITY" for alternative ways to set mobility models.

Recombination Model Flags

AUGER	Specifies Auger recombination (see Equation 3-386).
AUGGEN	Specifies that the Auger recombination model will be used as a generation and a recombination term.
AUTOBBT2	Alternative to **AUTOBBT**.
CDL and **CONCDL**	Enables the Coupled defect level recombination model. CONCDL additionally enables concentration dependence of the recombination lifetimes used in this model.
CONSRH	Specifies Shockley-Read-Hall recombination using concentration dependent lifetimes (see Equations 3-337, 3-350, and 3-351).
CONSRH.LAW	Enables the concentration dependent Shockley-Read-Hall recombination model using the alternate parameter set given in Table 3-78.
HNSAUG	Enables Auger recombination model that has co-efficients depending on Lattice temperature and carrier concentration (see Equations 3-403 and 3-405).
JTAT	Enables Local Trap assisted tunneling model with parameters on **MODELS** statement.
JTAT.DENSITY	Trap density for JTAT model.
JTAT.LEVEL	Trap energy for JTAT model.
JTAT.M2	Matrix element squared for JTAT model.
KERR.AUG	Enables the Kerr auger recombination model.
KLAAUG	Enables Klaassen's model for concentration dependent auger coefficients (see Equations 3-384, 3-385, and 3-386).
KAUGCN, **KAUGCP**, **KAUGDN**, and **KAUGDP**	Specify parameters of Klaassen's Auger recombination model (see Equations 3-208 and 3-209).
KLASRH	Enables Klaassen's model for concentration dependent lifetimes for Shockley Read Hall recombination (Equations 3-352 and 3-353).
KOSTER	Enables a modification to the Langevin recombination rate whereby the rate is dependent on the minimum mobility as described in (Equations 3-352 and 3-353).
KSRHCN, **KSRHCP**, **KSRHGN**, **KSRHGP**, **KSRHTN**, and **KSRHTP**	Specify parameters of Klaassen's SRH recombination model (see Equations 3-352 and 3-353).
NI.FERMI	Includes the effects of Fermi statistics into the calculation of the intrinsic concentration in expressions for SRH recombination.

OPTR	Selects the optical recombination model (see Equation 3-382). When this parameter is specified, the COPT parameter of the **MATERIAL** statement should be specified.
PIC.AUG	Enables the carrier and doping concentration dependent Auger rate coefficient model given by Equations 3-383 and 3-404.
RICHTER.AUG	Enables the Richter auger recombination model.
S.DISSOC	Enables the singlet dissociation model described in Section 16.3.3 "Exciton Dissociation".
SCHSRH	Enables Scharfetter concentration dependent lifetime model.
SRH	Specifies Shockley-Read-Hall recombination using fixed lifetimes (see Equation 3-348).
SRH.EXPTEMP	Enables the alternative model for the Lattice Temperature dependence of the recombination lifetimes.
TAT.LOCAL	When used with the ITAT/RTAT models, this causes the tunnel model to take into account tunnelling into localized defects in the semiconductor.
TAT.NLCURRS	When used with the self-consistent ITAT/RTAT models, this puts non-local couplings into the current continuity equations to try to improve convergence.
TAT.NLDERIVS	When used with the self-consistent ITAT/RTAT models, this puts non-local couplings into the Poisson equation to try to improve convergence.
TRAP.AUGER	Enables the dependence of recombination lifetime on carrier density as given by Equations 3-352 and 3-365.
TRAP.COULOMBIC	Specifies that the Poole-Frenkel barrier lowering for columbic wells will be used for traps.
TRAP.HURKX	Specifies that the trap-assisted tunneling enhancement factor used in the CONTINUOUS **DEFECTS** and **INTDEFECTS** models applied only to the recombination/generation model.
TRAP.JTAT	Enables Local Trap assisted tunneling model with parameters on **TRAP**/**DOPING** statements.
TRAP.TUNNEL	Specifies that the trap-assisted tunneling model will be used for traps.
TAT.NLDEPTH	Allows you to specify the trap energy used in the TAT.NONLOCAL model. Its units are in eV and how it is interpreted depends on whether the flag TAT.RELEI is specified. If TAT.NLDEPTH is not specified, the trap energy is the same as the Intrinsic Fermi energy.

TAT.NONLOCAL	Enables the non-local tunneling model in the calculation of the field effect enhancement factors. Setting this flag will also set the TRAP.TUNNEL flag.
TAT.RELEI	This flag affects how the TAT.NLDEPTH parameter is used to calculate the trap energy. See the "Trap-Assisted Tunneling" on page 125 for details.
ZAMDMER	Enables the electron lifetime model for TRAP states in LT-GaAs.

Direct Tunneling Flags

An alternative set of direct quantum tunneling models for calculating gate current can be set using DT.CUR on the **MODELS** statement. If you set this parameter, then select the particular model using DT.METH. The allowed values of DT.METH are 1, 2, and 3. The default value of DT.METH is 1.

DT.METH=1	Selects a Fowler-Nordheim type model, with a correction for the trazepoidal, rather than the triangular shape of the potential barrier.
DT.METH=2	Selects a WKB model with the assumption of a linearly varying potential. It involves a calculation of the classical turning points and integration between those limits.
DT.METH=3	Selects an exact formula for a trapezoidal barrier that involves Airy function solutions.

You may also need to specify the type of tunneling taking place by using at least one of the following flags.

DT.CBET	Specifies electron tunneling from conduction band to conduction band.
DT.CUR	Enables the self-consistent version of Direct Tunneling.
DT.VBHT	Specifies holes tunneling from valence band to valence band.
DT.VBET	Specifies Band-to-Band tunneling.
DTPP	Specify this along with DT.CUR to enable the Post-Processing version of Direct tunneling.

Note: Specifying DT.CUR on the **MODELS** statement will invoke the self-consistent version of these models. You can also specify DT.CUR and associated parameters on the **SOLVE** statement, which will invoke the post processing version of this model for the particular **SOLVE** statement.

Note: The effective masses used in the tunneling calculations are the default or specified effective masses for the materials in question, unless you set the ME.DT and MH.DT parameters on the **MATERIAL** statement.

Generation Model Flags

A0.TNLC	Specifies the initial “twist” angle for the twisted neumatic liquid crystal model.
A.BBT.SCHENK	This is the schenk band-to-band tunneling parameter (see Equation 3-470).
AUTOBBT	Causes band-to-band tunneling parameters to be calculated from first principals (see Equations 3-461 and 3-482).
B.BBT.SCHENK	This is the Schenk band-to-band tunneling parameter (see Equation 3-489).
BBT.ALPHA	Specifies the statistical factor scaling used in the Hurkx band-to-band tunneling model (see Equation 3-488).
BBT.DEHURKX	Specifies that the statistical factor used in the Hurkx band-to-band tunneling model will be modified according to Equation 3-486.
BBT.DJHURKX	Specifies that the statistical factor used in the Hurkx band-to-band tunneling model will be modified according to Equation 3-487.
BBT.FORWARD	Avoids convergence problems (if using BBT.NONLOCAL in a forward biased junction).
BBT.HURKX	Enables the Hurkx model (see Equation 3-476).
BBT.KANE	Specifies a Band-to-Band tunneling model according to Kane.
BBT.KL	Specifies a band-to-band tunneling model according to Klaassen.
BBT.NONLOCAL	Enables the non-local band-to-band tunneling model. No other band-to-band tunneling models should be enabled in the same region.
BBT.NLDERIVS	Enables the inclusion of non-local derivatives in the Jacobian matrix (if you set BBT.NONLOCAL).
BBT.SHURKX	Specifies that the Schenk statistical factor will be used in the band-to-band tunneling model (see Equation 3-480).
BBT.STD	Specifies a standard band-to-band tunneling model (see Equation 3-480). The alias for this parameter is **BTBT**.
BBT1	This is an alias for **BBT.KL**.
BBT2	This is an alias for **BBT.STD**.
BBT3	This is an alias for **BBT.HURKX**.
BH.FNONOS	This is the barrier height parameter for the FNONOS model.
BTBT	This is an alias for **BBT.KANE**.

Note: The specification of either DEVDEG.E, DEVDEG.H, or DEVDEG.B will initialize the density of electron (hole) like traps on the interface and will clear out the trapped electron (hole) density on the interface if any.

E.BENDING	Specifies that electron band bending will be taken into account for electron injection.
E.FGCGTUN	Enables the Floating Gate to Control Gate (FGCG) tunneling model for electrons.
E.SC.FGCGTUN	Enables the self-consistent version of the Floating Gate to Control Gate (FGCG) tunneling model for electrons.
ETA.FNONOS	This is the trap capture efficiency for the FNONOS model.
FERRO	Enables the ferroelectric permittivity model (see Section 3.6.8 "The Ferroelectric Permittivity Model").
FG.RATIO	Specifies the ratio of the floating gate extent in the Z direction to the channel width in Z direction for 2D simulation.
FN.CUR	Enables both FNORD and FNHOLES.
FNHOLES	Enables self-consistent analysis of hole Fowler-Nordheim currents.
FNONOS	Enables model for gate tunneling in SONOS structures.
F.TRAP.TUNNEL	Specifies the name of a file containing a C-interpreter function to simulate trap assisted tunneling. The alias for this parameter is F.TRAP.DIRAC.
F.TRAP.COULOMBIC	Specifies the name of a file containing a C-interpreter function to simulate trap assisted tunneling with Coulombic wells.
FNORD	Selects a self-consistent Fowler-Nordheim tunneling model for electrons (see Equation 3-501). This is the recommended approach for calculating Fowler-Nordheim current.
FNHPP	Selects a post-processing Fowler-Nordheim tunneling model for holes (see Equation 3-502). Generally, the Fowler-Nordheim current does not cause convergence problems so this approach is not required.
FNPP	Selects a post-processing Fowler-Nordheim tunneling model for electrons (see Equation 3-501). Generally, the Fowler-Nordheim current does not cause convergence problems so this approach is not required.
GATE1	Enables both HEI and HHI.
GATE2	This is an alias for **GATE1**.
GEIGER	Enables the Geiger model.
GEIGER.MINFIELD	Applies to the path calculation in the GEIGER model. If the search for the path finds a section with a field below this value, then the search will terminate.

GIAR	Enables the graded-index anti-reflective model for the real index of the region.
H.BENDING	Specifies that hole band bending will be taken into account for hole injection (see Equation 3-509).
H.FGCGTUN	Enables the Floating Gate to Control Gate (FGCG) tunneling model for holes.
H.SC.FGCGTUN	Enables the self-consistent version of the Floating Gate to Control Gate (FGCG) tunneling model for holes.
HEI	Specifies the calculation of hot electron injection into oxides. This parameter can be used to simulate MOS gate current or EPROM programming. In transient mode, the oxide current is self-consistently added to the floating gate charge (see Equation 3-504).
HHI	Specifies the calculation of hot hole current into an oxide in a similar manner to HEI (see Equation 3-505).
HW.BBT.SCHENK	This is the schenk band-to-band tunneling parameter (see Equation 3-470).
II.NODE	This is an alias for **OLDIMPACT**.
II.VALDI	This is an alias for **VALDINOCI**.
IMPACT	Invokes the empirical impact ionization model with ionization coefficients taken from [108]. More rigorous impact ionization models may be specified with the **IMPACT** statement.
ITAT.HJ.EL	Enables the Type 2 heterojunction trap assisted tunneling model for electrons.
ITAT.HJ.HO	Enables the Type 2 heterojunction trap assisted tunneling model for holes.
ITAT.PP.EL	Enables the Inelastic Trap Assisted Tunneling model for electrons with no coupling to current continuity equations.
ITAT.PP.HO	Enables the Inelastic Trap Assisted Tunneling model for holes with no coupling.
ITAT.SC.EL	Enables the Inelastic Trap Assisted Tunneling model for electrons with coupling to current continuity equations.
ITAT.SC.HO	Enables the Inelastic Trap Assisted Tunneling model for holes with coupling to current continuity equations.
MASS.TUNNEL	Specifies the effective mass for band to band tunneling (see Equations 3-103 and 3-104).
MIM.ITAT.EL	Enables the Ielmini Trap assisted Tunneling model for electrons in a Metal-Insulator-Metal structure.
MIM.ITAT.HO	Enables the Ielmini Trap assisted Tunneling model for holes in a Metal-Insulator-Metal structure.
MIM.RTAT	Enables the Ielmini Trap assisted Tunneling model for electrons and holes in a Metal-Insulator-Metal structure.

MIMSLICEPTS	Sets the number of points in the interpolation mesh using for tunneling calculation in MIMTUM and MIM.ITAT models.
MIMTUN	Enables the interelectrode tunneling model for electrons and holes in a Metal-Insulator-Metal structure.
MIMTUN.BBT	Enables a model for the interelectrode band-to-band tunneling in a Metal-Insulator-Metal structure.
MIMTUN.EL	Enables the interelectrode tunneling model for electrons in a Metal-Insulator-Metal structure.
MIMTUN.HO	Enable the interelectrode tunneling model for holes in a Metal-Insulator-Metal structure.
N.ANISO	Specifies that the region can be modeled with an anisotropic real index of refraction.
N.CONCANNON	Enables the Concannon gate current model for electrons.
NEARFLG	Specifies the model used for oxide transport when HEI, HHI or FNSONOS are used. The default is false which sets a purely drift based model assigning gate current to the electrodes where the electric field lines through the oxide terminate. Setting NEARFLG replaces this model with one assuming the gate current is flowing to the electrode physically nearest the point of injection. This assumes a purely diffusion transport mechanism. Setting NEARFLG for devices with only one gate or a coarse mesh is advised.
NEWIMPACT	Specifies that impact ionization should use the form described in Equation 3-414.
N.HOTSONOS	Alternative hot electron gate current model to be used with DYNASONOS model.
N1.GIAR	Specifies the starting isotropic index for a graded-index anti-reflective layer.
N2.GIAR	Specifies the ending isotropic index for a graded-index anti-reflective layer.
NE.TNLC	Specifies the constant extra-ordinary index for a twisted nematic liquid crystal region.
NE1.GIAR	Specifies the starting extra-ordinary index for a twisted anisotropic graded-index anti-reflective layer.
NE2.GIAR	Specifies the ending extra-ordinary index for a twisted anisotropic graded-index anti-reflective layer.
NO1.GIAR	Specifies the starting extra-ordinary index for a twisted anisotropic graded-index anti-reflective layer.
NO2.GIAR	Specifies the ending extra-ordinary index for a twisted anisotropic graded-index anti-reflective layer.
NO.TNLC	Specifies the constant ordinary index for a twisted nematic liquid crystal region.

NT.FNONOS	This is the trap area density for saturation mode of the FNONOS model.
OLDIMPACT	Specifies that the impact ionization should use the form described in Equation 3-413. The alias for parameter is **II.NODE**.
P.CONCANNON	Enables the Concannon gate current model for holes.
P.HOTSONOS	Alternative hot electron gate current model to be used with DYNASONOS model.
QT.FILE	Overrides the transmission probability profile by providing a custom profile, e.g. "QT.FILE = transmission.dat". For format details see Section 6.2.4 “Non-local Quantum Barrier Tunneling Model”.
QTNL.DERIVS	Enables non-local couplings in the Jacobian matrix in the self-consistent direct quantum tunneling models.
QTNLSC.BBT	Enables band-to-band mode of the self-consistent direct quantum tunneling model.
QTNLSC.EL	Enables the self-consistent direct quantum tunneling model for electrons.
QTNLSC.HO	Enables the self-consistent direct quantum tunneling model for holes.
QTREGION	Restricts the particular non-local tunneling model to the specified QTREGION. If omitted, then the tunneling model is enabled for all QTREGIONs
QTUNN.BBT	Enables the band-to-band mode of the direct quantum tunneling model. This does not work with the quantum confined variant of quantum tunneling.
QTUNN.DIR	specifies the tunneling current direction if you specify a rectangular mesh using the **QTX.MESH** and **QTY.MESH** statements. A value of 0 means the Y direction (default) and a value of 1 means the X direction.
QTUNN.EL	Enables direct quantum tunneling model for electrons (see Section 3.6.7 “Gate Current Models”).
QTUNN.HO	Enables direct quantum tunneling model for holes.
QTUNN	This is the same as specifying QTUNN.EL, QTUNN.HO, and QTUNN.BBT. These models calculate the quantum tunneling current through an oxide.
QTUNNSC	Enables the self-consistent versions of the direct quantum tunneling models for gate current. These include the tunneling current self-consistently in the carrier continuity equations. QTUNNSC is the same as specifying QTNLSC.EL, QTNLSC.HO, and QTNLSC.BBT.
RTAT.MAJQFL	When using RTAT.SC to connect two semiconducting regions, this flag will improve stability by using the majority carrier's quasi-Fermi level on each side of the barrier. Otherwise, both minority carrier and majority carrier quasi-Fermi levels are used.

RTAT.PP	Enables the Inelastic Trap Assisted Tunneling model for electrons and holes, including recombination, with no coupling to current continuity equations.
RTAT.SC	Enables the Inelastic Trap Assisted Tunneling model for electrons and holes, including recombination, with coupling to current continuity equations.
SBT	This is an alias for **UST**.
SCHENK.BBT	This is the schenk band-to-band tunneling model.
SCHENK.EL	This is the schenk post-processing oxide tunneling current for electrons.
SCHENK.HO	This is the schenk post-processing oxide tunneling current for holes.
SCHENK	This is the schenk post-processing oxide tunneling current for bipolar.
SCHENK.EL	This is the schenk self-consistent oxide tunneling current for electrons.
SCHENK.HO	This is the schenk self-consistent oxide tunneling current for holes.
SCHKSC	This is the schenk self-consistent oxide tunneling current for bipolar.
SHAPEOX	Specifies which algorithm will be used for calculating quantum tunneling current on an non-rectangular mesh. If this parameter is not specified, then the current is calulated in a direction normal to each segment of the semiconductor/oxide interface. If SHAPEOX is specified, then the minimum distance is found between each segment and a contact or a polysilicon region. The quantum tunneling current is calculated along the direction that gives this minimum distance.
SHJ.EL	Enables the non-local heterojunction tunneling model for electrons. See Section 6.2.3 “Non-local Heterojunction Tunneling Model”.
SHJ.HO	Enables the non-local heterojunction tunneling model for holes. See Section 6.2.3 “Non-local Heterojunction Tunneling Model”.
SHJ.NLDERIVS	Enables non-local derivatives for the Newton solver for SHJ models.
SIS.EL	Enables non-local quantum barrier tunneling model for electrons. See Section 6.2.4 “Non-local Quantum Barrier Tunneling Model”.
SIS.HO	Enables non-local quantum barrier tunneling model for holes. See Section 6.2.4 “Non-local Quantum Barrier Tunneling Model”.
SIS.NLDERIVS	Enables non-local derivatives into Newton solver for SIS models.
SIS.REVERSE	Causes the SIS.EL and SIS.HO models to use the quasi-Fermi levels at the start and end of the mesh slices. This may increase stability in some cases.
SIS.TAT	Enables Trap-Assisted-Tunnelling for structures that have an insulator layer between two semiconductors.
SISTAT.TN	Basic electron recombination time for SIS.TAT model.
SISTAT.TP	Basic hole recombination time for SIS.TAT model.

TAT.POISSON	When used with ITAT/RTAT model, this causes the trap ionized densities to be included in the Poisson equation solution.
TAT.SLICEPTS	When used with ITAT/RTAT model, this controls the number of discrete points used in each slice of the Quantum Tunnel region.
TNLC	Enables the twisted nematic liquid crystal model.
TR.TNLC	Specifies the twist rate in degrees per micron for a twisted nematic liquid crystal region.
TSQ.ZAIDMAN	Parameter for the Zaidman model.
UST	Enables the universal Schottky tunneling model (see Section 3.5.2 "Schottky Contacts"). The alias for this parameter is **SBT**.
VALDINOCI	Enables the Valdinoci impact ionization model. See Equations 3-399, 3-432, and 3-440. The alias for this parmeter is **II.VALDI**.
ZAIDMAN	Specifies the tunneling model for surface field emission.

Carrier Statistics Model Flags

BGN	Specifies bandgap narrowing model (see Equation 3-46) with Slotboom default values.
BGN2	Specifies bandgap narrowing model (see Equation 3-46) with Klaassen default values.
BGN.ALAMO	Enables del Alamo Bandgap Narrowing model (see Section 3.2.11 "Bennett-Wilson and del Alamo Bandgap Models").
BGN.BENNETT	Enables Bennett-Wilson Bandgap Narrowing model (see Section 3.2.11 "Bennett-Wilson and del Alamo Bandgap Models").
BGN.KLAASSEN	Has same effect as BGN2.
BGN.LIND	Enables the Lindefelt Bandgap narrowing model for Si and SiC. See Section 3.2.13 "Lindefelt Bandgap Narrowing Model".
BGN.SLOTBOOM	Has same effect as BGN.
BGN.SCHENK	Enables the bandgap narrowing model described in Section 3.2.12 "Schenk Bandgap Narrowing Model".
BOLTZMANN	Specifies that Boltzmann carrier statistics be used (see Equations 3-25 and 3-26).
FERMIDIRAC	Specifies that Fermi-Dirac carrier statistics be used (see Equations 3-25–3-99).
INCOMPLETE	Accounts for incomplete ionization of impurities in Fermi-Dirac statistics (see Equations 3-73 and 3-74).

INC.ION	Specifies that Equations 3-78 and 3-79 should be used for calculations of incomplete ionization.
IONIZ	Accounts for complete ionization of heavily doped silicon when using INCOMPLETE.
UBGN	Enables the universal bandgap narrowing model given in Equation 3-50.

Quantum Model Parameters

2DXY.SCHRO	This is the flag that tells Atlas to solve a 2D Schrodinger equation in XY plane when you also specify the SCHRODINGER or P.SCHRODINGER flag. When it is off, 1D Schrodinger equations will be solved at each slice perpendicular to X axis.
ALT.SCHRO	Specifies usage of an alternate Schrodinger solver.
BC.QUANT	Specifies Dirichlet boundary conditions for all insulator interfaces for the quantum transport equation.
BQP.N	Enables the BQP model for electrons.
BQP.P	Enables the BQP model for holes.
BQP.NGAMMA	Sets the value of BQP γ factor for electrons.
BQP.NALPHA	Sets the value of BQP α factor for electrons.
BQP.NEUMANN	Uses explicitly enforced Neumann boundary conditions instead of the implicit implementation.
BQP.PALPHA	Sets the value of BQP α factor for holes.
BQP.PGAMMA	Sets the value of BQP γ factor for holes.
BQP.QDIR	Sets the direction of principal quantization.
CALC.FERMI	Specifies that the Fermi level used in calculation of the quantum carrier concentration in the Schrodinger-Poisson model is calculated from the classic carrier concentrations.
CALC.PARTITION	Activates the calculation of partition function over all exciton-polaron states: both dark and bright states. This is the proper way to normalize Boltzmann occupation factors. If you leave this parameter as false (its default), the partition function includes only the bright states.
CAPT.AUGER	Enables Quantum Confined Auger recombination for WELL.CAPT model.
CAPT.SRH	Enables Quantum Confined SRH recombination for WELL.CAPT model.
CAPT.TRAP	Enables Quantum Confined Trap recombination for WELL.CAPT model.
DEC.C1	This is a conduction band shift for 1st pair of electron valleys.
DEC.C2	This is a conduction band shift for 2nd pair of electron valleys.
DEC.C3	This is a conduction band shift for 3rd pair of electron valleys.

DEC.D2	This is a conduction band shift for Δ2 electron valleys if DEC.C1 is not set.
DEC.D4	This is a conduction band shift for Δ4 electron valleys if DEC.C2 is not set.
DEV.ISO	This is a conduction band shift for isotropic electron band, when NUM.DIRECT=1.
DEV.HH	This is a valence band shift for heavy holes.
DEV.LH	This is a valence band shift for light holes.
DEV.ISO	This is a valence band shift for isotropic hole band, when NUM.BAND=1.
DEV.SO	This is a valence band shift for split-off holes.
DG.N	This is an alias for **QUANTUM** (see Section 14.3 "Density Gradient (Quantum Moments Model)").
DG.P	This is an alias for P.QUANTUM (see Section 14.3 "Density Gradient (Quantum Moments Model)").
DGN.GAMMA	Specifies the fit factor for electrons for the electron density gradient model (see Equation 14-17).
DGP.GAMMA	Specifies the fit factor for holes for the hole density gradient model (see Equation 14-18).
DGLOG	Specifies that the gradient of the log of concentration is to be used in the calculation of the quantum potential in the density gradient model (see Equation 14-17).
DGROOT	Specifies that the gradient of the root of the concentration is to be used in the calculation of the quantum potential in the density gradient model (see Equation 14-18).
EIGENS	Specifies the number of eigenstates solved for by the Poisson-Schrodinger solver (see Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model").
EF.TOL	Controls the eigenvector tolerance on numerical accuracy for the alternate eigenvalue solver enabled by the NEW.EIG parameter.
EQUIL.NEGF	Specified that the region will be trated as quasi-equilibrium in planar NEGF model.
ESIZE.NEGF	Sets the number energy grid points in the NEGF solver.
ETA.NEGF	An imaginary optical potential, added to Hamiltonian of quasi-equilibrium regions in planar NEGF model.
EV.TOL	Controls eigenvalue tolerance on numerical accuracy for the alternate eigenvalue solver enabled by the NEW.EIG parameter.
FAST.EIG	Enables a computationally efficient version of the eigen-value solver.
FIX.BQN	Specifies that solutions will use the last calculated or loaded values of the electron quantum potential even if BQP.N is not in effect.
FIX.BQP	Specifies that solutions will use the last calculated or loaded values of the electron quantum potential even if BQP.P is not in effect.

FIXED.FERMI	Specifies that a constant Fermi level is used along each individual slice in the Y direction for carrier concentration calculation in the Schrodinger-Poisson Model.
HANSCHQM	Turns on the Hansch quantum effects approximation model for N channel MOS devices (see Section 14.5.1 “Hansch’s Model”).
HOLSTEIN	Activates the Holstein model for calculation of optical emission and absorption spectra of organic materials.
LED	Specifies that the region is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 12 “LED: Light Emitting Diode Simulator”.
N.DORT	Turns on the Van Dort quantum effects approximation model for N channel MOS devices (see Section 14.5.2 “Van Dort’s Model”).
N.NEGF_PL	Activates planar NEGF solver for electrons. Solution is done in all 1D slices.
N.NEGF_PL1D	Activates planar NEGF solver for electrons. Solution is done for the first 1D slice and copied to all other slices.
N.QUANTUM	This is an alias for **QUANTUM** (see Section 14.3 “Density Gradient (Quantum Moments Model)”).
NACC.BETA	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron accumulation described in Section 14.5.2 “Van Dort’s Model”.
NACC.DORT	Enables the Van Dort MOS quantum correction model for electron accumulation described in Section 14.5.2 “Van Dort’s Model”.
NACC.EC	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron accumulation described in Section 14.5.2 “Van Dort’s Model”.
NACC.EXP	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron accumulation described in Section 14.5.2 “Van Dort’s Model”.
NACC.N	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron accumulation described in Section 14.5.2 “Van Dort’s Model”.
NEGF_CMS	Enforces coupled mode space NEGF approach.
NEGF_MS	Activates a mode-space NEGF solver. See Chapter 14 “Quantum: Quantum Effect Simulator” for more information.
NEGF_UMS	Enforces uncoupled mode space NEGF approach.
NPRED.NEGF	Sets a number of predictor iterations in the predictor-corrector scheme of Schrodinger and NEGF solvers.
NEW.SCHRO	Specifies that the new non-rectilinear self-consistent Schrodinger-Poisson solver will be used for an Atlas mesh (see Section 14.2 “Self-Consistent Coupled Schrodinger Poisson Model”). When this is specified, the Schrodinger solver mesh should also be specified using the **SPX.MESH** and **SPY.MESH** statements.
NEW.EIG	Enables an alternate eigenvalue solver that obtains eigenvectors using the `INVERSE ITERATION` method.

NINV.BETA	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron inversion described in Section 14.5.2 “Van Dort’s Model”.
NINV.DORT	Enables the Van Dort MOS quantum correction model for electron inversion described in Section 14.5.2 “Van Dort’s Model”.
NINV.EC	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron inversion described in Section 14.5.2 “Van Dort’s Model”.
NINV.EXP	Specifies a default parameter value for the Van Dort MOS quantum correction model for electron inversion described in Section 14.5.2 “Van Dort’s Model”.
NUM.BAND	Specifies the number of valence bands to retain for Schrodinger Poisson solutions.
NUM.DIRECT	Specifies the number of conduction band directions in the *k* space to retain for Schrodinger Poisson solutions.
ORTH.SCHRO	Specifies the critical value used for determination of the orthogonality of eigenvectors (wavefunctions) resulting from Schrodinger equation solutions. The orthogonality measurement is done by taking the sum of the product of individual samples from two eigenvectors in question and dividing the sum by the number of samples. If that measure exceeds the value specified by ORTH.SCHRO, then an error message will be issued during any subsequent output of structure files. This message indicates that you should carefully check any wavefunction data contained in the structure file and make adjustments as necessary.
OX.MARGIN	This is the maximum electron penetration into oxide. You can use it with OX.SCHRO in non-planar semiconductor-oxide interface or nonrectangular channel crossection (Atlas 3D) to reduce the number of nodes used in the solution of the Schrodinger equation.
OX.POISSON	Calculates the quasi-Fermi levels in insulators for Schrodinger Poisson solutions.
OX.SCHRO	Specifies that the band edges of insulators are included in the solution of Schrodinger's Equation in self-consistent solutions of the Schrodinger-Poisson Model (see Section 14.2 “Self-Consistent Coupled Schrodinger Poisson Model”).
P.DORT	Turns on the Van Dort quantum effects approximation model for P channel MOS devices (see Section 14.5.2 “Van Dort’s Model”).
P.NEGF_PL	Activates planar NEGF solver for holes. Solution is done in all 1D slices.
P.NEGF_PL1D	Activates planar NEGF solver for holes. Solution is done for the first 1D slice and copied to all other slices.
P.SCHRODING	Computes the self consistent Schrodinger Poisson for holes.
PACC.BETA	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole accumulation described in Section 14.5.2 “Van Dort’s Model”.
PACC.DORT	Enables the Van Dort MOS quantum correction model for hole accumulation described in Section 14.5.2 “Van Dort’s Model”.

PACC.EC	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole accumulation described in Section 14.5.2 “Van Dort’s Model”.
PACC.EXP	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole accumulation described in Section 14.5.2 “Van Dort’s Model”.
PACC.N	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole accumulation described in Section 14.5.2 “Van Dort’s Model”.
PIEZO.SCALE	This is an alias for **POLAR.SCALE**.
PIEZOELECTRIC	This is an alias for **POLARIZATION**.
PINV.BETA	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole inversion described in Section 14.5.2 “Van Dort’s Model”.
PINV.DORT	Enables the Van Dort MOS quantum correction model for hole inversion described in Section 14.5.2 “Van Dort’s Model”.
PINV.EC	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole inversion described in Section 14.5.2 “Van Dort’s Model”.
PINV.EXP	Specifies a default parameter value for the Van Dort MOS quantum correction model for hole inversion described in Section 14.5.2 “Van Dort’s Model”.
POLARIZATION	Enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. The alias for this parameter is **PIEZOELECTRIC**. See Section 3.6.11 “Polarization in Wurtzite Materials”.
POLAR.CHARG	Specifies polarization charge densities to replace the density calculated using Equations 3-375 and 3-388.
POLAR.SCALE	Specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the **REGION** statement. The alias for this parameter is PIEZO.SCALE. See Section 3.6.11 “Polarization in Wurtzite Materials”.
POST.SCHRO	Specifies to calculate the Schrodinger Equation (see Section 14.2 “Self-Consistent Coupled Schrodinger Poisson Model”) for quantum effects as a post processing step.
PSP.SCALE	Specifies a constant scale factor similar to POLAR.SCALE but applied only to spontaneous polarization.
QMINCONC	Specifies the minimum carrier concentration considered in the gradient calculations of the density gradient model or in Poisson's Equation in the self-consistent Schrodinger-Poisson Model (see Sections 14.2 “Self-Consistent Coupled Schrodinger Poisson Model” and 14.3 “Density Gradient (Quantum Moments Model)”).
QCRIT.NEGF	Sets a convergence criterium for potential in case of NEGF or Schrodinger solver.
QDOSTOL	The user-defined tolerance for DOS when performing adaptive refinement of Brillouin zone in the k.p case and the energy grid in the case of **HOLSTEIN** model.

QSPEC.EMAX	The user-defined maximum energy for computing the optical emission/absorption spectra. This is an alias to **KP.CHIEMIN**.
QSPEC.EMIN	The user-defined minimum energy for computing the optical emission/absorption spectra. This is an alias to **KP.CHIEMAX**.
QSPEC.RES	The user-defined resolution for optical spectra. This is an alias to **KP.CHIRES**.
QUANTUM	Enables the density gradient (see Section 14.3 "Density Gradient (Quantum Moments Model)").
QVERBOSITY	Defines the amount of detail printed to screen by the calculations performed by quantum library. 0 prints nothing, 1 prints minimal information, and 2 prints full information about total radiatve rates gain and quasi-particle densities in each region
QW.GEOM	This is an alias for **SL.GEOM**.
QWELL	Specifies that the region is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.
QWNUM	Optionally sets the number of the quantum well region.
QX.MAX, QX.MIN	Specify the minimum and maximum extent of the Poisson-Schrodinger solver along the X axis direction (see Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model").
QY.MAX, QY.MIN	Specify the minimum and maximum extent of the domain of the Schrodinger solver (see Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model").
SCHRO	Enables the Poisson-Schrodinger solver (see Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model"). This can be used for zero carrier solutions only (specified by `METHOD CARRIERS=0`). This is typically combined with the `EIGENS` parameter to control the number of eigenstates calculated. This is a 1D solver that is solved within the mesh between limits set by `QX.MIN` and `QX.MAX`.
SL.GEOM	Specifies the superlattice geometry as either `1DY` (default) or `1DX`.
SL.NUMBER	Assigns the index for a regions membership in a indexed superlattice.
SLATT	Enables the superlattice model for a region.
SLATT.NUMBER	This is an alias for **SL.NUMBER**.
SP.BAND	This is a flag that turns on a banded matrix solver used to solve the 2D Schrodinger equation in Atlas 3D. Otherwise, a full matrix solver is used.
SP.CHECK	Enables checking of orthagonality of wavefunctions for Schrodinger - Poisson solutions.

SP.DIR	Specifies direction of quantum confinement in Schrodinger equation and used in materials with single electron band but with anistropic effective mass. It is active only when you set SCHRODINGER and NUM.DIRECT=1 on the **MODELS** statement. The values of effective mass(es) in quantization direction(s) and DOS effective mass are give below. In Atlas, quantization is in Y direction: SP.DIR=0 my = MC, dos_mass= MC SP.DIR=1 my = ML, dos_mass= $\sqrt{MT1 \cdot MT2}$ SP.DIR=2 my = MT1, dos_mass= $\sqrt{ML \cdot MT2}$ SP.DIR=3 my = MT2, dos_mass= $\sqrt{MT1 \cdot MT2}$ In Atlas, quantization is in XY plane (with SCHRO.2DXY on MODELS statement): SP.DIR=0 (mx,my)= (MC,MC), dos_mass= MC SP.DIR=1 (mx,my)= (MT1,ML), dos_mass= MT2 SP.DIR=2 (mx,my)= (MT2,MT1), dos_mass= ML SP.DIR=3 (mx,my)= (ML,MT2), dos_mass= MT1 In Atlas 3D, quantization is in YZ plane: SP.DIR=0 (my,mz)= (MC,MC), dos_mass= MC SP.DIR=1 (my,mz)= (ML,MT2), dos_mass= MT1 SP.DIR=2 (my,mz)= (MT1,ML), dos_mass= MT2 SP.DIR=3 (my,mz)= (MT2,MT1), dos_mass= ML
SP.EXCORR	Activates exchange-correlation correction in Schrodinger solvers.
SP.FAST	Activates a fast product-space approach in a 2D Schrodinger solver.
SP.GEOM	Sets a dimensionality and direction of a Schrodinger solver. See Chapter 14 "Quantum: Quantum Effect Simulator" for more information.
SP.SMOOTH	Enables smoothing of quantum carrier calculations. When you set the SP.SMOOTH parameter on the **MODELS** statement, the SP solver will use the degeneracy factor and effective mass evaluated at the location of the maximum value of the primary wave function for calculation of carrier concentration. This removes any discontinuities due to abrupt changes in mass or degeneracy.
WELL.CAPT	Enables the quantum well capture escape model through multiple phonon emission.
WELL.DENERGY	Energy spacing for numerical integration of spontaneous emission spectrum to radiative recombination.
WELL.INPLANE	Enables in-plane 2D carrier drift diffusion transport for the quantum well capture escape model.

WELL.NX **WELL.NY** **WELL.NZ**	Specifies the number of grid points in x, y, and z directions for Schrodinger equation in a quantum well region.
WELL.SEPARATE	Forces each subband to have its own Fermi level for the quantum well capture escape model.
WELL.WBTUNN	Enables tunneling between confined states and bulk carriers for the quantum well capture secape model.
WELL.WWSPONT	Enables raditative transitions between neighboring wells in the quantum well capture escape model.
WELL.WWTUNN	Enables interwell tunneling between confined states for the quantum well capture escape model.

Multiband k.p Model Flags

KP.ADAPTIVE	Turns on the adaptive generation of *k* points with the refinement criterion set by convergence of the DOS to KP.DOSTOL within each element of the mesh in *k*-space.
KP.CHIEMIN	Specifies the minimum of the energy interval over which the optical response is to be calculated.
KP.CHIEMAX	Specifies the maximum of the energy interval over which the optical response is to be calculated.
KP.CHIRES	Resolution for the energy grid over which the optical response (e.g., gain and absorption) are computed.
KP.CNLDOS	Specifies the number of uniformly placed slices in the calculated energy range of conduction bands. The mean local density of states over the energy intervals between slices is reported in the structure file.
KP.COPYPARAMS	Overrides the values of the **k.p** parameters in the models' own internal database. The parameters are copied from the main Atlas program or as specified in the input deck.
KP.CV2	Activates the block diagonalized **k.p** model with HH, LH, and conduction band in one block, and the other block is treated as having identical spectrum. Block degeneracy holds strictly only for potentials with reflection symmetry. But, it is a good approximation when structural inversion asymmetry is small.
KP.CV3	Activates the block diagonalized **k.p** model with HH, LH, SO, and conduction band in one block, and the other block is treated as having identical spectrum.
KP.CV4	Activates the **k.p** model with spin resolved HH, LH, and conduction bands.
KP.CV6	Activates the full 8-band **k.p** model.
KP.CVCOUPLED	Turns off the conduction-valence coupling for band structure calculation in the KP.CV3 model.

KP.DOS.CRANGE	Specifies the size of the energy interval for calculating the density of states of the conduction bands. If unspecified, the interval is set as the energy window in whcih the WELL.CNBS many bands contribute at all k-points between 0 and KP.KMAX. The minimum of the energy window for is set below the bottom of minimum conduction band edge.
KP.DOSTOL	The convergence criterion for DOS. This is ignored if KP.ADAPTIVE is FALSE.
KP.DOSRES	The resolution for the energy grid over which DOS is computed.
KP.FEM.HERMITE	Activates the third order Hermite polynomial basis for finite element expansion of wavefunctions.
KP.FEM.INTERFACE	Activates wavefunction derivative matching at each cell boundary in the finite element modeling.
KP.FEM.LAGRANGE	Activates the second order polynomial basis for finite element expansion of wavefunctions.
KP.GAUSSIAN	Activates the Gaussian form of lineshape broadening of optical response with broadening given by KP.CHIBROAD.
KP.KMAX	The maximum value of *k* in the **k.p** model.
KP.INTERSUB	Activates the inclusion of intersubband transitions to the optical response. It essentially forces KP.CHIEMIN=0.
KP.LORENTZ	Activates the Lorentz form of lineshape broadening of optical response with broadening given by KP.CHIBROAD.
KP.NUMKPT	The initial number of points for the *k*-point mesh. It is also the final number if KP.ADAPTIVE is not set (default).
KP.SOLVERTOL	The tolerance for the Arpack solver.
KP.V2	Activates the Block diagonalized HH/LH model with explicit calculation performed only on one block. The remaining block is treated as having identical spectrum. This holds strictly only for symmetric potentials. But, this is a good approximation when structural inversion asymmetry is small.
KP.V3	Activates the Block diagonalized 6 valence band (HH, LH, SO) model with explicit calculation for only one block. See KP.V2 for cases when this is useful.
KP.V4	Activates the 4 valence band valence model with HH and LH bands.

KP.V6	Activates the 6 valence band model with HH, LH, and SO bands.
KP.VNLDOS	Specifies the number of uniformly placed slices in the calculated energy range of valence bands. The mean local density of states over the energy intervals between slices is reported in the structure file.
KP.DOS.VRANGE	Specifies the size of the energy interval for calculating the density of states of the valence bands. If unspecified, the interval is set as the energy interval in which the WELL.VNBS many bands contribute at all k-points between 0 and KP.KMAX. The maximum of the energy window is set above the top of maximum valence band edge.

Energy Balance Simulation Flags

A.TEMP	This is an alias for **HCTE**.
E.TAUR.GONZALEZ	Enables the Gonzalez energy balance relaxation time model for electrons. See Section 3.4.4 “Temperature Dependence of Relaxation Times”.
E.TAUR.VAR	Specifies that electron temperature dependent energy relaxation time is used. Use the TRE.T1, TRE.T2, TRE.T3, TRE.W1, TRE.W2, and TRE.W3 parameters on the **MATERIAL** statement to specifies the energy relaxation time.
ET.MODEL	This is an alias for **HCTE**.
H.TAUR.VAR	Specifies that hole temperature dependent energy relaxation time is used. Use TRH.T1, TRH.T2, TRH.T3, TRH.W1, TRH.W2, and TRH.W3 parameters on the **MATERIAL** statement to specifies the energy relaxation time.
HCTE	Specifies that both electron and hole temperature will be solved. The aliases for this parameter are **A.TEMP** and **ET.MODEL**.
HCTE.EL	Specifies that electron temperature will be solved.
HCTE.HO	Specifies that hole temperature will be solved.
F.KSN	Specifies the name of a file containing a C-Interpreter function specifying the electron Scattering Law Exponent as a function of electron energy. This is the value of Scattering Law Exponent. Values other than 0 or -1 are permitted.
F.KSP	Specifies the name of a file containing a C-Interpreter function specifying the hole Scattering Law Exponent as a function of hole energy. This is the value of Scattering Law Exponent. Values other than 0 or -1 are permitted.
GR.HEAT	Includes generation/recombination heat source terms in the lattice heat flow equation.
H.TAUR.GONZALEZ	Enables the Gonzalez energy balance relaxation time model for holes. See Section 3.4.4 “Temperature Dependence of Relaxation Times”.
JOULE.HEAT	Includes the Joule heat source terms in the lattice heat flow equation.
KSN	Specifies which hot carrier transport model will be used for electrons. KSN=0 selects the hydrodynamic model and KSN=-1 selects the energy balance model.

KSP	Specifies which hot carrier transport model will be used for holes. KSP=0 selects the hydrodynamic model and KSP=-1 selects the energy balance model.
TAUMOB	Specifies the dependence of relaxation times with carrier temperature in the mobility definition. If TAUMOB is specified, the values of **MATERIAL** statement parameters TAUMOB.EL and TAUMOB.HO are dependent on the carrier temperature.
TAUTEM	Specifies the dependence of relaxation times with carrier temperature. If TAUTEM is specified, the values of **MATERIAL** statement parameters TAUREL.EL and TAUREL.HO are dependent on carrier temperature.
N.TEMP or **HCTE**	Specifies that the electron temperature equation will be solved.
P.TEMP or **HCTE.HO**	Specifies that the hole temperature equation will be solved.
PASSLER	Enables Passler's bandgap versus temperature model.
PT.HEAT	Includes Peltier-Thomson heat sources in the lattice heat flow equation.

Lattice Heating Simulation Flags

F.PDN	Uses C-Interpreter function for phonon drag contribution to electron thermopower.
F.PDP	Uses C-Interpreter function for phonon drag contribution to hole thermopower.
HEAT.FULL	Enables all thermal sources and sinks (Joule heat, generation-recombination heat, and Peltier Thomson heat).
HEAT.PETHOM	This can be used to turn off the Peltier-Thomson heat source in the HEAT.FULL option.
LAT.TEMP **L.TEMP**	Specifies that the lattice temperature equation will be solved. For lattice heating simulation there must be at least one thermal contact defined using the **THERMCONTACT** statement.
PCM	Specifies that the phase change material temperature dependent resistivity is to be used.
PCM.LOG	Specifies that for phase change materials the interpolation for resistivity between the low temperature value (PCM.CTC and PCM.CRHO) and the high value (PCM.ATC and PCM.ARHO) is logarithmic.
PHONONDRAG	Enables the phonon drag contribution to thermopower.

Note: These parameters should only be specified if Giga or Giga 3D is enabled on your system.

Defect Generation Model

BEAM.LID	Specifies that the generation rate for the light induced defect generation model is calculated only from active **BEAM** statements, which have the LID parameter specified.
FILE.LID	Specifies the file name where the dangling bond density of states distribution, as a function of energy, will be stored.
GENERATE.LID	Specifies the generation rate for the light induced defect generation model.
TIME.EP	Specifies the bias stress time value used in the amphoteric defect generation model.
TIME.LID	Specifies the time value used in the light induced defect generation model.
TRAP.LID	Specifies the light induced defect generation model will be used.

Magnetic Field Model

BX	This is the X Component of Magnetic Field Vector (Tesla).
BY	This is the Y Component of Magnetic Field Vector (Tesla).
BZ	This is the Z Component of Magnetic Field Vector (Tesla).
MAGMIN.N	Stability parameter for electron equation (see Section 3.11 “Carrier Transport in an Applied Magnetic Field”).
MAGMIN.P	Stability parameter for hole equation (Section 3.11 “Carrier Transport in an Applied Magnetic Field”).
R.ELEC	This is the Hall scattering factor for electrons .
R.HOLE	This is the Hall scattering factor for holes.

Model Macros

BIPOLAR	Selects a default set of models which are used when simulating bipolar devices. The bipolar models are CONMOB, FLDMOB, BGN, CONSRH, and AUGER. If LAT.TEMP is also specified on the **MODELS** statement, or the TEMPERATURE parameter differs from 300 Kelvin by more than 10 Kelvin, then the ANALYTIC model is used instead of CONMOB.
BIPOLAR2	Selects an alternative default set of models which are used when simulating bipolar devices. The bipolar models are FERMI, KLA, KLASRH, KLAAUG, FLDMOB, and BGN.
ERASE	Specifies a default set of models which are used to simulate EEPROM erasure. When it’s specified, the MOS, FNORD, IMPACT, and BBT.KL models will be used.
GANFET	Enables models convenient for simulation for GaN based FETs. These models include POLAR, CALC.STRAIN, SRH, ALBRCT.N, and POLAR.SCALE=1.0.

MOS	Specifies a default set of models for MOS devices. The MOS models are CVT, SRH and FERMI.
MOS2	Specifies a default set of models for MOS devices. The MOS2 models are KLA, SHI, SRH, FERMI and BGN.
PROGRAM	Specifies a default set of models used when writing to EEPROMS. When PROGRAM is specified, the MOS, HEI, and IMPACT models will be used.

General Parameters

ADACHI	Forces the material parameters for InGaAsP (see Section 6.4.5 "In(1-x)Ga(x)As(y)P(1-y) System") for InGaP.
ALPHA.DOS	Specifies that density of states data should be extracted from the absorption spectra during output of complex index of refraction using OUT.INDEX on the **MATERIAL** statement.
ALL	Enables solution for both electrons and holes.
ASUB	Specifies the substrate lattice constant for strain calculations as described in Section 3.6.10 "Epitaxial Strain Tensor Calculation in Wurtzite".
BB.GAMMA	Specify the band-to-band tunneling parameters (see Equation 3-480).
BBT.GAMMA	This is an alias for **BB.GAMMA**..
BOUND.TRAP	Enables modeling of traps coincident with ohmic boundaries.
CARRIERS	Specifies the number of carriers to simulate, which should be 0, 1 or 2. The alias is **NUMCARR**.
CHUANG	This is an alias for the **WZ.KP** parameter.
CUBIC35	Specifies to use the material dependent band parameter model discussed in Section 6.4.1 "Cubic III-V Semiconductors".
DEVICE	Specifies which device in MixedMode simulation the **MODELS** statement should apply to. The synonym for this parameter is **STRUCTURE**.
DRIFT.DIFF	Specifies that the drift-diffusion transport model is to be used. This implies that the electron and hole carrier temperature equations will not be solved.
ELECTRONS	Specifies that the electron continuity equation is included in simulation.
FC.ACCURACY	This is the maximum ratio of the change in integrated photogeneration to the total that is acceptable during iterations for calculation of frequency conversion material emission (see Section 11.11 "Frequency Conversion Materials (2D Only)").
FETIS	Specifies that the default material parameters for GaN/AlN/IaN, AlGaN, and InGaN are what is used by the FETIS [302] simulator.
FREQ.CONV	Enables the frequency conversion material model.

F.PCM	Specfies the file name of a C-Interpreter function that models the resistivity of a conductor as a function of time, temperature, and history.
G.EMPIRICAL	Enables the empirical gain model from Section 3.9.4 “The Empirical Gain Model” (GAINMOD=2).
G.STANDARD	Enables the gain model from Section 3.9.3 “The Standard Gain Model” (GAINMOD=1).
H.ATOM	Enables the atomic hydrogen transport model for use with multistate trap model.
H.MOLE	Enables the molecular hydrogen transport model for use with multistate trap model.
HOLE	Specifies that the hole continuity equation is included in simulation.
ISHIKAWA	Enables Ishikawa's strain model for InGaAs/InGaAsP/InGaP (see Section 3.9.12 “Ishikawa's Strain Effects Model”).
K.P	Enables using the k*p model effective masses and band edge energies for drift-diffusion simulation.
LI	This is an alias for the **ZB.TWO** parameter.
LORENTZ	Enables Lorentz gain broadening.
MATERIAL	Specifies which material from Table B-1 will apply to the **MODELS** statement. If a material is specified, then all regions defined as being composed of that material will be affected.
MS.DISS	Specifies the upper limit for the integration of the geminate pair distance distribution function in terms of integer multiples of the geminate pair distance parameter A.SINGLET (i.e., the upper limit is given by MS.DISS*A.SINGLET).
NAME	Specifies which region will be applied to the **MODELS** statement. Note that the name must match the name specified in the NAME parameter of the **REGION** statement or the region number.
NS.DISS	Specifies the number of samples to take in evalution of the integral of the geminate pair distance distribution function.
NSPECIES	Number of ionic species to be simulated up to a maximum of 3 (or 2 if Giga is enabled).
PATRIN	Enables the strain dependence of bandgap in the Raykanan silicon absorption model in Equation 3-612.
PCH.INS	Specifies that polarization charges will be included along interfaces with insulators or with the outside domain.
PFREQ.CONV	Specifies that the ray trace for emissions from a frequency conversion material are periodic with respect to the boundaries at the minimum and maximum x coordinates.
PRT.BAND	Enables run-time output of regional band parameter model information.

PRT.COMP	Enables run-time output of regional composition information.
PRT.FLAGS	Enables run-time output of regional model flag information.
PRT.IMPACT	Enables run-time output of regional impact ionization model information.
PRT.RECOM	Enables run-time output of regional recombination model information.
PRT.THERM	Enables run-time output of regional thermal model information.
PRINT	Prints the status of all models, a variety of coefficients, and constants. We recommended that you include this parameter in all Atlas runs.
PRINT.REGION	Prints the model status and coefficients for the specified region.
PRINT.MATERIAL	Prints the model status and coefficients for the specified material.
RAJKANAN	Enables the analytic absorption model for silicon given by Equation 3-612.
REGION	Specifies the region number for this **MODELS** statement. The alias is **NUMBER**.
SDISS.TYPE	Specifies the geminate pair distribution type as follows: 1 - delta function, 2 - Gaussian, 3 - exponential, and 4 - r-squared exponential. See Section 16.3.3 "Exciton Dissociation".
SPEC.NAME	Specifies the name of a file to collect the emission spectrum from the Laser simulator.
SPECIES1.A	Composition number of ionic species 1 (see Section 3.15 "Generic Ion Transport Model").
SPECIES2.A	Composition number of ionic species 2 (see Section 3.15 "Generic Ion Transport Model").
SPECIES3.A	Composition number of ionic species 3 (see Section 3.15 "Generic Ion Transport Model").
SPECIES1.Z	Charge multiple of ionic species 1 (see Section 3.16 "Generic Ion Reaction Model").
SPECIES2.Z	Charge multiple of ionic species 2 (see Section 3.16 "Generic Ion Reaction Model").
SPECIES3.Z	Charge multiple of ionic species 3 (see Section 3.16 "Generic Ion Reaction Model").
SPONTANEOUS	Includes the radiative recombination rates derived from the LI, YAN or CHUANG model into the drift diffusion.

Note: You need additional computational resources since the radiative rate must be numerically integrated for each grid point.

STRESS	Enables the silicon stress dependent bandgap model from Section 3.6.12 "Stress Effects on Bandgap in Si".
STRUCTURE	This is a synonym for **DEVICE**.
SUBSTRATE	This is a logical parameter indicating that the specified region should be considered as the substrate for strain calcualtions described in Section 3.6.10 "Epitaxial Strain Tensor Calculation in Wurtzite".
TAYAMAYA	Enables the model from Section 3.9.5 "Takayama's Gain Model" (GAINMOD=3).
TEMPERATURE	Specifies the temperature in Kelvin.
TLU.INDEX	Specifies that the Tauc-Lorentz dielectric function (see Section 3.10.3 "Tauc-Lorentz Dielectric Function with Optional Urbach Tail Model for Complex Index of Refraction") should be used for calculation of the complex index of refraction.
UNSAT.FERRO	Enables unsaturated loop modeling for ferroelectrics.
WZ.KP	Enables the WZ.KP gain/radiative recombination model (GAINMOD=5).
YAN	This is an alias for the **ZB.ONE** parameter.
ZB.ONE	Enables the ZB.ONE gain/radiative recombination model (GAINMOD=5).
ZB.TWO	Enables the ZB.TWO gain/radiative recombination model (GAINMOD=5).

Model Dependent Parameters

ARORA Model Parameters

The parameters that can only be used with the ARORA model (see Equations 3-209 and 3-210) are **AR.MU1N**, **AR.MU2N**, **AR.MU1P**, and **AR.MU2P**.

BBT.KL and BBT.STD Model Parameters

The parameters used with the BBTKL model are BB.A1 and BB.B. The BB.A2 and BB.B parameters are used with the BBT.STD model. The BB.A1 and BB.A2 parameters are the pre-exponential coefficients in the band-to-band tunneling models. The BB.B parameter is the exponential coefficient used in both models. The default value of BB.B depends on which model is chosen (see Equation 3-480).

CCSMOB Model Parameters

The **CCS.EA**, **CCS.EB**, **CCS.HA**, and **CCS.HB** parameters describe the dependence of mobility on doping, carrier concentration, and temperature (see Equations 3-213 – 3-217).

FLDMOB Model Parameters

B.ELECTRONS	This is used in the field-dependent mobility (see Equation 3-323). See also BETAN on the **MOBILITY** statement.
B.HOLES	This is used in the field-dependent mobility expression (see Equation 3-324). See also BETAP on the **MOBILITY** statement.
E0	This is used in the field dependent mobility model for EVSATMOD=1 (see Equation 6-65).
EVSATMOD	Specifies which parallel field dependent mobility model (see Equation 3-323 and Equation 6-67) should be used for electrons. The value of EVSATMOD should be assigned as follows: • 0 - Use the standard saturation model (Equation 3-323). You also can apply the option parameter, **MOBTEM.SIMPL** (see the "Carrier Temperature Dependent Mobility" on page 209 for more information). • 1 - Use the negative differential velocity saturation model (Equation 6-67). • 2 - Use a simple velocity limiting model. In most cases, the default value of 0 should be used.
HVSATMOD	Specifies which parallel field dependent mobility model (see Equation 3-324 and Equation 6-68) should be used for holes. The value of HVSATMOD should be assigned as follows: • 0 - Use the standard saturation model (Equation 3-324). • 1 - Use the negative differential velocity saturation model (Equation 6-68). • 2 - Use a simple velocity limiting model. In most cases, the default value of 0 should be used.

Fowler-Nordheim Tunneling Model Parameters

The parameters used in this model are **F.AE** and **F.BE**. F.AE is the pre-exponential factor and F.BE is the exponential coefficient (see Equation 3-501).

WATT or SURFMOB Model Parameters

The parameters that can be used with the WATT model, include: **ALN1**, **ALN2**, **ALN3**, **ALP1**, **ALP2**, **ALP3**, **ETAN**, **ETAP**, **MREFN1**, **MREFN2**, **MREFN3**, **MREFP1**, **MREFP2**, and **MREFP3** (see Equations 3-305 and 3-318).

TFLDMB1 and TFLDMB2 Model Parameters

ACC.SF	Specifies the accumulation saturation factor which describes the ratio of the majority carrier concentration in the accumulation layer before and after bending of conductivity and valence bands.
INV.SF	Specifies the inversion saturation factor which describes the ratio of the majority carrier concentration in the inversion layer before and after the bending of conductivity and valance bands.
OX.BOTTOM	Specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor.
OX.LEFT	Specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor.
OX.RIGHT	Specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor.

CONCANNON Model Parameters

CGATE.N	Specifies an empirical tuning factor for electrons in the Concannon gate current model.
CGATE.P	Specifies an empirical tuning factor for holes in the Concannon gate current model.
PEFF.N	Specifies an effective barrier height for electrons in the Concannon gate current model. This is an alias for `IG.EB0`.
PEFF.P	Specifies an effective barrier height for holes in the Concannon gate current model. This is an alias for `IG.HB0`.
THETA.N	Specifies the critical rejection angle for electrons in the Concannon gate current model.
THETA.P	Specifies the critical rejection angle for electrons in the Concannon gate current model.
C0	Specifies the electron distribution weight factor in the Concannon gate current model.
CHIA	Specifies the electron distribution function constant in the Concannon gate current model.
CHIB	Specifies the electron distribution function constant in the Concannon gate current model.
CHI.HOLES	Specifies the hole distribution function constant in the Concannon gate current model.
ENERGY.STEP	Specifies the energy step for numeric integration in the Concannon gate current model.

INFINITY	Specifies the highest energy in numeric integration in the Concannon gate current model.
PATH.N	Specifies the mean free path in the oxide for electrons in the Concannon gate current model.
PATH.P	Specifies the mean free path in the oxide for holes in the Concannon gate current model.

HEI Model Parameters

The parameters that may be used with the HEI model include: IG.ELINR, IG.HLINR, IG.ELINF, IG.HLINF, IG.EBETA, IG.HBETA, IG.EETA, IG.HETA, IG.EEOX, IG.HEOX, IG.EB0, IG.HB0.

Table 22-3 lists aliases for parameters of the HEI model.

Table 22-3 HEI model aliases

Parameter	Alias	Parameter	Alias
IG.ELINF	**LAMHN**	**IG.EBETA**	**BARLN**
IG.ELINR	**LAMRN**	**IG.HBETA**	**BARLP**
IG.HLINF	**LAMHP**	**IG.EETA**	**TUNLN**
IG.HLINR	**LAMRP**	**IG.HETA**	**TUNLP**

N.DORT and P.DORT Model Parameters

B.DORT	User-specifiable model parameter for the Van Dort quantum effects approximation model.
D.DORT	User-specifiable model parameter for the Van Dort quantum effects approximation model.

BBT.KANE Model Parameters

The parameters used with the BBT.KANE model are BBT.A_KANE (aliases BB.A_KANE or A.BTBT), BBT.B_KANE (aliases BB.B_KANE or B.BTBT), and BBT.GAMMA (alias **BB.GAMMA**).

LASER Simulation Parameters

CAVITY.LENGTH	Specifies the cavity length in the longitudinal direction (in mm).
GAINMOD	Specifies the local optical gain model to be used. GAINMOD=0 specifies that no optical gain model will be used. GAINMOD=1 specifies that the complex frequency-dependent gain model will be used (see Section 3.9.3 "The Standard Gain Model"). GAINMOD=2 specifies that the simple gain model will be used (see Section 3.9.4 "The Empirical Gain Model"). GAINMOD=3 specifies that Takayama's gain model will be used (see Section 3.9.5 "Takayama's Gain Model"). GAINMOD=5 specifies that the Yam's, Li's, or Chuang's gain model will be used (see Sections 3.9.7 "Unstrained Zincblende Models for Gain and Radiative Recombination" and 3.9.10 "Strained Wurtzite Three-Band Model for Gain and Radiative Recombination").
LAS.ABSOR_SAT	Enables the non-linear absorption loss saturation model.
LAS.ABSORPTION	Enables the absorption loss model in Laser.
LAS.EFINAL **LAS.EINIT**	Specify the lower and upper photon energies. Laser will calculate multiple longitudinal photon rates within this range. Using wide ranges can slow down simulation.
LAS.ESEP	Specifies the photon energy separation. If this is not specified, Laser will automatically calculate the number of longitudinal modes based on the cavity length and the energy range.
LAS.ETRANS	Specifies that the Helmholtz solver should use the transverse mode with the eigen-energy closes to the energy specified by the LAS.TRANS_ENERGY parameter.
LAS.FCARRIER	Enables the free carrier loss model in Laser.
LAS.GAIN_SAT	Enables the non-linear gain saturation model.
LAS.ITMAX	Specifies the maximum number of iterations allowed for Laser simulation at each bias point.
LAS.LOSSES	Specifies the total losses in Equation 9-28.
LAS.MIRROR	Specifies the percentage facet reflectivity (both facets are assumed to have this value of reflectivity) for the mirror loss in Laser. 100% reflectivity is equivalent to no mirror loss.
LAS.NMODE	Specifies the number of transverse modes simulated.
LAS.NX	Specifies the number of discrete samples in the laser mesh in the X direction.
LAS.NY	Specifies the number of discrete samples in the laser mesh in the Y direction.

LAS.OMEGA	Specifies the lasing frequency to be used in Equation 9-1. If model 2 is used for simulation, then this parameter specifies an estimate of the lasing frequency. Instead of using this parameter, PHOTON.ENERGY can be used to specify photon energy.
LAS.REFLECT	Specifies that the Helmholtz equation should be solved with the Y axis (X=0) as an axis of symmetry.
LAS.RF	Specifies the front mirror reflectivity in percent.
LAS.RR	Specifies the rear mirror reflectivity in percent.
LAS.SIN	Specifies an initial photon density in the fundamental lasing mode. This value provides an initial guess for subsequent iterations. This parameter is used only when the single frequency model has been selected.
LAS.SPECSAVE	The spectrum file will be saved after every LAS.SPECSAVE Laser solution steps.
LAS.SPONTANEOUS	Enables the physically based model for spontaneous emission to be used.
LAS.TIMERATE	Specifies that the time dependent photon rate equation will be used in a transient laser simulation.
LAS.TOLER	Specifies the desired accuracy in photon areas.
LAS.TRANS_ENERGY	Specifies the transverse mode energy for mode selection with LAS.ETRANS set.
LAS.XMAX **LAS.XMIN** **LAS.YMAX** **LAS.YMIN**	Specify the maximum and minimum limits on the laser Helmholtz solution mesh.
LAS_MAXCH	Specifies the maximum allowable relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.
LASER	Enables the Laser simulation.
LMODE	Specifies that multiple longitudinal models are to be taken into account during the Laser simulation.
PHOTON.ENERGY	Specifies the energy of photons. If model 2 is used for simulation, this parameter specifies only an initial estimate of the photon energy. Instead of using this parameter, LAS.OMEGA can be used to specify the lasing frequency.
SPEC.NAME	Specifies the name of a spectrum file, which Laser will produce for each bias point if the LMODES parameter has been specified.

Exciton Model Flags

EXCITONS	Specifies that singlet and triplet exciton continuity equations will be solved.
F.KEXCITON	Specifies the file name of a C-Interpreter function that calculates position dependent parameters that effect both the singlet and triplet continuity equations.
F.KSINGLET	Specifies the file name of a C-Interpreter function that calculates position dependent parameters that effect the singlet continuity equation.
F.KTRIPLET	Specifies the file name of a C-Interpreter function that calculates position dependent parameters that effect the triplet continuity equation.
F.GENSINGLET	Specifies the file name of a C-Interpreter function that models the generation of singlet excitons
F.GENTRIPLET	Specifies the file name of a C-Interpreter function that models the generation of triplet excitons.
SINGLET	Specifies that the singlet exciton continuity equation will be solved.
TRIPLET	Specifies that the triplet exciton continuity equation will be solved.

Model Selection Example

This example selects concentration and field dependent mobilities, SRH recombination, and Auger recombination. This is a typical model set for bipolar simulation. The simulation temperature is 290K.

```
MODELS CONMOB FLDMOB SRH AUGER TEMP=290
```

Confirming Model Selection

To echo back model selections, parameters and material constants use

```
MODELS PRINT
```

Note: For the best model selection for different applications, see the Standard Examples set (see also Section 2.4 "Accessing The Examples" for more information about to how to access these examples).

22.38 MQW

MQW defines multiple quantum well structures in an existing device structure. This statement should follow all other structure definition statements (i.e., **MESH**, **REGION**, **ELECTRODE**, and **DOPING**).

Syntax

```
MQW <parameters>
```

Parameter	Type	Default	Units
ACCEPTORS	Real	0.0	cm-3
ALT.SCHRO	Logical	False	
ASTR	Real	0.0	
BSTR	Real	0.0	
CSTR	Real	0.0	
DELTA	Real	0.341	eV
DONORS	Real	0.0	cm-3
DSTR	Real	0.0	
EPSILON	Real	0.0	
ESTR	Real	0.0	
FSTR	Real	0.0	
GAIN0	Real	1.0	
GAMMA0	Real	2.0×10-3	eV
GSCALE	Real	1.0	
ISHIKAWA	Logical	False	
LI	Logical	False	
LORENTZ	Logical	False	
MATERIAL	Character		
MC	Real	0.067	
MHH	Real	0.62	
MLH	Real	0.087	
MSTAR	Real	0.053	
NBS	Integer	15	
NWELL	Integer	1	

NX	Integer	10	
NY	Integer	10	
RESOLVE	Logical	True	
SPONTANEOUS	Logical	False	
STRAIN	Real	0.0	
STABLE	Integer	0	
SY	Real		μm
TAU.IN	Real	3.3×10^{-13}	seconds
TE.MODES	Logical	True	
WB	Real		μm
WW	Real		μm
XCOMP	Real	0.0	
XMAX	Real		μm
XMIN	Real		μm
YAN	Logical	False	
YCOMP	Real	0.0	
YMAX	Real		μm
YMIN	Real		μm

Description

The MQW regions will be added to Atlas when you specify the MQW statement. This statement is necessary to correctly model the recombination and gain effects of these MQW regions.

The multiple quantum well structure is restricted to place the quantum wells only parallel to the X axis (perpendicular to the Y axis). Also, if more than one well is specified, then each well will have the same width defined by the WW parameter. Each barrier between wells will then have the same width defined by the WB parameter.

The MQW structure is located within the rectangle defined by the XMIN, XMAX, YMIN, and YMAX parameters. In the Y direction, the wells are centered within the span of YMIN to YMAX. Make sure that the difference between YMIN and YMAX is larger than the sum of the well width (WW) and the barrier width (WB) multiplied by the number of wells (NWELL). In the X direction, the wells extend the entire length from XMIN to XMAX.

The quantum well material is specified by the MATERIAL parameter, modified by the composition fractions: XCOMP and YCOMP. The barrier material is given by the material specified in the region, where the wells are defined.

This implementation solves the Schrodinger's equation (see Equations 13-2 and 14-3) for each quantum well, in order to calculate the quantum well-bound state energies and carrier

distributions in the wells. You can specify the number of bound states to use for this calculation with the NBS parameter. For laser gain calculations, only the lowest bound state energy in the conduction band and the highest bound state energy in the valence band are used. These bound state energies feed directly into the gain model and spontaneous emission models are shown in Section 3.9 “Optoelectronic Models”

MQW Parameters

ACCEPTORS	Specifies a uniform density of ionized acceptors in the quantum wells.
ALT.SCHRO	Specifies that the alternative Schrodinger solver should be used in the calculations.
ASTR, **BSTR**, **CSTR**, **DSTR**, **ESTR**, and **FSTR**	Specify coefficients to calculate the effective masses in the strained quantum well.
DELTA	Specifies the spin orbital splitting energy in the quantum well.
DONORS	Specifies the uniform density of ionized donors in the quantum wells.
EPSILON	Specifies the high frequency permittivity to used in calculating the gain and radiative recombinations.
GAIN0	Scaling factor for the MQW gain models.
GAMMA0	Specifies the width in eV (Note: GAMMA0=h/TAU.IN).
GSCALE	Specifies a scale factor to be multiplied by the optical gain.
ISHIKAWA	Specifies that the Ishikawa model [141] is used to account for strain effects (see Section 3.9.12 “Ishikawa's Strain Effects Model” for more information).
LI	Specifies that the Li model [49] be used for spontaneous emission in multiple quantum wells. The DELTA, MSTAR, MC and MHH parameters can be used with this model.
LORENTZ	This is the Lorentzian line broadening model for MQW gain. The half width of the Lorentzian shape function is controlled by the GAMMA0 parameter of the **MQW** statement. You can also set the TAU.IN parameter to specify the intraband relaxation time in seconds. For more information about the Lorentzian line broadening model, see Section 3.9.11 “Lorentzian Gain Broadening”.
MATERIAL	Specifies the name of the material to be used in the wells. The barrier material is taken from whatever material is already in the structure at the MQW location.
MC	This is the conduction band effective mass.
MHH	This is the heavy hole effective mass.
MLH	This is the light hole effective mass.
MSTAR	Disperses energy in the conduction band. This doesn’t necessarily equate to the conduction band effective mass, see [363].

NBS	Defines the number of bound states used in solving the Schrodinger equation.
NWELL	Specifies the number of quantum wells.
NX and **NY**	Specify the number of mesh points in the X and Y direction for resolving the MQW structure. Make sure that these are specified large enough to resolve the individual quantum wells. NY is defined as:

$$NY = \left[\frac{YMAX - YMIN}{\langle YGridSpacing(\mu m)\rangle}\right] + 1 \qquad 22\text{-}5$$

RESOLVE	Specifies whether the band edge discontinuities at the well edges are resolved by the device equations.
SPONTANEOUS	Specifies whether the quantum well spontaneous recombination is included in the device continuity equations.
STABLE	Specifies the table row to be used for coefficients to calculate the effective masses in strained quantum wells.
STRAIN	Specifies the strain percentage in the quantum well.
SY	Specifies the maximum mesh spacing to be applied in the Y direction through the quantum wells.
TAU.IN	Specifies the intraband relaxation time in seconds (GAMMA0=h/TAU.IN).
TE.MODES	Specifies whether TE or TM modes will be used in calculating the asymmetrical factors in the LI model.
WB	Specifies the individual barrier width between wells in microns.
WW	Specifies the individual quantum well width in microns.
XCOMP	Defines the X composition fraction for the wells for ternary and quaternary materials.
XMIN, **XMAX**, **YMIN**, **YMAX**	Specify a rectangular box where the quantum wells are to be simulated. The units of these parameters are in microns.
YCOMP	Defines the Y composition fraction for the wells for quaternary materials.
YAN	Specifies that the Yan model [363] be used for spontaneous emission in multiple quantum wells. The DELTA, MSTAR, MC, and MHH parameters can be used with this model (GAINMOD=5).

22.39 NITRIDECHARGE

NITRIDECHARGE specifies the parameters describing the properties of traps in Silicon Nitride in the DYNASONOS model and the PF.NITRIDE model.

The parameters apply to all Silicon Nitride regions.

Syntax

```
NITRIDECHARGE [<params>]
```

Parameter	Type	Default	Units
NT.N	Real	0.0	cm^3
	Real	0.0	cm^3
TAU.N	Real	1.0×10^{300}	s
TAU.P	Real	1.0×10^{300}	s
ELEC.DEPTH	Real	-999.0	eV
HOLE.DEPTH	Real	-999.0	eV
SIGMAN.P	Real	1.0×10^{-15}	cm^2
SIGMAP.N	Real	1.0×10^{-14}	cm^2
SIGMAT.N	Real	1.0×10^{-16}	cm^2
SIGMAT.P	Real	1.0×10^{-14}	cm^2
PF.BARRIER	Real		eV
PF.A	Real	1.0×10^{5}	s
PF.B	Real	1.0e13	Hertz

Description

The NITRIDE charge allows you to set up the properties of electron and hole traps in Siliocn Nitride. It is used in the DYNASONOS model and also the steady state version of the PF.NITRIDE model.

Trap Density

NT.N	Sets the value of the uniform density of electron traps (acceptor states) in the Silicon Nitride.
	Sets the value of the uniform density of hole traps (donor states) in the Silicon Nitride.

Cross-Section

SIGMAP.N	A factor in determining the rate at which the acceptor states capture electrons from the conduction band.
SIGMAN.P	A factor in determining the rate at which the donor states capture holes from the valence band.
SIGMAN.P	A factor in determining the rate at which the acceptor states capture holes from the valence band.
SIGMAP.N	A factor in determining the rate at which the donor states capture electrons from the conduction band.

Trap Depth

ELEC.DEPTH	Energy below conduction band edge of acceptor traps.
HOLE.DEPTH	Energy above valence band of donor traps.

Lifetimes

TAU.N	Lifetime for electron detrapping from acceptor states to conduction band.
TAU.P	Lifetime for hole detrapping from donor states to valence band.
PF.A	Constant of proportionality for the ohmic part of the electron generation rate in the PF.NITRIDE model.
PF.B	Constant of proportionality for electron detrapping rate when Poole-Frenkel model is enabled.
PF.BARRIER	If specified, it overrides ELEC.DEPTH in calculation of electron detrapping rate when Poole-Frenkel model is enabled.

22.40 ODEFECTS

ODEFECTS actives the bandgap defect model for organic polymer materials and sets the parameter values. This model can be used when simulating organic polymer devices using the Organic Display and Organic Solar modules (see Chapter 16 "Organic Display and Organic Solar: Organic Simulators").

Syntax

```
ODEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	False	
DEVICE	Character		
DFILE	Character		
DOPANT	Logical	True	
DOPING.PERCENT	Real		%
EA	Real	0	eV
ED	Real	0	eV
F.ODEFECTS	Character		
F.OTFTACC	Character		
F.OTFTDON	Character		
HA	Real	0	cm^{-3}
HD	Real	0	cm^{-3}
HOPPING	Logical	False	
INT_LIM1	Real	0.0	eV
INT_LIM2	Real	0.0	eV
MATERIAL	Character		
NA	Real	0	cm^{-3}
ND	Real	0	cm^{-3}
NIA	Real	0	cm^{-3}
NID	Real	0	cm^{-3}
NUMA	Real	12	
NUMD	Real	12	

Parameter	Type	Default	Units
REGION	Real	All	
SIGAE	Real	1.0×10^{-16}	cm^2
SIGAH	Real	1.0×10^{-14}	cm^2
SIGDE	Real	1.0×10^{-14}	cm^2
SIGDH	Real	1.0×10^{-16}	cm^2
SIGMAA	Real	0	eV
SIGMAD	Real	0	eV
SIGMAIA	Real	0	eV
SIGMAID	Real	0	eV
STRUCTURE	Character		
TCA	Real	0.0	K
TCD	Real	0.0	K
TFILE	Character		
X.MIN	Real		μm
X.MAX	Real		μm
Y.MIN	Real		μm
Y.MAX	Real		μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

AFILE	Specifies the acceptor state density output file.
CONTINUOUS	Specifies that the continuous defect model will be used.
DEVICE	Specifies which device in MixedMode simulation will apply to the ODEFECTS statement. The synonym for this parameter is STRUCTURE.
DFILE	Specifies the donor state density output file.
DOPANT	Specifies that the defects defined using NA and ND are to be used in the dopant singlet exciton calculation. Only one dopant is allowed per region.
DOPING.PERCENT	Specifies the doping percentage.

EA	Specifies the energy shift between the intrinsic and dopant states for acceptor-like traps.
ED	Specifies the energy shift between the intrinsic and dopant states for donor-like traps.
F.ODEFECTS	Specfies the name of a file containing a C-Interpreter function describing the **ODEFECTS** statements as a function of position.
F.OTFTACC	Specifies the C-Interpreter file for the acceptor states energy distrubution function.
F.OTFTDON	Specifies the C-Interpreter file for the donor states energy distrubution function.
HA	Specifies the total acceptor-like trap density.
HD	Specifies the total donor-like trap density.
HOPPING	Specifies the effective transport hopping energy model is to be used.
INT_LIM1	Specifies the lower limit for the numerical integration of the CONTINUOUS model.
INT_LIM2	Specifies the upper limit for the numerical integration of the CONTINUOUS model.
MATERIAL	Specifies which material will apply to the **ODEFECTS** statement. If a material is specified, then all regions defined as being composed of that material will be affected.
NA	Specifies the total dopant density for acceptor-like traps.
ND	Specifies the total dopant density for donor-like traps.
NIA	Specifies the total intrinsic dopant density for acceptor-like traps.
NID	Specifies the total intrinsic dopant density for donor-like traps.
NUMA	Specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.
NUMD	Specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.
REGION	Specifies the region to which the **ODEFECTS** statement applies.
SIGAE	Specifies the electron capture cross-section for acceptor traps.
SIGAH	Specifies the hole capture cross-section for acceptor traps.
SIGDE	Specifies the electron capture cross-section for donor traps.
SIGDH	Specifies the hole capture cross-section for donor traps.
SIGMAA	Specifies the Gaussian width for the acceptor-like traps dopant distribution.
SIGMAD	Specifies the Gaussian width for the donor-like traps dopant distribution.
SIGMAIA	Specifies the Gaussian width for the acceptor-like traps intrinsic distribution.

SIGMAID	Specifies the Gaussian width for the donor-like traps intrinsic distribution.
STRUCTURE	This is a synonym for **DEVICE**.
TCA	Specifies the characteristic temperature for the acceptor-like trap states.
TCD	Specifies the characteristic temperature for the acceptor-like trap states.
TFILE	Specifies the file name where the acceptor and donor state density distributions will be stored.
X.MIN **X.MAX** **Y.MIN** **Y.MAX** **Z.MIN** **Z.MAX**	Specify the bounding box for the **ODEFECTS** statement.

22.41 OINTDEFECTS

OINTDEFECTS actives the bandgap interface defect model for organic polymer materials and sets the parameter values. This model can be used when simulating organic polymer devices using the OTFT module (see Chapter 16 "Organic Display and Organic Solar: Organic Simulators").

Syntax

```
OINTDEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	False	
DEVICE	Character		
DFILE	Character		
DOPANT	Logical	True	
EA	Real	0	eV
ED	Real	0	eV
F.OINTDEFECTS	Character		
F.OTFTACC	Character		
F.OTFTDON	Character		
HA	Real	0	cm^{-2}
HD	Real	0	cm^{-2}
HOPPING	Logical	False	
INT_LIM1	Real	0.0	eV
INT_LIM2	Real	0.0	eV
INTMATERIAL	Character		
INTNAME	Character		
INTNUMBER	Character		
INTREGION	Character		
MATERIAL	Character		
NA	Real	0	cm^{-3}
ND	Real	0	cm^{-3}
NIA	Real	0	cm^{-3}

Parameter	Type	Default	Units
NID	Real	0	cm^{-3}
NUMA	Real	12	
NUMD	Real	12	
R1MATERIAL	Character		
R1NAME	Character		
R1NUMBER	Integer		
R1REGION	Character		
R2MATERIAL	Character		
R2NAME	Character		
R2NUMBER	Integer		
R2REGION	Character		
REGION	Real	All	
S.I	Logical	True	
S.M	Logical	False	
S.S	Logical	False	
S.X	Logical	False	
SIGAE	Real	1.0×10^{-16}	cm^2
SIGAH	Real	1.0×10^{-14}	cm^2
SIGDE	Real	1.0×10^{-14}	cm^2
SIGDH	Real	1.0×10^{-16}	cm^2
SIGMAA	Real	0	eV
SIGMAD	Real	0	eV
SIGMAIA	Real	0	eV
SIGMAID	Real	0	eV
STRUCTURE	Character		
TCA	Real	0.0	K
TCD	Real	0.0	K
TFILE	Character		

Parameter	Type	Default	Units
X.MIN	Real		μm
X.MAX	Real		μm
Y.MIN	Real		μm
Y.MAX	Real		μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

AFILE	Specifies the acceptor state density output file.
CONTINUOUS	Specifies that the continuous defect model will be used.
DEVICE	Specifies which device in MixedMode simulation will apply to the OINTDEFECTS statement. The synonym for this parameter is STRUCTURE.
DFILE	Specifies the donor state density output file.
DOPANT	Specifies that the defects defined using NA and ND are to be used in the dopant singlet exciton calculation. Only one dopant is allowed per region.
EA	Specifies the energy shift between the intrinsic and dopant states for acceptor-like traps.
ED	Specifies the energy shift between the intrinsic and dopant states for donor-like traps.
F.OINTDEFECTS	Specfies the name of a file containing a C-Interpreter function describing the OINTDEFECTS statements as a function of position.
F.OTFTACC	Specifies the C-Interpreter file for the acceptor states energy distrubution function.
F.OTFTDON	Specifies the C-Interpreter file for the donor states energy distrubution function.
HA	Specifies the total acceptor-like trap density.
HD	Specifies the total donor-like trap density.
HOPPING	Specifies the effective transport hopping energy model is to be used.
INT_LIM1	Specifies the lower limit for the numerical integration of the CONTINUOUS model.
INT_LIM2	Specifies the upper limit for the numerical integration of the CONTINUOUS model.
INTMATERIAL	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, INTMATERIAL="oxide/silicon".
INTNAME	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, INTNAME="channel/substrate".

INTNUMBER	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, `INTNUMBER="2/3"`.
INTREGION	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, `INTREGION="channel/substrate"`.
MATERIAL	Specifies which material will apply to the **OINTDEFECTS** statement. If a material is specified, then all regions defined as being composed of that material will be affected.
NA	Specifies the total dopant density for acceptor-like traps.
ND	Specifies the total dopant density for donor-like traps.
NIA	Specifies the total intrinsic dopant density for acceptor-like traps.
NID	Specifies the total intrinsic dopant density for donor-like traps.
NUMA	Specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.
NUMD	Specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.
R1MATERIAL **R2MATERIAL**	Specifies that the interface models defined on this command should be applied to every interface between the two given materials. For example, `R1MATERIAL=oxide R2MATERIAL=silicon`.
R1NAME **R2NAME**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region name). For example, `R1NAME=channel R2NAME=substrate`.
R1NUMBER **R2NUMBER**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by their region number). For example, `R1NUMBER=2 R2NUMBER=3`.
R1REGION **R2REGION**	Specifies that the interface models defined on this command should be applied to every interface between the two given regions (where the regions are defined by either their region name or their region number). For example, `R1REGION=channel R2REGION=substrate`.
REGION	Specifies the region to which the **OINTDEFECTS** statement applies.
S.I	Specifies that the **OINTDEFECTS** statement should apply to semiconductor-insulator interfaces.
S.M	Specifies that the **OINTDEFECTS** statement should apply to semiconductor-metal interfaces.
S.S	Specifies that the **OINTDEFECTS** statement should apply to semiconductor-semiconductors interfaces.

S.X	Specifies that the **OINTDEFECTS** statement should apply to semiconductor-insulator interfaces, including those to the outside domain.
SIGAE	Specifies the electron capture cross-section for acceptor traps.
SIGAH	Specifies the hole capture cross-section for acceptor traps.
SIGDE	Specifies the electron capture cross-section for donor traps.
SIGDH	Specifies the hole capture cross-section for donor traps.
SIGMAA	Specifies the Gaussian width for the acceptor-like traps dopant distribution.
SIGMAD	Specifies the Gaussian width for the donor-like traps dopant distribution.
SIGMAIA	Specifies the Gaussian width for the acceptor-like traps intrinsic distribution.
SIGMAID	Specifies the Gaussian width for the donor-like traps intrinsic distribution.
STRUCTURE	This is a synonym for **DEVICE**.
TCA	Specifies the characteristic temperature for the acceptor-like trap states.
TCD	Specifies the characteristic temperature for the acceptor-like trap states.
TFILE	Specifies the file name where the acceptor and donor state density distributions will be stored.
X.MIN **X.MAX** **Y.MIN** **Y.MAX** **Z.MIN** **Z.MAX**	Specify the bounding box for the **OINTDEFECTS** statement.

22.42 OPTIONS

OPTIONS sets options for an entire run.

Syntax

```
OPTIONS  [<parameters>]
```

Parameter	Type	Default	Units
CINT.CHAR	Character		
CINT.DLL	Logical	False	
CINT.DOUBLE	Real	0	
CINT.INT	Real	0	
CINT.MODE	Integer		
CINT.PARAM	Character		
ERROR.FILE	Character		
ERROR.MAXIMUM	Real		
ERROR.MINIMUM	Real		
FAIL.QUIT	Logical	False	
NORMAL	Logical	True	
OUTLOGS	Logical	False	
QUIET	Logical	False	
TRANSITION	Logical	False	
VERBOSE	Logical	False	

Description

CINT.CHAR	Specifies the value of a C-Interpreter global character parameter. CINT.PARAM must be defined along with CINT.CHAR and only one C-Interpreter global parameter can be defined per **OPTIONS** statement. The value of CINT.CHAR can be accessed in a C-Interpreter function by using CIget_Global_Char(). The function prototype is `char* CIget_Global_Char(char *param_name);`
CINT.DLL	Specifies that Atlas should treat subsequent C-Interpreter files with the extension ".so" (on UNIX based machines) or ".dll" (on Windows based machines) as shared object libraries instead of C code files. The shared object libraries will be dynamically loaded at run-time.
CINT.DOUBLE	Specifies the value of a C-Interpreter global double parameter. CINT.PARAM must be defined along with CINT.DOUBLE and only one C-Interpreter global parameter can be defined per **OPTIONS** statement. The value of CINT.DOUBLE can be accessed in a C-Interpreter function by using CIget_Global_Double(). The function prototype is `double CIget_Global_Double(char *param_name);`
CINT.INT	Specifies the value of a C-Interpreter global integer parameter. CINT.PARAM must be defined along with CINT.INT and only one C-Interpreter global parameter can be defined per **OPTIONS** statement. The value of CINT.INT can be accessed in a C-Interpreter function by using CIget_Global_Int(). The function prototype is `int CIget_Global_Int(char *param_name);`
CINT.MODE	Specifies the C-Interpreter compilation mode. If CINT.MODE is set to 0, all subsequent C-Interpreter files will be compiled in optimize mode. If CINT.MODE is set to 1, all subsequent C-Interpreter files will be compiled in command-line debug mode. If CINT.MODE is set to 2, all subsequent C-Interpreter files will be compiled in GUI debug mode. The default compliation mode is optimize.
CINT.PARAM	Specifies the name of a C-Interpreter global parameter. Either CINT.CHAR, CINT.INT, or CINT.DOUBLE must be defined along with CINT.PARAM. Only one C-Interpreter global parameter can be defined per **OPTIONS** statement.
ERROR.FILE	Specifies a file that contains the equation errors. This file will be written at every iteration.
ERROR.MAXIMUM	Specifies the maximum error value for the data in ERROR.FILE.
ERROR.MINIMUM	Specifies the minimum value of equation update and RHS errors for ERROR.FILE.
FAIL.QUIT	If enabled, it will cause the simulation to quit if the number of cutbacks exceeds the number specified by the MAXTRAP parameter of the **METHOD** statement and TRAP is enabled.
parameters	This is one or more of the run control parameters described below. These parameters, which are not normally used, specify debugging options.

NORMAL	This is the default specification for run-time output filtering. At this setting Atlas prints out the most relevant information (e.g., mesh statistics, terminal voltages, currents, warnings, and error messages).
OUTLOGS	Specifies that complex index of refraction data is to be automatically output into loggiles similar to the use of OUT.INDEX on the **MATERIAL** statement.
QUIET	Specifies the maximum of filtering of run-time output.
TRANSITION	Specifies that during calculation of bound state energies for optical transitions. The allowable transitions are printed to the run-time output.
VERBOSE	Specifies the minimum filtering of run-time output. You should specify VERBOSE if you want output of residual norms.

Example

```
options cint.param=filename cint.char="a.in"
options cint.param=debug_flag cint.int=1
options cint.param="time" cint.double=10.5


char *filename;
int debug_flag;
double time;
filename = CIget_Global_Char("filename");
debug_flag = CIget_Global_Int("debug_flag");
time = CIget_Global_Double("time");
```

The following syntax can be used to provide extra debugging output.

```
OPTION VERBOSE
```

22.43 OUTPUT

OUTPUT allows you to specify the data that will be stored in standard structure format files.

Syntax

```
OUTPUT <parameters>
```

Parameter	Type	Default	Units
ANGLE	Logical	False	
AREA	Logical	False	
BAND.PARAM	Logical	False	
BAND.TEMP	Logical	false	
CHARGE	Logical	False	
CON.BAND	Logical	False	
CONTACT	Integer		
D.BBT	Logical	False	
DELTAV	Real	0.1	
DEVDEG	Logical	False	
E.FIELD	Logical	True	
E.LINES	Logical	False	
E.MOBILITY	Logical	False	
E.TEMP	Logical	True	
E.VELOCITY	Logical	False	
EFIELD	Logical	True	
EIGENS	Integer	5	
ERRORS	Logical	False	
ESIZEOUT.NEGF	Real	500	
EX.FIELD	Logical	True	
EX.VELOCITY	Logical	False	
EY.FIELD	Logical	True	
EY.VELOCITY	Logical	False	
EZ.FIELD	Logical	True	
EZ.VELOCITY	Logical	False	
FLOWLINES	Logical	False	

Parameter	Type	Default	Units
GAUSSIAN.BAND	Logical	False	
H.MOBILITY	Logical	False	
H.TEMP	Logical	True	
H.VELOCITY	Logical	False	
HCTE.JOULE	Logical	False	
HEI	Logical	False	
HHI	Logical	False	
HX.VELOCITY	Logical	False	
HY.VELOCITY	Logical	False	
HZ.VELOCITY	Logical	False	
IMPACT	Logical	True	
INAME	Character		
INT.CHARGE	Logical	False	
INV.AREA	Logical	False	
INV.ANGLE	Logical	False	
J.CONDUC	Logical	True	
J.DISP	Logical	False	
J.DRIFT	Logical	False	
J.DIFFUSION	Logical	False	
J.ELECTRON	Logical	True	
J.HOLE	Logical	True	
J.TOTAL	Logical	True	
JX.CONDUC	Logical	False	
JX.ELECTRON	Logical	True	
JX.HOLE	Logical	True	
JX.TOTAL	Logical	True	
JY.CONDUC	Logical	False	
JY.ELECTRON	Logical	True	
JY.HOLE	Logical	True	

Parameter	Type	Default	Units
JY.TOTAL	Logical	True	
JZ.CONDUC	Logical	True	
JZ.ELECTRON	Logical	True	
JZ.HOLE	Logical	True	
JZ.TOTAL	Logical	True	
KP.BASISPROJ	Logical	False	
KP.BULKBANDS	Logical	False	
KP.VMATRIX	Logical	False	
KSN	Logical	False	
KSP	Logical	False	
L.TEMPER	Logical	True	
LAS.ASESPEC	Logical	False	
LAS.DBRSPEC	Logical	False	
LAS.GAINSPEC	Logical	False	
LAS.LONGMODES	Logical	False	
LAS.MODALREFL	Logical	False	
LRATIO	Real	1.0	
MINSET	Logical	False	
N.LINES	Integer	20	
NLTAT.GAMMA	Logical	False	
NOISE	Logical	False	
NOISE.IMP	Logical	False	
NOISE.ALL	Logical	False	
OPT.INTENS	Logical	False	
OX.CHARGE	Logical	False	
OLD.AVG	Logical	True	
ONEFILEONLY	Logical	True	
P.QUANTUM	Logical	False	
PERMITTIVITY	Logical	False	

Parameter	Type	Default	Units
PHOTOGEN	Logical	True	
POLAR.CHARGE	Logical	False	
QFN	Logical	True	
QFP	Logical	True	
QSS	Logical	False	
QTUNN.BBT	Logical	False	
QTUNN.EL	Logical	False	
QTUNN.HO	Logical	False	
RECOMB	Logical	True	
SCHOTTKY	Logical	False	
SONOS.RATES	Logical	False	
T.QUANTUM	Logical	False	
TAURN	Logical	False	
TAURP	Logical	False	
TOT.DOPING	Logical	False	
TRAPS	Logical	True	
TRAPS.FT	Logical	False	
U.AUGER	Logical	False	
U.BBT	Logical	False	
U.LANGEVIN	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
U.TRANTRAP	Logical	False	
U.TRAP	Logical	False	
VAL.BAND	Logical	False	
VECTORS	Logical	False	
X.COMP	Logical	True	
Y.COMP	Logical	True	

Description

BAND.PARAM	Specifies that the band parameters (Eg, ni, Nc, Nv, and χ) are included in the standard structure file.
BAND.TEMP	Outputs the following temperature dependent band parameters. • Sqrt(electron concentration * hole concentration) (cm^{-3}) • Conduction band effective density of states (cm^{-3}) • Valence band effective density of states (cm^{-3}) • Energy band gap (eV) • Conduction Band Energy (eV) • Valence Band Energy (eV)
D.BBT	Specifies that the D factor from the BBT.HURKX model will be included in the standard structure file.
CHARGE	Specifies that the net charge will be included in the standard structure file.
CON.BAND	Specifies that the conduction band edge will be included in the standard structure file.
DEVDEG	Causes the distribution of acceptor/donor like traps on the interface, hot electron/hole current density on the interface, and trapped electron/holes to be written to the structure file.
E.FIELD or **EFIELD**	Specifies that total electric field will be included in the standard structure file.
E.LINES	Specifies the electric field lines will be included in the standard structure file.
E.MOBILITY	Specifies that electron mobility will be included in the standard structure file.
E.TEMP	Specifies that the electron temperature will be included in the standard structure file.
E.VELOCITY	Specifies that the total electron velocity will be included in the standard structure file.
EIGENS	Specifies the maximum number of eigen energies and eigen functions to be written to the structure file from a Poisson- Schrodinger solution.
ERRORS	Specifies that the equation errors will be included in the standard structure file.
ESIZEOUT.NEGF	Sets the number of energy grid points in the output file where transmission, DOS, current and carrier densities versus energy are stored.
EX.FIELD	Specifies that the x-component of electric field will be included in the standard structure file.
EX.VELOCITY	Specifies that the x-component of electron velocity will be included in the standard structure file.
EY.FIELD	Specifies that the y-component of electric field will be included in the standard structure file.

EY.VELOCITY	Specifies that the y-component of electron velocity will be included in the standard structure file.
EZ.FIELD	Specifies that the z-component of the electric field will be included in the standard structure file.
EZ.VELOCITY	Specifies that the z-component of the electron velocity will be included in the standard structure file.
FLOWLINES	Specifies that the current flowlines will be included in the standard structure file.
H.MOBILITY	Specifies that hole mobility will be included in the standard structure file.
H.TEMP	Specifies that the hole temperature will be included in the standard structure file.
H.VELOCITY	Specifies that the total hole velocity will be included in the standard structure file.
HCTE.JOULE	Specifies that the volumetrically averaged Joule heating will be included in the standard structure file.
HEI	Specifies that the hot electron current density will be included in the standard structure file.
HHI	Specifies that the hot hole current density will be included in the standard structure file.
HX.VELOCITY	Specifies that the x-component of hole velocity will be included in the standard structure file.
HY.VELOCITY	Specifies that the y-component of hole velocity will be included in the standard structure file.
HZ.VELOCITY	Specifies that the z-component of the hole velocity will be included in the standard structure file.
GAUSSIAN.BAND	Specifies that the Gaussian density of states are included in the standard structure file. These are Ntc, Sigc, GNtc, GSigc, GDEc, Ntv, Sigv, GNtv, GSigv, and GDEv.
IMPACT	Specifies that the impact ionization rate will be included in the standard structure file.
INT.CHARGE	This is a synonym for **QSS**.
J.CONDUC	Specifies that the total conduction current density will be included in the standard structure file.
J.DISP	Specifies that the total displacement current density will be included in the standard structure file.
J.ELECTRON	Specifies that the total electron current density will be included in the standard structure file.

J.HOLE	Specifies that the total hole current density will be included in the standard structure file.
J.TOTAL	Specifies that the total current density will be included in the standard structure file.
JX.CONDUC	Specifies that the x-component of the total conduction current density will be included in the standard structure file.
J.DRIFT	Specifies that the drift current density will be included in the standard structure file.
J.DIFFUSION	Specifies that diffusion current density will be included in the standard structure file.
JX.ELECTRON	Specifies that the x-component of electron current density will be included in the standard structure file.
JX.HOLE	Specifies that the x-component of hole current density will be included in the standard structure file.
JX.TOTAL	Specifies that the x-component of total current density will be included in the standard structure file.
JY.CONDUC	Specifies that the y-component of the total conduction current density will be included in the standard structure file.
JY.ELECTRON	Specifies that the y-component of electron current density will be included in the standard structure file.
JY.HOLE	Specifies that the y-component of hole current density will be included in the standard structure file.
JY.TOTAL	Specifies that the y-component of total current density will be included in the standard structure file.
JZ.CONDUC	Specifies that the z-component of the conduction current density will be included in the standard structure file.
JZ.ELECTRON	Specifies that the z-component of the electron current density will be included in the standard structure file.
JZ.HOLE	Specifies that the z-component of the hole current density will be included in the standard structure file.
JZ.TOTAL	Specifies that the z-component of the total current density will be included in the standard structure file.
KP.BASISPROJ	Activates the output of the projection of eigenstates onto the bulk basis as a function of *k* into the band structure file.
KP.BULKBANDS	Activates the output of bulk band structure of each region treated by the **k.p** models.

KP.VMATRIX	Activates the output of the square of the velocity matrix elements as a function of *k* in the band structure file. The velocity matrix elements are normalized to the bulk velocity matrix element.
KSN	Specifies that electron Scattering Law Exponent is to be written to any saved structure file.
KSP	Specifies that hole Scattering Law Exponent is to be written to any saved structure file.
L.TEMPER	Specifies that lattice temperature will be included in the standard structure file.
LAS.ASESPEC	Specifies that the Amplified Spontaneous Emission spectrum is written to the spectrum file.
LAS.DBRSPEC	Instructs Atlas to also output the reflectivity spectrum for the DBR mirror in the spectrum file specified at the **SAVE** statement. The DBR spectrum is just the reflectivity over the DBR portion of the resonator.
LAS.GAINSPEC	Specifies that the average gain over the region where non-zero gain exists is written to the spectrum file.
LAS.LONGMODES	Specifies that in the log file, the modal quantities are written separately for each longitudinal mode. The default behavior is to add quantities over all longitudinal modes of each transverse mode.
LAS.MODALREFL	Specifies that model reflectance is written to the log file for both front and rear facets.
MINSET	This is the minimum set of data (potential, carrier concentration, and electric field) that will be included in the standard structure file.
NLTAT.GAMMA	Specifies that the Nonlocal trap-assisted-tunneling Gamma factors are written to any saved structure file. The values written have been interpolated onto the device mesh.
ONEFILEONLY	Specifies that the file specified by **ERROR.FILE** will be overwritten at each direction.
OPT.INTENS	Specifies that optical intensity is included in the standard structure file.
OX.CHARGE	Specifies that fixed oxide charge is include in the standard structure file.
P.QUANTUM	Specifies that the Bohm quantum potential is included in the solution file.
PERMITTIVITY	Specifies the dielectric permittivity is saved.
PHOTOGEN	Specifies that the photogeneration rate will be included in the standard structure file.
POLAR.CHARGE	Specifies that polarization charge will be included in the structure file.
QFN	Specifies that the electron quasi-fermi level will be included in the standard structure file.

QFP	Specifies that the hole quasi-fermi level will be included in the standard structure file.
QSS	Specifies that the surface charge will be included in the standard structure file.
QTUNN.BBT	Specifies that the direct quantum tunneling band-to-band current density at each interface node is written to any saved structure file.
QTUNN.EL	Specifies that the direct quantum tunneling electron current density at each interface node is written to any saved structure file. If Fowler-Nordheim tunneling is enabled instead of direct quantum tunneling, then the electron current density due to Fowler-Nordheim tunneling is output.
QTUNN.HO	Specifies that the direct quantum tunneling hole current density at each interface node is written to any saved structure file. If Fowler-Nordheim tunneling is enabled instead of direct quantum tunneling, then the hole current density due to Fowler-Nordheim tunneling is output.
RECOMB	Specifies that the recombination rate will be included in the standard structure file.
SCHOTTKY	Specifies that recombination velocities and barrier lowering will be included in the standard structure file.
SONOS.RATES	This causes the output of four quantities to the structure file. These are the Generation and Recombination rates for the trapped electron states and the trapped hole states in the DYNSAONOS model. This requires the DYNASONOS or BESONOS models.
T.QUANTUM	Specifies that quantum temperatture from the density gradient model is included in the solution file.
TAURN	Specifies that electron relaxation times are to be written to any saved structure file.
TAURP	Specifies that hole relaxation times are to be written to any saved structure file.
TOT.DOPING	Specifies that total doping will be included in the standard structure file.
TRAPS	Specifies that trap density information will be included in the standard structure file.
TRAPS.FT	Specifies that the trap probability of occupation will be included in the standard structure file.
U.AUGER	Specifies that the Auger component of recombination is to be written to solution files.
U.BBT	Specifies that the band to band tunneling rate will be included in the standard structure file.
U.LANGEVIN	Specifies that the Langevin recombination rate will be included in the standard structure file.

U.RADIATIVE	Specifies that the radiative component of recombination is to be written to solution files.
U.SRH	Specifies that the SRH component of recombination is to be written to solution files.
U.TRANTRAP	Specifies that both electron recombination rate and the hole recombination rate into transient traps will be included in the Silvaco standard structure file.
U.TRAP	Outputs the reaction rates of the quantities specified in the **REACTION** statements. The quantities are output in units of cm^{-3} s^{-1} and a positive value means that the quantity is consumed by the reaction. For electrons and holes, this is the same as a recombination rate.
VAL.BAND	Specifies that the valence band edge will be included in the standard structure file.
VECTORS	Specifies that only vector components will be included in the standard structure file.
X.COMP	Specifies that the composition fraction, x, is to be written to solution files.
Y.COMP	Specifies that the composition fraction, y, is to be written to solution files.

Ionization Integral Parameters

INAME	Specifies the name of a contact for which electric field lines are calculated.
CONTACT	Specifies a contact number for which electric field lines are calculated.
LRATIO	Specifies the spacing ratio between adjacent electric field lines. Defaults to 1.0 for uniform spacing.
N.LINES	Specifies the number of electric field lines.
DELTAV	Since the electric field is near zero at the contact, the electric field line calculations begin at a distance from the contact at which the contact voltage has changed by DELTAV. Defaults to 0.1 V.

Note: See Section 22.58 "SOLVE" and the on-line examples for instructions on using ionization integrals.

NOISE Parameters

NOISE	Selects the total local noise source for output.
NOISE.IMP	Selects the impedance fields for output.
NOISE.ALL	Selects everything for output. Currently, the local noise source, the impedance fields, the short-circuit current Green's function, the individual microscopic noise sources, and the individual local noise sources.

Averaging Parameters for Vector Quantities

OLD.AVG	Specifies that the current and field quantities will be averaged using an older algorithm (from version 3.0.0.R and back). By default the new method is used.
ANGLE	Specifies that averaging of current and fields will be weighted by the size of the angle of triangles intersecting at the node.
INV.ANGLE	Specifies that averaging of current and fields will be weighted by the inverse of the size of the angle of triangles intersecting at the node.
AREA	Specifies that averaging of current and fields will be weighted by the areas of triangles intersecting at the node.
INV.AREA	Specifies that averaging of current and fields will be weighted by the inverse of areas of triangles intersecting at the node.

Note: Certain quantities that can be output into the structure file and subsequently displayed using TonyPlot need special mention. These quantities are evaluated within Atlas along the links between grid points. They are represented in the structure file at the grid points themselves. As such these quantities are subject to averaging. In particular, electric field and currents are averaged so as to take into account the vector nature of these values. Mobility is simply summed up over all the links surrounding the grid point and divided by the total number of links. Carrier velocities are derived by dividing the averaged current by the carrier density at the grid point and the fundamental electron charge, q.

An Example of Combining OUTPUT with SOLVE and SAVE

The **OUTPUT** statement is often used in conjunction with the **SAVE** statement. The following statement lines specify that current flowlines and electron velocity components are saved in all subsequent standard structure solution files.

```
OUTPUT FLOWLINES EX.VELO EY.VELO

SOLVE  PREVIOUS V5=2 OUTF=data1.str MASTER

SAVE  OUTF=data2.str
```

22.44 PHOTOGENERATE

PHOTOGENERATE statement specifies photogenetion of carriers inside the device. It ensures TMA compatibility with the PHOTOGEN statement.

Syntax

```
PHOTOGENERATE <parameters>
```

Parameter	Type	Default	Units
A1	Real	0.0	cm^{-3}
A2	Real	0.0	cm^{-3}/um
A3	Real	0.0	cm^{-3}
A4	Real	0.0	um^{-1}
CONSTANT	Real	0.0	cm^{-3}
EXPONENT	Real	0.0	um^{-1}
FACTOR	Real	0.0	cm^{-3}
LINEAR	Real	0.0	cm^{-3}/um
RADIAL	Real	∞	um
R.CHAR	Real	∞	um
X.END	Real	Min x of device	um
X.MAX	Real	Max x of device	um
X.MIN	Real	Min x of device	um
X.ORIGIN	Real	Min x of device	um
X.START	Real	Min x of device	um
Y.END	Real	Max y of device	um
Y.MAX	Real	Max y of device	um
Y.MIN	Real	Min y of device	um
Y.ORIGIN	Real	Min y of device	um
Y.START	Real	Min y of device	um

Description

A1	This is the synonym for **CONSTANT**. Added for TMA compatibility.
A2	This is the synonym for **LINEAR**. Added for TMA compatibility.
A3	This is the synonym for **FACTOR**. Added for TMA compatibility.
A4	This is the synonym for **EXPONENT**. Added for TMA compatibility.
CONSTANT	Specifies the constant photogeneration rate.
EXPONENT	Specifies the exponential coefficient for depth dependent photogeneration.
FACTOR	Specifies the pre-exponential factor for depth dependent photogeneration.
LINEAR	Specifies the linear factor for depth dependent photogeneration.
RADIAL	Specifies the characteristic radial distance for Gaussian dependence of photogeneration rate in the direction perpendicular to the line along which the carriers are injected.
R.CHAR	This is the synonym for **RADIAL**. Added for TMA compatibility.
X.END	Specifies the X coordinate of the end of the line along which photogenerated carriers are injected.
X.MAX	This is the maximum X coordinate at which photogeneration occurs. Generation of carriers is zero for x>X.MAX.
X.MIN	This is the minimum X coordinate at which photogeneration occurs. Generation of carriers is zero for x<X.MIN.
X.ORIGIN	Specifies the X coordinate of the origin of the line along which photogenerated carriers are injected.
X.START	This is the synonym of **X.ORIGIN**. Added for TMA compatibility.
Y.END	Specifies the Y coordinate of the end of the line along which photogenerated carriers are injected.
Y.MAX	This is the maximum Y coordinate at which photogeneration occurs. Generation of carriers is zero for y>Y.MAX.
Y.MIN	This is the minimum Y coordinate at which photogeneration occurs. Generation of carriers is zero for y<Y.MIN.
Y.ORIGIN	Specifies the Y coordinate of the origin of the line along which photogenerated carriers are injected.
Y.START	This is the synonym of **Y.ORIGIN**. Added for TMA compatibility.

22.45 PHOTONICS

Defines new photonic devices as a sub-domains of the entire device structure for stand-alone as well as coupled simulation of optical modes. After the first definition, devices are accessed by specifying their number or name on this statement.

Syntax

```
PHOTONICS [<parameters>]
```

Parameter	Type	Default	Units
ALL	Logical	False	
ARPACK.MODE	Integer	2	
BB.SNAP	Logical	False	
COMPLEX	Logical	True	
COUPLING	Integer		
DISPERSION	Logical	False	
DISP.FILE	Character		
E.FINAL	Real		eV
E.INIT	Real		eV
GETMESH	Logical	False	
MESHRES	Real	0.1	
MESH.INFO	Logical	True	
MESH.TYPE	Integer	1	
MODE.INFO	Logical	False	
NAME	Character		
NDISP	Integer	1	
NEFF	Real		
NEFF.SORT	Character	LR	
NEFF.TYPE	Character	LM	
NMODES	Integer	1	
NMODES.TRANS	Integer	1	
NUM	Integer		
OMEGA	Real		rad/s
OMEGA.FINAL	Real		rad/s

Parameter	Type	Default	Units
OMEGA.INIT	Real		rad/s
PHOTON.ENERGY	Real		eV
PRINT.MESH	Logical	False	
REMESH	Logical	False	
SX	Real	0.1	um
SY	Real	0.1	um
WAVELENGTH	Real		um
WAVELENGTH.FINAL	Real		um
WAVELENGTH.INIT	Real		um
XMAX	Real		um
XMIN	Real		um
YMAX	Real		um
YMIN	Real		um

Description

ALL	Applies the PHOTONICS statement to all devices defined.
ARPACK.MODE	Selects between one of the two arpack modes for the eigenvalue search.
BB.SNAP	If true, snaps the bounding box to the nearest mesh lines of the Atlas mesh. This is useful in avoiding some meshing issues that arise at very high resolution.
COMPLEX	This flag indicates that the complex dielectric function is used in Maxwell equations. If false, the real part of the complex dielectric produced by Atlas is used.
COUPLING	Value of 0 indicates a device uncoupled from electrical bias and is thus solved only once. Value of 1 indicates a coupled device whose modes are solved with updated dielectric at each bias point.
DISPERSION	Indicates that only the eigenvalues are calculated, and results are written to a log file as the calculation progresses (see DISP.FILE) below
DISP.FILE	Specifies the name of the log file storing the results of dispersion calculation. If not specified, the default name is generated from "<name>_dispersion.log", where <name> is the device name.
E.FINAL	Maximum photon energy for solving modes over an interval.
E.INIT	The minimum of the photon energy interval in which to solve for modes.

GETMESH	Indicates that the mesh is generated from nodes on the Atlas mesh within the bounding box specified via XMIN, XMAX, YMIN, and YMAX.
MESHRES	Controls the maximum density of mesh lines near boundaries when MESH.TYPE=2.
MESH.INFO	Prints the summary of the mesh generated for the device.
MESH.TYPE	Selects between two types of meshes. Type 1 meshes each dielectric region uniformly and Type 2 meshes with Chebyshev points, with shifting (MESH.RES) for stability in finite differencing. The Chebyshev points create higher mesh density near region boundaries and less density in the middle of the region.
MODE.INFO	Prints the summary of modes found in table format.
NAME	Selects the device with specified name if NUM is not given, or applies the name to the device if NUM of an existing device is provided. If device is not found, then NAME creates a new device with the name and a unique NUM.
NDISP	The number of wavelengths (or photon energies) in the mode dispersion calculation. The spacing is uniform on wavelength scale if WAVELENGTH.INIT and WAVELENGTH.FINAL is specified. It is uniform on energy scale if E.INIT and E.FINAL or OMEGA.INIT and OMEGA.FINAL are specified.
NEFF	Specifies that NMODES number of modes to be calculated are those that have effective index closest to NEFF than all other modes. This has an effect only if ARPACK.MODE=2.
NEFF.SORT	Specifies the type of sorting applied to the eigenvalues. The possible values are LM (descending magnitude), LR (descending real part), SM (ascending magnitude), SR (ascending real part), LI (descending imaginary), and SI (ascending imaginary).
NEFF.TYPE	Selects the part of the eigenvalue spectrum searched. The possible values are LM, LR, LI, SM, SR, SI, and BE. See ARPACK User's Guide for explanation of these.
NMODES	The number of total modes. This parameter will differ from NMODES.TRANS only in 3D or in coupled devices. Currently,both parameters have the same effect.
NMODES.TRANS	The number of transverse modes (the only kind for 2D at present) to find.
NUM	Selects the device by ID to apply the **PHOTONICS** statement to. If ID is not found among existing devices, NUM will then create a new device with properties on the statement.
OMEGA	Same behavior as PHOTON.ENERGY except specified in rad/s.
OMEGA.FINAL	Same behavior as E.FINAL except specified in rad/s.
OMEGA.INIT	Same behavior as E.INIT except specified in rad/s.

PHOTON.ENERGY	Single photon energy at which to solve for modes. Applies only if E.INIT and E.FINAL are not specified.
PRINT.MESH	Writes the mesh to the file "<name>_mesh.dat", where name is the name specified via the NAME parameter or the internally generated name "PhotonicDevice_<ID>".
REMESH	Performs the meshing of a device (which is specified by NUM or ALL).
SX	Sets the approximate mesh spacing in the x-direction. The actual mesh spacing depends on the region boundaries and MESH.TYPE parameter.
SY	Sets the approximate mesh spacing in the y-direction. The actual mesh spacing depends on the region boundaries and MESH.TYPE parameter.
WAVELENGTH	Single wavelength at which to solve for modes. Applies only if WAVELENGTH.INIT and WAVELENGTH.FINAL are not specified.
WAVELENGTH.FINAL	The maximum wavelength for solving modes over a wavelength range.
WAVELENGTH.INIT	The minimum of the wavelength interval in which to solve for modes.
XMAX	Sets the upper x limit of the rectangular bounding box for the device
XMIN	Sets the lower x limit of the rectangular bounding box for the device.
YMAX	Sets the upper y limit of the rectangular bounding box for the device.
YMIN	Sets the lower y limit of the rectangular bounding box for the device.

Examples

```
PHOTONICS NAME="waveguide1" NUM=1 XMIN=-0.6 XMAX=0.6 YMIN=-0.4
YMAX=0.3
```

defines a device inside the rectangle -0.6 < x < 0.6 and -0.4 < y < 0.3. The device can be accessed later in the same deck with its NUM parameter. Thus,

```
PHOTONICS NUM=1 NMODES=5 NEFF=3.4 WAVELENGTH=0.6
```

sets the device number 1 for calculating 5 transverse modes near the effective index of 3.5 and photon wavelength of 0.6 microns. The solution will be performed at the first **SOLVE** statement that follows. Note that **PHOTONICS** statement itself does not execute a solution. A photonics only solution can be executed by specifying **SOLVE PHOTONICS**.

If you want to use the Atlas mesh within the bounding box too, set the GETMESH flag.

```
PHOTONICS NUM=1 XMIN=-0.6 XMAX=0.6 YMIN=-0.4 YMAX=0.3 GETMESH
```

uses the Atlas mesh lines to define the mesh for optical mode solver.

22.46 PML

The **PML** statement allows definition of perfectly matched layers for finite-difference time-domain (FDTD) analysis.

Syntax

```
PML [parameters]
```

Parameter	Type	Default	Units
ALL	Logical	False	
ALPHA	Real	0.0	1/cm
BACK	Logical	False	
BEAM	Integer	All	
BOTTOM	Logical	True	
DEGREE	Integer	2	
FRONT	Logical	False	
IMAG.INDEX	Real	0.0	
LEFT	Logical	False	
MATERIAL	Character		
NSAMP	Integer		
NXX	Real	0.0	
NYY	Real	0.0	
NZZ	Real	0.0	
PERMEABILITY	Real	1.0	
PERMITTIVITY	Real	1.0	
R90	Real	0.0	
REAL.INDEX	Real	1.0	
TOP	Logical	False	
RIGHT	Logical	False	
WIDTH	Real	0.0	μm
XDIR	Logical	False	
YDIR	Logical	False	

Description

ALL	Specifies that the PML is applied to all sides (TOP, BOTTOM, LEFT, and RIGHT) of the FDTD domain.
ALPHA	Specifies the maximum absorption coefficient of the PML.
BACK	Specifies that the PML is added at the XY plane at the minimum value of z.
BEAM	Specifies the index for the beam that the PML will be applied. If **BEAM** is unspecified, then PML will be applied to all previously defined beams.
BOTTOM	Specifies that the PML is added at the XZ plane at the maximum value of y.
DEGREE	Specifies the degree of Equation 11-68
FRONT	Specifies that the PML is added at the XY plane at the maximum of z.
IMAG.INDEX	Specifies the imaginary part of the complex index of refraction to use in the direction transverse to the PML.
LEFT	Specifies that the PML is added at the YZ plane at the minimum X coordinate.
MATERIAL	Specifies the material name to use to determine the real part of the complex index of refraction versus wavelength for the PML.
NSAMP	Specifies the number of samples in the PML in the direction of the coordinate axis specified by X.DIR or Y.DIR.
NXX **NYY** **NZZ**	Specify the axial indices for an anisotropic PML. These parameters are only used if ANISO is specified on the **BEAM** statement.
PERMEABILITY	Specifies the relative permeability to use within the PML if NSAMP is unspecified it is calculated from the WIDTH.
PERMITTIVITY	Specifies the relative permittivity to use within the PML.
R90	Specifies the desired value of the normal reflection coefficient. This is R(Θ) at Θ=90° in Equation 11-69.
REAL.INDEX	Specifies the real part of the complex index of refraction of the PML.
RIGHT	Specifies that the PML is added at the YZ plane at the maximum X coordinate.
TOP	Specifies that the PML is added at the XZ plane at the minimum value of y.
WIDTH	Specifies the thickness of the PML in microns.
XDIR	Specifies that the PML will be added at the maximum and minimum x coordinates.
YDIR	Specifies that the PML will be added at the maximum and minimum y coordinates.

22.47 PROBE

PROBE allows you to output the value of several distributed quantities to the log file. The value at a specified location or the minimum, maximum, or integrated value within a specified area of the device will be saved to the log file at each bias or time point.

Note: **PROBE** is the most accurate way to determine the value of many parameters calculated by Atlas. Parameters stored on node points in the structure files for TonyPlot are often interpolated and subject to noise.

Syntax

```
PROBE[MIN|MAX|INTEGRATED|x=<n>y=<n>z=<n>
[DIR=<n>]][POLAR=<n>]<parameters>
```

Parameter	Type	Default	Units
ACC.CTRAP	Logical	False	
ACC.TRAP	Logical	False	
ALPHAN	Logical	False	
ALPHAP	Logical	False	
APCURRENT	Logical	False	
AUGER	Logical	False	
AVERAGE	Logical	False	
BACK	Real	maximum z coord	µm
BAND	Integer	1	
BAND.GAP	Logical	False	
BEAM	Integer		
BND.ENER	Logical	False	
BOTTOM	Real	maximum y coord	µm
CAPT.SRH	Logical	False	
CAPT.AUGER	Logical	False	
CAPT.N.CONC	Logical	False	
CAPT.P.CONC	Logical	False	
CHARGE	Logical	False	
COMPLIANCE	Real		
CON.BAND	Logical	False	
CONCACC.CTRAP	Logical	False	

Parameter	Type	Default	Units
CONCACC.TRAP	Logical	False	
CONCDON.CTRAP	Logical	False	
CONCDON.TRAP	Logical	False	
CONDUCTIVITY	Logical	False	
CURDVSE	Logical	False	
DENSVSE	Logical	False	
DEVICE	Character		
DIR	Real	0.0	degrees
DON.CTRAP	Logical		
DON.TRAP	Logical		
DOPANT.EXCITON	Logical	False	
DOPRAD.EXCITON	Logical	False	
DOSVSE	Logical	False	
E.TRANTRAP	Logical	False	
EMAX	Real		eV
EMIN	Real		eV
ENERGY	Real		eV
EXCITON	Logical	False	
FILENAME	Character	"TransvsE"	
FIELD	Logical	False	
FRONT	Real	minimum z coord	µm
FTACC.CTRAP	Logical	False	
FTACC.TRAP	Logical	False	
FTDON.CTRAP	Logical	False	
FTDON.TRAP	Logical	False	
FT.MSC	Logical	False	
GENERATION	Logical	False	
GR.HEAT	Logical	False	
H.ATOM	Logical	False	

Parameter	Type	Default	Units
H.CONC	Logical	False	
H.MOLE	Logical	False	
H.TRANTRAP	Logical	False	
INT.CHARGE	Logical	False	
INTEGRATED	Logical	False	
INTENSITY	Logical	False	
J.CONDUCTION	Logical	False	
J.DISP	Logical	False	
J.ELECTRON	Logical	False	
J.HOLE	Logical	False	
J.PROTON	Logical	False	
J.TOTAL	Logical	False	
JOULE.HEAT	Logical	False	
KX	Logical	False	
KY	Logical	False	
KZ	Logical	False	
LAYER.ABSO	Logical	False	
LAYER.REFL	Logical	False	
LAYER.TRAN	Logical	False	
LMAX	Real		μm
LMIN	Real		μm
LANGEVIN	Logical	False	
LASER.INTENSITY	Logical	False	
LASER.GAIN	Logical	False	
LASER.MODE	Real		
LAT.TEMP	Logical	False	
LEFT	Real	minimum x coord	μm
LUMINOUS.INTENSITY	Logical	False	
MATERIAL	Character		

Parameter	Type	Default	Units
MAX	Logical	False	
MIN	Logical	False	
MAX.XITION	Logical	False	
MIN.XITION	Logical	False	
MSC.STATE	Integer	1	
N.CONC	Logical	False	
N.MOB	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
NWELL	Logical	False	
P.CONC	Logical	False	
P.MOB	Logical	False	
P.TEMP	Logical	False	
PERMITTIVITY	Logical	False	
PHOTOGEN	Logical	False	
POLAR	Real	90	degrees
POLARIZATION	Logical	False	
POTENTIAL	Logical	False	
PT.HEAT	Logical	False	
PWELL	Logical	False	
QF	Logical	False	
QFN	Logical	False	
QFP	Logical	False	
R.BBT	Logical	False	
R.TRAP	Logical	False	
RADIATIVE	Logical	False	
RAUGER	Logical	False	
REACTION.ELEC	Logical	False	
REACTION.HOLE	Logical	False	

Parameter	Type	Default	Units
REACTION.SP1	Logical	False	
REACTION.SP2	Logical	False	
REACTION.SP3	Logical	False	
RECOMBIN	Logical	False	
REGION	Integer		
RESISTIVITY	Logical	False	
RIGHT	Real	maximum x coord	µm
RNAME	Character		
SINGLET.CURRENT	Logical	False	
SINGLET.VELOCITY	Logical	False	
SONOS.CHARGE	Logical	False	
SPECIES1	Logical	False	
SPECIES2	Logical	False	
SPECIES3	Logical	False	
SPONT.EMISS	Logical	False	
SRH	Logical	False	
STATE	Integer	1	
STRUCTURE	Character		
THERMAL	Logical	False	
THETA	Real	0	degrees
TOTAL.HEAT	Logical	False	
TOP	Real	minimum y coord	µm
TRIPLET.CURRENT	Logical	False	
TRIPLET.VELOCITY	Logical	False	
U.SP1TRAP	Logical	False	
U.SP2TRAP	Logical	False	
U.SP3TRAP	Logical	False	
TRANSMISSION	Logical	False	
V.MAG	Logical	False	

Parameter	Type	Default	Units
V.THETA	Logical	False	
V.X	Logical	False	
V.XANG	Logical	False	
V.Y	Logical	False	
V.Z	Logical	False	
V.ZANG	Logical	False	
V.ZETA	Logical	False	
VAL.BAND	Logical	False	
VEL.ELECTRON	Logical	False	
VEL.HOLE	Logical	False	
WAVE.FUN	Logical	False	
WAVELENGTH	Logical	False	
X	Real		μm
X.MAX	Real	maximum x coord	μm
X.MIN	Real	minimum x coord	μm
Y	Real	maximum x coord	μm
Y.MAX	Real	maximum x coord	μm
Y.MIN	Real	minimum y coord	μm
Z.MAX	Real	maximum z coord	μm
Z.MIN	Real	minimum z coord	μm
ZETA	Real	0	degrees

Description

ACC.CTRAP	Specifies that the continuous acceptor trap density of states is to be probed.
ACC.TRAP	Specifies that the acceptor trap density of states is to be probed. If multiple trap levels are present the ENERGY parameter should be set to the desired energy level that is to be probed.
ALPHAN	Specifies that the electron impact ionization coefficient is to be probed. The DIR parameter should also be specified as this is a vector quantity.
ALPHAN and ALPHAP	Specify that the electron or hole impact ionization constant alpha is to be probed.

ALPHAP	Specifies that the hole impact ionization coefficient is to be probed. The DIR parameter should also be specified as this is a vector quantity.
APCURRENT	Specifies that the available photocurrent is probed.
AUGER	Specifies that Auger recombination rate is probed.
AVERAGE	Similar to INTEGRATED but calculates the average value of the probe over the specified region.
BAND	This is the number of the electron or hole band characterized by effective mass. Generally, the electron valley or hole band is characterized by values of effective mass (mx, my, mz) along main axes, which are given below for different values of BAND on the PROBE statement and the SCHRODINGER, P.SCHRODINGER, NUM.DIRECT, NUM.BAND and SP.DIR parameters on the **MODELS** statement. (mx, my, mz) (MT1, ML, MT2) (BAND=1 AND SCHRODINGER AND NUM.DIRECT>1) (MT2, MT1, ML) (BAND=2 AND SCHRODINGER AND NUM.DIRECT>1) (ML, MT2, MT1) (BAND=3 AND SCHRODINGER AND NUM.DIRECT=3) (MC, MC, MC) (BAND=1 AND SCHRODINGER AND NUM.DIRECT=1) (MHH, MHH, MHH) (BAND=1 AND P.SCHRODINGER AND NUM.BAND>1) (MLH, MLH, MLH) (BAND=2 AND P.SCHRODINGER AND NUM.BAND>1) (MSO, MSO, MSO) (BAND=3 AND P.SCHRODINGER AND NUM.BAND=3) (MV, MV, MV) (BAND=1 AND P.SCHRODINGER AND NUM.BAND=1) If BAND<0 , it defaults to BAND=1. If BAND>NUM.DIRECT and SCHRODINGER, it defaults to BAND=NUM.DIRECT. If BAND>NUM.BAND and P.SCHRODINGER, it defaults to BAND=NUM.BAND. For materials with one band but with anisotropic electron effective mass, you can change the effective mass in the case of BAND=1 and SCHRODINGER=1 and NUM.DIRECT=1 using the SP.DIR parameter on the **MODELS** statement.
BAND.GAP	Specifies that the bandgap is to be probed. If the probe is located in a equation well, the probe returns the energy associated with the minimum allowable transition.

BEAM	Specifies a beam number to focus the application of the probe to that specific beam.
BND.ENER	Probes the bound state energy characterized by the BAND and STATE parameters.
CAPT.SRH	Probes the Quantum Confined SRH recombination rate in a quantum well in Capture-Escape model.
CAPT.AUGER	Probes the Quantum Confined Auger recombination rate in a quantum well in Capture-Escape.
CAPT.N.CONC	Probes the total (Quantum confined + bulk) electron concentration in Capture-Escape model.
CAPT.P.CONC	Probes the total (Quantum confined + bulk) hole concentration in Capture-Escape model.
CHARGE	Specifies that the net charge is to be probed.
COMPLIANCE	Specifies the probed quantity compliance value. When the probed quantity reaches the specifies value, the **SOLVE** statement (or **.DC** or **.TRAN** statement in MixedMode) will terminate and will execute the next statement.
CON.BAND	Probes the conduction band.
CONCACC.CTRAP	Specifies that the continuous acceptor trap concentration is to be probed.
CONCACC.TRAP	Specifies that the acceptor trap concentration is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all acceptor energy levels will be probed.
CONCDON.CTRAP	Specifies that the continuous donor trap concentration is to be probed.
CONCDON.TRAP	Specifies that the donor trap concentration is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all donor energy levels will be probed.
CONDUCTIVITY	Specifies that the conductivity of metals and semiconductors is probed. The conductivity in insulators is returned as 0.
CURDVSE	Stores current density versus energy for each solution of NEGF solver.
DENSVSE	Stores carrier density versus energy for each solution of NEGF solver.
DEVICE	Specifies which device in MixedMode simulation the **PROBE** statement should apply to. The synonym for this parameter is **STRUCTURE**.
DIR	Specifies the direction relative to the X axis in degrees associated with certain directed quantities. These quantities include FIELD, N.MOB, P.MOB, and POLARIZATION.
DON.CTRAP	Specifies that the continuous donor trap density of states is to be probed.

DON.TRAP	Specifies that the donor trap density of states is to be probed. If multiple trap levels are present the ENERGY parameter should be set to the desired energy level that is to be probed.
DOPANT.EXCITON	Specifies that the singlet dopant exciton density is to be probed.
DOPRAD.EXCITON	Specifies that the singlet dopant exciton radiative rate is to be probed.
DOSVSE	Stores density of states (DOS) versus energy for each solution of NEGF solver.
E.TRANTRAP	Specifies that the electron recombination rate into transient traps is to be probed.
EMAX and **EMIN**	Specifies the range of energies to search for peak emission wavelength when the WAVELENGTH parameter is set on the PROBE statement.
FILENAME	Specifies the name of file where Transmission. DOS, carrier, and current density versus energy are stored.

Note: The algorithm used finds the triangle in the mesh containing the specified X and Y values. Then the value of the DIR parameter is used to find which edge of the triangle lies in the direction nearest that value.

EXCITON	Specifies that the exciton density is probed.
FIELD	Specifies that a value of electric field is probed. The DIR parameter should also be specified if FIELD is used.
FT.MSC	Probes the occupation fraction of the internal state of the multistate traps given by MSC.STATE.
FTACC.CTRAP	Specifies that the continuous trap probability of occupation is to be probed.
FTACC.TRAP	Specifies that the acceptor trap probability of occupation is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all acceptor energy levels will be probed.
FTDON.CTRAP	Specifies that the continuous donor trap probability of occupation is to be probed.
FTDON.TRAP	Specifies that the donor trap probability of occupation is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all donor energy levels will be probed.
GENERATION	Specifies that the generation rate due to impact ionization is probed.
GR.HEAT	Requests a **PROBE** of Generation-Recombination heat in Giga.
H.ATOM	Specifies that the atomic hydrogen density is to be probed.
H.CONC	Specifies that proton concentration is probed.

H.MOLE	Specifies that the molecular hydrogen density is to be probed.
H.TRANTRAP	Specifies that the hole recombination rate into transient traps is to be probed.
INT.CHARGE	Specifies that the total interface charge is to be probed.
INTEGRATED	Specifies that the probed value is to be integrated over all mesh points inside a box defined with the parameters X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN and Z.MAX.
INTENSITY	Evaluates optical intensity of an optical beam in Luminous. If this is a multispectral source, this value accounts for the subsampling, while the intensity already in the logfile will only reflect the integrated input spectrum.
J.CONDUCTION	Requests a **PROBE** of Conduction current (J.ELEC+J.HOLE).
J.DISP	Requests a **PROBE** of Displacement current.
J.ELECTRON	Requests a **PROBE** of Electron Current density.
J.HOLE	Requests a **PROBE** of Hole Current density.
J.PROTON	Specifies that the proton current density is probed.
J.TOTAL	Requests a **PROBE** of Total Current density (J.TOTAL+J.DISP).
JOULE.HEAT	Requests a **PROBE** of Joule-Heating in Giga.
KX, **KY**, and **KZ**	Specify directions in k space associated with probing conduction and valence band bound state energies. See also **NWELL** and **PWELL**.
LANGEVIN	Specifies that the Langevin recombination rate is probed.
LAYER.ABSO LAYER.REFL LAYER.TRAN	Specify that the absorption, reflection, or transmission coefficient is probed for Luminous ray trace or transfer matrix. When specifying one of these values, you also need to specify the **BEAM** and **REGION** parameters.
LASER.GAIN	Specifies that the probe will operate on the Laser optical gain. For more information about Laser, see Chapter 9 “Laser: Edge Emitting Simulator”.
LUMINOUS.INTENSITY	Specifies that the probe will operate on the Laser optical intensity. For more about information Laser, see Chapter 9 “Laser: Edge Emitting Simulator”.
LASER.MODE	Specifies the optical mode for LASER.INTENSITY and LASER.GAIN.
LAT.TEMP	Specifies that the probe will operate on lattice temperature.
LMAX, **LMIN**	Specifies the range of wavelengths to search for peak emission wavelength when the WAVELENGTH parameter is set on the **PROBE** statement.

LUMINOUS.INTENSITY	Specifies that the probe will operate on the Luminous optical intensity. For more information about Luminous, see Chapter 11 "Luminous: Optoelectronic Simulator".
MATERIAL	This is assigned to a specific material name that will act to direct the probe to only address regions composed of the specified material.
MAX	Specifies that the probe will find the maximum value on the mesh.
MIN	Specifies that the probe will find the minimum value on the mesh.
MAX.XITION	Specifies that the maximum allowable transition between bound state energies is to be probed. If there are no allowable transitions between bound state energies, the band gap is probed.
MIN.XITION	Specifies that the minimum allowable transition between bound state energies is to be probed. If there are no allowable transitions between bound state energies, the band gap is probed.
MSC.STATE	Specifies which of the 5-internal states of the multistate traps are probed.
NAME	Sets a character string that allows you to specify the description displayed by TonyPlot.
N.CONC	Specifies that the probe will operate on electron concentration.
N.MOB	Specifies that the probe will operate on the electron mobility. The DIR parameter should also be specified if N.MOB is used.
N.TEMP	Specifies that the probe will operate on electron temperature.
NWELL	Specifies that the a specified conduction bound state energy is to be probed. To specify the bound state, you must also specify which bound state is to be probed with the STATE parameter (where 0 specifies the lowest bound state) and the direction in k space using the KX, KY or KZ parameters.
P.CONC	Specifies that the probe will operate on hole concentration.
P.MOB	Specifies that the probe will operate on the hole mobility. The DIR parameters should also be specified if P.MOB is used.
P.TEMP	Specifies that the probe will operate on hole temperature.
PERMITTIVITY	Specifies that material permittivity is probed.
PHOTOGEN	Specifies that photogeneration or SEU generation rate is probed.
POLAR	Specifies the direction of the desired component of a vector probe along with DIR. POLAR is the angle from the z-axis. This cannot be used with THETA/ZETA parameters.
POLARIZATION	Specifies that the probe will operate on ferroelectric polarization.The DIR parameter should also be specified if POLARIZATION is used.
POTENTIAL	Specifies that the probe will operate on electrostatic potential.

PT.HEAT	Requests a **PROBE** of Peltier-Thomson Heat in Giga.
PWELL	Specifies that the a specified valence bound state energy is to be probed. To specify the bound state, you must also specify which bound state is to be probed with the STATE parameter (where 0 specifies the highest bound state) and the direction in k space using the KX, KY or KZ parameters.
QF	Specifies that the fixed oxide charge is to be probed.
QFN	Specifies that the probe will operate on the electron quasi-Fermi level.
QFP	Specifies that the probe will operate on the hole quasi-Fermi level.
R.BBT	Specifies tha the band-to-band tunneling rate is to be probed.
R.TRAP	Specifies that the trap recombination is to be probed.
RADIATIVE	Specifies that the probe will operate on radiative recombination rate.
REACTION.ELEC	Specifies that the **PROBE** will give the consumption rate of electrons in a **REACTION**.
REACTION.HOLE	Specifies that the **PROBE** will give the consumption rate of holes in a **REACTION**.
REACTION.SP1	Specifies that the **PROBE** will give the consumption rate of ionic species 1 in a **REACTION**.
REACTION.SP2	Specifies that the **PROBE** will give the consumption rate of ionic species 2 in a **REACTION**.
REACTION.SP3	Specifies that the **PROBE** will give the consumption rate of ionic species 3 in a **REACTION**.
RECOMBIN	Specifies that the probe will operate on net recombination rate.
REGION	Limits the application of the probe to a specific region of a specified index.
RESISTIVITY	Specifies that the resistivity of metals and semiconductors is probed. The resistivity in insulators is returned as 0.
SINGLET.CURRENT	Specifies that the probe will calculate the flow of singlet excitons. This is a vector probe.
SINGLET.VELOCITY	Specifies that the probe will calculate the velocity of singlet excitons. This is a vector probe.
SONOS.CHARGE	Specifies that the net trapped carrier density in a Silicon-Nitride region is to be probed. The net density is defined as the electron density minus the hole density.
SPECIES1	Specifies that the **PROBE** will operate on ionic species 1.
SPECIES2	Specifies that the **PROBE** will operate on ionic species 2.
SPECIES3	Specifies that the **PROBE** will operate on ionic species 3.

SPONT.EMISS	Probes electron-hole band-band optical transition rate.
SRH	Specifies that the probe will operate on SRH recombination rate.
STATE	This is the number of the bound state. It can take values from 1 to the maximum number of bound states, which is specified by the EIGENS parameter on the **MODELS** statement.
STRUCTURE	This is a synonym for **DEVICE**.
THERMAL	Specifies that the structure temperature calculated using the Thermal 3D module will be probed (see Chapter 18 "Thermal 3D: Thermal Packaging Simulator").
THETA	Specifies the direction of the desired component of a vector probe along with ZETA. THETA is the angle in the xz-plane from the x-axis towards the z-axis. This cannot be used with DIR/POLAR parameters.
TOTAL.HEAT	Requests a PROBE of Total heat in Giga (GR.HEAT+PT.HEAT+JOULE.HEAT).
TRANSMISSION	Stores Transmission versus energy for each solution of NEGF solver. See Chapter 14 "Quantum: Quantum Effect Simulator" for more information.
TRIPLET.CURRENT	Specifies that the probe will calculate the flow of triplet excitons. This is a vector probe.
TRIPLET.VELOCITY	Specifies that the probe will calculate the velocity of triplet excitons. This is a vector probe.
U.SP1TRAP	Gives the overall recombination rate into multistate traps of ionic species 1, or atomic hydrogen if the atomic hydrogen transport model (H.ATOM) has been enabled on the **MODELS** statement.
U.SP2TRAP	Gives the overall recombination rate into multistate traps of ionic species 2, or molecular hydrogen if the atomic hydrogen transport model (H.ATOM) has been enabled on the **MODELS** statement.
U.SP3TRAP	Gives the overall recombination rate into multistate traps of ionic species 3.
VAL.BAND	Probes the valence band.
V.MAG	Returns the magnitude of a vector probe; a DIR parameter will be ignored.
V.THETA	Returns the angle (in degrees) of a vector probe in the xz-plane relative to the x-axis.
V.X	Returns the x-component of a vector probe; a DIR parameter will be ignored.
V.XANG	Returns the angle (in degrees) of a vector probe in the xy-plane relative to the x-axis; a DIR parameter would be ignored.
V.Y	Returns the y-component of a vector probe; a DIR parameter would be ignored.

V.Z	Returns the z-component of a vector probe; a DIR parameter would be ignored.
V.ZANG	Returns the angle (in degrees) of a vector probe relative to the z-axis; a DIR parameter would be ignored.
V.ZETA	Returns the angle (in degrees) of a vector probe relative to the y-axis.
VEL.ELECTRON	Specifies that the probe will operate on electron velocity.
VEL.HOLE	Specifies that the probe will operate on hole velocity.
WAVE.FUN	Probes the wavefunction characterized by the BAND and STATE parameters.
WAVELENGTH	Specifies that the radiative emission wavelength is probed.
ZETA	Specifies the direction of the desired component of a vector probe along with THETA . ZETA is the angle from the y-axis. This cannot be used with the DIR/POLAR parameters.

Region Parameters

BACK	This is a synonym for **Z.MAX**.
BOTTOM	This is a synonym for **Y.MAX**.
FRONT	This is a synonym for **Z.MIN**.
LEFT	This is a synonym for **X.MIN**.
RIGHT	This is a synonym for **X.MAX**.
TOP	This is a synonym for **Y.MIN**.
X, Y	Specifies the location of a point probe.
X.MAX	Specifies the maximum X coordinate for the minimum, maximum, or integrated probe.
X.MIN	Specifies the minimum X coordinate for the minimum, maximum, or integrated probe.
Y.MAX	Specifies the maximum Y coordinate for the minimum, maximum, or integrated probe.
Y.MIN	Specifies the minimum Y coordinate for the minimum, maximum, or integrated probe.
Z.MAX	Specifies the maximum Z coordinate for the minimum, maximum, or integrated probe.
Z.MIN	Specifies the minimum Z coordinate for the minimum, maximum, or integrated probe.

Example of Probing the Maximum Value

The following line will cause the maximum electron concentration on the grid to be output to the log file:

```
PROBE NAME=peak_electrons MAX N.CONC
```

Probing at a Location Example

This syntax will cause the potential at the location X=0.5 Y=0.1 to be output to the log file.

```
PROBE NAME=mypotential X=0.5 Y=0.1 POTENTIAL
```

Vector Quantity Example

For vector quantities, we can only probe a single value per **PROBE** statement. The desired component is indicated with the direction parameter, DIR, a specific Cartesian component with V.X, V.Y, or V.Z, or a specific Spherical Polar component with V.MAG, V.XANG, or V.ZANG. These two lines allow a lateral mobility and vertical field in a MOSFET.

```
PROBE NAME=channel_mobility x=1 y=0.001 N.MOB DIR=0

PROBE NAME=channel_field x=1 y=0.001 FIELD DIR=90
```

In Atlas 3D, you can add a POLAR parameter to the **PROBE** statement. Its default value is 90°, which means the direction lies within the XY plane. POLAR gives the angle respect to the Z axis. A value of zero means that the probed direction is along the Z axis. A value of POLAR between 0 and 90° will **PROBE** along a direction vector with a z-component of COS(POLAR).

```
PROBE NAME=electroncurrent dir=45 polar=60 j.electron  x=0.5 y=0.5
z=0.0
```

This probes the electron current at the specified position along the direction (0.61237, 0.61237, 0.5).

22.48 QTREGION

This allows you to specify one or more polygonal regions as regions where non-local band-to-band tunneling will occur. Each **QTREGION** statement sets up a quadrilateral region. The polygon is built up from one or more quadrilaterals. Atlas deletes the mesh inside the polygonal regions and replaces it with another, which is more suitable for the non-local band-to-band tunneling calculation. The modified mesh is also the mesh to be outputted by any **SAVE** statements.

Syntax

```
QTREGION NUMBER [X1 Y1] X2 Y2 X3 Y3 [X4 Y4]
PTS.TUNNEL| STNL.BEG STNL.END | F.RESOLUTION
PTS.NORMAL | SNRM.BEG SNRM.END
```

Parameter	Type	Default	Units
DEVICE	Character		
F.RESOLUTION	Character		
NUMBER	Real	1	
PTS.NORMAL	Real	-1	
PTS.TUNNEL	Real	-1	
SNRM.BEG	Real	0	μm
SNRM.END	Real	0	μm
STNL.BEG	Real	0	μm
STNL.END	Real	0	μm
STRUCTURE	Character		
X1	Real	0	μm
X2	Real	0	μm
X3	Real	0	μm
X4	Real	0	μm
Y1	Real	0	μm
Y2	Real	0	μm
Y3	Real	0	μm
Y4	Real	0	μm

Description

DEVICE	Specifies which device the statement applies in the mixed mode simulation. The synonym for this parameter is **STRUCTURE**.
F.RESOLUTION	Specifies a text file that contains a column of numbers between 0 and 1. This determines the mesh spacing in the tunneling direction and is an alternative to PTS.TUNNEL. The numbers give the mesh points relative to the corners, 0.0 maps to the starting corner, and 1.0 maps to the ending corner. Intermediate mesh points are calculated according to a linear mapping from the supplied coordinates in [0,1].
NUMBER	This is the index of the polygonal region that the **QTREGION** statement applies to.
PTS.NORMAL	This is the number of points in quadrilateral in direction perpendicular to tunneling direction. This dictates a regular mesh spacing along the sides from corner 1 to corner 2 and from corner 3 to corner 4. See Figure 3-9 for an example.
PTS.TUNNEL	This is the number of points in quadrilateral in direction parallel to tunneling direction. This dictates a regular mesh spacing along the sides from corner 1 to corner 4 and from corner 2 to corner 3.
SNRM.BEG	Along with SNRM.END, is one way of specifying the mesh spacing in the direction normal to tunneling. SNRM.BEG specifies the mesh spacing at the corner 1 (4). SNRM.END specifies it at the corner 2 (3).
SNRM.END	See **SNRM.BEG**.
STNL.BEG	Along with STNL.END, is one way of specifying the mesh spacing in the direction parallel to tunneling. STNL.BEG specifies the mesh spacing at the corner 1 (2). STNL.END specifies it at the corner 4 (3).
STNL.END	See **STNL.BEG**.
STRUCTURE	This is a synonym for **DEVICE**.
X1	This is the X coordinate of the first corner of the quadrilateral.
X2	This is the X coordinate of the second corner of the quadrilateral.
X3	This is the X coordinate of the third corner of the quadrilateral.
X4	This is the X coordinate of the fourth corner of the quadrilateral.
Y1	This is the Y coordinate of the first corner of the quadrilateral.
Y2	This is the Y coordinate of the second corner of the quadrilateral.
Y3	This is the Y coordinate of the third corner of the quadrilateral.
Y4	This is the Y coordinate of the fourth corner of the quadrilateral.

Note: The corners must be specified in counter-clockwise order.

Examples

The following four **QTREGION** statements make up a polygonal region similar in shape to that in Figure 3-10. You need to specify all four corners in the first statement.

Subsequent statements may omit (x1, y1) and (x4, y4). In the tunneling direction, only one specification of mesh spacing in the tunneling direction is allowed per polygonal region. In this case, a regular spacing with 51 points is requested.

```
QTREGION NUMBER=1 PTS.TUNNEL=51 X1=0.0 Y1=0.45 X2=0.9 Y2=0.45
X3=0.9 Y3=0.55 X4=0.0 Y4=0.55 SNRM.BEG=0.2 SNRM.END=0.02


QTREGION NUMBER=1 PTS.NORMAL=9 X1=0.9 Y1=0.45 X2=0.94 Y2=0.38
X3=1.03 Y3=0.45 X4=0.9 Y4=0.55


QTREGION NUMBER=1 PTS.NORMAL=9 X2=0.975 Y2=0.25
X3=1.075 Y3=0.25


QTREGION NUMBER=1 PTS.NORMAL=3 X2=0.975 Y2=0.0
X3=1.075 Y3=0.0 SNRM.BEG=0.05 SNRM.END=0.1
```

The following example uses a data file, `mesh.dat`, to specify the mesh spacing in the tunneling direction.

```
QTREGION NUMBER=2 X1=0.0 Y1=1.55 X2=0.8 Y2=1.55 \
X3=0.8 Y3=1.65 X4=0.0 Y4=1.65 SNRM.BEG=0.2 SNRM.END=0.02
F.RESOLUTION=MESH.DAT
```

The `mesh.dat` file should have a format such as

```
0.0
0.2
0.4
0.45
0.5
0.55
0.6
0.8
1.0
```

In the direction normal to tunneling, the spacing is 0.2 at one end and 0.02 at the other.

22.49 QTX.MESH, QTY.MESH

QTX.MESH and **QTY.MESH** allow you to specify a rectangular mesh in order to calculate quantum tunneling current in either the X direction or Y direction. It is required in some situations for the direct quantum tunneling model for gate insulators. It is also required for non-local band-to-band and trap assisted tunneling models.

Syntax

```
QTX.MESH NODE=<n> LOCATION=<n>

QTX.MESH SPACING=<n> LOCATION=<n>

QTY.MESH NODE=<n> LOCATION=<n>

QTY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE	Specifies the mesh line index. These mesh lines are assigned consecutively.
LOCATION	Specifies the location of the grid line.
SPACING	Specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

Example

```
qtx.mesh loc=0.0 spac=0.25
qtx.mesh loc=1.0 spac=0.25


qty.mesh loc=1.5 spac=0.05
qty.mesh loc=2.0 spac=0.01
qty.mesh loc=3.0 spac=0.01
qty.mesh loc=3.5 spac=0.05
```

will set up a mesh in the rectangle bounded by x=0.0, x=1.0, y=1.5 and y=3.5. The mesh is finer in the Y direction because it is intended to calculate the tunneling current in this direction.

22.50 QUIT

QUIT stops execution of Atlas.

Syntax

QUIT

Synonyms

STOP

END

EXIT

Description

The **QUIT** statement may be placed anywhere in an input file. Atlas will stop execution upon encountering the **QUIT** statement. All input lines after the occurrence of the QUIT statement will be ignored for that execution of Atlas.

Note: To quit and immediately restart Atlas inside of DeckBuild, use the **GO ATLAS** statement. Full details on the **GO** syntax statement are found in the DeckBuild User's Manual.

22.51 REACTION

REACTION specifies the desired reactions between ionic species and between ionic species and free carriers. The reaction rates are also specified.\\

Syntax

```
REACTION [BEFORE.<X>AFTER.<X>] [Rate parameters]
```

See Sections 3.15 “Generic Ion Transport Model” and 3.16 “Generic Ion Reaction Model”for full description.

Parameter	Type	Default	Units
AFTER.N	Integer	0	
AFTER.P	Integer	0	
AFTER.SP1	Integer	0	
AFTER.SP2	Integer	0	
AFTER.SP3	Integer	0	
BEFORE.N	Integer	0	
BEFORE.P	Integer	0	
BEFORE.SP1	Integer	0	
BEFORE.SP2	Integer	0	
BEFORE.SP3	Integer	0	
EQUIRATE	Real	0.0	$cm^{-3}s^{-1}$
FORWARD.EA	Real	0.0	eV
FORWARD.RATE	Real	0.0	See description
REVERSE.EA	Real	0.0	eV
REVERSE.RATE	Real	0.0	See description

Description

You specify reactants and products with the parameters BEFORE.<X> and AFTER.<X> described in first description table below. You specify reaction rates with the parameters described in the second table below.

AFTER.N	Number of electrons produced by reaction.
AFTER.P	Number of holes produced by reaction.
AFTER.SP1	Number of SPECIES1 ions produced by reaction.
AFTER.SP2	Number of SPECIES2 ions produced by reaction.
AFTER.SP3	Number of SPECIES3 ions produced by reaction.

BEFORE.N	Number of electrons consumed by reaction.
BEFORE.P	Number of holes consumed by reaction.
BEFORE.SP1	Number of SPECIES1 ions consumed by reaction.
BEFORE.SP2	Number of SPECIES2 ions consumed by reaction.
BEFORE.SP3	Number of SPECIES3 ions consumed by reaction.
EQUIRATE	Fixed reaction rate in $cm^{-3}s^{-1}$.
FORWARD.EA	Activation energy in eV of forward reaction rate.
FORWARD.RATE	Forward reaction rate constant. The forward rate itself can depend on any product of concentrations, and the rate constant should give overall rate dimensions of $cm^{-3}s^{-1}$.
REVERSE.EA	Activation energy in eV of reverse reaction rate.
REVERSE.RATE	Reverse reaction rate constant. The reverse rate itself can depend on any product of concentrations, and the rate constant should give overall rate dimensions of $cm^{-3}s^{-1}$.

Examples

The set of reactions

$$2H \Leftrightarrow H_2$$

$$H + e^- \Leftrightarrow H^-$$

$$H^- + h^+ \Leftrightarrow H$$

can be represented by the following set of **REACTION** statements

```
REACTION BEFORE.SP1=2 AFTER.SP2=1
REACTION BEFORE.SP1=1 BEFORE.N=1 AFTER.SP3=1
REACTION BEFORE.SP3=1 BEFORE.P=1 AFTER.SP1=1
```

where SPECIES1 is H, SPECIES2 is H_2, and SPECIES3 is H^-.

22.52 REGION

REGION specifies the location of materials in a previously defined mesh. Every triangle must be defined as a material.

Syntax

```
REGION NUMBER=<n> <material> [<position>]
```

Parameter	Type	Default	Units
ABS	Logical	False	
A.MAX	Real		degrees
A.MIN	Real		degress
A-SILICO	Logical	False	
ACCEPTORS	Real	0.0	cm^{-3}
ALGAAS	Logical	False	
ALINAS	Logical	False	
ASUB	Real	1.0	Å
BOTTOM	Logical	False	
CALC.STRAIN	Logical	False	
COMPX.BOTTOM	Real	0.0	
COMPY.BOTTOM	Real	0.0	
COMPX.TOP	Real	0.0	
COMPY.TOP	Real	0.0	
CONDUCTOR	Logical	False	
CONVERT	Logical	False	
DEC.C1	Real	0.0	eV
DEC.C2	Real	0.0	eV
DEC.C3	Real	0.0	eV
DEC.D2	Real	0.0	eV
DEC.D4	Real	0.0	eV
DEC.ISO	Real	0.0	eV
DEV.HH	Real	0.0	eV
DEV.LH	Real	0.0	eV
DEV.SO	Real	0.0	eV

Parameter	Type	Default	Units
DEV.ISO	Real	0.0	eV
DEVICE	Character		
DIAMOND	Logical	False	
DONORS	Real	0.0	cm^{-3}
ETA.NEGF	Real	0.0066	eV
EQUIL.NEGF	Logical	False	
FIXED.FERMI	Logical	False	
FN.SIS	Logical	False	
GAAS	Logical	False	
GAASP	Logical	False	
GERMANIU	Logical	False	
GRAD.12	Real		μm
GRAD.23	Real		mm
GRAD.34	Real		μm
GRAD.41	Real		μm
GRADED	Logical	False	
HELMHOLTZ	Logical	True	
HGCDTE	Logical	False	
IGNORE	Logical	False	
INGAAS	Logical	False	
INAS	Logical	False	
INASP	Logical	False	
INGAP	Logical	False	
INHERIT	Logical	False	
INP	Logical	False	
INSULATO	Logical	False	
IX.LOW	Integer	left of structure	
IX.HIGH	Integer	right of structure	
IX.MAX	Integer	right of structure	

Parameter	Type	Default	Units
IX.MIN	Integer	left of structure	
IY.LOW	Integer	top of structure	
IY.HIGH	Integer	bottom of structure	
IY.MAX	Integer	top of structure	
IY.MIN	Integer	bottom of structure	
IZ.HIGH	Real		
IZ.LOW	Real		
IZ.MAX	Integer		
IZ.MIN	Integer		
LED	Logical	False	
MATERIAL	Character		
MODIFY	Logical	False	
NAME	Character		
NA.BOTTOM	Real	0.0	cm^{-3}
NA.TOP	Real	0.0	cm^{-3}
ND.BOTTOM	Real	0.0	cm^{-3}
ND.TOP	Real	0.0	cm^{-3}
NITRIDE	Logical	False	
NUMBER	Integer		
NX	Real	2	
NY	Real	2	
OXIDE	Logical	False	
OXYNITRI	Logical	False	
P1.X	Real		μm
P1.Y	Real		μm
P2.X	Real		μm
P2.Y	Real		μm
P3.X	Real		μm
P3.Y	Real		μm

Parameter	Type	Default	Units
P4.X	Real		μm
P4.Y	Real		μm
PIEZOELECTRIC	Logical	False	
PIEZO.SCALE	Real	1.0	
POLAR.SCALE	Real	1.0	
POLARIZATION	Logical	False	
POLYSILICON	Logical	False	
PSP.SCALE	Real	1.0	
QT.EXTENT	Real	0.0	microns
QTREGION	Integer	0	
QT.XDIR	Logical	False	
QT.YDIR	Logical	False	
QWELL	Logical	False	
QWNUM	Integer	-1	
R.MAX	Real		μm
R.MIN	Real		μm
S.OXIDE	Logical	False	
SAPPHIRE	Logical	False	
SCHRO	Logical	True	
SEMICOND	Logical	False	
SELF	Logical	False	
SI3N4	Logical	False	
SIC	Logical	False	
SIGE	Logical	False	
SILICON	Logical	False	
SIO2	Logical	False	
SL.GEOM	Character	1DY	
SL.NUMBER	Integer		
SLATT	Logical	False	

Parameter	Type	Default	Units
SLATT.NUMBER	Integer		
STAY	Logical	False	
STRAIN	Real	0.0	
STR.BOT	Real	0.0	
STR.TOP	Real	0.0	
SUBSTRATE	Logical	False	
SX	Real	0.0	
SY	Real	0.0	
THICKNESS	Real	0.0	μm
TOP	Logical	False	
TRAPPY	Logical	False	
USER.GROUP	Character		
USER.MATERIAL	Character		
WELL.CNBS	Integer	1	
WELL.DEBUG	Logical	False	
WELL.FIELD	Logical	True	
WELL.GAIN	Real	1.0	
WELL.NX	Integer	10	
WELL.NY	Integer	10	
WELL.NZ	Integer	10	
WELL.OVERLAP	Real		
WELL.VNBS	Integer	1	
X.COMP	Real	0.0	
X.MAX	Real	right of structure	μm
X.MIN	Real	left of structure	μm
X.MOLE	Real	0.0	
Y.MOLE	Real	0.0	
Y.COMP	Real	0.0	
Y.MIN	Real	top of structure	μm

Parameter	Type	Default	Units
Y.MAX	Real	bottom of structure	μm
Z.MAX	Real		
Z.MIN	Real		
ZEROBQP	Logical	False	
ZNSE	Logical	False	
ZNTE	Logical	False	

Description

n	Specifies a region number from 1 to 200.
material	This is one or more of the material names described below.
position	This is one or more of the position parameters described below.
IGNORE	Specifies that the specified region is to be igorned in the equation assembly.
SL.GEOM	Specifies the superlattice geometry as either 1DY (default) or 1DX.
SL.NUMBER	Assigns the index for a regions membership in a indexed superlattice.
SLATT	Enables the superlattice model for a region.
SLATT.NUMBER	This is an alias for SL.NUMBER.

Region Parameters

ABS	Specifies that quantum confinement is included in the calculation of absorption or imaginary index of refraction. To use this model, you must also specify QWELL on the REGION statement. See Section 11.8.3 "Quantum Well Absorption".
ACCEPTORS	Specifies a uniform density of ionized acceptors in the region.
ASUB	Specifies the substrate lattice constant for strain calculations as described in Section 3.6.10 "Epitaxial Strain Tensor Calculation in Wurtzite".
CALC.STRAIN	Specifies that the strain in the region is calculated from the lattice mismatch with adjacent regions.
COMPX.BOTTOM COMPY.BOTTOM COMPX.TOP COMPY.TOP	Specify the composition fractions at the top and bottom of a region when linearly graded. Linear grading is specified by the GRADED parameter. See Section 2.6.4 "Automatic Meshing (Auto-meshing) Using The Command Language".
CONDUCTOR	Specifies that the region is to be simulated as a conductor. This means that the conduction equation is solved for the region.
CONVERT	This is an alias for MODIFY.

DEC.C1	This is a conduction band shift for 1st pair of electron valleys.
DEC.C2	This is a conduction band shift for 2nd pair of electron valleys.
DEC.C3	This is a conduction band shift for 3rd pair of electron valleys.
DEC.D2	This is a conduction band shift for Δ2 electron valleys if DEC.C1 is not set.
DEC.D4	This is a conduction band shift for Δ4 electronvalleys if DEC.C2 is not set.
DEC.ISO	This is a conduction band shift for isotropic electron band, when NUM.DIRECT=1.
DEV.HH	This is a valence band shift for heavy holes.
DEV.LH	This is a valence band shift for light holes.
DEC.ISO	This is a valence band shift for isotropic hole band, when NUM.BAND=1.
DEV.SO	This is a valence band shift for split-off holes.
DEVICE	Specifies which device the **REGION** statement along with the MODIFY parameter should apply to in MixedMode simulations.
DONORS	Specifies a uniform density of ionized donors in the region.
ETA.NEGF	An imaginary optical potential, added to Hamiltonian of quasi-equilibrium regions in planar NEGF model.
EQUIL.NEGF	Specified that the region will be trated as quasi-equilibrium in planar NEGF model.
GRAD.<n>	Specifies the compositional gradings for heterojunctions along each side of the region rectangle or quadrilateral. The value of the GRAD parameters specifies the distance at which the composition fraction reduces to zero. A value of 0.0 specifies that the heterojunction is abrupt. The value of <n> can be the numbers: 12, 23, 34, and 41. The meanings of these numbers is down below. • 12 = top surface • 23= right hand side • 34= bottom surface • 41 = left hand side These correspond to the point indices around the rectangular region working clockwise from top left.
GRADED	Specifies that the region has linear compositional or doping variation or both. See also **COMPX.BOTTOM**, **COMPY.BOTTOM**, **COMPX.TOP**, **COMPY.TOP**, **ND.BOTTOM**, **ND.TOP**, **NA.BOTTOM**, and **NA.TOP**.
HELMHOLTZ	Allows (default) or prevents a solution of vector Helmholtz equation in the region.

INHERIT	Specifies that if the region is added to an existing structure file, it will inherit the noded value (such as doping) from any overwritten regions.
LED	Specifies that the region is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 12 "LED: Light Emitting Diode Simulator".
MATERIAL	Specifies the material used for the region. Valid material names are listed in Table B-1. All materials are divided into three classes: semiconductors, insulators and conductors.

Note: You can specify the following logical parameters to indicate the region material instead of assigning the MATERIAL parameter: **SILICON**, **GAAS**, **POLYSILI**, **GERMANIU**, **SIC**, **SEMICOND**, **SIGE**, **ALGAAS**, **A-SILICO**, **DIAMOND**, **HGCDTE**, **INAS**, **INGAAS**, **INP**, **S.OXIDE**, **ZNSE**, **ZNTE**, **ALINAS**, **GAASP**, **INGAP**, **INASP**, **OXIDE**, **SIO2**, **NITRIDE**, **SI3N4**, **INSULATO**, **SAPPHIRE**, and **OXYNITRI**.

MODIFY	This is used to modify the characteristics of regions imported into the simulator. See Section 2.6.5 "Modifying Imported Regions". The alias for this parameter is **CONVERT**.
NAME	Specifies the name of the region. The name can be used in the **MODELS**, **MATERIAL**, and **IMPACT** statements to provide regionally dependent models. This name is just a label. It doesn't imply any material parameter settings.
ND.BOTTOM **ND.TOP** **NA.BOTTOM** **NA.TOP**	Specify the doping concentrations at the top and bottom of a region when linearly graded. Linear grading is specified by the GRADED parameter. See Section 2.6.4 "Automatic Meshing (Auto-meshing) Using The Command Language".
NUMBER	Assigns a region number. Multiple **REGION** lines with the same number can be used to define region shapes made from several rectangles.
PIEZO.SCALE	This is an alias for **POLAR.SCALE**.
PIEZOELECTRIC	This is an alias for **POLARIZATION**.
POLARIZATION	Enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. The alias for this parameter is **PIEZOELECTRIC**. See Section 3.6.11 "Polarization in Wurtzite Materials".
POLAR.SCALE	Specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the **REGION** statement. The alias for this parameter is PIEZO.SCALE. See Section 3.6.11 "Polarization in Wurtzite Materials".
PSP.SCALE	Specifies a constant scale factor similar to POLAR.SCALE but applied only to spontaneous polarization.
QT.EXTENT	When QTREGION flag is enabled, this restricts a QTREGION to within QTEXTENT microns of a type heterojunction.

QTREGION	This designates the region as a quantum barrier for the quantum barrier tunnelling models SIS.EL and SIS.HO. The index of the QTREGION is also set with this parameter. For further information, see Section 6.2.4 “Non-local Quantum Barrier Tunneling Model”.
QT.XDIR	Forces quantum tunneling to be in x-direction through QTREGION
QT.YDIR	Forces quantum tunneling to be in y-driection through QTREGION
QWELL	Specifies that the region is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.
QWNUM	Optionally sets the number of the quantum well region.
SCHRO	Allows (default) or prevents a solution of Schrodinger in the region. Switch it off, if the region is to be treated classically.
STRAIN	Specifies the strain in the region. (Negative strain is compressive).
STR.BOT **STR.TOP**	Specify the strain at the bottom and top of the region for calculations of the k.p model. The strain values between the top and bottom of the region are linearly interpolated.
SUBSTRATE	This is a logical parameter indicating that the specified region should be considered as the the substrate for strain calculations as described in Section 3.6.10 “Epitaxial Strain Tensor Calculation in Wurtzite”.
TRAPPY	Used only with the BESONOS model. This indicates that a Silicon-Nitride region will be able to trap electrons and holes.
USER.GROUP	Specifies the material group for the user-defined material. USER.GROUP can be either SEMICONDUCTOR, INSULATOR, or CONDUCTOR.
USER.MATERIAL	Specifies a user-defined material name. The specified material name can be any name except that of a default material, such as Silicon. You should define each material with an accompanying definition in the **MATERIAL** statement. You can define up to 50 user-defined materials.
WELL.CNBS **WELL.VNBS**	Specify the number of bound states retained for calculation of radiative recombination or gain if the region is treated as a quantum well as specified by the QWELL parameter.
WELL.FIELD	Specifies that the calculations of bound state energies should include the effects of the local field.
WELL.GAIN	Specifies a constant scale factor multiplied by the calculated gain to give the net gain used for certain optoelectronic calculations.
WELL.NX **WELL.NY** **WELL.NZ**	Specifies the number of grid points in x, y, and z directions for Schrodinger equation in a quantum well region.
WELL.OVERLAP	Specifies the wavefunction overlap integral. If WELL.OVERLAP is not specified, then the overlap integral is calculated from the wavefunctions.

Note: The highest region number takes precedence if **REGION** definitions overlap.

ZEROBQP	Specifies that in the given region the Bohm quantum potential is not solved and is assigned to zero.

Position Parameters

You can use grid indices to define a region only when the mesh is rectangular. To define a region with a rectangular mesh, use the **X.MESH** and **Y.MESH** statements to specify grid indices.

You can also use the IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters to specify x and y mesh line number values.

Note: To add regions to irregular meshes, such as those from Athena, specify the boundaries with the X.MAX, X.MIN, Y.MAX, and Y.MIN parameters.

A.MAX	Specifies the maximum angle of a 3D cylindrical region.
A.MIN	Specifies the minimum angle of a 3D cylindrical region.
BOTTOM	Specifies that the region is to be added at the bottom (starting at the maximum previously specified Y coordinate and extending in the positive Y direction) of the structure.
FN.SIS	Applies to the Fowler-Nordheim model for an interface to the insulator region having this attribute. Any Fowler-Nordheim current being evaluated is not assigned to a contact. Instead, it is assumed to pass from one semiconductor region to another through the tagged region.
IX.HIGH	Specifies the maximum x value of the grid index. The alias for this parameter is **IX.MAX**.
IX.LOW	Specifies the minimum x value of the grid index. The alias for this parameter is **IX.MIN**.
IY.HIGH	Specifies the maximum y value of the grid index. The alias for this parameter is IY.MAX.
IY.LOW	Specifies the minimum y value of the grid index. The alias for this parameter is **IY.MIN**.
IZ.HIGH	Specifies the maximum z value of the grid index. The alias for this parameter is **IZ.MAX**.
IZ.LOW	Specifies the minimum z value of the grid index. The alias for this parameter is **IZ.MIN**.
IX.MAX **IX.MIN** **IY.MAX** **IY.MIN** **IZ.MAX** **IZ.MIN**	These are aliases for **IX.HIGH**, **IX.LOW**, **IY.HIGH**, **IY.LOW**, **IZ.HIGH**, and **IZ.LOW**.

NX	Specifies the number of uniformly spaced mesh lines to be added to resolve the region in the X direction.
NY	Specifies the number of uniformly spaced mesh lines to be added to resolve the region in the Y direction.
STAY	This is used in automeshing to create material variations in the X direction. See Section 2.6.4 "Automatic Meshing (Auto-meshing) Using The Command Language".
SX	Specifies the spacing between mesh lines to be applied at the edges of the region in the X direction.
SY	Specifies the spacing between mesh lines to be applied at the edges of the region in the Y direction.
P1.X	X coordinate of 1st corner of **REGION**.
P1.Y	Y coordinate of 1st corner of **REGION**.
P2.X	X coordinate of 2nd corner of **REGION**.
P2.Y	Y coordinate of 2nd corner of **REGION**.
P3.X	X coordinate of 3rd corner of **REGION**.
P3.Y	Y coordinate of 3rd corner of **REGION**.
P4.X	X coordinate of the last corner of **REGION**.
P4.Y	Y coordinate of the last corner of **REGION**.
THICKNESS	Specifies the thickness of the region in the Y direction. This parameter must accompany the specification of the TOP or BOTTOM parameters.
TOP	Specifies that the region is to be added at the top (starting at the minimum previously specified Y coordinate and extending in the negative Y direction) of the structure.
R.MAX	Specifies the maximum radius of a 3D cylindrical region.
R.MIN	Specifies the minimum radius of a 3D cylindrical region.
X.MAX	Specifies the maximum x-boundary.
X.MIN	Specifies the minimum x-boundary.
X.COMP, **X.MOLE**	This is the composition fraction (X) for a region with a composition dependent cations (e.g., AlGaAs).
Y.COMP, **Y.MOLE**	This is the composition fraction (Y) for a region with a composition dependent anions(e.g., InGaAsP).
Y.MAX	Specifies the maximum y-boundary.

Y.MIN	Specifies the minimum x-boundary.
Z.MIN	Specifies the minimum z-boundary.
Z.MAX	Specifies the maximum z-boundary.

Grid Indices Example

Define a silicon region extending from nodes 1 to 25 in the X direction and from nodes 1 to 20 in the Y direction.

```
REGION NUM=1 IX.LO=1 IX.HI=25 IY.LO= 1 IY.HI=20 MATERIAL=SILICON
```

Non-Rectangular Region Example

Define a region that's composed of two separate rectangular areas. Note that REGION statements are cumulative.

```
REGION NUM=1 IX.LO=4 IX.HI=5 IY.LO=1 IY.HI=20 MATERIAL=OXIDE
REGION NUM=1 IX.LO=1 IX.HI=30 IY.LO=1 IY.HI=37 MATERIAL=OXIDE
```

Typical MOS Example

Define regions for a typical MOS structure.

```
REGION NUM=1 Y.MAX=0 MATERIAL=OXIDE
REGION NUM=2 Y.MIN=0 MATERIAL=SILICON
```

3D Region Definition Example

Define a cube of oxide within a region silicon in 3D.

```
REGION NUM=1 MATERIAL=SILICON
REGION NUM=2 Y.MAX=0.5 X.MIN=0.5 \
X.MAX=1.0 Z.MIN=0.5 Z.MAX=1.0 MATERIAL = OXIDE
```

Graded Heterojunction Definition Example

Define a graded heterojunction of AlGaAs/GaAs.

```
REGION NUM=1 MATERIAL=GaAs Y.MIN=1
REGION NUM=2 MATERIAL=AlGaAs Y.MAX=0.9 X.COMP=0.2 GRAD.34=0.1
```

In this case, the area between y=0.9 and 1.0 is graded in composition from 0.2 to 0.0. The Y.MAX parameter refers to the bottom of the uniform composition region. The actual bottom of the AlGaAs region is Y.MAX+GRAD.34.

Graded x.comp and y.comp Material Definition Example

Define a graded x.comp and y.comp region in Atlas using the compx.top, compx.bottom, compy.top, and the compy.bottom keywords on the **REGION** statement. The following example shows the use of compx.top, compx.bottom functions. Similar use, however, can be made of the compy.top and compy.bottom keywords.

You can use the following code to create a graded x.comp AlGaAs region where the x.comp falls off linearly in value from 0.4 to 0.2 from the top of the region to the bottom of the region.

```
region num=1 material=algaas compx.top=0.4 compx.bottom=0.1
```

22.53 REGRID

REGRID allows you to refine a crude mesh. A triangle is refined when the chosen variable changes by more than one specified criteria.

Syntax

```
REGRID RATIO=<n> <var> [<lp>] [<cp>] [<io>]
```

Parameter	Type	Default	Units
ABSOLUTE	Logical	False	
ASCII	Logical	False	
CHANGE	Logical	True	
COS.ANGLE	Real	2.0	
DOPFILE	Character		
DOPING	Logical		cm^{-3}
E.TEMP	Logical	False	
EL.FIELD	Logical		V/cm
ELECTRON	Logical		cm^{-3}
H.TEMP	Logical	False	
HOLE	Logical		cm^{-3}
IGNORE	Integer		
IN.GREEN	Character		
LOCALDOP	Logical	False	
LOGARITHM	Logical	False	
MAX.LEVEL	Integer	1+ maximum level of grid	
MIN.CARR	Logical		cm^{-3}
NET.CARR	Logical		cm^{-3}
NET.CHRG	Logical		cm^{-3}
PISCES.OUT	Logical	False	
OUT.GREEN	Character	value of OUTFILE + tt	
OUTFILE	Character		
POTENTIAL	Logical		V
QFN	Logical		V

Parameter	Type	Default	Units
QFP	Logical		V
REGION	Integer		All
SMOOTH.KEY	Integer		
STEP	Real		
X.MAX	Real	Right	μm
X.MIN	Real	Left	μm
Y.MAX	Real	Bottom	μm
Y.MIN	Real	Top	μm

Description

RATIO STEP	Specifies the maximum allowed variance across one element.
var	This is one of the variable parameters described below. The selected parameter is used as the basis for regridding.
lp	This is one of more of the location parameters described below. These parameters are used to select the areas which are to be refined.
cp	This is one or more of the control parameters. These parameters are used to control the plotted output.
io	This is one or more of the File I/O parameters.

Variable Parameters

DOPING	Selects net doping.
E.TEMP	Select electron temperature.
EL.FIELD	Selects electric field.
ELECTRON	Selects electron concentration.
H.TEMP	Selects hole temperature.
HOLE	Selects hole concentration.
MIN.CARR	Selects minority carrier concentration.
NET.CARR	Selects net carrier concentration.
NET.CHRG	Selects net charge.

POTENTIAL	Selects mid-gap potential.
QFN	Selects the electron quasi-Fermi level.
QFP	Selects the hole quasi-Fermi level.

Location Parameters

If no location parameters are specified, refinement will include:

- All regions for potential and electric field regrids.
- All semiconductor regions for regrids which depend on the other variables.

IGNORE	Specifies regions that are not to be refined.
REGION	Specifies regions which are refined according to the specified criterion. Other regions may be refined to maintain well-shaped triangles.
X.MAX	Uses device coordinates to specify the maximum x-value for refinement.
X.MIN	Uses device coordinates to specify the minimum x-value for refinement.
Y.MAX	Uses device coordinates to specify the maximum y-value for refinement.
Y.MIN	Uses device coordinates to specify the minimum y-value for refinement.

Control Parameters

ABSOLUTE	Specifies that the absolute value of the variable be used.
CHANGE	Determines whether to use the magnitude or the difference of a triangle variable as the refinement criterion. This parameter defaults to "difference".
COS.ANGLE	Limits the creation of obtuse angles in the mesh by specifying obtuse criterion. If this parameter is used, nodes are added to the mesh so that the number of obtuse triangles is reduced.

Note: Be careful when using the COS.ANGLE parameter. Recommended values are from 0.8 to 0.95. Smaller values may dramatically increase the number of nodes.

LOCALDOP	Specifies that if minority carrier concentration exceeds local doping, the grid will be refined. This parameter is used in conjunction with minority carrier regrids.
LOGARITHM	Specifies a logarithmic refinement scale. Since many of the quantities may become negative, numerical problems are avoided by using $log(x)=sign(x)\cdot log10(1+\|x\|)$. If you wish to obtain the true logarithm of a quantity, the ABSOLUTE parameter must be specified before the LOGARITHM parameter is specified. The absolute value of a quantity is computed first, thereby eliminating negative arguments.
MAX.LEVEL	Specifies the maximum level of any triangle relative to the original mesh. This parameter defaults to one more than the maximum level of the grid, but can be set to a smaller value to limit refinement. Values less than or equal to zero are interpreted relative to the current maximum grid level.
SMOOTH.KEY	Specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows: 1. Triangle smoothing: All region boundaries remain fixed. 2. Triangle smoothing: Only material boundaries are maintained. 3. Node averaging. 4. Improved triangle smoothing method: This method uses diagonal flipping to reduce the number of obtuse triangles. 5. Aligns triangles with electric field gradient. Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSUPREM4. Option 4 is similar to option 1, but option 4 usually creates less obtuse triangles.

File I/O Parameters

ASCII	Specifies that mesh files and triangle trees will be written in an ASCII rather than a binary format. This parameter has no effect on the device doping file (see the **DOPFILE** parameter).
DOPFILE	Specifies the name of a file, which contains device doping information. This file is created on the **DOPING** statement. Specifying DOPFILE avoids linear interpolation of doping values at newly created grid points by using the initial doping specification to apply doping to the new grid points.
IN.GREEN	Specifies a triangle tree for the mesh which will be used in this regrid. If this parameter is not specified, the program will look for a file with the same name as the current mesh plus, *tt*, at the end. If no such file exists, the program will not use a triangle tree for the previous mesh.
MASTER.OUT	Saves mesh and doping information in a standard structure file format.

OUTFILE	Specifies the name of a standard structure output file where mesh information will be stored. This parameter must be specified if the mesh is to be used for subsequent runs.
OUT.GREEN	Specifies the name of the file that holds the history of the triangle tree. This history is used in further regrid steps.
PISCES.OUT	Saves mesh and doping information in a binary PISCES-II format. Files in this format cannot be displayed in TonyPlot.

Doping Regrid Examples

Starting with an initial grid, we refine twice so that all triangles with large doping steps are refined.

```
REGRID LOG DOPING RATIO=6 OUTF=grid1 DOPF=dopxx SMOOTH=4
REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx SMOOTH=4
```

A similar effect could be obtained with just one regrid statement.

```
REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx MAX.LEVEL=2
```

In both cases, two levels of refinement are performed. The first example is recommended because new doping information is introduced at each level of refinement. This produces better refinement criterion and fewer triangles.

Potential Regrid Example

Next, an initial solution is produced and triangles which exhibit large potential steps are refined.

```
SOLVE INIT
REGRID POTENTIAL RATIO=0.2 OUTF=grid3.str SMOOTH=4
```

Re-initializing after Regrid example

Often you need to re-solve the same bias point after a **REGRID** using the following style of syntax.

```
SOLVE VDRAIN=3.0
REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str
SOLVE PREV
```

Occasionally, you need to quit and restart Atlas with the new mesh. To do this, type syntax such as:

```
SOLVE VDRAIN=3.0
REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str
go atlas
MESH INF=mygrid.str
```

After this **MESH** statement all models, material parameters and numerical methods have to be re-specified before any **SOLVE** statement.

22.54 SAVE

SAVE saves all node point information into an output file.

Note: In all cases the region boundaries, electrodes, mesh, and doping are saved. If a **SOLVE** statement has preceded the **SAVE** statement all electrical data from the last solution is stored.

Syntax

```
SAVE OUTFILE=<filename> [MASTER]
```

Parameter	Type	Default	Units
1.FRAC.YIELD	Character		
1.INT.YIELD	Character		
1.INT.FRAC.YIELD	Character		
1.YIELD	Character		
ABSORB	Logical	False	
AMBIEN.IMAG	Real	0.0	
AMBIEN.REAL	Real	1.0	
ANGLE.OUTPUT	Real	30.0	degrees
ANGPOWER	Character		
COMPARE	Logical	False	
COUPLING3D	Logical	False	
CUTPLANE	Logical	False	
D.ORIENT	Real	0	
DIPOLE	Logical	False	
DDMS.LOG	Logical	False	
DOS.MAXN	Real	100	
EDGE_ONLY	Logical	False	
EDGE_REFLECT	Logical	False	
EMAX	Real		eV
EMIN	Real		eV
EMIT.BOT	Logical	True	
EMIT.TOP	Logical	True	
F.RHO.DIPOLE	Character		

Parameter	Type	Default	Units
FAST	Logical	False	
FRAC.YIELD	Character		
HORIZONTAL	Real	2/3	
I.DXMIN	Real	1.00E-005	
I.MAXDOUBLE	Real	10	
I.MINI	Real	0	
I.RELERR	Real	1.00E-005	
INCOHERENT	Logical	False	
INT.FRAC.YIELD	Character		
INT.PARA.INTEGRAL	Character		
INT.PERP.INTEGRAL	Character		
INT.QEFF	Real	1	
INT.YIELD	Character		
INCOHERENT	Logical	False	
INCOH.TOP	Logical	False	
INTERFERE	Logical	False	
INT.OPDOS	Character		
INT.YIELD	Character		
K.SURFACE	Real	0	
K.SUBSTRATE	Real	0	
KP.OUTFILE	Character		
L.WAVE	Real	0.8	μm
LMAX	Real		μm
LMIN	Real		μm
MASTER	Character	True	
MAT.SURFACE	Character		
MAT.SUBSTRATE	Character		
MERGE	Logical	False	
MIR.BOTTOM	Logical	False	

Parameter	Type	Default	Units
MIN.POWER	Real	1×10^{-4}	
MIR.TOP	Logical	False	
MQWELLS	Character		
MUMAZIMUTH	Real	1	
N.SURFACE	Real	1	
N.SUBSTRATE	Real	1	
NEAR.FIELD	Character		
NEGF.LOG	Logical	False	
NORMALIZE.ANGULAR	Logical	False	
NORMALIZE.PL	Logical	False	
NORMALIZE.SPEC	Logical	False	
NSAMP	Integer	100	
NUMRAYS	Integer	180	
OPDOS	Character		
OPSM.OP1	Integer	1	
OPSM.OP2	Integer	0	
OPSM.POW1	Logical	False	
OPSM.RESET	Logical	False	
OUT.CENTERWISE	Logical	False	
OUT.FILE	Character		
OUTFILE	Character		
OUT.SPEC	Character		
PARA.INTEGRAL	Character		
PERP.INTEGRAL	Character		
PATTERNS	Character		
PISCES	Logical	False	
POLAR	Real	0.0	
PRINT	Logical	False	
PRINT.EVERY	Logical	False	

Parameter	Type	Default	Units
RAYPLOT	Character		
READ.1.FRAC.YIELD	Character		
READ.1.YIELD	Character		
READ.OPDOS	Character		
REAL.RI	Logical	False	
REFLECTS	Integer	0	
SIDE	Logical	False	
SMOOTHNESS	Real	8.0	
SOURCE.TERM	Logical	False	
SOURCE_TERM	Logical	False	
SPEC.INTERSUB	Character		
SPECT.ANGLE	Character		
SPECTRUM	Character		
STRUCTURE	Character	False	
SUM.COLOR	Character		
SUM.COLOR.OUT	Character		
SURFACE_ONLY	Logical	False	
T.INITSTEP	Real	0.01	um
T.MINSTEP	Real	0.01	um
T.RELERR	Real	0.02	
T.RHO.DIPOLE	Character		
TEMPER	Real	300.0	
TMM.GF	Logical	False	
TR.MATRIX	Logical	False	
TRAP.FILE	Character		
UNI.POW	Real	0	W
UNITSOURCE	Logical	False	
USER.SPECT	Character		
X	Real	0.0	μm

Parameter	Type	Default	Units
X.CUTPLANE	Real	0.0	μm
XMAX	Real	0.0	μm
XMIN	Real	0.0	μm
XNUM	Integer	0	
Y	Real	0.0	μm
Y.CUTPLANE	Real	0.0	μm
YIELD	Character		
YMAX	Real	0.0	μm
Y.MAX	Real	0.0	μm
YMIN	Real	0.0	μm
Y.MIN	Real	0.0	μm
YNUM	Integer	0	
Z	Real	0.0	μm
Z.CUTPLANE	Real		μm
ZMAX	Real	0.0	μm
Z.MAX	Real	0.0	μm
ZMIN	Real	0.0	μm
Z.MIN	Real	0.0	μm

Description

ABSORB	Takes absorption into account in reverse ray-tracing (see Section 12.6 "Reverse Ray-Tracing"). The absorption is assumed to be constant specified for each material by the imaginary part of the refractive index.
AMBIEN.IMAG	Specifies the imaginary index of refraction for the ambient outside the device domain for LED reverse ray tracing.
AMBIEN.REAL	Specifies the real index of refraction for the ambient outside the device domain for LED reverse ray tracing.
ANGLE.OUTPUT	Specifies the angular interval in degrees where spectrum versus angle are written to the file specified by the SPECT.ANGLE parameter.
ANGPOWER	Enables the reverse ray-tracing algorithm (see Section 12.6 "Reverse Ray-Tracing") for analysis of output coupling of light from the structure of a Light Emitting Diode. ANGPOWER specifies the name of the output file for the angular power density vs. output angle dependence.

COUPLING3D	Calculates an output coupling coefficient, assuming a cylindrically symmetric light output with the light source located on the axis of symmetry (see Section 12.6 “Reverse Ray-Tracing”). Takes into account the 3D nature of the problem with output coupling calculation.
CUTPLANE	Specifies that a 2D cutplane will be saved from a 3D structure to the file specified by the OUTFILE parameter. The direction of the cutplane depends on the value specified on either the X.CUTPLANE, Y.CUTPLANE or Z.CUTPLANE parameter. If you specify the X.CUTPLANE parameter, the output cutplane will be perpendicular to the X axis. If you specify the Y.CUTPLANE parameter, the output cutplane will be perpendicular to the Y axis. If you specify the Z.CUTPLANE parameter, the output cutplane will be perpendicular to the Z axis.
DIPOLE	Specifies a particular angular distribution of the internal radiating field that corresponds to a preferred in-plane orientation of dipoles often relevant to OLED devices (see Section 12.6 “Reverse Ray-Tracing”).
DDMS.LOG	Specifies that in case of DDMS model (see Section 14.9 “Drift-Diffusion Mode-Space Method (DD_MS)”), an additional log file with subband-resolved quantities will be stored along with structure file.
EDGE_ONLY	Specifies that for LED reverse ray tracing only the rays exiting from the edges are considered. Rays exiting at the top and bottom surfaces are not shown.
EDGE_REFLECT	Specifies the rays in reverse ray tracing reaching the edges of the device are totally reflected.
EMAX and **EMIN**	Specify the energy range for saving a spectrum file.
EMIT.BOT	Same functionality as **EMIT.TOP** but applied to the emission from the bottom-most surface.
EMIT.TOP	When True, Atlas includes the emission from top surface in the output power spectral density written to the file specified by **SPECT.ANGLE**.
FAST	Specifies that a fast algorithm should be used for LED reverse ray tracing. In this case, the ray tracing is performed from region boundary to region boundary rather than the default element by element trace.
HORIZONTAL	Specifies the proportion of horizontally oriented dipoles for the dipole emission model.
INCOHERENT	Specifies that an incoherent material in LED/OLED structure is used for the purpose of combined transfer matrix method/reverse ray tracing (TTM+RRT) optical analysis.
INCOH.TOP	When specified with the SOURCE.TERM parameter, this allows a combination of transfer matrix method in thin coherent layers and ray tracing in thick incoherent layers for LED reverse ray tracing.

INTERFERE	Specifies that rays originating in the same point are fully coherent in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”). In this case, the phase information upon propagation is preserved. Phase change upon reflection is also considered. Thus, interference of rays exiting the device at the same angle are taken into account.
KP.OUTFILE	Root of the filename for output of spectrum, bandstructure, and DOS from the **k.p** calculation
L.WAVE	Specifies the wavelength for reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
LMAX and **LMIN**	Specify the range of wavelength for saving spectrum files.
MASTER	Specifies that the output file will be written in a standard structure format. Files in this format can be plotted in TonyPlot.
MIN.POWER	Specifies the minimum relative power of a ray (see Section 12.6 “Reverse Ray-Tracing”). The ray is not traced after its power falls below MIN.POWER value. This is useful to limit the number of rays traced. The default value is MIN.POWER=1e-4.
MIR.TOP	Specifies that the top surface of the device be treated as an ideal mirror in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
MIR.BOTTOM	Specifies that the bottom surface of the device be treated as an ideal mirror in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
NEAR.FIELD	Specifies the output file name where LED near field distribution is saved. A file for each surface where light exits is created.
NEGF.LOG	Specifies that spectra of transmission, DOS, density and and curent obtained from NEGF model need to be saved.
NORMALIZE.ANGULAR	The parameter normalizes each spectrum stored in the file specified by **SPECT.ANGLE** to its own maximum.
NSAMP	Specifies the number of samples to use for a spectrum plot.
NUMRAYS	Specifies the number of rays starting from the origin in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”). The default is 180. Acceptable range is 36-3600.
OUT.FILE	This is a synonym for **OUTFILE**.
OUTFILE	Specifies the name of an output file name. The synonym for this parameter is **OUT.FILE**.
OUT.SPEC	Specifies the filename where the angle integrated spectrum is stored.
PISCES	Specifies that the output file will be written in the original PISCES-II format.

PATTERNS	Specifies a character string representing the root of the file names where near and far field patterns are written for Laser or VCSEL. The near field pattern file is appended with the string `.nfp` and the far field pattern file is appended with the string `.ffp`.
POLAR	Specifies polarization of the emitted photons in degrees while linearly polarized light is assumed (see Section 12.6 "Reverse Ray-Tracing"). Parallel (TM-mode, `POLAR=90.0`) and perpendicular (TE-mode, `POLAR=0.0`) polarizations result in significantly different output coupling values. Use `POLAR=45.0` if there is no preferred direction of polarization of emitted photons (unpolarized light emission).
RAYPLOT	Specifies the name of the output file containing the information on each ray exiting the device in single origin reverse ray-tracing (see Section 12.6 "Reverse Ray-Tracing"). This file is only created when the single origin for all rays is assumed. The information includes ray output angle, relative ray power (TE and TM-polarization and total), and initial internal angle at the origin (only if `INTERFERE` parameter is unspecified). 0° angle corresponds to the rays in the X axis direction. 90° angle corresponds to the rays in the Y axis direction.
REFLECTS	Specifies a number of reflections to be traced for each ray in reverse ray-tracing (see Section 12.6 "Reverse Ray-Tracing"). The default value is `REFLECTS=0`. The maximum allowed value is `REFLECTS=10`. Setting the number of reflections to 3 or 4 is often a good choice.
SIDE	Specifies that the rays reaching the sides of the device are terminated there and do not contribute to the total light output (see Section 12.6 "Reverse Ray-Tracing").
SOURCE.TERM	Indicates that dipole source terms are combined with transfer matrix calculation of light output for LEDs. The alias is **SOURCE_TERM**.
SPEC.INTERSUB	Specifies a file name for saving the spectrum of intersubband transitions.
SPECT.ANGLE	Specifies the output file name where LED spectrum is written as a function of emission angle.
SPECTRUM	Specifies the name of a file for saving spectrum plots.
STRUCTURE	This is a synonym for **SAVE**.
SURFACE_ONLY	Specifies that for LED reverse ray tracing only the rays exiting from the top and bottom surfaces are considered. Rays exiting at the edges are not shown.
TEMPER	This is the temperature (needed for using appropriate refractive indexes of the materials in reverse ray-tracing). The default setting of 300 K will be used if `TEMPER` is unspecified.
TRAP.FILE	Specifies a file name to which all trap densities at the location specified by X,Y, and Z are to be saved.

TR.MATRIX	Specifies that the transfer matrix method should be used to handle thin film LEDs.
X	This is the X coordinate of light origin for a single point reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
X.CUTPLANE	Specifies the X coordinate for a YZ cutplane. The region and nodal data from the nearest X plane to this X coordinate will be saved in the structure file.
XMAX	This is the maximum X coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
XMIN	This is the minimum X coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
XNUM	Specifies the number of points along X axis within a rectuangular area in multiple origin reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
Y	This is the Y coordinate of light origin for a single point reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
Y.CUTPLANE	Specifies the Y coordinate for a XZ cutplane. The region and nodal data from the nearest Y plane to this Y coordinate will be saved in the structure file.
YMAX	This is the maximum Y coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
YMIN	This is the minimum Y coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
YNUM	Specifies the number of points along Y axis within a rectuangular area in multiple origin reverse ray-tracing (see Section 12.6 “Reverse Ray-Tracing”).
Z	Specifies the Z coordinate.
Z.CUTPLANE	Specifies the Z coordinate for a XY cutplane. The region and nodal data from the nearest Z plane to this Z coordinate will be saved in the structure file.

Optical S-Matrix Parameters

1.FRAC.YIELD	The file to save unity fractional yield results.
1.INT.FRAC.YIELD	The file to save integrated unity fractional yield results.
1.INT.YIELD	The file to save integrated unity yield results.

1.YIELD	The file to save unity yield results.
ANGLE.OUTPUT	Output angle for yield and fractional yield calculations.
D.ORIENT	Orientation of the dipoles.
DOS.MAXN	Upper limit of the *D* integral.
F.RHO.DIPOLE	The file that contains a C-Interpreter function "rhodipole".
FRAC.YIELD	The file to save fractional yield results.
INCOHERENT	Indicates to TMM-GF solver to perform incoherent-coherent layer mix computation.
INT.OPDOS	The file to save integrated *D* results.
INT.PARA.INTEGRAL	The file to save the parallel dipole integrand (as a function of parallel wavevector) integrated over dipole position results.
INT.PERP.INTEGRAL	The file to save the perpendicular dipole integrand (as a function of parallel wavevector) integrated over dipole position results.
INT.QEFF	The internal quantum efficiency of the dipoles.
INT.FRAC.YIELD	The file to save integrated fractional yield results.
INT.YIELD	The file to save integrated yield results.
I.DXMIN	The minimum step in D and yield integrals.
I.MAXDOUBLE	The parameter to limit number of points in *D* and yield integrals.
I.MINI	The parameter to control the *D* and yield integrals.
I.RELERR	The parameter to control number of points in *D* and yield integrals.
K.SUBSTRATE	Imaginary part of constant refractive index of the substrate.
K.SURFACE	Imaginary part of constant refractive index of the surface.
L.WAVE	Wavelength if calculating at a single wavlength.
LMAX	Maximum wavelength if calculating at a range of wavelengths.
LMIN	Minimum wavlength if calculating at a range of wavelengths
LOG.COLOR	This gives the name of a file to save all the color results that have been evaluated during an Optical S-Matrix simulation.
MAT.SUBSTRATE	Substrate material (for non-constant substrate refractive index)
MAT.SURFACE	Surface material (for non-constant surface refractive index).
MERGE	If MERGE is True then newly calculated OPDOS, YIELD, and FRAC.YIELD data are integrated with the data already in the named files.

MUMAZIMUTH	specifies the number of azimuthal angles to be considered in a 3D FDTD calculation.
NORMALIZE.PL	Specifies that the PL spectrum is normalized in a wavelength scan.
NORMALIZE.SPEC	Specifies that the resulting spectrum is to be multiplied by the photon energy and scaled to make the spectrum maximum equal to 1.
N.SUBSTRATE	Real part of constant substrate refractive index.
N.SURFACE	Real part of constant surface refractive index.
OPDOS	The file to save the *D* data to.
OPSM.OP1	Ordinal of first calculation to output
OPSM.OP2	The number of calculations to skip between outputs.
OPSM.POW1	Indicates yield and fractional yield calculations should be done at unity power.
OPSM.RESET	Deletes the internal database of previous calculations
OUT.CENTERWISE	Specifies whether the output spectra are saved for each individual emitting center for a multi-source optical simulation.
PARA.INTEGRAL	The file to save the parallel dipole integrand (as a function of parallel wavevector) results.
PERP.INTEGRAL	The file to save the perpendicular dipole integrand (as a function of parallel wavevector) results.
PRINT	Prints the physical 1D structure being simulated.
PRINT.EVERY	Prints the refractive indexes for the photon being simulated.
READ.1.FRAC.YIELD	The file to read previously calculated unity fractional yield results from.
READ.1.YIELD	The file to read previously calculated unity yield results from.
REAL.RI	Forces the refractive index of materials close to the dipole to be real.
READ.OPDOS	The file to read previously calculated *D* results from.
SUM.COLOR	This gives a comma-separated list of files that contain spectral intensity and sums the results to calculate a total color value for an LED. For example, `SAVE SUM.COLOR="file1.dat, file2.dat, file3.dat"`.
SUM.COLOR.OUT	The file to save the SUM.COLOR result to.
T.INITSTEP	Initial wavelength step if calculating at a range of wavelengths.
T.MINSTEP	Minimum wavelength step if calculating at a range of wavelengths.

TMM.GF	Specifies that the TMM Green's function method is to be used.
T.RELERR	Error condition to check if we need to calculate at more wavelengths.
T.RHO.DIPOLE	The file that contains a table of dipole concentration as a function of position.
UNI.POW	Manually specify the output power from a region. UNI.POW=1 would be the same as defining OPSM.POW1. UNI.POW=5e-3 would lead to a uniform power generation of 5 mW.
UNITSOURCE	Adds unit power at the origin defined by X and Y on the same **SAVE** statement for all wavelengths.
USER.SPECT	The file to read user defined dipole emission spectrum from.
X	The X position of dipole.
Y	The Y position of dipole if calculating at a single position.
YIELD	The file to save the yield results to.
YMAX	The maximum dipole position if calculating at several positions.
YMIN	The minimum dipole position if calculating at several positions.
YNUM	The number of dipole positions.

Basic Save Example

```
SOLVE V1=5
SAVE OUTF=data1.str
```

is equivalent to

```
SOLVE V1=5 OUTF=data1.str MASTER
```

Save Example with User-Defined Output

In the second example, the SAVE and **OUTPUT** commands are used to produce two output files for the same bias. The **OUTPUT** statement selects which data will be stored in each file. The first file (data1.str) contains the default contents, total electric field, and components of electron velocity. The second file (data2.str) contains components of hole velocity and band edge potentials. Note that the EX.VELO and EY.VELO parameters are used to prevent electron velocity components from being stored in file data2.

```
OUTPUT E.FIELD EX.VELO EY.VELO
SAVE OUTF=data1.str
OUTPUT HX.VELO HY.VELO CON.BAND VAL.BAND ^EX.VELO ^EY.VELO
```

```
SAVE OUTF=data2.str
```

Note: You can customize the contents of the saved file by using the **OUTPUT** statement.

22.55 SET

The **SET** statement is used to define variables for substitution into Atlas syntax. **SET** commands are executed by DeckBuild.

Note: Full documentation of the **SET** statement is found in the DeckBuild User's Manual.

Numeric Variable Example

Define a numerical variable. Use it in a calculation and substitute it into the Atlas syntax for `REGION` definition

```
SET MYLENGTH=0.1
SET HALFLENGTH= $"MYLENGTH"*0.5
...
REGION NUM=1 MATERIAL=SILICON X.MIN=$"HALFLENGTH" X.MAX=$"MYLENGTH"
```

String Variable Example

Define a string variable to use as part of a filename

```
SET LABEL=testcase1
..
LOG OUTF=$"LABEL".log
..
SAVE OUTF=bias_$"LABEL"_25.str
```

This will produce files called `testcase1.log` and `bias_testcase1_25.str`.

22.56 SINGLEEVENTUPSET

SINGLEEVENTUPSET specifies the values of parameters used in Single Event Upset modeling.

Syntax

```
SINGLEEVENTUPSET <parameters>
```

Parameter	Type	Default	Units
A1	Real		
A2	Real		
A3	Real		
A4	Real		
B1	Real		
B2	Real		
B3	Real		
B4	Real		
B.DENSITY	Real	0	cm^{-3}
BEAM.RADIUS	Real	0	μm
DENSITY	Real		cm^{-3}
DEVICE	Character		
STRUCTURE	Character		
ENTRYPOINT	Real	vector	μm
EXITPOINT	Real	vector	μm
F.SEU	Character		
PCUNITS	Logical	False	
RADIALGAUSS	Logical	False	
RADIUS	Real		μm
RESCALE	Logical	False	
TFINAL.SEU	Real		
T0	Real		s
TC	Real		s
TF	Real		s
UNIFORM	Logical	False	

Description

A1, **A2**, **A3**, and **A4**	The first set of parameters for the length dependence of the charge generation pulse.
B1, **B2**, **B3**, and **B4**	The second set of parameters for the length dependence of the charge generation pulse.
B.DENSITY	Specifies the number of electron-hole pairs per unit volume or generated charge in pico Coulumbs per micron if the PCUNITS parameter is specified.
BEAM.RADIUS	This is the radius of the beam where the generation rate is maintained constant. Beyond this point the generation will decay by either an exponential or by a Gaussian function.
DENSITY	Specifies the number of electron-hole pairs per unit volume generated along the alpha particle track.
DEVICE	Specifies which device the **SINGLEEVENTUPSET** statement applies to in MixedMode. The synonym for this parameter is **STRUCTURE**.
ENTRYPOINT	Specifies the X, Y, and Z coordinates of the beginning of the alpha particle track. The specified point should belong to the semiconductor region.
EXITPOINT	Specifies the X, Y, and Z coordinates of the end of the alpha particle track. The specified point should belong to the semiconductor region.
F.SEU	Specifies the name of C-Interpreter function describing the SEU generation rate as a function of position and time.
PCUNITS	Sets the units of B.DENSITY to be pC per micron.
RADIALGAUSS	Specifies the Gaussian radial dependence of the charge generation pulse. By default, the exponential dependence is used.
RADIUS	Specifies the radius of the alpha particle track.
RESCALE	This causes nodal generation rates to be scaled by the ratio of the integral of the analytic generation rate divided by the numerically integrated value.
STRUCTURE	This is a synonym for **DEVICE**.
TFINAL.SEU	Specifies the finish time for the track, this defaults to the finish time of the transient simulation if TFINAL.SEU is not specified.
T0	Specifies the peak in time of the charge generation pulse.
TC	Specifies the width of the charge generation pulse.
UNIFORM	Specifies that a uniform generation rate corresponding to that specified by the DENSITY parameter is applied.

SEU Example

This statement specifies a track path, radius and density:

```
SINGLEEVENTUPSET ENTRYPOINT="1.5,2.0,0.0"\
                 EXITPOINT="1.0,1.0,4.0 RADIUS=0.05" \
                 DENSITY=1E18
```

Note: For user-defined Single Event Generation profiles, the C-Interpreter function `F.SEU` on the **SINGLEEVENTUPSET** statement can be used.

22.57 SOLAR

The **SOLAR** statement is used to simplify standard analysis of solar cells.

Syntax

```
SOLAR <parameters>
```

Parameter	Type	Default	Units
ANODE	Integer		
AZIMUTH	Real	0.0	Degrees
BEAM	Integer		
DV	Integer	0.05	Volts
ELEVATION	Real	90.0	Degrees
IA	Character		
IV	Character		
IW	Character		
LATTITUDE	Real	35.0	Degrees
MAX.ANGLE	Real	85.0	Degrees
MAX.WAVE	Real	microns	
MIN.DV	Real	0.002	Volts
MIN.WAVE	Real	microns	
RTO	Character		
STEP.ANGLE	Real	5.0	Degrees
STEP.WAVE	Real	microns	
VANODE	Real	0.5	Volts

Description

ANODE	Specifies the electrode index of the "anode". The "anode" is the electrical contact where positive voltages are applied to extract the standard current-voltage characteristic of a solar cell. If the ANODE parameter is left unspecified, the simulator will search for a contact named "anode".
AZIMUTH	Specifies the solar cell azimuth angle (angle with respect to due south) for orbital simulation of annual energy collection.
BEAM	Specifies the beam index of the beam used as a solar source. If the BEAM parameter is left unspecified, the simulator will automatically choose the first beam defined on a **BEAM** statement.
DV	Specifies the initial voltage step increment for extraction of the standard IV characteristic.
ELEVATION	Specifies the solar cell elevation angles (angle from the horizon) for orbital simulation of annual energy collection.
IA	Specifies the name of a file for collection of efficiency versus angle data. You must specify the IA parameter to perform the efficiency versus angles extraction.
IV	Specifies the name of a file for collection of the standard current-voltage characteristics of the solar cell. You must specify the IV parameter to perform standard current voltage extraction including the extraction of Voc, Isc, Pmax, and FF.
IW	Specifies the name of a file for collection of solar cell efficiency versus wavelength. If you specify this parameter, you must also specify the MIN.WAVE, MAX.WAVE, and STEP.WAVE parameters.
LATTITUDE	Specifies the lattitude of the solar cell for orbital simulation of the annual energy collection.
MAX.ANGLE	Specifies the maximum angle for angular efficiency collection (see **IA**). This value must be less than 90° and greater than 0°.
MAX.WAVE	Specifies the maximum wavelength for collection of spectral response of the solar cell (see **IW**).
MIN.DV	Specifies minimum voltage step at open circuit conditions for the extraction of standard solar cell current-voltage characteristics (see **IV**).
MIN.WAVE	Specifies the minimum wavelength for collection of spectral response of the solar cell (see **IW**).
RTO	Specifies the name of a file where the solar cell run-time output will separately be written.

STEP.ANGLE	Specifies the angle step in degrees for angular efficiency collection (see **IA**).
STEP.WAVE	Specifies the wavelength step for collection of spectral response of the solar cell (see **IW**).
VANODE	Specifies the anode operating voltage for extraction of angular response (see **IA**) or spectral response (see **IW**) or both. If the current-voltage characteristics are extracted (see **IV**) the voltage at maximum power, Pmax, will be used as the operating voltage.

22.58 SOLVE

SOLVE instructs Atlas to perform a solution for one or more specified bias points.

Syntax

```
SOLVE [<ion>] <dc> [<fp>][<ep>][<tp>][<ac>][<photo>] [<thermal>]
```

Parameter	Type	Default	Units
AC.ANALYSIS	Logical	False	
AFINAL	Real	85.0	Degrees
ANAME	Character		
ANGLE	Real	0.0	Degrees
ASCII	Logical	False	
ASTEP	Real	5.0	Degrees
AUTO	Logical	True	
B<n>	Real	0.0	W/cm^2
BEAM	Integer		
BRANIN	Logical	True	
C.IMAX	Real	10^{-4}	A
C.IMIN	Real	-10^{-4}	A
C.VSTEP	Real	0.0	V
C.VMAX	Real	5.0	V
C.VMIN	Real	-5.0	V
CNAME	Character		
COMPLIANCE	Real		
CONTINUE	Logical	False	
CURVETRACE	Logical	False	
CYCLES	Integer	1	
CYCLIC.BIAS	Logical	False	
CYCLIC.RELAX	Real	0.2	
CYCLIC.TOL	Real	1.0×10^{-5}	
DDMS.LOG	Logical	False	
DECAY	Real		s

Parameter	Type	Default	Units
DELTAV	Real	0.1	V
DEVDEG.GF.E	Logical	False	
DEVDEG.GF.H	Logical	False	
DIRECT	Logical	False	
DT	Real	0	s
DT.CBET	Logical	False	
DT.CUR	Logical	False	
DT.METH	Real	1	
DT.VBET	Logical	False	
DT.VBHT	Logical	False	
E.CHECK	Integer		
E.COMPLIANCE	Real		
E.CRIT	Real	1.0×10^{-8}	
ERROR.FILE	Character		
ERROR.ONEFILEONLY	Logical	True	
ELECTRODE	Integer		
EMAX	Real	0.0	eV
EMIN	Real	0.0	eV
FERRODAMP	Real	1.0	
ENDRAMP	Real		s
FERRODAMP	Real	1.0	
FREQUENCY	Real		Hz
FREEZEBQPNI	Logical	False	
FREEZESPECIES	Logical	False	
FSTEP	Real	0	Hz
GRAD	Logical	False	
I<n>	Real		A/μm
IFINAL	Real		A/μm
IMPACT.I	Logical	False	

Parameter	Type	Default	Units
IMULT	Logical	False	
INAME	Character		
INDEX.CHECK	Logical	False	
INT.INCREMENT	Logical	False	
INITIAL	Logical	False	
ION.CRIT	Real	1.0	
ION.ELEC	Integer	see Description	
IONIZINT	Logical	False	
IONLINES	Integer	50	none
IONSTOP	Logical	True	
ISTEP	Real	0.0	A/μm
IT.SAVE	Character		
ITAT.FILE	Character		
KP.BANDS.RECALC	Logical	True	
KP.OFF	Logical	False	
KP.POTENTIAL.AVG	Logical	False	
KP.PREV	Logical	False	
LAMBDA1	Real		μm
LIT.STEP	Real		W/cm^2
LMAX	Real	0.0	μm
LMIN	Real	0.0	μm
LOCAL	Logical	False	
LRATIO	Real	1.0	
L.WAVE	Real		μm
MASTER	Logical	False	
MAX.INNER	Integer	25	
MLOCAL	Logical	False	
MONOTONIC	Logical	False	
MULT.FREQ	Logical	False	

Parameter	Type	Default	Units
N.BIAS	Real		V
NAME	Character		
NB1	Real		V
NB2	Real		V
NB3	Real		V
NB4	Real		V
NB5	Real		V
NB6	Real		V
NB7	Real		V
NB8	Real		V
NEGATIVE	Logical	False	
NEGF.LOG	Logical	False	
NFSTEPS	Integer	0	
NIT.N	Real	0.0	cm^{-3}
NIT.P	Real	0.0	cm^{-3}
NIT.XMAX	Real	right hand side	μm
NIT.XMIN	Real	left hand side	μm
NIT.YMAX	Real	top of region	μm
NIT.YMIN	Real	bottom of region	μm
NLAYERS	Real	15	
NOCONTACT	Logical	False	
NOCURRENT	Logical	False	
NOINTERFACE	Logical	False	
NOISE.SS	Logical	False	
NSAMP	Integer	100	
NSTEPS	Integer	0	
OFFSET.ANGLE	Real	0.0	Degrees
ONEFILEONLY	Logical	See Description	
OUT.FILE	Character		

Parameter	Type	Default	Units
OUTFILE	Character		
P.BIAS	Real		
PB1	Real		V
PB2	Real		V
PB3	Real		V
PB4	Real		V
PB5	Real		V
PB6	Real		V
PB7	Real		V
PB8	Real		V
PIEZSCALE	Real	1.0	
POSITIVE	Logical	False	
POWER<n>	Real		Ω
POWERFINAL	Real		Ω
PREVIOUS	Logical	False	
PROJ.MOD	Logical	False	
PROJECT	Logical	False	
PULSE.WIDTH	Real		s
Q<n>	Real		C/μm
QFACTOR	Real	1.0 or previous value	
QFINAL	Real		C/μm
QSCV	Logical	False	
QSTEP	Real		C/μm
RAMPTIME	Real		s
RAMP.LIT	Logical	False	
RELATIVE	Logical	False	
SCAN.SPOT	Integer		
S.OMEGA	Real	1.0	
SINUAMP.COMP	Real	0.0	

Parameter	Type	Default	Units
SINUVAR.COMP	Real	0.0	
SONOS	Logical	False	
SOR	Logical	False	
SPECTRUM	Character		
SQPULSE	Logical	False	
SP.LAST	Logical	False	
SS.LIGHT	Real	0.001	W/cm^2
SS.PHOT	Logical	False	
TD1 ... TD10	Real		S
T<n>	Real		K
T.COMP	Real	0	K
T.SAVE	Real	0.0	s
TABLE	Character		
TDELAY	Real	0.0	s
TFALL	Real	0.0	s
TRISE	Real	0.0	s
TSAVE.MULT	Real	1.0	
TEMPFINAL	Real		K
TERMINAL	Integer	all contacts	
TIMESPAN	Real	∞	seconds
TOLERANCE	Real	1.0×10^{-5}	
TRANS.ANALY	Logical	False	
TSTOP	Real		
TWOFILESONLY	Logical	False	
V<n>	Real		V
VFINAL	Real		V
VMULT	Logical	False	
VSTEP	Real	0.0	V
VSS	Real	0.1	KT/Q

Parameter	Type	Default	Units
WAVE<n>	Logical	False	
WFINAL	Real		microns
WAVEFORMS	Character		
WSTEP	Real		microns

Description

Each **SOLVE** statement must specify an initial bias condition. Once any DC condition has been solved, either a transient or AC analysis may be performed. You may also solve for carrier generation due to incident light under DC, or AC analysis transient conditions.

dc	This is one or more of the DC bias parameters
fp	This is one or more of the file parameters
ep	This is one or more of the initial guess or estimate parameters. Estimate parameters are used to specify how the initial approximation for the solution is to be obtained.
tp	This is one or more of the transient parameters. These parameters are used to specify data for transient analysis.
ac	This is one or more of the AC parameters. AC parameters are used to specify data for AC analysis.
ion	This is a set of the ionization integral parameters.
NEGF.LOG	Specifies that spectra of transmission, DOS, density and and curent obtained from NEGF model need to be saved.
photo	This is one or more of the photogeneration parameters. Photogeneration parameters are used to specify illumination data.
SP.LAST	Allows to solve Schrodinger equation in the last point of a bias sweep, as a post processing step after classical solution is obtained. When the parameter is off, a Schrodinger solution for each bias point is performed.
therm	This is one or more of the thermal parameters. Thermal parameters are used for obtaining solutions in Thermal 3D.

DC Parameters

C.IMAX	Specifies the maximum current for curve tracing.
C.IMIN	Specifies the minimum current for curve tracing.
C.VSTEP	Specifies the initial voltage step for curve tracing.
C.VMAX	Specifies the maximum voltage for curve tracing.
C.VMIN	Specifies the minimum voltage for curve tracing.

CONTINUE	This is an alias for **CURVETRACE**.
CURVETRACE	Initiates curve tracing. See also the **CURVETRACE** statement. The alias for this parameter is **CONTINUE**.
DDMS.LOG	Specifies that in case of DDMS model (see Section 14.9 "Drift-Diffusion Mode-Space Method (DD_MS)"), an additional log file with subband-resolved quantities will be stored along with structure file. You must also set MASTER and OUT.
DEVDEG.GF.E	Enables the Boltzmann transport equation solver for electrons using the latest potential distribution.
DEVDEG.GF.H	Enables the Boltzmann transport equation solver for holes using the latest potential distribution.
ELECTRODE	Specifies electrodes that you wish to add voltage and current increments (VSTEP and ISTEP) to. If n electrodes are to be stepped, ELECTRODE should be an n-digit integer, where each of the digits is a separate electrode number. See also the NAME parameter.
FREEZEBQPNI	Forces the Bohm Quantum Potential full Newton solver to use the stored value of intrinsic carrier density. It is recommended to use this on all solutions except equilibrium solutions.
FREEZESPECIES	This prevents the solution of the generic species transport equations on a steady state solve. This is appropriate if the timscale for the ionic transport is much longer than the assumed observation time of the device.
I<name> or **I(<name>)**	Specifies the applied current for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel, and ground. No other user-defined names are allowed. This parameter is used when current boundary conditions are selected (see **CONTACT**).
I<n>	Specifies the terminal current for electrode, n. This parameter is used when current boundary conditions are selected (see **CONTACT**). Normally, I defaults to the current from the previous bias point. It is more usual to use electrode names rather than numbers. This parameter is superseded by I<name>.
IFINAL	Specifies the final current value for a set of bias increments. If IFINAL is specified, either ISTEP or NSTEPS must be specified.
IMULT	Specifies that the current (for current boundary conditions) be multiplied by ISTEP rather than incremented.
ISTEP	Specifies a current increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If ISTEP is specified, either IFINAL or NSTEPS must also be specified.

KP.BANDS.RECALC	Specifies that each corrector call to **k.p** solver should recalculate the entire band structure with the updated potentials and carrier densities.
KP.OFF	Turns off the **k.p** calculation for the solve where it is specified
KP.POTENTIAL.AVG	Specifies that the electrostatic potential in the quantum region be replaced by its average value over the region.
KP.PREV	Turns off the **k.p** calculation but uses the results from the previous **k.p** calculation. If no previous calculation is available, a calculation is initiated.
N.BIAS	Specifies fixed electron quasi-Fermi potentials if electron continuity is not being solved. If N.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. But if FIX.QF is set in the METHOD statement, the quasi-Fermi levels will be set to the maximum bias.
NB<n>	Allows region by region specification of N.BIAS. The n index corresponds to the region index for which the specified value of electron quasi-fermi level applies.
NAME	Specifies that the named electrode is to be ramped. Custom electrode names are supported by name. See also the **V<name>** parameter.
NSTEPS	Specifies the number of DC bias increments.
P.BIAS	Specifies fixed hole quasi-Fermi potentials if hole continuity is not being solved. If P.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. But if FIX.QF is set in the **METHOD** statement, the quasi-Fermi levels will be set to the minimum bias.
PB<n>	Allows region by region specification of P.BIAS. The n index corresponds to the region index for which the specified value of hole quasi-fermi level applies.
Q<name> or **Q(<name>)**	Specifies the charge on a named electrode. These names are as follows: gate, gg, fgate, cgate, ngate, and pgate. No other user-defined names are allowed.This parameter is used when floating or charge boundary conditions are selected (see **CONTACT**).
Q<n>	Specifies the charge on electrode number, n. It is more usual to use electrode names rather than numbers. This parameter is superseded by Q<name>.
QFINAL	Specifies the final charge for a set of bias increments. If QFINAL is specified, either QSTEP or NSTEPS must also be specified.
QSCV	Specifies the Quasi-static Capacitance calculation. This only has meaning when the **SOLVE** statement is ramping a bias. This requires a small voltage increment between solutions for best results.
QSTEP	specifies a charge increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If QSTEP is specified, either QFINAL or NSTEPS must also be specified.

TIMESPAN	Specifies the simulated time for a bias ramp in the HEIMAN model. The ramp is assumed to be linear in time, and so the time taken to change by VSTEP volts is TIMESPAN times VSTEP divided by Total Bias Change.
V<name> or **V(<name>)**	Specifies the bias voltage for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel, and ground. No other user-defined names are allowed.
V<n>	Specifies the bias voltage for electrode, n. Normally, Vn, defaults to the potential from the previous bias point. It is more usual to use electrode names rather than numbers. This parameter is superseded by V<name>.
VFINAL	Specifies the final voltage for a set of bias increments. If VFINAL is specified, either VSTEP or NSTEPS must also be specified.
VMULT	Specifies that the voltage (for voltage boundary conditions) be multiplied by VSTEP rather than incremented.
VSTEP	Specifies a voltage increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If VSTEP is specified, you must either specify VFINAL or NSTEPS.

File Output Parameters

EMAX and **EMIN**	Specify the energy range for output of spectral data in the file named by the SPECTRUM parameter.
ERROR.FILE	Specifies the name of a file that contains the equation errors. This file will be written at every iteration for the current **SOLVE** statement.
ERROR.ONEFILEONLY	Specifies that the file specified by ERROR.FILE will overwritten at each iteration.
IMPACT.I	Specifies that impact ionization rate should be saved to the structure file. If impact ionization is not enabled, these results will not be self-consistent.
INT.INCREMENT	Specifies that the last character of OUTFILE will not be incremented for each structure file. Instead an integer corresponding to the bias point number will be appended to the OUTFILE along with .str.
IT.SAVE	Specifies the name of an output file where the structure information will be stored after every GUMMEL or NEWTON iteration. The ASCII code of the last non-blank character of the supplied file name will be incremented by one for each iteration, resulting in an unique file per iteration.
ITAT.FILE	Gives the filename stem for log files that contain information about the Ielmini Inelastic Trap-Assisted-Tunneling model (see "Ielmini Inelastic Trap Assisted Tunneling Model" on page 299).

LMAX and **LMIN**	Specify the wavelength range for spectral output in the file named by the SPECTRUM parameter.
MASTER	Specifies that the output file selected by the OUTFILE parameter will be written in a standard structure file rather than in the older PISCES-II binary format.
ONEFILEONLY	Specifies that only one filename will be used to save solutions during a bias ramp. If this parameter is specified, filename incrementing described under OUTFILE is not applied. This parameter is true by default unless the NAME or ELECTRODE parameters or transient simulations specified. When CURVETRACE is used, you need to explicitly set this parameter to false using ^ONEFILEONLY to save a different filename at each bias point.
OUT.FILE	This is an alias for **OUTFILE**.
OUTFILE	Specifies the name of the output file where bias point solution information will be stored. If an electrode is stepped so that more than one solution is generated by the **SOLVE** statement, the ASCII code of the last non-blank character of the supplied file name will be incremented by one for each bias point in succession, resulting in a unique file per bias point. The alias for this parameter is **OUT.FILE**.
SPECTRUM	Specifies the name of an output file where LED spectral data are stored after each step.
TD1 ... TD10	These specify the elapsed times at which the effects of degradation are calculated in the General framework degradation model. An output file is created for each specified elapsed time for further analysis.

Note: The output file specified by OUTFILE has a limit of 132 characters on a UNIX system and eight characters on a PC system.

Initial Guess Parameters

FERRODAMP	Specifies a scaling factor which can be used to attain initial convergence of the Ferroelectric model. This parameter can be ramped from a small positive value up to a value of 1.0 on subsequent SOLVE statements.
INITIAL	Sets all voltages to zero. If this parameter is not specified for the first bias point in a given structure a **SOLVE** INIT statement is automatically inserted. The **SOLVE** INIT statement is always solved in zero carrier mode with no external elements attached. It is used to provide a good initial guess to subsequent solutions.
LOCAL	Specifies that the initial approximation should use local values of quasi-Fermi levels. See Section 21.6 "Initial Guess Strategies" for more detailed information.
MLOCAL	Specifies the modified local initial guess. This parameter is used when solved for carrier temperatures in the energy balance models. If any energy balance model is specified, MLOCAL defaults to true.

NOCURRENT	Specifies an initial guess strategy that solves the non-linear Poisson equation and uses this solution as the initial guess. This will typically give good results for situations where the quasi-fermi levels are almost flat and the current is consequently low. Under these conditions, you can solve almost any desired bias point directly.
PREVIOUS	Specifies that the previous solution as the initial approximation.
PROJ.MOD	Gives an alternative to the PROJ initial guess strategy in the case of one carrier type solution by making a different update for the unsolved carrier concentration.
PROJECT	Specifies that an extrapolation from the last two solutions will be used as an initial approximation. This parameter may be used if there are two or more existing solutions and equivalent bias steps are taken on any electrodes that are changed.

Note: If no initial guess parameters are specified, Atlas will use PROJECT wherever possible.

QFACTOR	Specifies as scaling factor to improve initial guesses when the QUANTUM model is used. This parameter should be ramped slowly from zero to unity at the start of quantum effect simulations.

Compliance Parameters

Compliance parameters define a current limit for a DC bias ramp or transient simulation. You can terminate the simulation by monitoring the current and checking against a user-defined limit.

COMPLIANCE	Sets a limit on the current from the electrode which has been specified by the CNAME or E.COMPLIANCE parameter. When the COMPLIANCE value is reached, any bias ramp is stopped and the program continues with the next line of the input file. The COMPLIANCE parameter is normally specified in A. If the GRAD parameter is specified, COMPLIANCE is specified in A/V.
E.COMPLIANCE	Specifies the electrode number to be used by the COMPLIANCE parameter. See also **CNAME**.
CNAME	Specifies the name of the electrode used by the COMPLIANCE.
GRAD	Specifies that the compliance value is a current/voltage gradient, and not a current value.
E.CHECK	Specifies the electrode number for compliance testing for monotonicity, positiveness or negativeness.

MONOTONIC	Specifies that current in the electrode associated with the index specified on the E.CHECK parameter must remain monotonic during a static ramp. If this condition is violated, the solution ramp is immediately exited.
POSITIVE	Specifies that current in the electrode associated with the index specified on the E.CHECK parameter must remain positive during a static ramp. If this codition is violated, the solution ramp is immediately exited.
NEGATIVE	Specifies that current in the electrode associated with the index specified on the E.CHECK parameter must remain negative during a static ramp. If this codition is violated, the solution ramp is immediately exited.

Transient Parameters

CYCLES	Specifies the number of periods to be simulated (both FREQUENCY and TRANS.ANALY must be specified when this parameter is used). The synonym for this parameter is PERIODS.
CYCLIC.BIAS	Specifies that a cyclic bias [309] simulation is being performed. Cyclic biasing allows the effects of repeated trapezoidal/square transient pulses (see SQPULSE) on the output characteristics to be determined, without having to do the extremely large numbers of cycles. When CYCLIC.BIAS is specified, Atlas will calculate a series of trapezoidal/square pulses using the SQPULSE parameters. At the end of the third cycle (and at all subsequent cycles) the values of the potential (V), electron concentration (n), hole concentration (p) and trap probability of occupation (f_T) will be modified according to the equation:

$$x(k+1) = x(k) - \text{CYLIC}\cdot\text{RELAX}*((x(k)-x(k-1))*dx(k)/(dx(h)-dx(k-1)) \quad \text{22-6}$$

where x is a parameter such as V, n, p, or f_T at every node point, $x(k)$ is the updated value calculated from the equation, k is the cycle number (with k+1 being the new update for parameter x), CYCLIC.RELAX is the relaxation factor, $\delta x(k)$ is the difference between the simulated values of x at the start and beginning of the current cycle, while $\delta x(k\text{-}1)$ is the equivalent for the previous cycle.

Steady state cyclic convergence is determined by comparing the normalizing sum of the updated values of V, n, p, and f_T with a tolerance value (CYCLIC.TOL).

CYCLIC.RELAX	Specifies the CYCLIC.BIAS relaxation factor (the recommended range for this parameter to ensure stable convergence is between 0.2 and 1).
CYCLIC.TOL	Specifies the CYCLIC.BIAS tolerance factor.
DECAY	Specifies the time constant used for defining an exponential change in bias for transient simulation.
ENDRAMP	Applies any bias changes as linear ramps. ENDRAMP specifies the exact end of the ramp in running time (i.e., the ramp will start at $t=t_0$ and end at t=ENDRAMP).

NSTEPS	This can be used to signal the end of the run (i.e., the final time would be $t = t +$ NSTEPS*DT). You can use this parameter instead of the TSTOP parameter.
PULSE.WIDTH	Specifies the time constant used for sinusoidal non-linear time domain simulation. If SQPULSE is specified, PULSE.WIDTH is the width of the trapezoidal/square pulse not including the rise and fall times.
RAMPTIME	Applies any bias changes as linear ramps. RAMPTIME specifies a ramp interval in seconds (i.e., the ramp will begin at $t=t_0$ and ends at $t=t_0$ +RAMPTIME).
RELATIVE	Specifies that the TSTOP value is relative to the current time value.
SINUAMP.COMP	Specifies that sinusoidal amplitude compliance will be used. The transient simulation will stop if the amplitude of a sinusoidal waveform is less than the value of SINUAMP.COMP.
SINUVAR.COMP	Specifies that sinusoidal variance compliance will be used. The transient simulation will stop if the change in amplitude of a sinusoidal waveform is less than SINUVAR.COMP.
SQPULSE	Specifies that multiple trapezoidal/square transient pulses will be applied. The pulse is controlled by the TDELAY, TRISE, PULSE.WIDTH, TFALL, and FREQUENCY parameters.
T.COMP	Specifies a temperature for temperature compliance. Once the specified temperature is obtained at some location on the mesh the solution process is discontinued.
T.SAVE	Specifies a time increment at which the device structure files are saved.

Note: Actual time steps may not correspond to the user-specified increment.

TABLE	Specifies the filename of a table of time and bias values for the specified electrode. The following format must be used for the TABLE file.
	`time1 bias1`
	`time2 bias2`
	`time3 bias3`
	`....  ....`
	`end`
	The time value should be in seconds. The value of the bias should be in the same units as displayed in Atlas (i.e., Volts for a voltage controlled electrode, A/um for a current controlled electrode, and A for a current controlled electrode in Device 3D or Atlas with the WIDTH parameter specified on the **MESH** statement). There's no limit to the number of time/bias pairs that can be defined in the table file. The keyword, END, must be placed after the last time/bias pair. Linear interpolation will be used to determine the bias for time points that aren't specified in the table file.

Note: An electrode must also be specified when using the TABLE parameter. The electrode can be specified either by name (e.g., NAME=anode) or by number (e.g., ELECTRODE=1).

TDELAY	Specifies the time delay before the first cycle of multiple trapezoidal/square transient pulses (SQPULSE) will be applied.
TFALL	Specifies the fall time for trapezoidal/square transient pulses (SQPULSE).
TRISE	Specifies the rise time for trapezoidal/square transient pulses (SQPULSE).
TRANS.ANALY	Specifies that transient analysis is being performed and that a sinusoid with a frequency, specified by the FREQUENCY parameter, should be applied.
TSAVE.MULT	Specifies a multiplier the save time increment, T.SAVE, is multiplied by after each time a structure is saved.
TSTEP or **DT**	Specifies the time-step to be used. For automatic time-step runs, DT is used to select only the first time step (see Section 22.35 "METHOD").
TSTOP	Specifies the end of the time interval. If simulation begins at $t=t_0$ and RELATIVE is specified, it will end at t=TSTOP+t_0.
WAVE<n>	Specifies that the WAVEFORM, n (where n is between 1 and 10), will be enabled for the current **SOLVE** statement. This parameter should appear on every SOLVE statement where the specified WAVEFORM is to be applied.

Note: You can solve more than one WAVEFORM at the same time.

WAVEFORMS	Specifies multiple WAVEFORMs to be enabled for the current **SOLVE** statement. The WAVEFORM numbers should be specified as a comma separated list. For example, SOLVE WAVEFORMS= "1, 2, 12" TFINAL=1e-6. This will enable WAVEFORM numbers 1, 2, and 12.

AC Parameters

AC.ANALYSIS	Specifies that AC sinusoidal small-signal analysis should be performed after solving for the DC condition. The full Newton method must be used for this analysis. This is typically specified with the statement: **METHOD** NEWTON CARRIERS=2
ANAME	Specifies the name of the electrode to which the AC bias will be applied. See also TERMINAL. If no ANAME is specified, all electrodes have AC bias applied in turn.
AUTO	Selects an automatic AC analysis method. This method initially uses SOR. The DIRECT method will be used if convergence problems occur. We strongly recommend the use of the AUTO parameter.
DIRECT	Selects the direct AC solution method. This method is robust at all frequencies, but slow.

FREQUENCY	Specifies the AC analysis frequency. Analysis may be repeated at a number of different frequencies (without solving for the DC condition again) by specifying FSTEP. FREQUENCY can also be used to specify a sinusoidal or square pulse frequency for transient simulations.
FSTEP	Specifies a frequency increment which is added to the previous frequency. If MULT.FREQ is specified, the frequency will be multiplied by FSTEP.
MAX.INNER	Specifies the maximum number of SOR iterations.
MULT.FREQ	Specifies that the frequency will be multiplied by FSTEP, instead of added to FSTEP.
NFSTEPS	Specifies the number of times that the frequency is to be incremented by FSTEP.
S.OMEGA	Specifies the SOR parameter. This parameter is not the AC frequency.
SOR	Selects the SOR AC solution method. Although SOR is fast, it should only be used when you are performing simulations at low frequencies. Low frequency can be defined here as at least an order of magnitude below the cutoff frequency.
TERMINAL	Specifies the electrode number to which the AC bias will be applied. Although more than one contact number may be specified (via concatenation), each will be solved separately. Each contact that is specified generates a column of the admittance matrix. If no TERMINAL is specified, all electrodes have AC bias applied in turn. See also **ANAME**.
TOLERANCE	Specifies SOR convergence criterion.
VSS	Specifies the magnitude of the applied small-signal bias. The approach used for small-signal analysis constructs a linear problem based on derivatives calculated during Newton iterations, so adjusting VSS will generally not affect the results.

NOISE Parameters

BRANIN	Specifies that Branin's method should be used in the calculation of the impedance field. The direct method (which is the alternative to Branin's method) is very slow. Therefore, we recommend that BRANIN is never set to False.
NOINTERFACE	Specifies that any element which is adjacent to an interface should be ignored in the noise calculation.
NOCONTACT	Specifies that any element which is adjacent to a contact should be ignored in the noise calculation.
NOISE.SS	Specifies that a small-signal noise simulation should be performed.

Direct Tunneling Parameters

An alternative set of direct quantum tunneling models for calculating gate current can be set using DT.CUR on the **SOLVE** statement. If you set this parameter, then select the particular

model using DT.METH. The allowed values of DT.METH are 1, 2 and 3. The default value of DT.METH is 1.

DT.METH=1	Selects a Fowler-Nordheim type model, with a correction for the trazepoidal, rather than the triangular shape of the potential barrier.
DT.METH=2	Selects a WKB model with the assumption of a linearly varying potential. It involves a calculation of the classical turning points and integration between those limits.
DT.METH=3	Selects an exact formula for a trapezoidal barrier that involves Airy function solutions.

You can also need to specify the type of tunneling taking place by using at least one of the following flags.

DT.CBET	Specifies electron tunneling from conduction band to conduction band.
DT.VBHT	Specifies holes tunneling from valence band to valence band.
DT.VBET	Specifies Band-to-Band tunneling.

Note: Specifying DT.CUR on the **SOLVE** statement invokes the post-processing version of Direct tunneling calculation of Gate current. Specifying the same parameters on the **MODELS** statement enables a self-consistent version.

Ionization Integral Parameters

Ionization parameters are used to calculate ionization integrals. No calculation will take place unless the IONIZINT parameter is specified.

DELTAV	Since the electric field can be near zero at electrodes, the electric field line calculations begin at a distance from the electrode. A potential contour is drawn around the electrode at a distance where the potential is DELTAV less than the applied bias on the contact. Defaults to 0.1 V, but will typically need to be increased especially for power devices and heavily doped contact regions.
E.CRIT	Specifies the minimum electric field used to calculate integration integrals. Field lines will terminate when the field drops below E.CRIT.
INAME	Specifies the electrode name from which the electric field lines are calculated. The default is to use the same as electrode as specified in NAME.
ION.CRIT	Specifies the critical value of the ionization integral used by IONSTOP to terminate the simulation. When the critical value is reached, any bias ramp will be terminated and the next line of the input file will be executed.
ION.ELEC	Specifies the electrode from which the electric field lines are calculated. This parameter defaults to the electrode which is being ramped (if any).
IONIZINT	Enables the calculation of ionization integrals along electric field lines.

IONLINES	Specifies the number of electric field lines to be calculated. Ionization integrals are calculated along each line.
IONSTOP	Stops the bias ramp if integral is greater than 1.0
LRATIO	Specifies the ratio between the starting points of the electric field lines. An LRATIO value of between 0.5 and 1.5 is recommended. LRATIO=1 means that the spacing between the starting points of the electric field lines is equal. LRATIO < 1 means more lines start towards the left hand side of the structure. LRATIO > 1 means more lines start towards the right hand side.

Photogeneration Parameters

All these parameters require Luminous to be licensed for correct operation.

AFINAL	Specifies the final angle for angular dependence analysis.
ANGLE	This is an alias for **OFFSET.ANGLE**.
ASTEP	Specifies the step size for angular analysis.
B<n>	Specifies the optical spot power associated with optical beam number, n. The beam number must be an integer from 1 to 10.
BEAM	Specifies the beam number of the optical beam when AC angular or spectral photogeneration analysis is performed. Unlike the ELECTRODE parameter, this parameter can only be used to specify a single beam.
INDEX.CHECK	Specifies that the real and imaginary refractive indices used in the ray tracing of each beam will be printed to the run-time output. This parameter can be used as confirmation of the input of user-defined refractive indices or to check the default parameters. The indices are only printed when you perform the ray trace at the first reference to a given beam on the **SOLVE** statement.
L.WAVE	Specifies the luminous wavelength to use to calculate the luminous power estimate for radiative recombination. If a positive value of L.WAVE is specified the luminous power will be saved in any log file specified for that solution.
LAMBDA1	Specifies the wavelength of the optical source (beam) number 1 for this solution. This parameter can be used to perform analysis of spectral response as a function of wavelength.
LIT.STEP	Selects the light intensity increment of all optical beams which have been specified. This parameter is used when light intensity varies by stepping (similar to the VSTEP parameter). If LIT.STEP is specified, the NSTEP parameter should be used to select the number of steps.
OFFSET.ANGLE	Specifies the offset angle for Luminous angular response modeling (see Section 11.12.15 “Obtaining Angular Response”). The angle of propogation is the sum of the angle specified on the BEAM statement and the value of this parameter.

RAMP.LIT	Specifies that the light intensity is to be ramped when transient simulations are performed. If RAMP.LIT is specified, transient mode parameters such as RAMPTIME, TSTEP, and TSTOP must also be specified. The RAMP.LIT parameter affects all specified optical beams (i.e., all beams are ramped).
SCAN.SPOT	Specifies a beam number for spot scanning. Spot scanning requires you to specify the RAYS parameter of the **BEAM** statement. With this specification, the incident light is split into the user-specified number of rays. During the spot scan, solutions are obtained with the beam energy applied to each of the rays in sequence.
SS.LIGHT	Specifies the intensity in the small signal part of the optical beam when small signal AC photogeneration analysis is performed.
SS.PHOT	Specifies that small signal AC analysis will be performed on the optical beam selected by the BEAM parameter. When SS.PHOT is specified, other AC parameters (e.g., FREQUENCY, FSTEP, MAX.INNER, MULT.FREQ, NFSTEPS, S.OMEGA, and TOLERANCE) should also be specified. You don't need to specify the AC.ANALYSIS parameter when performing small signal AC analysis on optical beams.
WFINAL	Specifies the final wavelength in a spectral analysis ramp.
WSTEP	Specifies the wavelength step size in as spectral analysis ramp.

SONOS Parameters

NIT.N	Specifies the trapped electron concentration in a Silicon-Nitride region for the steady-state **SOLVE**. This can be used in conjunction with the SONOS models or independently from them.
NIT.P	Specifies the trapped hole concentration in a Silicon-Nitride region for the steady-state **SOLVE**. This can be used in conjunction with the SONOS models or independently from them.
NIT.XMAX	Maximum x-ccordinate for applying density specified by NIT.N or NIT.P.
NIT.XMIN	Minimum x-coordinate for applying density specified by NIT.N or NIT.P.
NIT.YMAX	Maximum y-ccordinate for applying density specified by NIT.N or NIT.P.
NIT.YMIN	Minimum y-coordinate for applying density specified by NIT.N or NIT.P.
SONOS	Only used in the case of **SOLVE** INIT with a SONOS structure. Helps prevent **SOLVE** INIT introducing large carrier concentrations in a charged Silicon Nitride region.

Thermal3D Parameters

POWER<n>	Specifies the power in watts on the n-th heat source.
POWERFINAL	Specifies the final power for a linearly ramped heat source.
T<n>	Specifies the temperature in K on the n-th heat sink.
TEMPFINAL	Specifies the final temperature for a linearly ramped heat sink.

For more information about these parameters, see also Section 18.4 “Obtaining Solutions In THERMAL3D”.

DC Conditions Example

The following statement solves for a defined bias point and saves the solution to output file, *OutA*. The voltages on electrodes other than the gate will keep the value from the previous **SOLVE** statement.

```
SOLVE VGATE=0.1 OUTFILE=OutA
```

Bias Stepping Example

In the next example, the bias stepping is illustrated. The two **SOLVE** statements produce the following bias conditions.

Bias Point	Vgate	Vdrain	Vsub
1	0.0	0.5	-0.5
2	1.0	0.5	-0.5
3	2.0	0.5	-0.5
4	3.0	0.5	-0.5
5	4.0	0.5	-0.5
6	5.0	0.5	-0.5

The solutions for these bias points will be saved to the files: OUT1, OUTA, OUTB, OUTC, OUTD, and OUTE. The initial approximation for each bias point is obtained directly from the preceding solution.

For bias points 4, 5, and 6, the program will use a PROJ to obtain an initial approximation. Since, starting with bias point 4, both of its preceding solutions (bias points 2 and 3) only had the same electrode bias (number 1) altered.

```
SOLVE Vdrain=.5 Vsub=-.5 OUTF=OUT1

SOLVE Vgate=1 VSTEP=1 VFINAL=5  NAME=gate OUTF=OUTA
```

Transient Simulation Example

The following sequence is an example of a time dependent solution. The **METHOD** statement specifies second-order discretization, automatic time-step selection, and an automated Newton-Richardson procedure.

The first **SOLVE** statement then computes the solution for a device with 1V on the base electrode and 2V on the collector in steady-state. The second **SOLVE** statement specifies that the base electrode is to be ramped to 2V over a period of 10 ns and is left on until 25 ns. Each solution is written to a file. The name of the file is incremented in a manner similar to that described for a DC simulation (UP1, UP2, and so on). Note that an initial time step had to be specified in this statement.

The third **SOLVE** statement ramps ther base down from 2V to - 0.5V in 20 ns (end of ramp is at =45 ns). The device is then solved at this bias for another 55 ns (out to 100 ns). Each solution is again saved in a separate file (DOWN1, DOWN2, and so on).

No initial timestep was required since one had been estimated from the last transient solution from the previous **SOLVE** statement.

Finally, the fourth **SOLVE** statement performs the steady-state solution at Vbe = -0.5V and Vce = 2V.

```
METHOD 2ND TAUTO AUTONR

SOLVE  Vbase=1 Vcollector=2

SOLVE  Vbase=2 DT=1E-12 TSTOP=25E-9 RAMPTIME=10E-9 OUTF=UP1
```

```
SOLVE  Vbase= –0.5 TSTOP=100E-9 RAMPTIME=20E-9 OUTF=DOWN1
SOLVE  Vbase= –0.5 Vcollector=2
```

AC Analysis Example

The following example illustrates an application of the **SOLVE** statement for AC analysis. It is assumed that the device has three electrodes. This analysis is being performed after DC conditions are solved at V1 = 0V, 0.5V, 1.0V, 1.5V, and 2.0V. A series of 10mV AC signals with frequencies of 1 MHz, 10 MHz, 100 MHz, 1 GHz, 10 GHz, and 100 GHz are applied to each electrode in the device.

Ninety AC solutions will be performed (5 X 6 X 3 = 90).

```
SOLVE V1=0 V2=0 V3=0 VSTEP=0.5 NSTEPS=4 ELECT=1 \
AC FREQ=1E6 FSTEP=10 MULT.F NFSTEP=5 VSS=0.01
```

Photogeneration Examples

The following statement simulates two DC optical beams incident on the device with optical spot powers of 15 and 25 W/cm^2.

```
SOLVE B1=15 B3=25
```

The next example shows how DC spot power of the two optical beams can be stepped simultaneously. Beam #1 will increase to 35 W/cm^2 and beam #3 will increase to 45 W/cm^2.

```
SOLVE LIT.STEP=5.0 NSTEP=4
```

In the next example, the beams are ramped in the time domain. Beam #1 is ramped down to 0 W/cm^2 and beam #3 is ramped up to 100 W/cm^2. The duration of the ramp is 1 ns. After the ramp, simulation will continue for 4 ns.

```
SOLVE B1=0 B3=100 RAMP.LIT TSTEP=2E-11 RAMPTIME=1E-9 TSTOP=5E-9
```

Next, the small signal response of a single beam is analyzed. First, Atlas will solve the DC characteristics at the specified optical spot powers. Then, the AC response of beam #1 will be calculated at a frequency of 10 MHz.

```
SOLVE B1=10 B3=20 BEAM=1 FREQUENCY=1e7 SS.PHOT SS.LIGHT=0.01
```

Frequency stepping is used to look at the small signal AC frequency response of one of the beams. AC response is calculated at frequencies from 1kHz to 100MHz at each decade. The MULT.F parameter is used to geometrically increase the frequency.

```
SOLVE B1=10 B3=5 BEAM=1 SS.PHOT SS.LIGHT=0.01 \
      MULT.F FREQUENCY=1E3 FSTEP=10 NFSTEP=6
```

The following illustrates the specification of analysis of angular response of beam number 2 at angles from 0 to 80° in 1° increments.

```
SOLVE BEAM=2 ANGLE=0 AFINAL=80 ASTEP=1
```

Similarly, the following illustrates the specification of analysis of spectral response over the wavelength range from 1 to 3 microns in steps of 0.05 microns.

```
SOLVE BEAM=1 LAMBDA=1 WFINAL=3 WSTEP=0.05
```

Ionization Integral Example

Ionization integrals are used to estimate the breakdown voltage from analysis of the electric field. They can be used in zero carrier mode providing much faster simulation that conventional breakdown analysis. The ionization syntax uses the parameter IONIZ to enable the ionization integral calculation. An equipotential contour is calculated at a potential DELTAV from the contact NAME (or INAME). Electric field lines are started at distances along this potential contour. IONLINES sets the number of lines and LRATIO is the ratio of the distance between the starting points of the lines. The following syntax is for a PMOS transistor. See the on-line examples for other cases using ionization integrals.

```
IMPACT SELB

METHOD CARR=0

SOLVE VDRAIN=-1 VSTEP=-1 VFINAL=-20.0 NAME=DRAIN \

IONIZ IONLINES=50 LRATIO=0.9 DELTAV=1.2
```

Note: There are over 300 on-line examples supplied with Atlas to provide examples of sequences of **SOLVE** statements applied to practical problems for a variety of device technologies.

22.59 SPX.MESH, SPY.MESH, SPZ.MESH

S<n>**.MESH** specifies the location of grid lines along the <n>-axis in a rectangular mesh used in self-consistent Schrodinger-Poisson Model simulation (see Section 14.2 "Self-Consistent Coupled Schrodinger Poisson Model"). The syntax is equivalent for all directions. **SPX.MESH** and **SPY.MESH** should always be set or nor set together, while **SPZ.MESH** is always optional.

Syntax

```
SPX.MESH NODE=<n> LOCATION=<n>

SPX.MESH SPACING=<n> LOCATION=<n>

SPY.MESH NODE=<n> LOCATION=<n>

SPY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE	Specifies the mesh line index. These mesh lines are assigned consecutively.
LOCATION	Specifies the location of the grid line.
SPACING	Specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

Note: The mesh defined in these statements for the Self-Consistent Schrodinger-Poisson Model is entirely separate from the electrical device simulation mesh defined on the **MESH** statement.

Setting Locally Fine Grids Example

This example shows how to space grid lines closely around a topological feature such as a junction at y=0.85 microns.

```
SPY.MESH  LOC=0.0  SPAC=0.2

SPY.MESH  LOC=0.85 SPAC=0.01

SPY.MESH  LOC=2    SPAC=0.35
```

22.60 SYMBOLIC

Note: In versions 3.0 and greater, the **SYMBOLIC** statement is no longer needed. All functions have been moved to the **METHOD** statement.

22.61 SPREAD

SPREAD distorts rectangular grids defined by Atlas in the vertical direction to follow surface and junction contours.

Note: The use of this parameter is not recommended by Silvaco.

Syntax

```
SPREAD LEFT|RIGHT WIDTH=<r> UPPER=<i> LOWER=<i>
Y.LOWER|THICKNESS [<options>]
```

Parameter	Type	Default	Units
ENCROACH	Real	1.0	
GRADING	Real	1.0	
GR1	Real	1.0	
GR2	Real	1.0	
LEFT	Logical	False	
LOWER	Integer		
MIDDLE	Integer	Halfway between UPPER and LOWER	
RIGHT	Logical	False	
THICKNESS	Real		μm
UPPER	Integer		
VOL.RATIO	Real	0.44	
WIDTH	Real		μm
Y.LOWER	Real		μm
Y.MIDDLE	Real	0.50	μm

Description

SPREAD can reduce the grid complexity of specific simulations. Since the SPREAD statement is somewhat complicated, we suggest that you follow the supplied examples carefully until you're confident to understand the workings of this statement.

Mandatory Parameters

LEFT	Distorts the left side of the grid. If LEFT is specified, RIGHT must not be specified.
LOWER	Specifies the lower y-grid line above which distortion will take place.
RIGHT	Distorts the right side of the grid. If RIGHT is specified, LEFT must not be specified.
THICKNESS	Specifies the thickness of the distorted region. Unless VOL.RATIO is set to 0 or 1, THICKNESS will usually move the positions of both the UPPER and LOWER grid lines. The Y.LOWER and THICKNESS parameters define the distorted grid region. Only one of these parameters should be specified.
UPPER	Specifies the upper y-grid line under which distortion will take place.
WIDTH	Specifies the width from the left or right edge (depending on whether LEFT or RIGHT is selected) of the distorted area. The actual X coordinate specified by WIDTH (min[*x*]+ WIDTH for LEFT, max[*x*] + WIDTH for RIGHT) will lie in the middle of the transition region between distorted and undistorted grid regions.
Y.LOWER	Specifies the physical location in the distorted region to which the line specified by LOWER will be moved. The line specified by UPPER is not moved. The Y.LOWER and THICKNESS parameters define the distorted grid region. Only one of these parameters should be specified.

Optional Parameters

ENCROACH	Defines the abruptness of the transition between a distorted and non-distorted grid. The transition region becomes more abrupt with smaller ENCROACH factors (the minimum is 0.1).

Note: Depending on the characteristics of the undistorted grid, long, thin, or obtuse triangles may result if too low an ENCROACH value is used.

GRADING	Specifies a grid ratio which produces a non-uniform grid in the distorted region. This parameter is identical to the RATIO parameter in the X.MESH and Y.MESH statements.
GR1 and **GR2**	These may be used instead of the GRADING parameter. GR1 and GR2 may be specified in conjunction with MIDDLE (y grid line) and Y.MIDDLE (location) so that GR1 specifies grading in the spread region from UPPER to MIDDLE, and GR2 specifies grading from MIDDLE to LOWER.
MIDDLE	Specifies the y-grid line that serves as a boundary between the grading specified by GR1 and the grading specified by GR2.
VOL.RATIO	Specifies the ratio of the downward displacement of the lower grid line to the net increase in thickness. The default (0.44) should be used for oxide-silicon interfaces. VOL.RATIO is ignored if Y.LOWER is specified.
Y.MIDDLE	Specifies the physical location in the distorted grid to which the line specified by MIDDLE will be moved.

Examples

This example spreads a uniform 400 Å of oxide to 1000 Å on the left side of the device. This increases oxide thickness by 600 Å. Because VOL.RATIO is not specified, the default (0.44) is used. Therefore, 0.44 X 600 = 264 Å of the net increase will lie below the original 400 Å and 0.56×600 = 336 Å of the net increase will lie above the original 400 Å. The width of the spread region is 0.5 μm, and the oxide taper is quite gradual because of the high encroachment factor. The grid is left uniform in the spread region.

```
# *** Mesh definition ***
MESH NX=30 NY=20 RECT
X.M N=1  L=0
X.M N=30 L=5
Y.M N=1  L=-.04
Y.M N=5  L=0
Y.M N=20 L=1 R=1.4
# *** Thin oxide ***
REGION IY.H=5 OXIDE NUM=1
# *** Silicon substrate ***
REGION IY.L=5 SILICON NUM=2
# *** Spread ***
SPREAD LEFT W=0.7 UP=1 LO=4 THICK=0.1 MID=2 Y.MID=0.05
```

In the second example, the right side of the grid is distorted in order to follow a junction contour. The initial grid is assumed to be above. Y.LOWER is used so that there is no increase in the size of the device, just grid redistribution. When Y.LOWER is set to the junction, choose the ENCROACH parameter so that the lower grid line (LOWER=10) follows the junction as closely as possible. The grid is graded so that grid lines are spaced closer together as they approach the junction. Because the point specified by WIDTH lies in the middle of the transition region, it should be chosen to be slightly larger than the width of the doping box.

```
# *** Doping ***
DOPING UNIFORM N.TYPE CONC=1E15
DOPING GAUSS P.TYPE X.LEFT=1.5 X.RIGHT=2 \
PEAK= CONC=1.e19  RATIO=.75 JUNC=0.3
# *** Spread ***
SPREAD RIGHT W=0.7 UP=1 LO=4 THICK=0.3 MID=2 Y.MID=0.10
```

22.62 SYSTEM

SYSTEM allows execution of any UNIX command within an input file

Note: The **SYSTEM** statement is executed by DeckBuild and is fully documented in the DeckBuild User's Manual. To enable **SYSTEM** command, select **Category: Options** in the DeckBuild Main Control menu.

Examples

The following command will remove all files `test*.str` before a **SOLVE** statement where the `OUTF` parameter is used.

```
system \rm -rf test*.str

SOLVE  .... OUTF=test0
```

The `system` command and the UNIX commands are case sensitive.

UNIX commands may be concatenated on a single line using the semicolon (`;`) operator. For example to run a third party program that reads and writes Silvaco format files with fixed names `input.str` and `output.str`.

```
SAVE OUTF=mysave.str

system mv mysave.str input.str; source myprog.exe; mv output.str
       myrestart.str

EXTRACT INIT INF=myrestart.str
```

The UNIX re-direct symbol > is not supported by the `system` command. The UNIX `echo` and `sed` syntax can be used instead to output values or variables to a given filename. For example to save the extracted value of variable `$myvariable` to the file `myfile`.

```
system echo $"myvariable" | sed -n " w myfile"
```

22.63 THERMCONTACT

THERMCONTACT specifies the position and properties of thermal contacts. This statement must be used when lattice heating solutions are specified using Giga or Giga 3D.

Syntax

```
THERMCONTACT NUMBER=<n> <position> [EXT.TEMPER=<n>] [ALPHA=<n>]
```

Parameter	Type	Default	Units
ALPHA	Real		W/(cm2·K)
BLACKBODY	Logical	False	
BOUNDARY	Logical	True	
DEVICE	Character		
ELEC.NUMBER	Integer		
EXT.TEMPER	Real	300	K
F.CONTEMP	Character		
NAME	Character		
NUMBER	Integer	1	
STEFAN	Real	5.67051×10^{-12}	$W/(cm^{-2}K^{-4})$
STRUCTURE	Character		
TEMPERATURE	Real	300	K
X.MAX	Real	Right side of structure	µm
X.MIN	Real	Left side of structure	µm
Y.MAX	Real	Bottom of structure	mµ
Y.MIN	Real	Top of structure	µm
Z.MAX	Real	Back	µm
Z.MIN	Real	Front	µm

Description

At least one **THERMCONTACT** statement must be specified when simulating lattice heating effects (MODELS LAT.TEMP). The **THERMCONTACT** statement must appear in the input deck before any **METHOD** statement.

ALPHA	Specifies the reverse value of thermal resistance ($a=1/R_{TH}$).
BLACKBODY	Adds blackbody radiation to the conductive thermal boundary conditions.
EXT.TEMPER	Specifies the external temperature. The synonym for this parameter is **TEMPERATURE**.

F.CONTEMP	Specifies the name of a file containing a C-Interpreter function describing the contact temperature as a function of time.
NUMBER	Specifies a thermal contact number from 1 to 20. Contact numbers should be specified in increasing order. This parameters must be specified on all **THERMCONTACT** statements.
position	This is a set of the position parameters described below. Use either the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters to specify the exact position of the contact or the ELEC.NUMBER parameter to specify an electrode number that coincides with the thermal contact.
STEFAN	The Stefan-Boltzmann constant for blackbody radiations. For non-ideal emitters you set an appropriate value.

Position Parameters

BOUNDARY	Specifies which parts of the thermal contact the thermal boundary conditions are applied at. See Section 8.2.7 "Thermal Boundary Conditions" for a full description.
NAME	Specifies an electrode name that the thermal contact is coincident with. See Section 22.15 "ELECTRODE" for more information.
DEVICE	Specifies which device in MixedMode simulation the **THERMCONTACT** statement applies to. The synonym for this parameter is **STRUCTURE**.
ELEC.NUMBER	Specifies an electrode number that the thermal contact is coincident with.
STRUCTURE	This is a synonym for **DEVICE**.
X.MAX	Specifies the right edge of the contact.
X.MIN	Specifies the left edge of the contact.
Y.MAX	Specifies the bottom edge of the contact.
Y.MIN	Specifies the top edge of the contact.
Z.MAX	Specifies the location of the rear edge of the thermal contact.
Z.MIN	Specifies the location of the front edge of the thermal contact.

Coordinate Definition Example

A thermal contact is located where Y coordinate values range from 10 μm to the bottom side of the structure and X coordinate values range from the left edge of the structure to the right edge of the structure (be default). The external temperature is set to 300K and a thermal resistance of 1 is added. Thus, the temperature at y = 10 μm will be greater than 300K once lattice heating effects occur.

```
THERMCONTACT NUM=1 Y.MIN=10 EXT.TEMP=300 ALPHA=1
```

Setting Thermal and Electrical Contacts Coincident Example

The next statement line creates a thermal contact at the location of electrode #4. An external temperature of 400K is specified.

```
THERMCONTACT NUM=2 ELEC.NUM=4 EXT.TEMP=400
```

Note: Location and Parameters of thermal contacts are not stored in the Atlas solution files. Therefore, **THERMCONTACT** statements must be defined in each Atlas run involving lattice heating.

22.64 TITLE

TITLE specifies the title (up to 29 characters) that will appear in the standard output. If used, the **TITLE** command should be the first statement in the Atlas input file.

Syntax

```
TITLE <string>
```

Example

This example causes the text, `*** CMOS p-channel device ***`, to be printed at the top of all Atlas printouts and screen displays.

```
TITLE *** CMOS p-channel device ***
```

Note: **TITLE** cannot be used with the automatic Athena to Atlas interface feature of DeckBuild.

22.65 TONYPLOT

TONYPLOT starts the graphical post-processor TonyPlot

Note: The **TONYPLOT** statement is executed by DeckBuild and is fully described in the DeckBuild User's Manual.

Examples

All graphics in Atlas is performed by saving a file and loading the file into TonyPlot. The **TONYPLOT** command causes Atlas to automatically save a structure file and plot it in TonyPlot. The TonyPlot window will appear displaying the material boundaries. Use the **Plot:Display** menu to see more graphics options.

This command will display the file, `myfile.str`.

```
tonyplot -st myfile.str
```

This command will overlay the results of `myfile1.str` and `myfile2.str`.

```
tonyplot -overlay myfile1.str myfile2.str
```

Note: For documentation of the extensive features of TonyPlot for graphical display and analysis, see TonyPlot User's Manual.

22.66 TRAP

TRAP activates bulk traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
TRAP DONOR|ACCEPTOR E.LEVEL=<r> DENSITY=<r> DEGEN=<v>
<capture parameters> REGION|NUMBER
```

Parameter	Type	Default Value	Units
ACCEPTOR	Logical	False	
DEGEN.FAC	Real	undefined	
DENSITY	Real		cm^{-3}
DEVICE	Character		
DONOR	Logical	False	
E.LEVEL	Real		eV
E0N	Real		s^{-1}
E0P	Real		s^{-1}
F.E0N	Character		
F.E0P	Character		
F.DENSITY	Character		
FAST	Logical	False	
GATECONTACT	Character		
HJ.TNL.TRAP	Logical	False	
MIDGAP	Logical	False	
MATERIAL	Character		
NMP4.DET	Logical	False	
NMP4.EA	Real	1.3	eV
NMP4.EA.SD	Real	0.0	eV
NMP4.ED	Real	0.2	eV
NMP4.ED.SD	Real	0.0	eV
NMP4.EPST2S	Real	0.3	eV
NMP4.EPST2S.SD	Real	0.0	eV
NMP4.EPS1S1	Real	1.0	eV

Parameter	Type	Default Value	Units
NMP4.EPS1S2.SD	Real	0.0	eV
NMP4.EPS2S2	Real	0.3	eV
NMP4.EPS2S2.SD	Real	0.0	eV
NMP4.ET1	Real	0.0	eV
NMP4.ET1.SD	Real	0.0	eV
NMP4.ET2	Real	0.0	eV
NMP4.ET2.SD	Real	0.0	eV
NMP4.GAMMA	Real	10^{-9}	cm/V
NMP4.GAMMA.SD	Real	0.0	cm/V
NMP4.NIT	Real	0.0	cm^{-2}
NMP4.NU	Real	10^{13}	Hz
NMP4.NU.SD	Real	0.0	eV
NMP4.NUP	Real	10^{13}	Hz
NMP4.NUP.SD	Real	0.0	eV
NMP4.R1S2	Real	1.0	eV
NMP4.R1S2.SD	Real	0.0	eV
NMP4.R12S	Real	1.0	eV
NMP4.R12S.SD	Real	0.0	eV
NMP4.S1S2	Real	2.5	eV
NMP4.S1S2.SD	Real	0.0	eV
NMP4.S12S	Real	2.5	eV
NMP4.S12S.SD	Real	0.0	eV
NMP4.SAMPLES	Real	1	
NMP4.STO	Logical	False	
NMP4.WKB	Logical	True	
NUMBER	Integer	0	
REGION	Integer	0	
SIGN	Real		cm^2

Parameter	Type	Default Value	Units
SIGP	Real		cm^2
STRUCTURE	Character		
SUBCONTACT	Character		
TAT.TRAP	Logical	False	
TAUN	Real		s
TAUP	Real		s
X.MAX	Real	Maximum X coord	µm
X.MIN	Real	Minimum X coord	µm
Y.MAX	Real	Maximum Y coord	µm
Y.MIN	Real	Minimum Y coord	µm
Z.MAX	Real	Maximum X coord	µm
Z.MIN	Real	Minimum Z coord	µm

Description

ACCEPTOR	Specifies an acceptor-type trap level.
DEGEN.FAC	Specifies the degeneracy factor of the trap level used to calculate the density.
DENSITY	Sets the maximum density of states of the trap level.
DEVICE	Specifies which device the statement applies to in MixedMode simulation. The synonym for this parameter is STRUCTURE.
DONOR	Specifies a donor-type trap level.
E.LEVEL	Sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge. For donors, it depends on the valence band edge.
E0N	Specifies the trap emission rate for elections.
E0P	Specifies the trap emission rate for holes.
F.E0N	Specifies the name of a file containing a C-Interpreter function describing the trap emission rate for electrons as a function of time.
F.E0P	Specifies the name of a file containing a C-Interpreter function describing the trap emission rate for holes as a function of time.
F.DENSITY	Specifies the name of a file containing a C-Interpreter function describing the density of donor/acceptor traps as a function of position.
FAST	Specifies that the trap densities are static even during transient simulation (i.e., they reach equilibrium instantaneously).

GATECONTACT	Sets the gate material involved in charge trapping of the four-state NMP model (i.e. polysilicon or metal).
HJ.TNL.TRAP	Specifies that the TRAPS are to have a tunneling component added to their capture and emission terms when used with the ITAT.HJ.EL and ITAT.HJ.HO models for Type-2 heterojunctions.
MATERIAL	Specifies which material from the table in Appendix B "Material Systems" will apply to the **TRAP** statement. If a material is specified, then all regions defined as being composed of that material will be affected.
MIDGAP	Specifies that the energy of the discrete trap be set to the middle of the bandgap.
NMP4.DET	Enables the deterministic four-state NMP model.
NMP4.EA	Activation energy of the interface reaction.
NMP4.EA.SD	Variation parameter for NMP4.EA.
NMP4.ED	Reaction energy of the interface reaction.
NMP4.ED.SD	Variation parameter for NMP4.ED.
NMP4.EPST2S	Energy difference between state 2' and 2.
NMP4.EPST2S.SD	Variation parameter for NMP4.EPST2S.
NMP4.EPS1S1	Energy barrier from state1' to state 1.
NMP4.EPS1S2.SD	Variation parameter for NMP4.EPS1S1.
NMP4.EPS2S2	Energy difference between state 2' and 2.
NMP4.EPS2S2.SD	Variation parameter for NMP4.EPST2S.
NMP4.ET1	Trap level for charge capture/emission.
NMP4.ET1.SD	Variation parameter for NMP4.ET1.
NMP4.ET2	Trap level for the NMP transition $1 \leftrightarrow 2'$.
NMP4.ET2.SD	Variation parameter for NMP4.ET2.
NMP4.GAMMA	Coefficient for field dependent thermal activation energy.
NMP4.GAMMA.SD	Variation parameter for NMP4.GAMMA.
NMP4.NIT	Areal density of sites for the interface reaction.
NMP4.NU	Attempt frequency for the structural relaxation.
NMP4.NU.SD	Variation parameter for NMP4.NU.
NMP4.NUP	Attempt frequency for the interface reaction.
NMP4.NUP.SD	Variation parameter for NMP4.NUP.

NMP4.R1S2	Quantity which determines the NMP barrier from state 2 to state 1'.
NMP4.R1S2.SD	Variation parameter for NMP4.R1S2.
NMP4.R12S	Quantity which determines the NMP barrier from state 2' to state 1.
NMP4.R12S.SD	Variation parameter for NMP4.R12S.
NMP4.S1S2	Relaxation energy of the NMP process $1' \leftrightarrow 2$.
NMP4.S1S2.SD	Variation parameter for NMP4.S1S2.
NMP4.S12S	Relaxation energy of the NMP process $1 \leftrightarrow 2'$.
NMP4.S12S.SD	Variation parameter for NMP4.S12S.
NMP4.SAMPLES	Number of sample traps per point for stochastic models.
NMP4.STO	Enables the stochastic four-state NMP model.
NMP4.WKB	Specifies whether the transmission coefficient is calculated as a WKB factor.
NUMBER	This is the synonym for **REGION**.
REGION	Specifies which region the traps apply to. If unspecified, the traps apply to all regions.
STRUCTURE	This is a synonym for **DEVICE**.
SUBCONTACT	Sets the substrate material involved in charge trapping of the four-state NMP model (usually silicon).
TAT.TRAP	Causes the trap level to be used in the ITAT/RTAT model. The trap must be located in an insulator or wide bandgap semiconductor quantum barrier.
X.MAX	Specifies the maximum X coordinate of a box where traps are to be applied.
X.MIN	Specifies the minimum X coordinate of a box where traps are to be applied.
Y.MAX	Specifies the maximum Y coordinate of a box where traps are to be applied.
Y.MIN	Specifies the minimum Y coordinate of a box where traps are to be applied.
Z.MAX	Specifies the maximum Z coordinate of a box where traps are to be applied.
Z.MIN	Specifies the minimum Z coordinate of a box where traps are to be applied.

Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN	Specifies the capture cross section of the trap for electrons.

SIGP	Specifies the capture cross section of the trap for holes.
TAUN	Specifies the lifetime of electrons in the trap level.
TAUP	Specifies the lifetime of holes in the trap level.

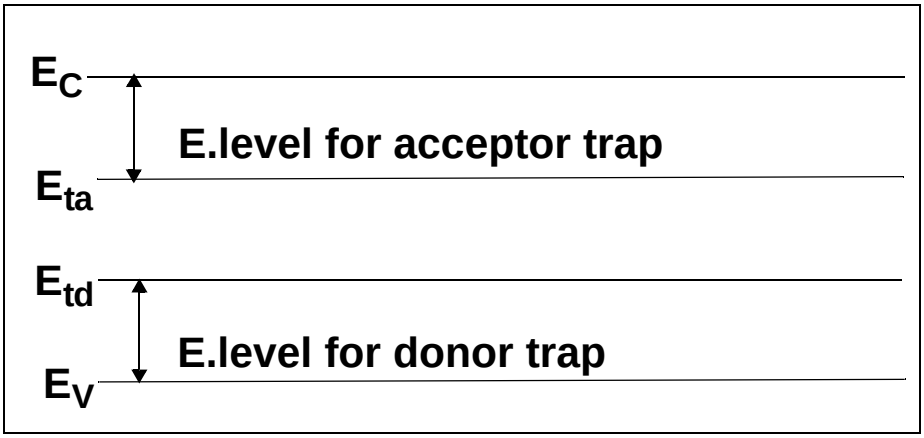


Figure 22-8: Acceptor and Donor Trap Energy Levels

Multiple Trap Level Definition Example

The following example sets three discrete trap levels within the silicon bandgap. These trap levels will capture carriers, slowing the switching speed of any device. In this example the capture cross sections are used to define the properties of each trap.

```
trap e.level=0.49 acceptor density=2.e15 degen=12 \
sign=2.84e-15 sigp=2.84e-14
trap e.level=0.41 acceptor density=1.e15 degen=12 \
sign=7.24e-16 sigp=7.24e-15
trap e.level=0.32 donor density=1.e15 degen=1 \
sign=1.00e-16 sigp=1.00e-17
```

Note: For distributed trap levels, see Section 22.10 "DEFECTS". For interface traps, see Section 22.25 "INTTRAP". Also, spatial TRAP distributions in X, Y, or Z directions must be defined in the **DOPING** statement.

22.67 UTMOST

UTMOST converts data from a Atlas logfile format into an Utmost logfile format.

Note: Utmost 12.3.4 or later can read Atlas format logfiles directly in batch mode. This statement is obsolete and its use is not recommended.

Syntax

```
UTMOST BIP|DIODE|MOS <input> OUTFILE=<fn> WIDTH=<n>
[<elec>] <cntrl>
[TRANSLATE INFILE1=<filename>][<analysis>] [<parasites>]
```

Parameter	Type	Default	Units
AC	Logical	False	
ANODE	Integer	1	
APPEND	Logical	False	
BASE	Integer		
BIPBIP	Logical	False	
BULK	Integer		
CATHODE	Integer		
COLLECTOR	Integer		
DEVICE	Integer	1	
DIODE	Logical	False	
DRAIN	Integer		
EMITTER	Integer		
GATE	Integer		
INFILE	Character		
INTERNAL	Logical	False	
LENGTH	Real	1.0	μm
MESFET	Logical	False	
MINSTEP	Real	0.01	V
MOS	Logical	False	
OUTFILE	Character		
POLARITY	Integer	1 (n-type)	
ROUTINE	Integer	1	

Parameter	Type	Default	Units
SOURCE	Integer		
TEMPERATURE	Real	300	K
WELL	Integer		
WIDTH	Real	1.0	μm

Description

Specify a technology parameter (BIP, DIODE, or MOS), an input file (INFILE1=), and an output file (OUTFILE=). You can also specify one or more of the optional electrode parameters.

INFILE	This is used to convert up to nine Atlas logfiles into a single Utmost logfile. The Atlas logfiles must be specified in the form: INFILE1=<fn> INFILE2=<fn> INFILE3=<fn> . . . where fn is the name of the logfile that you wish to convert.
OUTFILE	Specifies the name of a file where I-V data will be stored in an Utmost file format.
APPEND	Specifies that I-V data should be appended to the output file specified by OUTFILE2.
WIDTH	This is used to specify the width of the device. Electrode current is multiplied by the value of WIDTH before being saved in the logfile.

Technology Parameters

BIP	Create an Utmost biploar transistor log file.
DIODE	Creates an Utmost diode log file.
MOS	Creates an Utmost MOSFET log file.
MESFET	Creates an Utmost MESFET log file.

Electrode Parameters

Different electrode parameters may be specified with different technologies.

BIP	Technology.
BASE	Specifies the base electrode number.
COLLECTOR	Specifies the collector electrode number.
EMITTER	Specifies the emitter electrode number.
POLARITY	Indicates the device type. POLARITY=1 specifies npn. POLARITY=-1 specifies pnp.
MOS and MESFET	Technologies

Note: If you have used the NAME parameter of the **ELECTRODE** statement to assign standard electrode names (bulk, drain, gate, and source), you don't need to re-specify these names in the **UTMOST** statement.

BULK or **SUBSTRATE**	Specifies the bulk electrode number.
DRAIN	Specifies the drain electrode number.
GATE	Specifies the gate electrode number.
POLARITY	Indicates the device type. POLARITY=1 specifies nmos. POLARITY=-1 specifies pmos.
SOURCE	Specifies the source electrode number.

Control Parameters

The optional control parameters are used to specify what type of information will be converted to an Utmost logfile.

AC	Specifies that input logfiles contain AC parameters and that the Utmost routine numbers refer to the Utmost capacitance routines.
DEVICE	Specifies the device (or row) number used to identify different devices in Utmost. The synonym for this parameter is ROW.
INTERNAL	Specifies that internal contact voltage will be used instead of applied bias.
LENGTH	Specifies the length of the device.
MINSTEP	Specifies the minimum voltage step between data points.
TEMPERATURE	Specifies the simulation temperature.
ROUTINE	Specifies which Utmost routine number data will be saved for. The following routines are supported.

BIP Technology

ROUTINE=1	Specifies I_C vs. V_{CE} curves. Each input file holds I-V data for a solution with a fixed base current and V_{CE} stepped. (Utmost IC/VCE routine).
ROUTINE=9	Specifies a B_F vs. V_{BE} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BE} stepped. (Utmost BF vs. IC routine)
ROUTINE=10	Specifies a B_F vs. V_{CE} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BC} stepped. (Utmost BR routine)
ROUTINE=14	Specifies a I_C,I_B vs. V_{BE} (Gummel) plot. Each input file holds I-V data for a solution with constant V_{CE} and V_{BE} stepped. (Utmost gummel routine)
ROUTINE=15	Specifies a I_E,I_B vs. V_{BC} (Reverse Gummel) plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BC} stepped. (Utmost rgummel routine)
ROUTINE=29	Specifies I_E vs. V_{EC} plot. Each input file holds I-V data for a solution with a fixed base current and V_{EC} stepped. (Utmost IE/VEC routine)

MOS Technology

ROUTINE=1	Specifies a I_D vs. V_{DS} plot. V_B is constant and V_G is stepped. (Utmost ID/VD-VG routine)
ROUTINE=2	Specifies a I_D vs. V_{GS} plot. V_D is constant and V_B is stepped. (Utmost ID/VG-VB routine)
ROUTINE=3	Specifies the L_{EFF}, *R*SD routine. Two devices of different lengths are needed. The I_D/V_{GS} data should be used.
ROUTINE=5	Specifies a I_D vs. V_{GS} plot. V_D is constant and V_B=0. (Utmost VTH routine)
ROUTINE=9	Specifies a I_D vs. V_{GS} plot. V_D is constant and V_B is stepped. (Utmost NSUB routine).
ROUTINE=10	Specifies the IS routine. The forward diode characteristics of the drain to bulk junction are required.
ROUTINE=11	Specifies the LAMBDA routine. An I_D/V_D curve is required.
ROUTINE=12	Specifies the ETA routine. A set of I_D/V_{GS} data for different VDS is required.
ROUTINE=13	Specifies the VMAX routine. A set of I_D/V_{GS} curves for small changes in V_{DS} are required.
ROUTINE=14	Specifies the U0 routine. A set of I_D/V_{GS} data is required.
ROUTINE=26	Specifies a I_D vs. V_{GS} plot. V_B is constant and V_D is stepped. (Utmost ID/VG-VD routine).

MOS Capacitances

The following routines may be specified if both MOS and AC parameters are selected.

ROUTINE=1	Specifies the CGSO routine. C_{GS} vs. V_{GS} data is required.
ROUTINE=2	Specifies the CGDO routine. C_{GD} vs. V_{GS} data is required.
ROUTINE=5	Specifies the CJ routine. C_{BD} vs. V_{DS} data is required.
ROUTINE=6	Specifies the CJSW routine. C_{BD} vs. V_{DS} data is required.
ROUTINE=12	Specifies the CJ/CJSW routine. C_{GD} vs. V_{GS} data is required for two different size drain areas.

Diode Technology

ROUTINE=1	Specifies an I vs. V plot. V is stepped. (`UTMOST ID` vs. `VD` routine).

22.68 VCSEL

The **VCSEL** statement specifies certain VCSEL simulation parameters.

Syntax

```
VCSEL <parameters>
```

Parameter	Type	Default	Units
ABSORPTION	Logical	False	
CHECK	Logical	False	
EFINAL	Real		eV
EINIT	Real		eV
FCARRIER	Logical		
INDEX.BOTTOM	Real	1.0	
INDEX.TOP	Real	1.0	
ITMAX	Integer	30	
MAXCH	Real	2.5	
LOSSES	Real	0.0	
NEFF	Real	1.0	
NMODE	Integer	1	
NSPEC	Integer	100	
OMEGA	Real		
PERTURB	Logical	False	
PHOTON.ENERGY	Real		eV
PROJ	Logical	False	
TAUSS	Real	0.05	s
TOLER	Real	0.01	
VCSEL.CHECK	Logical	False	
VCSEL.INCIDENCE	Integer	1	

Description

ABSORPTION	Enables the absorption loss model in VCSEL.
CHECK	Enables reflectivity test simulation of the VCSEL structure
EFINAL and **EINIT**	Specify the lower and upper photon energies in reflectivity test. The default values are EINIT=0.8 PHOTON.ENERGY and EFINAL=1.2 PHOTON.ENERGY.
INDEX.BOTTOM	Specifies the refractive index of the medium below the structure. The default value is 1.0.
INDEX.TOP	Specifies the refractive index of the medium above the structure. The default value is 1.0.
ITMAX	Sets maximum number of external VCSEL iterations during photon density calculation. The default value is 30.
MAXCH	Specifies the maximum allowed relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.
LOSSES	Specifies additional losses in Chapter 10 "VCSEL Simulator", Equation 10-16.
NEFF	Specifies the effective refractive index in Chapter 10 "VCSEL Simulator", Equation 10-11.
NMODE	Specifies the number of transverse modes of interest. The default value is NMODE=1.
NSPEC	Specifies the number of sampling points between EINIT and EFINAL in reflectivity test. The default value is NSPEC=100.
OMEGA	Specifies the reference frequency (see Equation 10-2 in Chapter 10 "VCSEL Simulator"). The reference frequency can change after the reflectivity test to correspond to the resonant frequency of the structure.
PERTURB	Enables faster solution of the Helmholtz equation. When it is specified, the program solves the longitudinal wave equation only once. The perturbational approach is used to account for the refractive index changes.
PHOTON.ENERGY	Specifies the reference photon energy. The reference photon energy can change after reflectivity test to correspond to the resonant photon energy of the structure. This parameter is related to OMEGA parameter.
PROJ	Enables faster solution of the photon rate equations below threshold. You should disable this solution scheme when the bias reaches lasing threshold. To do this, specify ^PROJ in a **VCSEL** statement.
TAUSS	This is an iteration parameter used in the calculation of photon densities. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.

TOLER	Sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.
VCSEL.CHECK	This is the same as **CHECK**.
VCSEL.INCIDENCE	Specifies the direction of light incident on the structure. VCSEL.INCIDENCE=1 is the light incident from the top. VCSEL.INCIDENCE=0 is the light incident from the bottom. VCSEL.INCIDENCE=2 or >2 means both directions of light incidence are considered. By default, light is incident from the top of the structure.

22.69 WAVEFORM

WAVEFORM defines transient waveforms, which can then be enabled by specifying WAVE1 .. WAVE10 on the **SOLVE** statement.

Syntax

```
WAVEFORM [<parameters>]
```

Parameter	Type	Default	Units
AMPLITUDE	Real	0	
BEAM	Real		
CONTINUE	Logical	True	
DECAY	Real	0	
ELEC.NAME	Character		
ELEC.NUMBER	Real	0	
F.WAVE	Character		
FREQUENCY	Real	0	Hz
HIGH.BIAS	Real	0	
LOW.BIAS	Real	0	
NUMBER	Real	0	
OFFSET	Logical	False	
PERIODS.NUMBER	Real		
PHASE	Real	0	Deg
PRBS	Logical	False	
PULSE	Logical	False	
PWL	Character		
PWLFILE	Character		
SINUSOID	Logical	False	
TABLE	Character		
TDELAY	Real	0	S
TFALL	Real	0	S
TPERIOD	Real	0	S

Parameter	Type	Default	Units
TRISE	Real	0	S
WIDTH	Real	0	S
WORD	Real	0	S

Description

AMPLITUDE	Specifies the sinusoidal amplitude.
BEAM	Specifies the optical BEAM number to which the WAVEFORM parameters will apply.
CONTINUE	Specifies that the waveform will continue (i.e. retain the phase) across mulitple SOLVE statements. This is the default.
DECAY	Specifies an optional decay value.
ELEC.NAME	Specifies the electrode name.
ELEC.NUMBER	Specifies the electrode number (either ELEC.NAME or ELEC.NUMBER must be specified).
F.WAVE	Specifies the C-Interpreter file. The function prototype is: `int wave(double time, double dt, double *bias)` where time is the transient time, dt is the time step and bias is the bias calculated by the C-Interpreter function.
FREQUENCY	Specifies the frequency.
HIGH.BIAS	Specifies the high (pulsed) bias for PRBS and PULSE waveforms.
LOW.BIAS	Specifies the low (initial) bias for PRBS and PULSE waveforms.
NUMBER	Specifies the waveform number. This must be specified on each WAVEFORM statement.
OFFSET	Specifies that a phase offset is to be used. If you use a non-zero phase in a sinusoidal definition, then SOLVE statement will go immediately to this bias. For example, if PHASE=90, AMPLITUDE=1 and the DC bias is 1V, the transient SOLVE statement will start at 2V and vary from 2V to 0V. If you specify OFFSET, then the bias will vary from 1V to -1V.
PERIODS.NUMBER	Specifies the maximum number of sinusoidal periods (default is unlimited).
PHASE	Specifies the phase.
SPRBS	Specifies that a pseudo-random bit sequence will be generated.
PULSE	Specifies that a pulse will be generated.

PWL	Specifies a piecewise linear list of time and bias value points. The time values must be in ascending order (e.g., PWL="1e-9,1 2e-9, 2 3e-9, 3" specifies that the bias with change to 1V at 1ns, 2V at 2ns, and 3V at 3ns). The PWL list must not be more than 132 characters long. Longer PWL list should be defined in a file and **PWLFILE** used instead.
PWLFILE	Specifies a file containing time and bias piecewise linear data. The time values must be in ascending order

SINUSOID	Specifies that a sinusoidal waveform will be generated according to:
	`AMPLITUDE*exp(-time*DECAY)`
	`*sin(2*PI*(FREQUENCY*time+PHASE/360.0))`

TABLE	Specifies a file containing time and bias tabular data.
TDELAY	Specifies the time delay for PRBS and PULSE waveforms.
TDELAY	Specifies the fall time for PRBS and PULSE waveforms.
TPERIOD	Specifies the period time for PRBS and PULSE waveforms.
TRISE	Specifies the rise time for PRBS and PULSE waveforms.
WIDTH	Specifies the pulse width for PRBS and PULSE waveforms.
WORD	Specifies the word length for PRBS waveforms.

22.70 WAVEGUIDE

WAVEGUIDE defines geometry and physical models for a stand-alone optical mode simulation, which is done by solving the Helmholtz equation. For information about **WAVEGUIDE**, see Section 9.4 "WAVEGUIDE Statement".

Syntax

```
WAVEGUIDE <parameters>
```

Parameter	Type	Default	Units
DISP.NAME	Character	dispersion.log	
DISPERSION	Logical	False	
E.INIT	Real	-999	eV
E.FINAL	Real	-999	eV
FAR.NX	Integer	100	
FAR.NY	Integer	100	
LX.MAX	Real	999	μm
LX.MIN	Real	-999	μm
LY.MAX	Real	999	μm
LY.MIN	Real	-999	μm
NDISP	Integer	10	
NMODE	Integer	1	
NB.ITEREIG	Integer	4	
NEAR.NX	Integer	100	
NEAR.NY	Integer	100	
OMEGA	Real	-999	1/s
OMEGA.FINAL	Real	-999	1/s
OMEGA.INIT	Real	-999	1/s
PHOTON.ENERGY	Real	-999	eV
PRT.ITEREIG	Logical	False	
REFLECT	Logical	False	
V.HELM	Logical	False	

Description

DISP.NAME	Specifies the log file name for waveguide dispersion.
E.INIT	Sets the lower photon energy boundary for dispersion calculation.
E.FINAL	Sets the upper photon energy boundary for dispersion calculation.
FAR.NX	Sets the number of X-samples for far-field pattern.
FAR.NY	Sets the number of Y-samples for far-field pattern.
LX.MAX	Sets the upper X boundary of solution domain for vector Helmholtz solver.
LX.MIN	Sets the lower X boundary of solution domain for vector Helmholtz solver.
LY.MAX	Sets the upper Y boundary of solution domain for vector Helmholtz solver.
LY.MIN	Sets the lower Y boundary of solution domain for vector Helmholtz solver.
NB.ITEREIG	Sets the ratio of basis size of iterative Helmholtz eigensolver to the number of requested transverse modes, given by the NMODES parameter.
NDISP	Sets the number of photon energy samples for dispersion calculation.
NEAR.NX	Sets the number of X-samples for near-field pattern.
NEAR.NY	Sets the number of Y-samples for near-field pattern.
NMODE	Sets the number of requested optical modes in vector Helmholtz solver.
OMEGA	Sets the photon frequency in vector Helmholtz solver.
OMEGA.INIT	Sets the lower frequency boundary for dispersion calculation.
OMEGA.FINAL	Sets the upper frequency boundary for dispersion calculation.
PHOTON.ENERGY	Sets the photon energy in vector Helmholtz solver.
PRT.ITEREIG	Specifies that iterative Helmholtz eigensolver prints information about its internal parameters and convergence.
REFLECT	Specifies that the structure is mirror-symmetrical around x=0 axis.
V.HELM	Specifies that vector Helmholtz solver will be used for solution for transverse optical modes.

Appendix A
C-Interpreter Functions

A.1 Overview

Atlas has a built-in C language interpreter that allows many of the models contained in Atlas to be modified. To use the Silvaco C-Interpreter, (SCI), you must write C language functions containing analytic descriptions of the model. The C-Interpreter uses the ANSI standard definition of C. If you are not familiar with the C language, we suggest that you refer to any of the popular C language references such as the one written by Kernighan and Ritchie [152].

The function arguments of the C-Interpreter functions and return values are fixed in Atlas. Make sure that the arguments and the return values for the functions match those expected by Atlas. To help you, this release of Atlas includes a set of templates for the available functions. The C-Interpreter function template can be obtained by typing:

```
atlas -T filename
```

where filename is the name of a file where you want the template to be copied. You can also get the C-Interpreter function template by accessing the template file through DeckBuild.

This template file should be copied and edited when implementing user-defined C-Interpreter functions. To use the C-Interpreter function, specify the file name containing this function in the appropriate Atlas statement. The relevant statement names and parameters are listed in the template file. The template function prototypes are defined in the include file template.in.

The following example shows how the C-Interpreter function munsat can be used to describe velocity saturation for electrons. First, examine the template file and find the template for munsat. The template should look something like this:

```
/*
 * Electron velocity saturation model.
 * Statement: MATERIAL
 * Parameter: F.MUNSAT
 * Arguments:
 * tl    lattice temperature (K)
 * na    acceptor concentration (per cc)
 * nd    donor concentration (per cc)
 * e     electric field (V/cm)
 * v     saturation velocity (cm/s)
 * mu0   low field mobility (cm^2/Vs)
 * *mu   return: field dependent mobility (cm^2/Vs)
 * *dmde return: derivative of mu with e
 */
int munsat(double tl, double na, double nd, double e,
           double v, double mu0, double *mu, double *dmde)
{
    return(0);               /* 0 - ok */
}
```

Here, we see that the function is passed the lattice temperature, the acceptor and donor concentrations, the electric field, the saturation velocity, and the low field mobility. The function returns the field dependent mobility and the derivative of mobility with respect to electric field.

Note: It is important to properly calculate and return derivative values when specified, since the convergence algorithms in Atlas require these derivatives.

The return value is an error flag. In this case, it is set to zero indicating that the function was successful (`0`=OK, `1`=fail).

In this example, the C-Interpreter implements a standard model for velocity saturation. This particular model is already built into Atlas. If you want to compare the user-defined version with the standard version, you can activate the built-in model by setting the `B.ELECTRONS` parameter to 1.

To implement the model, two lines additional lines of C code in the body of the function are specified as follows:

```
/*
 * Electron velocity saturation model.
 * Statement: MATERIAL
 * Parameter: F.MUNSAT
 * Arguments:
 * tl    lattice temperature (K)
 * na    acceptor concentration (per cc)
 * nd    donor concentration (per cc)
 * e     electric field (V/cm)
 * v     saturation velocity (cm/s)
 * mu0   low field mobility (cm^2/Vs)
 * *mu   return: field dependent mobility (cm^2/Vs)
 * *dmde return: derivative of mu with e
 */


int munsat(double tl, double na, double nd, double e,
           double v, double mu0, double *mu, double *dmde)
{
  *mu  = mu0/(1.0+mu0*e/v);
  *dmde = (*mu)*(*mu)/v;
  return(0);                /* 0 - ok */
}
```

The function will then need to be stored in a file. For example, the function can be stored in the file: `test.lib`. The function is then introduced into a specific example by specifying the file name on the `F.MUNSAT` parameter in the **`MATERIAL`** statement as follows:

```
MATERIAL F.MUNSAT="test.lib"
```

In this case, you will also need to activate the calculation of the field dependent mobility by adding the `FLDMOB` parameter to a **`MODELS`** statement. For example:

```
MODELS FLDMOB
```

When the input deck is executed, your C-Interpreter functions will be used in place of the built in function. When trying this example, place `PRINT` statements (using the `printf C` command) in the function to check that the function is working correctly.

Table A-1 shows a complete list of all the interpreter functions available in Atlas.

Table A-1 Complete list of available C-Interpreter functions in Atlas

Statement	Parameter	Template	Description
`BEAM`	`F.IMAGE`	`image ()`	Specifies 2D relative intensity versus offset at the source perpendicular to the direction of propogation for ray tracing in Luminous 3D.
`BEAM`	`F.INTENSITY`	`intensity ()`	Optical intensity as a function of position and wavelength.
`BEAM`	`F.RADIATE`	`radiate ()`	Generation rate as a function of position.
`BEAM`	`F3.RADIATE`	`radiate3 ()`	Generation rate as a function of position (3D).
`BEAM`	`F.REFLECT`	`reflect ()`	Reflection coefficient.
`CONTACT`	`F.ETUNNEL`	`fetunnel ()`	Schottky contact electron tunneling.
`CONTACT`	`F.HTUNNEL`	`fhtunnel ()`	Schottky contact hole tunneling.
`CONTACT`	`F.WORKF`	`workf ()`	Electrical Field dependent workfunction.
`DEFECTS`	`F.DEFECTS`	`defects ()`	**`DEFECTS`** statement parameters as a function of position.
`DEFECTS`	`F.TFTACC`	`tftacc ()`	TFT acceptor trap density as a function of energy.

Table A-1 Complete list of available C-Interpreter functions in Atlas

Statement	Parameter	Template	Description
DEFECTS	F.TFTDON	tftdon ()	TFT donor trap density as a function of energy.
DEGRADATION	F.NTA	devdeg_nta ()	Interface acceptor trap density as a function of location.
DEGRADATION	F.NTD	devdeg_ntd ()	Interface donor trap density as a function of location.
DEGRADATION	F.SIGMAE	devdeg_sigmae ()	Interface electron capture cross-section.
DEGRADATION	F.SIGMAH	devdeg_sigmah ()	Interface hole capture cross-section.
DOPING	F.COMPOSIT	composition ()	Composition dependent composition fractions.
DOPING	F.DOPING	doping ()	Composition dependent net doping.
DOPING	F3.DOPING	doping3 ()	Composition dependent net doping.
IMPACT	F.IIAN	iian()	Electron impact ionization coefficient as a function of composition, temperature, and electric field.
IMPACT	F.IIAP	iiap()	Hole impact ionization coefficient as a function of composition, temperature, and electric field.
INTDEFECTS	F.INTDEFECTS	int defects ()	**INTDEFECTS** statement parameters as a function of position.
INTDEFECTS	F.TFTALL	int tftacc ()	TFT interface acceptor density as a function of energy.
INTDEFECTS	F.TFTDON	int tftdonor ()	TFT interface donor density as a function of energy.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
INTERFACE	F.QF	int_fixed_charge ()	Interface fixed charge as a function of position.
INTERFACE	F.HTE.N	hten()	Electron heterojunction thermionic emission current.
INTERFACE	F.HTE.P	htep()	Hole heterojunction thermionic emission current.
INTERFACE	F.HTFE.N	htfen()	Electron heterojunction thermionic field emission delta term.
INTERFACE	F.HTFE.P	htfep()	Hole heterojunction thermionic field emission delta term.
INTTRAP	F.DENSITY	int_acc_trap ()	Interface ACCEPTOR trap density as a function of position.
INTTRAP	F.DENSITY	int_don_trap ()	Interface DONOR trap density as a function of position.
INTTRAP	F.F0N	e0n_trap	Trap emission rate for electrons as a function of time.
INTTRAP	F.F0P	e0p_trap	Trap emission rate for holes as a function of time.
INTTRAP	F.MSCRATES	msctransitionrates()	Enables you to specify the transition rates between all internal states of a multistate trap.
IMPACT	F.EDIIN	ediin ()	Electron temperature coefficients for impact ionization.
IMPACT	F.EDIIP	ediip ()	Hole temperature coefficients for impact ionization.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
LASER	F.LGAIN	lgain()	Laser gain as a function of position, composition, carrier concentration, quasi-Fermi level and current.
LASER	F.MIRROR	mirror	Mirror facet reflectivities on a function of wavelength.
LASER	F.OMODE	omode()	Laser transverse optical mode distribution as a function of position.
LENS	F.LENS	lenslet()	Lenslet surface as a function of position.
MATERIAL	F.BANDCOMP	bandcomp ()	Temperature and composition dependent band parameters.
MATERIAL	F.BBT	bbt ()	Electric field and carrier concentration dependent band-to-band generation rate.
MATERIAL	F.CAPT.MUN	qwcaptmun()	Capture-escape model inplane electron mobility
MATERIAL	F.CAPT.MUP	qwcaptmup()	Capture-escape model inplane hole mobility
MATERIAL	F.BGN	bgn ()	Composition, temperature, and doping dependent band gap narrowing.
MATERIAL	F.CBDOSFN	cbdosfn	Electron temperature dependent effective conduction band density of states.
MATERIAL	F.CONMUN	conmun ()	Composition, temperature, position, and doping dependent electron mobility.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.CONMUP	conmup ()	Composition, temperature, position, and doping dependent hole mobility.
MATERIAL	F.COPT	copt ()	Radiative recombination rate as a function of composition and temperature.
MATERIAL	F.EPSILON	epsilon ()	Composition and temperature dependent permittivity.
MATERIAL	F.FERRO	ferro ()	Field dependent permittivity.
MATERIAL	F.GAUN	gaun ()	Electron Auguer rate as a function of composition and temperature.
MATERIAL	F.GAUP	gaup ()	Hole Auger rate as a function of composition and temperature.
MATERIAL	F.INDEX	index ()	Wavelength dependent complex index of refraction.
MATERIAL	F.MNSNDIFF	mnsndiff ()	Microscopic noise source for electron diffusion noise.
MATERIAL	F.MNSPDIFF	mnspdiff ()	Microscopic noise source for hole diffusion noise.
MATERIAL	F.MNSNFLICKER	mnsnflicker ()	Microscopic noise source for electron flicker noise.
MATERIAL	F.MNSPFLICKER	mnspflicker ()	Microscopic noise source for hole flicker noise.
MATERIAL	F.MUNSAT	munsat ()	Electron velocity saturation model.
MATERIAL	F.MUPSAT	mupsat ()	Hole velocity saturation model.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.NKEG	nkeg ()	Specifies the complext index of refraction as a function of band gap.
MATERIAL	F.PDN	fphonondragn()	Adds a phonon drag term to the electron thermopower in Giga. Function of lattice temperature and electron density.
MATERIAL	F.PDP	fphonondragp()	Adds a phonon drag term to the hole thermopower in Giga. Function of lattice temperature and hole density.
MATERIAL	F.RECOMB	recomb ()	Position, temperature, and concentration dependent recombination.
MATERIAL	F.SRH	srh()	This is intended for modeling SRH-like recombination rates, where the lifetimes depend on doping density and temperature. This can also be used for more general recombination rates.
MATERIAL	F.TAURP	taurp ()	Energy dependent hole relaxation time.
MATERIAL	F.TAUN	taun ()	Position, temperature, and doping dependent electron SRH lifetime.
MATERIAL	F.TAUP	taup ()	Position, temperature, and doping dependent hole SRH lifetime.
MATERIAL	F.TAURN	taurn ()	Energy dependent electron relaxation time.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.TCAP	TCAP()	Lattice Thermal Capacity as a function of lattice temperature, position, doping, and fraction composition.
MATERIAL	F.TCOND	TCOND()	Lattice Thermal Conductivity as a function of lattice temperature, position, doping, and fraction composition.
MATERIAL	F.VSATN	vsatn ()	Composition and temperature dependent electron saturation velocity.
MATERIAL	F.VSATP	vsatp ()	Composition and temperature dependent hole saturation velocity.
MATERIAL	F.VBDOSFN	vbdosfn	Hole temperature dependent effective valence band density of states.
MATERIAL/MOBILITY	F.ENMUN	endepmun	Electron mobility as a function of electron temperature and a few other choice variables.
MATERIAL/MOBILITY	F.ENMUP	endepmup	Hole mobility as a function of hole temperature and a few other choice variables.
MATERIAL/MOBILITY	F.LOCALMUN	localmun	Electron mobility with an explicit position dependence.
MATERIAL/MOBILITY	F.LOCALMUP	localmup	Hole mobility with an explicit position dependence.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
MATERIAL/MOBILITY	F.TOFIMUN	tofimun	Electron mobility with a dependence on both parallel and perpendicular electric field.
MATERIAL/MOBILITY	F.TOFIMUP	tofimup	Hole mobility with a dependence on both parallel and perpendicular electric field.
MODELS	F.ALPHAA	bulk_absorb()	Bulk absorption coefficient versus carrier density and photon energy.
MODELS	F.BFIELD	magneticfield()	Position dependent magnetic field vector (BX,BY,BZ).
MODELS	F.GENSINGLET	gensinglet()	Independent generation of singlet excitons.
MODELS	F.GENTRIPLET	gentriplet()	Independent generation of triplet excitons.
MODELS	F.KEXCITON	kexciton()	Position dependent singlet and triplet continuity equation coefficients.
MODELS	F.KSINGLET	ksinglet()	Position dependent singlet continuity equation coefficients.
MODELS	F.KSN	fksn()	Energy dependent electron Scattering Law Exponent.
MODELS	F.KSP	fksp()	Energy dependent hole Scattering Law Exponent.
MODELS	F.KTRIPLET	ktriplet()	Position dependent triplet continuity equation coefficients.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
MODELS	F.PCM	pcm()	Phase change material resistivity as a function of time step, temperature, previous temperature, and previous resistivity.
MODELS	F.PDN	fphonondragn()	Adds a phonon drag term to the electron thermopower in Giga. Function of lattice temperature and electron density.
MODELS	F.PDP	fphonondragp()	Adds a phonon drag term to the hole thermopower in Giga. Function of lattice temperature and hole density.
MODELS	F.TRAPTUNNEL	ftraptunnel	Energy and electric field dependent Dirac enhancement factor.
MODELS	F.TRAP.COULOMBIC	ftrapcoulombic	Energy and electric field dependent Coulombic enhancement factor.
ODEFECTS	F.ODEFECTS	odefects()	**ODEFECTS** statement parameter as a function of position.
OINTDEFECTS	F.OINTDEFECTS	ointdefects()	**OINTDEFECTS** statement parameter as a function of position.
SINGLEEVENTUPSET	F.SEU	seu()	Time and position dependent SEU generation.
TRAP	F.DENSITY	acc_trap()	ACCEPTOR trap density as a function of position.
TRAP	F.DENSITY	don_trap()	DONOR trap density as a function of position.
TRAP	F.F0N	e0n_trap	Trap emission rate for electrons as a function of time.

Table A-1 Complete list of available C-Interpreter functions in Atlas			
Statement	**Parameter**	**Template**	**Description**
TRAP	F.F0P	e0p_trap	Trap emission rate for holes as a function of time.

C-Interpreter functions have fixed inputs and outputs. However, certain C-Interpreter functions allow extra input arguments to be obtained by calling the functions CIgetNodalDouble() and CIgetNodalVoid(). If these functions is called within one of the supported functions, then extra information about the current node point is returned. The function prototype for CIgetNodalDouble is

```
double CIgetNodalDouble(CInodalType type);
```

where CInodalType is an enum. The values of CInodalType are described in Table A-2.

Table A-2 Arguments for the CIgetNodalDouble Function		
CInodalType Type	**Description**	**Units**
CI_NODAL_ACCEPTOR_DOPING	Acceptor doping concentration	cm^{-3}
CI_NODAL_ACTIVE_HYDROGEN_CONCENTRATION	Active hydrogen concentration	cm^{-3}
CI_NODAL_ACTIVE_OXYGEN_CONCENTRATION	Active oxygen concentration	cm^{-3}
CI_NODAL_DONOR_DOPING	Donor doping concentration	cm^{-3}
CI_NODAL_ELECTRON_CONCENTRATION	Electron concentration	cm^{-3}
CI_NODAL_ELECTRON_TEMPERATURE	Electron temperature	K
CI_NODAL_HOLE_CONCENTRATION	Hole concentration	cm^{-3}
CI_NODAL_HOLE_TEMPERATURE	Hole temperature	K
CI_NODAL_HYDROGEN_CONCENTRATION	Hydrogen concentration	cm^{-3}
CI_NODAL_INTERFACE_CHARGE	Interface charge	cm^{-2}
CI_NODAL_INTERFACE_QF	Fixed oxide interface charge	cm^{-2}
CI_NODAL_LATTICE_TEMPERATURE	Lattice temperature	K
CI_NODAL_INTERFACE_CHARGE	Total interface charge	cm^{-2}
CI_NODAL_MAGNITUDE_EFIELD	Electric field magnitude	$V\ cm^{-1}$
CI_NODAL_MAGNITUDE_ELECTRON_CURRENT	Electron current magnitude	A
CI_NODAL_MAGNITUDE_HOLE_CURRENT	Hole current magnitude	A
CI_NODAL_NET_DOPING	Net doping concentration	cm^{-3}
CI_NODAL_OPTICAL_INTENSITY	Optical intensity	$W\ cm^{-2}$

Table A-2 Arguments for the CIgetNodalDouble Function		
CInodalType Type	**Description**	**Units**
CI_NODAL_OXYGEN_CONCENTRATION	Oxygen concentration	cm^{-3}
CI_NODAL_PHOTOGENERATION_RATE	Photogeneration rate	cm^{-3} s
CI_NODAL_POTENTIAL	Potential	V
CI_NODAL_RECOMBINATION_RATE	Recombination rate	$cm^{-3}s^{-1}$
CI_NODAL_SINGLET_CONCENTRATION	Singlet exciton concentration	cm^{-3}
CI_NODAL_SPECIES_1_CONCENTRATION	Species 1 concentration	cm^{-3}
CI_NODAL_SPECIES_2_CONCENTRATION	Species 2 concentration	cm^{-3}
CI_NODAL_SPECIES_3_CONCENTRATION	Species 3 concentration	cm^{-3}
CI_NODAL_TOTAL_DOPING	Total doping concentration	cm^{-3}
CI_NODAL_TRIPLET_CONCENTRATION	Triplet exciton concentration	cm^{-3}
CI_NODAL_X_EFIELD	Electric field X-component	V cm^{-1}
CI_NODAL_X_COMPOSITION	X composition fraction	
CI_NODAL_X_ELECTRON_CURRENT	Electron current X-component	A
CI_NODAL_X_HOLE_CURRENT	Hole current X-component	A
CI_NODAL_X_POSITION	X-location	um
CI_NODAL_Y_COMPOSITION	Y composition fraction	
CI_NODAL_Y_EFIELD	Electric field Y-component	V cm^{-1}
CI_NODAL_Y_ELECTRON_CURRENT	Electron current Y-component	A
CI_NODAL_Y_HOLE_CURRENT	Hole current Y-component	A
CI_NODAL_Y_POSITION	Y-location	um
CI_NODAL_Z_EFIELD	Electric field Z-component	V cm^{-1}
CI_NODAL_Z_ELECTRON_CURRENT	Electron current Z-component	A
CI_NODAL_Z_HOLE_CURRENT	Hole current Z-component	A
CI_NODAL_Z_POSITION	Z-location	um

The supported C-Interpreter functions are `defects()`, `doping()`, `doping3()`, `e0n()`, `e0p()`, `intdefects()`, `intensity()`, `odefects()`, `ointdefects()`, `qf()`, `radiate()`, `tftacc()`, and `tftdon()`.

`CIgetNodalVoid` returns a void pointer to a C struct when it can be cast into the required data type. The function prototype for `CIgetNodalVoid` is

```
void *CIgetNodalVoid(CInodalVoidType type);
```

where CInodalVoidType is an enum. The values of CInodalVoidType are described in Table A-3.

Table A-3 Arguments for the CIgetNodalDouble Function

CInodalVoid Type	Description	Struct Type
CI_NODAL_VOID_DEFECTS	**DEFECTS** statement parameters.	CIdefectsStruct
CI_NODAL_VOID_INTDEFECTS	**INTDEFECTS** statement parameters.	CIdefectsStruct
CI_NODAL_VOID_INTERFACE	**INTERFACE** statement parameters.	CIinterfaceStruct

CIdefectsStruct is defined as:

```
typedef struct CIdefectsStruct
{
    int numa;                   /* number of acceptor levels */
    int numd;                   /* number of donor levels */

    double nta;                 /* acceptor tail density
                                   (cm^-3 eV^-1) */
    double wta;                 /* acceptor tail
                                   characteristic decay (eV) */
    double nga;                 /* acceptor bump (Gaussian)
                                   density (cm^-3 eV^-1) */
    double wga;                 /* acceptor bump characteristic
                                   decay (eV) */
    double ega;                 /* acceptor bump distribution
                                   peak energy (eV) */
    double ntd;                 /* donor tail density
                                   (cm^-3 eV^-1) */
    double wtd;                 /* donor tail characteristic
                                   decay (eV) */
    double ngd;                  /* donor bump (Gaussian) density
                                   (cm^-3 eV^-1) */
    double wgd;                  /* donor bump characteristic
                                    decay (eV) */
    double egd;                  /* donor bump distribution peak
                                    energy (eV) */
    double sigtae;               /* acceptor tail electron
                                    capture cross section
                                    (cm^2) */
    double sigtah;                /* acceptor tail hole
                                     capture-cross section
                                     (cm^2) */
```

```
    double siggae;                  /* acceptor bump electron
                                       capture-cross section
                                       (cm^2) */

    double siggah;                  /* acceptor bump hole
                                       capture-cross section
                                       (cm^2) */

    double sigtde;                  /* donor tail electron
                                       capture-cross section
                                       (cm^2) */

    double sigtdh;                  /* donor tail hole
                                       capture-cross section
                                       (cm^2) */

    double siggde;                  /* donor bump electron
                                       capture-cross section
                                       (cm^2) */

    double siggdh;                  /* donor bump hole
                                       capture-cross section
                                       (cm^2) */

CItrapStruct *acceptorTailStates;   /* acceptor tail states
                                       (size numa) */

 double acceptorTailDos;            /* total acceptor tail DOS
                                       (cm^-3) */

 double acceptorTailIonizedDensity; /* total acceptor tail ionized
                                       density (cm^-3) */

CItrapStruct *acceptorBumpStates;   /* acceptor bump states
                                       (size numa) */

 double acceptorBumpDos;            /* total acceptor bump DOS
                                       (cm^-3) */

 double acceptorBumpIonizedDensity; /* total acceptor bump ionized
                                       density (cm^-3) */

 double acceptorDos;                /* total acceptor DOS
                                       (cm^-3) */

 double acceptorIonizedDensity;     /* total acceptor ionized
                                       density (cm^-3) */

 CItrapStruct *donorTailStates;     /* donor tail states
                                       (size numd) */

 double donorTailDos;               /* total donor tail DOS
                                       (cm^-3) */

 double donorTailIonizedDensity;    /* total donor tail ionized
                                       density (cm^-3) */

 CItrapStruct *donorBumpStates;     /* donor bump states
                                       (size numd) */

double donorBumpDos;               /* total donor bump DOS
                                       (cm^-3) */

double donorBumpIonizedDensity;    /* total donor bump ionized
                                       density (cm^-3) */
```

```
   double donorDos;                  /* total donor DOS (cm^-3) */
   double donorIonizedDensity;       /* total donor ionized density
                                        (cm^-3) */

   } CIdefectsStruct;
```

CItrapStruct is defined as:

```
   typedef struct CItrapStruct
   {
       double energy;                 /* trap energy (eV) */
       double dos;                    /* trap DOS (cm^-3) */
       double ionizedDensity;         /* trap ionized density
                                         (cm^-3) */

   } CItrapStruct;
```

CIinterfaceStruct is defined as:

```
   typedef struct CIinterfaceStruct
   {
       double charge;          /* total interface charge (cm^-2) */
       double qf;              /* interface charge (cm^-2) */
       double sn;              /* electron surface recombination
                                  velocity (cm s^-1) */
      double sp;               /* hole surface recombination velocity
                                  (cm s^-1) */

   } CIinterfaceStruct;
```

CIgetStructureDouble returns the nodal data from previously saved Atlas structure files. The specified structure file must have the same mesh as the current device. The function prototype for CIgetStructureDouble is

```
   double CIgetStructureDouble(CIstructureType type, char *filename);
```

where CIstructureType is an enum and filename is the name of a structure file.

Table A-4 Arguments for the CIgetSturctureDouble Function

CIsturctureType Type	Description	Units
CI_STRUCTURE_ELECTRON_CONCENTRATION	Electron concentration	cm^{-3}
CI_STRUCTURE_ELECTRON_TEMPERATURE	Electron temperature	K
CI_STRUCTURE_HOLE_CONCENTRATION	Hole concentration	cm^{-3}
CI_STRUCTURE_HOLE_TEMPERATURE	Hole temperature	K

Table A-4 Arguments for the CIgetSturctureDouble Function

CIsturctureType Type	Description	Units
CI_STRUCTURE_INTERFACE_CHARGE	Total interface charge	cm^{-2}
CI_STRUCTURE_LATTICE_TEMPERATURE	Lattice temperature	K
CI_STRUCTURE_MAGNITUDE_EFIELD	Electric field magnitude	V cm^{-1}
CI_STRUCTURE_MAGNITUDE_ELECTRON_CURRENT	Electron current magnitude	A
CI_STRUCTURE_MAGNITUDE_HOLE_CURRENT	Hole current magnitude	A
CI_STRUCTURE_OPTICAL_INTENSITY	Optical intensity	W cm^{-2}
CI_STRUCTURE_PHOTOGENERATION_RATE	Photogeneration rate	cm^{-3} s
CI_STRUCTURE_POTENTIAL	Potential	V
CI_STRUCTURE_RECOMBINATION_RATE	Recombination rate	cm^{-3} s
CI_STRUCTURE_SINGLET_CONCENTRATION	Singlet concentration	cm^{-3}
CI_STRUCTURE_TRIPLET_CONCENTRATION	Triplet concentration	cm^{-3}
CI_STRUCTURE_X_EFIELD	Electric field X-component	V cm^{-1}
CI_STRUCTURE_X_ELECTRON_CURRENT	Electron current X-component	A
CI_STRUCTURE_X_HOLE_CURRENT	Hole current X-component	A
CI_STRUCTURE_Y_EFIELD	Electric field Y-component	V cm^{-1}
CI_STRUCTURE_Y_ELECTRON_CURRENT	Electron current Y-component	A
CI_STRUCTURE_Y_HOLE_CURRENT	Hole current Y-component	A
CI_STRUCTURE_Z_EFIELD	Electric field Z-component	V cm^{-1}
CI_STRUCTURE_Z_ELECTRON_CURRENT	Electron current Z-component	A
CI_STRUCTURE_Z_HOLE_CURRENT	Hole current Z-component	A

Appendix B
Material Systems

B.1 Overview

Atlas understands a library of materials for reference to material properties and models of various regions in the semiconductor device. These materials are chosen to represent those most commonly used by semiconductor physicists today. Blaze or Device 3D users will have access to all of these materials. S-Pisces users will have only access to Silicon and Polysilicon.

S-Pisces is designed to maintain backward compatibility with the standalone program, SPisces2 Version 5.2. In the SPisces2 syntax, certain materials can be used in the **REGION** statement just by using their name as logical parameters. This syntax is still supported.

B.2 Semiconductors, Insulators, and Conductors

All materials in Atlas are strictly defined into three classes: semiconductor materials, insulator materials, or conductors. Each class of materials has particular properties.

B.2.1 Semiconductors

All equations specified by your choice of models are solved in semiconductor regions. All semiconductor regions must have a band structure defined in terms of bandgap, density of states, affinity, and so on. The parameters used for any simulation can be echoed to the run-time output using **MODELS** PRINT. For complex cases with mole fraction dependent models, these quantities can be seen in TonyPlot by specifying **OUTPUT** BAND.PARAM and saving a solution file.

Any semiconductor region that is defined as an electrode is then considered to be a conductor region. This is typical for polysilicon gate electrodes.

B.2.2 Insulators

In insulator materials, only the Poisson and lattice heat equations are solved. Therefore for isothermal simulations, the only parameter required for an insulator is dielectric permittivity defined using **MATERIAL** PERM=<n>.

Materials usually considered as insulators (e.g., SiO_2) can be treated as semiconductors using Blaze, however all semiconductor parameters are then required.

B.2.3 Conductors

All conductor materials must be defined as electrodes, and all electrode regions are defined as conductor material regions. If a file containing regions of a material known to be a conductor are read in, these regions will automatically become un-named electrodes. As noted below, if the file contains unknown materials, these regions will become insulators.

During electrical simulation, only the electrode boundary nodes are used. Nodes that are entirely within an electrode region are not solved. Any quantities seen inside a conductor region in TonyPlot are spurious. Only optical ray tracing and absorption for Luminous and lattice heating are solved inside of conductor/electrode regions.

B.2.4 Unknown Materials

If a mesh file is read containing materials not in Table B-1, these will automatically become insulator regions with a relative permittivity of 3.9. All user-defined materials from Athena, regardless of the material name chosen, will also become such insulator materials.

B.2.5 Specifying Unknown or User-Defined Materials in Atlas

As mentioned previously, all materials in Atlas are classified as a semiconductor, an insulator, or a conductor. These classes are termed user groups. In order to correctly define a new material in Atlas, you must specify the name of the material, the user group it belongs to, and the known Atlas material it is to take as a default material. Once you set these three elements up in their appropriate places in the input deck, you can change the specific properties of them using **MATERIAL** statements as you would normally do.

A user group takes one of the following definitions:

```
user.group=semiconductor

user.group=insulator

user.group=conductor
```

The name of the material is specified using:

```
user.material=material_name
```

The default material that the new material is to be based on is specified using:

```
user.default=known_atlas_material_name
```

As an example of the syntax required to define a new material, define a new material called my_oxynitride. For simplicity, omit all the other code for the complete Atlas deck and only include code for deck structure purposes. Then, define a new insulator called my_oxynitride.

```
go atlas


mesh

.

.

.

region      num=1      x.min=1      x.max=2      y.min=0      y.max=2
user.material=my_oxynitirde


electrode ...


doping ...


material        material=my_oxynitride        user.group=insulator
user.default=oxide permittivity=9

.

.

.
```

After creating your new material in Atlas, include a print on your **MODELS** statement to echo the parameter values used for the material.

B.3 Atlas Materials

This section lists the materials in Atlas. The superscripted numbers (e.g., Silicon[(1)]) on these tables refer to the notes on the next page.

Table B-1 The Atlas Materials

Single Element Semiconductors			
Diamond	Germanium	Poly[(2)]	Silicon[(1)]
Binary Compound Semiconductors			
AlAs	AlN	AlP	AlSb
BeTe	CdS	CdSe	CdTe
GaAs[(3)]	GaP	GaN	GaSb
HgS	HgSe	HgTe	IGZO
InAs	InN	InP	InSb
PbS	PbSe	PbTe	
SiC-3C	SiC-4H	SiC-6H	ScN
SiGe	SnTe	ZnS	ZnSe
ZnO	ZnTe		
Ternary Compound Semiconductors			
AlAsSb	AlGaAs	AlGaP	AlGaN
AlGaSb	AlInN	AlPSb	AlPAs
CdZnTe	GaAsSb	GaAsP	GaSbAs
GaSbP	HgCdTe	InGaAs	InAlAs
InAsP	InAlP	InAlSb	InAsSb
InGaP	InGaN	InGaSb	InPSb
Quaternary Compound Semiconductors			
AlGaAsP	AlGaAsSb	AlGaNAs	AlGaNP
AlInNAs	AlInNP	$CuInGaSe_2$	InAlAsSb
InAlAsP	InAlGaAs	InGaAsP	InAlGaN
InAlGaP	InGaNAs	InGaNP	InGaAsSb
InPAsSb			

Table B-1 The Atlas Materials			
Insulators			
Air	Al_2O_3	Ambient	BPSG
BSG	HfO_2	$HfSiO_4$	La_2O_3
Nitride	Oxide	OxyNitride	Sapphire
Si3N4	SiO2	SiN	SnO_2
Ta_2O_5	TiO	TiO_2	Vacuum
ZrO_2			
Conductors[4]			
Aluminum	AlSi	AlSiCu	AlSiTi
Cobalt	Copper	CoSi	Conductor
Contact	GaSbTe	Gold	GST
In2O3	Iron	ITO	Lead
Molybdenum	MoSi	NiSi	Nickel
PaSi	Palladium	PCM	Platinum
Polysilicon[2]	PtSi	Silver	Tantalum
TaSi	Tin	TiSi	Titanium
TiON	TiW	Tungsten	WSi
ZrSi			
Organics			
Alq3 Organic	BAlq	CBP	CuPc
IGZO	Irppy	NPB	NPD
Pentacene	PPV	Tetracene	TPD

Notes

1: The material models and parameters of Silicon are identical to those of S-Pisces2 Version 5.2. Be aware that although these band parameters may be physically inaccurate compared to bulk silicon measurements, most other material parameters and models are empirically tuned using these band parameters.

2: Polysilicon is treated differently depending on how it is used. In cases where it's defined as an electrode, it's treated as a conductor. It can also be used as a semiconductor such as in a polysilicon emitter bipolars.

3: The composition of SiGe is the only binary compound that can be varied to simulate the effects of band gap variations.

4: Conductor names are only associated with electrodes. They are used for the specification of thermal conductivities and complex index of refraction and for display in TonyPlot.

B.3.1 Specifying Compound Semiconductors Rules

The rules for specifying the order of elements for compound semiconductors are derived from the rules used by the International Union of Pure and Applied Chemistry [223]. These rules are:

1. Cations appear before anions.
2. When more than one cation is present the order progresses from the element with the largest atomic number to the element with the smallest atomic number.
3. The order of anions should be the in order of the following list: B, Si, C, Sb, As, P, N, H, Te, Se, S, At, I, Br, Cl, O, and F.
4. The composition fraction x is applied to the cation listed first.
5. The composition y is applied to the anion listed first.

To accommodate popular conventions, there are several exceptions to these rules (see Table B-2).

Table B-2 Exceptions to the Specifying Compound Semiconductor Rules

Material	Description
`SiGe`	The composition fraction x applies to the Ge component. SiGe is then specified as $Si_{(1-x)}Ge_{(x)}$, an exception to rule #4.
`AlGaAs`	This is specified as $Al_{(x)}Ga_{(1-x)}As$. This is an exception to rule #2.
`InGaAsP (system)`	The convention $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ as set forth by Adachi [1] is used. This is an exception to rule #4.
InAlAs	The convention $In_{(1-x)}Al_{(x)}As$ as set forth by Ishikawi [141] is used. This is an exception to rule #4.
InAlGaN	Here we use the convention $In_yAlxGa_{(1-x-y)}N$. This is an exception to rule #4.
HgCdTe	Here we use the convention $Hg_{(1-x)}C_{(x)}dTe$. This is an exception to rule #4.
InAlN	Here we use the convention $In_{(1-x)}Al_{(x)}N$. This is an exception to rule #4.

B.4 Silicon and Polysilicon

The material parameters defaults for Polysilicon are identical to those for Silicon. The following paragraphs describe some of the material parameter defaults for Silicon and Polysilicon.

Note: A complete description is given of each model Chapter 3 "Physics". The parameter defaults listed in that chapter are all Silicon material defaults.

Table B-3 Band parameters for Silicon and Poly

Material	Eg300 eV	a	b	Nc300 per cc	Nv300 per cc	χ eV
Silicon	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17
Poly	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17

Table B-4 Static dielectric constants for Silicon and Poly

Material	Dielectric Constant
Silicon	11.8
Poly	11.8

The default mobility parameters for Silicon and Poly are identical in all cases. The defaults used depend on the particular mobility models in question. A full description of each mobility model and their coefficients are given in Section 3.6.1 "Mobility Modeling".

Table B-5 contains the silicon and polysilicon default values for the low field constant mobility model.

Table B-5 Lattice Mobility Model Defaults for Silicon and Poly

Material	MUN cm^2/Vs	MUP cm^2/Vs	TMUN	TMUP
Silicon	1000.0	500.0	1.5	1.5
Poly	1000.0	500.0	1.5	1.5

Table B-6 shows the silicon and polysilicon default values for the field dependent mobility model.

Table B-6 Parallel Field Dependent Mobility Model Parameters for Silicon and Poly

Material	BETAN	BETAP
Silicon	2	1
Poly	2	1

Table B-7 shows the default values used in the bandgap narrowing model for Silicon and Polysilicon.

Table B-7 Bandgap Narrowing Parameters for Silicon and Poly

Statement	Parameter	Defaults	Units
MATERIAL	BGN.E	6.92×10^{-3}	V
MATERIAL	BGN.N	1.3×10^{17}	cm^{-3}
MATERIAL	BGN.C	0.5	—

Table B-8 shows the default parameters for Schockley-Read-Hall (SRH) recombination.

Table B-8 SRH Lifetime Parameter Defaults for Silicon and Poly

Material	TAUN0 (s)	TAUP0 (s)	NSRHN (cm^{-3})	NSRHP (cm^{-3})
Silicon	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}
Poly	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}

The default parameters for Auger recombination are given in Table B-9.

Table B-9 Auger Coefficient Defaults for Silicon and Poly

Material	AUGN	AUGP
Silicon	2.8×10^{-31}	9.9×10^{-32}
Poly	2.8×10^{-31}	9.9×10^{-32}

Table B-10 shows the default values for the SELB impact ionization coefficients.

Table B-10 Impact Ionization Coefficients for Silicon and Poly

Parameter	Value
EGRAN	4.0×10^{5}
BETAN	1.0
BETAP	1.0
AN1	7.03×10^{5}
AN2	7.03×10^{5}
BN1	1.231×10^{6}
BN2	1.231×10^{6}

Table B-10 Impact Ionization Coefficients for Silicon and Poly	
Parameter	**Value**
AP1	6.71×10^{5}
AP2	1.582×10^{6}
BP1	1.693×10^{6}
BP2	2.036×10^{6}

Table B-11 shows the default Richardson cofficients.

Table B-11 Effective Richardson Coefficients for Silicon and Poly		
Material	**ARICHN ($A/cm^2/K^2$)**	**ARICHP ($A/cm^2/K^2$)**
Silicon	110.0	30.0
Poly	110.0	30.0

B.5 The $Al_{(x)}Ga_{(1-x)}As$ Material System

Table B-12 shows the default recombination parameters for AlGaAs.

Table B-12 Default Recombination Parameters for AlGaAs

Parameter	Value	Equation
TAUN0	1.0×10^{-9}	3-348
TAUP0	1.0×10^{-8}	3-348
COPT	1.5×10^{-10}	3-382
AUGN	5.0×10^{-30}	3-386
AUGP	1.0×10^{-31}	3-386

Table B-13 shows the default values for the SELB impact ionization coefficients used for GaAs. AlGaAs uses the same values as GaAs.

Table B-13 Impact Ionization Coefficients for GaAs

Parameter	Value
EGRAN	0.0
BETAN	1.82
BETAP	1.75
EGRAN	0.0
AN1	1.889×10^{5}
AN2	1.889×10^{5}
BN1	5.75×10^{5}
BN2	5.75×10^{5}
AP1	2.215×10^{5}
AP2	2.215×10^{5}
BP1	6.57×10^{5}
BP2	6.57×10^{5}

The default values for the effective Richardson coefficients for GaAs are 6.2875 $A/cm^2/K^2$ for electrons and 105.2 $A/cm^2/K^2$ for holes.

B.6 The $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ System

The default material thermal models for InGaAsP assumes lattice-matching to InP. The material density is then given by

$$\rho = 4.791 + 0.575y.composition + 0.138y.composition$$

The specific heat for InGaAsP is given by

$$C_p = 0.322 + 0.026y.composition - 0.008y.composition$$

The thermal resistivities of InGaAsP are linearly interpolated from Table B-14.

Table B-14 Thermal Resistivities for InGaAsP Lattice-Matched to InP

Composition Fraction y	Thermal Resistivity (deg(cm/w)
0.0	1.47
0.1	7.05
0.2	11.84
0.3	15.83
0.4	19.02
0.5	21.40
0.6	22.96
0.7	23.71
0.8	23.63
0.9	22.71
1.0	20.95

The default thermal properties for the ternary compounds in the InGaAsP system: $In_{(1-x)}Ga_{(x)}As$, $In_{(1-x)}Ga_{(x)}P$, $InAs_{(y)}P_{(1-y)}$, and $GaAs_{(y)}P_{(1-y)}$, are given as a function of composition fraction by linear interpolations from these binary compounds.

B.7 Silicon Carbide (SiC)

Table B-15 shows the default values for the SELB impact ionization coefficients used for SiC.

Table B-15 Impact Ionization Coefficients for SiC

Parameter	Value
EGRAN	0.0
BETAN	1.0
BETAP	1.0
AN1	1.66×10^{6}
AN2	1.66×10^{6}
BN1	1.273×10^{7}
BN2	1.273×10^{7}
AP1	5.18×10^{6}
AP2	5.18×10^{6}
BP1	1.4×10^{7}
BP2	1.4×10^{7}

Table B-16 shows the default thermal parameters used for both 6H-SiC and 3C-SiC.

Table B-16 Default Thermal Parameters for SiC

Parameter	Value	
	3C-SiC	6H-SiC
TCA	0.204	0.385

B.8 Material Defaults for GaN/InN/AlN System

Tables B-17 through B-20 show the default values for the band structure parameters for InN, GaN and AlN for each of the user-defined parameter sets. To choose the parameters sets, specify KP.SET1 (Table B-17), KP.SET2 (Table B-18), KP.SET3 (Table B-19), and KP.SET4 (Table B-20). By default, the parameters in Table B-17 are used.

Table B-17 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET1 (All Parameters Values are from [246] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN	GaN	AlN
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.11	0.20	0.33
Electron eff. mass (t)	m_{ct}	MTT	m0	0.11	0.18	0.25
Hole eff. mass param.	A1	A1		-9.24	-7.24	-3.95
Hole eff. mass param.	A2	A2		-0.60	-0.51	-0.27
Hole eff. mass param.	A3	A3		8.68	6.73	3.68
Hole eff. mass param.	A4	A4		-4.34	-3.36	-1.84
Hole eff. mass param.	A5	A5		-4.32	-3.35	-1.92
Hole eff. mass param.	A6	A6		-6.08	-4.72	-2.91
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.89	3.42	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.019	-0.164
Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01413	0.01913
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Elastic constant	C33	C33	GPa	200	392	382
Elastic constant	C13	C13	GPa	94	100	127

Table B-17 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET1 (All Parameters Values are from [246] unless otherwise noted.)						
Hydrostatic deformation potential	ac	AC	eV	-4.08	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-0.89	-0.89	-0.89
Shear deform. potential	D2	D2	eV	4.27	4.27	4.27
Shear deform. potential	D3	D3	eV	5.18	5.18	5.18
Shear deform. potential	D4	D4	eV	-2.59	-2.59	-2.29

Table B-18 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET2 (All Parameters Values are from [335] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN**	**GaN**	**AlN**
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.12	0.20	0.32
Electron eff. mass (t)	m_{ct}	MTT	m0	0.12	0.20	0.28
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95
Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27
Hole eff. mass param.	A3	A3		7.57	5.65	3.68
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92
Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.507	6.23

Table B-18 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET2 (All Parameters Values are from [335] unless otherwise noted.)						
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.019	-0.164
Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01413	0.01913
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Elastic constant	C33	C33	GPa	224	398	373
Elastic constant	C13	C13	GPa	92	106	108
Hydrostatic deformation potential	ac	AC	eV	-4.08	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-0.89	-3.0	-0.89
Shear deform. potential	D2	D2	eV	4.27	3.6	4.27
Shear deform. potential	D3	D3	eV	5.18	8.82	5.18
Shear deform. potential	D4	D4	eV	-2.59	-4.41	-2.29

Table B-19 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET3 (All Parameters Values are from [56] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN [335]**	**GaN**	**AlN**
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.12	0.20	0.32
Electron eff. mass (t)	m_{ct}	MTT	m0	0.12	0.20	0.28
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95 [335]
Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27 [335]
Hole eff. mass param.	A3	A3		7.57	5.65	3.68 [335]

Table B-19 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET3 (All Parameters Values are from [56] unless otherwise noted.)						
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84 [335]
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92 [335]
Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91 [335]
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.44	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.016	-0.0585
Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01213	0.020413
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112 [335]
Elastic constant	C33	C33	GPa	224	267	395
Elastic constant	C13	C13	GPa	92	158	120
Hydrostatic deformation potential	ac	AC	eV	-3.5	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-3.0	0.7	0.7
Shear deform. potential	D2	D2	eV	3.6	2.1	2.1
Shear deform. potential	D3	D3	eV	8.82	1.4	1.4
Shear deform. potential	D4	D4	eV	-4.41	-0.7	-0.7

Table B-20 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET4 (All Parameters Values are from [58] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN [335]**	**GaN**	**AlN**
Electron eff. mass (*z*)	m_{cz}	MZZ	m0	0.12	0.20	0.33
Electron eff. mass (*t*)	m_{ct}	MTT	m0	0.12	0.18	0.25
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95 [335]
Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27 [335]
Hole eff. mass param.	A3	A3		7.57	5.65	3.68 [335]
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84 [335]
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92 [335]
Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91 [335]
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.44	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.021	-0.0585
Crystal-field split energy	D2	DELTA2	eV	0.00113	0.01618	0.020413
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112 [335]
Elastic constant	C33	C33	GPa	224	392	395
Elastic constant	C13	C13	GPa	92	100	120
Hydrostatic deformation potential	ac	AC	eV	-35	-4.08	-4.08

Table B-20 Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET4 (All Parameters Values are from [58] unless otherwise noted.)						
Shear deform. potential	D1	D1	eV	-3.0	-0.89	-0.89
Shear deform. potential	D2	D2	eV	3.6	4.27	4.27
Shear deform. potential	D3	D3	eV	8.82	5.18	5.18
Shear deform. potential	D4	D4	eV	-4.41	-2.59	-2.29

Tables B-21 through B-23 show the default polarization parameters for InN, GaN and AlN for each of the user-defined parameter sets. To choose the parameter sets, specify POL.SET1 (Table B-21), POL.SET2 (Table B-22), or POL.SET3 (Table B-23) on the **MATERIAL** statement. By default, the parameter values in Table B-24 are used.

Table B-21 Polarization Parameters for InN, GaN and AlN Using POL.SET1 (All Parameters Values are from [246] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN**	**GaN**	**AlN**
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.042	-0.034	-0.09
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.81	0.67	1.5
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.41	-0.34	-0.53

Table B-22 Polarization Parameters for InN, GaN and AlN Using POL.SET2 (All Parameters Values are from [335] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN**	**GaN**	**AlN**
Lattice constant	a_0	ALATTICE	Å	3.545	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.032	-0.029	-0.081
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.97	1.27	1.79
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.57	-0.35	-0.5

Table B-23 Polarization Parameters for InN, GaN and AlN Using POL.SET3 (All Parameters Values are from [57] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN**	**GaN**	**AlN**
Lattice constant	a_0	ALATTICE	Å	3.54	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.032	-0.029	-0.081
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.97	0.73	1.46
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.57	-0.49	-0.6

B.9 Material Defaults for Compound Semiconductors

Tables B-24a and B-24b, B-24c and B-24d show the material defaults for binary compound semiconductors. Table B-25 shows Material defaults for ternary compound semiconductors.

Note: x.comp is set to 0.3. See Section 6.4 "Material Dependent Physical Models" for more information.

Table B-24a. Materials Defaults for binary compound semiconductors

Material	GaAs	GaP	CdSe	SnTe	InP	CdTe	ScN
Epsilon	13.2	11.1	10.6	0	12.5	10.9	0
Eg (eV)	1.42	2.26	1.74	0.18	1.35	1.5	2.15
Chi (eV)	4.07	4.4	0	0	4.4	4.28	0
Nc (per cc)	4.35E+17	1.76E+18	1.18E+18	1	5.68E+17	1	1
Nv (per cc)	8.16E+18	8.87E+18	7.57E+18	1	8.87E+18	1	1
ni (per cc)	2.12E+06	0.409	7.22E+03	0.0308	1.03E+07	2.51E-13	8.70E-19
Gc	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Lifetime (el)	1.00E-09	1	1	1	1	1	1
Lifetime (ho)	2.00E-08	1	1	1	1	1	1
Auger cn	5.00E-30	0	0	0	0	0	0
Auger cp	1.00E-31	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0
Copt	1.50E-10	0	0	0	0	0	0
An**	6.29	20.4	15.6	1.40E-11	9.61	1.40E-11	1.40E-11
Ap**	105	60.1	54.1	1.40E-11	60.1	1.40E-11	1.40E-11
betan	1.82	1	1	1	1	1	1
betap	1.75	1	1	1	1	1	1
egran	0	0	0	0	0	0	0
an1	1.90E+05	7.30E+04	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn1	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.90E+05	7.30E+04	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
an2	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	2.22E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	6.57E+05	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	2.22E+05	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	6.57E+05	2.40E+05	2.40E+05	2.04E+06	2.40E+05	2.04E+06	2.04E+06
Vsatn (cm/s) :	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s) :	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
mun (cm^2/Vs)	8000	300	800		4600	1050	
tmun	1	1.5	1.5		1.5	1.5	
mup (cm^2/Vs)	400	100			150	100	
tmup	2.1	1.5			1.5	1.5	

Table B-24b. Materials Defaults for binary compound semiconductors

Material	6H-SiC	4H-SiC	3C-SiC	InSb	HgS	InAs	HgSe	AlP
Epsilon	9.66	9.7	9.7	17.7	0	14.6	25	9.8
Eg (eV)	2.9	3.23	3.23	0.174	2.5	0.35	0	2.49
Chi (eV)	3.5	3.2	3.2	4.59	0	4.67	0	3.98
Nc (per cc)	7.68E+18	1.66E+19	6.59E+18	3.94E+16	1	9.33E+16	1	6.37E+18
Nv (per cc)	4.76E+18	3.30E+19	1.68E+18	7.12E+18	1	8.12E+18	1	1.48E+19
ni (per cc)	2.64E-06	1.73E-08	1.1	1.84E+16	9.99E-22	9.99E+14	0	0.0123
Gc	2	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Lifetime (el)	1.00E-07	1.00E-07	1.00E-07	1	1	1.00E-09	1	1
Lifetime (ho)	1.00E-07	1.00E-07	1.00E-07	1	1	2.00E-08	1	1
Auger cn	2.80E-31	5.00E-29	0	0	0	0	0	0
Auger cp	9.90E-31	9.90E-32	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0	0
Copt	0	0	0	0	0	0	0	0
An**	54.6	91.3	49.3	1.62	1.40E-11	1.40E-11	1.40E-11	48.2
Ap**	39.7	144	19.8	51.9	1.40E-11	1.40E-11	1.40E-11	84.5
betan	1	1	1	1	1	1	1	1
betap	1	1	1	1	1	1	1	1
egran	-1	-1	-1	-1	0	0	0	0
an1	1.66E+06	1.66E+06	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn1	1.27E+07	1.27E+07	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.66E+06	1.66E+06	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn2	1.27E+07	1.27E+07	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	5.18E+06	1.66E+06	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	1.40E+07	1.27E+07	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	5.18E+06	5.18E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	1.40E+07	1.40E+07	2.04E+06	2.40E+05	2.04E+06	2.40E+05	2.04E+06	2.04E+06
Vsatn (cm/s) :	1.90E+07	2.20E+07	2.00E+07	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s) :	1.00E+07	1.00E+07	1.00E+07	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06

Table B-24c. Material Defaults for binary compound semiconductors

Material	ZnS	HgTe	AlAs	ZnSe	PbS	BeTe	GaN
Epsilon	8.3	20	12	8.1	170	0	8.9
Eg (eV)	3.8	0	2.16	2.58	0.37	2.57	3.43
Chi (eV)	0	0	3.5	4.09	0	0	4.31
Nc (per cc)	6.35E+18	1	4.35E+17	1	3.14E+18	1	2.24E+18
Nv (per cc)	1	1	8.16E+18	1	3.14E+18	1	2.51E+19
ni (per cc)	3.02E-23	1	1.35	2.13E-22	2.45E+15	2.58E-22	1.06E-10
Gc	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters							
Lifetime (el)	1	1	1.00E-09	1	1	1	1.20E-08
Lifetime (ho)	1	1	2.00E-08	1	1	1	1.20E-08
Auger cn	0	0	5.00E-30	0	0	0	0
Auger cp	0	0	1.00E-31	0	0	0	0
Auger kn	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0
Copt	0	0	1.50E-10	0	0	0	0
An**	48.1	1.40E-11	8.05	1.40E-11	30	1.40E-11	24
Ap**	1.40E-11	1.40E-11	56.8	1.40E-11	30	1.40E-11	120
Impact Ionization Parameters							
betan	1	1	1.82	1	1	1	1
betap	1	1	1.75	1	1	1	1
egran	0	0	0	0	0	0	-1
an1	7.03E+05	7.03E+05	1.90E+05	7.03E+05	7.30E+04	7.03E+05	2.52E+08
bn1	1.23E+06	1.23E+06	5.75E+05	1.23E+06	1.23E+06	1.23E+06	3.41E+07
an2	7.03E+05	7.03E+05	1.90E+05	7.03E+05	7.30E+04	7.03E+05	2.52E+08
bn2	1.23E+06	1.23E+06	5.75E+05	1.23E+06	1.23E+06	1.23E+06	3.41E+07
ap1	6.71E+05	6.71E+05	2.22E+05	6.71E+05	6.71E+05	6.71E+05	5.37E+06
bp1	1.69E+06	1.69E+06	6.57E+05	1.69E+06	1.69E+06	1.69E+06	1.96E+07
ap2	1.58E+06	1.58E+06	2.22E+05	1.58E+06	1.58E+06	1.58E+06	5.37E+06
bp2	2.04E+06	2.04E+06	6.57E+05	2.04E+06	2.40E+05	2.04E+06	1.96E+07
Saturation Velocities							
Vsatn (cm/s)	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.91E+07
Vsatp (cm/s)	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Mobility Parameters							
mun (cm^2/Vs)	165	22000	1000	100	600		400
tmun	1.5	1.5	1.5	1.5	1.5		1.5
mup (cm^2/Vs)	5	100	100		700		8
tmup	1.5	1.5	1.5		1.5		1.5

Table B-24d. Material Defaults for binary compound semiconductor

Material	AlSb	ZnTe	PbSe	GaSb	PbTe	SiGe
Epsilon	11	9.7	250	15.7	412	11.8
Eg (eV)	1.52	2.28	0.26	0.72	0.29	0.78
Chi (eV)	3.65	3.5	0	4.06	4.6	4.17
Nc (per cc)	6.79E+18	1	1	5.68E+18	1.76E+18	1.92E+19
Nv (per cc)	6.35E+18	1	1	2.95E+18	2.24E+18	8.20E+18
ni (per cc)	1.12E+06	7.04E-20	0.00655	3.66E+12	7.28E+15	3.52E+12
Gc	2	2	2	2	2	2
Gv	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters						
Lifetime (el)	1	1	1	1	1	3.00E-05
Lifetime (ho)	1	1	1	1	1	1.00E-05
Auger cn	0	0	0	0	0	8.30E-32
Auger cp	0	0	0	0	0	1.80E-31
Auger kn	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0
Copt	0	0	0	0	0	0
An**	50.3	1.40E-11	1.40E-11	44.6	20.4	110
Ap**	48.1	1.40E-11	1.40E-11	28.8	24	30
Impact Ionization Parameters						
betan	1	1	1	1	1	1
betap	1	1	1	1	1	1
egran	0	0	0	0	0	4.00E+05
an1	7.30E+04	7.03E+05	7.03E+05	7.30E+04	7.30E+04	7.03E+05
bn1	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	7.30E+04	7.03E+05	7.03E+05	7.30E+04	7.30E+04	7.03E+05
bn2	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	2.40E+05	2.04E+06	2.04E+06	2.40E+05	2.40E+05	2.04E+06
Saturation Velocities						
Vsatn (cm/s)	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+07
Vsatp (cm/s)	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	7.98E+06
Mobility Parameters						
mun (cm^2/Vs)	200	7	1020	4000	6000	1430
mup (cm^2/Vs)	1.5	1.5	2.5	1.5	1.5	2.5
mup (cm^2/Vs)	550		930	1400	4000	480
tmup	1.5		2.5	1.5	1.5	2.5

Table B-25 Material Default for ternary compound semiconductors

Material	AlGaAs	GaSbP	InAlAs	GaAsP	InGaAs	GaSbAs	InAsP	HgCdTe
Epsilon	12.3	0	0	11.7	14.2	0	13.1	16.4
Band Parameters								
Eg (eV)	1.8	2.01	2.1	2.31	0.571	0.665	1.03	0.291
Chi (eV)	3.75	0	0	4.32	4.13	4.06	4.32	2.54
Nc (per cc)	4.35E+17	1	1	1.04E+18	1.15E+17	1	2.86E+17	7.58E+16
Nv (per cc)	8.16E+18	1	1	8.54E+18	8.12E+18	1	8.54E+18	1.02E+19
ni (per cc)	1.37E+03	1.24E-17	2.29E-18	0.125	1.56E+13	2.61E-06	3.56E+09	3.18E+15
Gc	2	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters								
Lifetime (el)	1.00E-09	1	1	1	1	1	1	1
Lifetime (ho)	2.00E-08	1	1	1	1	1	1	1
Auger cn	5.00E-30	0	0	0	0	0	0	0
Auger cp	1.00E-31	0	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0	0
Copt	1.50E-10	0	0	0	0	0	0	0
An**	8.05	1.40E-11	1.40E-11	14.4	3.32	1.40E-11	6.08	2.51
Ap**	56.8	1.40E-11	1.40E-11	58.6	56.6	1.40E-11	58.6	66.1
Impact Ionization Model Parameters (Selberherr Model)								
betan	1.82	1	1	1	1	1	1	1
betap	1.75	1	1	1	1	1	1	1
egran	0	0	0	0	0	0	0	0
an1	1.90E+05	7.03E+05	8.60E+06	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05
bn1	5.75E+05	1.23E+06	3.50E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.90E+05	7.03E+05	8.60E+06	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05

Note: `x.comp` is set to 0.3. See Section 6.4 "Material Dependent Physical Models" for more information.

Table B-26 shows material defaults for quaternary compound semiconductors.

Table B-26 Material Defaults for quaternary compound semiconductors

Material :	InGaAsP	AlGaAsP	AlGaAsSb	InGaNAs	InGaNP	AlGaNAs	AlGaNP	AlInNAs
Epsilon :	12.7	0	0	0	0	0	0	0
Band Parameters								
Eg (eV) :	1.27	0	0	0	0	0	0	0
Chi (eV) :	4.32	0	0	0	0	0	0	0
Nc (per cc) :	3.61E+17	1	1	1	1	1	1	1
Nv (per cc) :	8.54E+18	1	1	1	1	1	1	1
ni (per cc) :	3.72E+07	1	1	1	1	1	1	1
Gc :	2	2	2	2	2	2	2	2
Gv :	4	4	4	4	4	4	4	4
Ed (eV) :	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV) :	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters								
Lifetime (el):	1	1	1	1	1	1	1	1
Lifetime (ho):	1	1	1	1	1	1	1	1
Auger cn :	0	0	0	0	0	0	0	0
Auger cp :	0	0	0	0	0	0	0	0
Auger kn :	0	0	0	0	0	0	0	0
Auger kp :	0	0	0	0	0	0	0	0
Copt :	0	0	0	0	0	0	0	0
An** :	7.11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11
Ap** :	58.6	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11
Impact Ionization	Parameters	(Selberherr model)						
betan :	1	1	1	1	1	1	1	1
betap :	1	1	1	1	1	1	1	1
egran :	0	0	0	0	0	0	0	0

Table B-27 shows material defaults for organic materials.

Table B-27 Material Defaults for organic materials

Material	Organic	Pentacene	Alq3	TPD	PPV	Tetracene
Epsilon	11.8	11.8	3.4	11.8	11.8	11.8
Band Parameters						
Eg (eV)	1.08	1.08	2.4	1.08	1.08	1.08
Chi (eV)	4.17	4.17	4.17	4.17	4.17	4.17
Nc (per cc)	2.80E+19	2.80E+19	1.44E+20	2.80E+19	2.80E+19	2.80E+19
Nv (per cc)	1.04E+19	1.04E+19	1.44E+20	1.04E+19	1.04E+19	1.04E+19
ni (per cc)	1.45E+10	1.45E+10	0.998	1.45E+10	1.45E+10	1.45E+10
Gc	2	2	2	2	2	2
Gv	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045
An**	129	129	385	129	129	129
Ap**	66.8	66.8	385	66.8	66.8	66.8
Recombination Parameters						
taun0	1	1	1	1	1	1
taup0	1	1	1	1	1	1
etrap	0	0	0	0	0	0
nsrhn	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03
nsrhp	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03
ksrhtn	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
ksrhtp	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
ksrhcn	3.00E-13	3.00E-13	3.00E-13	3.00E-13	3.00E-13	3.00E-13
ksrhcp	1.18E-12	1.18E-12	1.18E-12	1.18E-12	1.18E-12	1.18E-12
ksrhgn	1.77	1.77	1.77	1.77	1.77	1.77
ksrhgp	0.57	0.57	0.57	0.57	0.57	0.57
nsrhn	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03
nsrhp	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03
augn	0	0	0	0	0	0
augp	0	0	0	0	0	0
augkn	0	0	0	0	0	0
augkp	0	0	0	0	0	0
kaugcn	1.83E-31	1.83E-31	1.83E-31	1.83E-31	1.83E-31	1.83E-31
kaugcp	2.78E-31	2.78E-31	2.78E-31	2.78E-31	2.78E-31	2.78E-31
kaugdn	1.18	1.18	1.18	1.18	1.18	1.18
kaugdp	0.72	0.72	0.72	0.72	0.72	0.72
copt	0	0	0	0	0	0
Band-to-Band Tunneling Parameters						
bb.a	4.00E+14	4.00E+14	4.00E+14	4.00E+14	4.00E+14	4.00E+14
bb.b	1.90E+07	1.90E+07	1.90E+07	1.90E+07	1.90E+07	1.90E+07
bb.gamma	2.5	2.5	2.5	2.5	2.5	2.5
mass.tunnel	0.25	0.25	0.25	0.25	0.25	0.25
me.tunnel	1.08	1.08	3.21	1.08	1.08	1.08
mh.tunnel	0.556	0.556	3.21	0.556	0.556	0.556
Saturation Velocities						
Vsatn (cm/s) :	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s) :	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06

Note: `x.comp` and `y.comp` are both set to 0.3. See Section 6.4 "Material Dependent Physical Models" for more information.

Table B-28 lists the default material parameters for metal oxides and various other semiconductors.

Table B-28 Material Defaults for Conductive Oxides and Others

Material	CMO	IMO	CdO	MgO	CdZnO	MgZnO	MgCdO	MgCdZnO	CNT	Se	IGZO
Epsilon	9.7	28.0	9.0	9.8	8.49	8.49	8.49	8.49	9.7	11.56	10.0
Band Parameters											
Eg(eV)	0.5	5.0							0.8	2.22	3.05
Chi(eV)	5.0	3.5							3.2	2.02	4.16
Nc(percc)	1.0e20	1.0e20	2.2e18	2.2e18	2.2e18	2.2e18	2.2e18	2.2e18	6.65e17	5.0e19	5.0e18
Nv(percc)	1.0e20	1.0e20	1.8e19	1.8e19	1.8e19	1.8e19	1.8e19	1.8e19	6.65e17	5.0e19	5.0e18
mun0 (cm2/Vs)	0.1	0.1	100.0							0.003	15
mup0 (cm2/Vs)	0.1	0.1	25.0							0.1	0.1

Table B-29 Lattice Parameters for Binary Zincblende Semiconductors [246]

Parameter	a0	da0/dT	C11	C12	b	av	ac
Unit	(A)	(10^{-5} A/K)	(GPa)	(GPa)	(eV)	(eV)	(eV)
GaAs	5.65325	3.88	1221	566	-2.0	1.16	-7.17
InP	5.8697	2.79	1011	561	-2.0	0.6	-6.0
AlAs	5.6611	2.90	1250	534	-2.3	2.47	-5.64
GaSb	6.0959	4.72	884.2	402.6	-2.0	0.8	-7.5
AlSb	6.1355	2.60	876.9	434.1	-1.35	1.4	-4.5
InAs	6.0583	2.74	832.9	452.6	-1.8	1.0	-5.08
GaP	5.4505	2.92	1405	620.3	-1.6	1.7	-8.2
AlP	5.4672	2.92	1330	630	-1.5	3.0	-5.7
InSb	6.4794	3.48	684.7	373.5	-2.0	0.36	-6.94

Table B-30 Luttinger Parameters for Zincblende Semiconductors

Parameter	$\gamma 1$	$\gamma 2$	$\gamma 3$
GaAs	6.98	2.06	2.93
InP	5.08	1.60	2.10
AlAs	3.76	0.82	1.42

Table B-30 Luttinger Parameters for Zincblende Semiconductors			
Parameter	**γ1**	**γ2**	**γ3**
GaSb	13.4	4.7	6.0
AlSb	5.18	1.19	1.97
InAs	20.0	8.5	9.2
GaP	4.05	0.49	2.93
AlP	3.35	0.71	1.23
InSb	34.8	15.5	16.5

B.10 Miscellaneous Semiconductors

The remainder of the semiconductors available have defined default parameter values to various degrees of completeness [1]. The following tables describe those parameter defaults as they exist. Since many of the material parameters are currently unavailable, we recommended that you should be careful when using these materials. Make sure the proper values are used

Note: You can use the **MODELS** PRINT statement to echo the parameters used to the run-time output.

Table B-31 Band Parameters for Miscellaneous Semiconductors

Material	Eg(0)eV	Eg(300)eV	a	b	m_c	m_v	χeV
Silicon							
Polysilicon							
Ge	0.7437		4.77×10^{-4}	235.0	0.2225	0.2915	4.0
Diamond		5.45	4.77×10^{-4}	0.0	(a)	(b)	7.2
6H-SiC	2.9	2.9	0.0	0.0	0.454	0.33	3.5
4H-SiC	3.26	3.26	0.0	0.0	0.76	1.20	4.0
3C-SiC	2.2	2.2	0.0	0.0	0.41	0.165	3.2
A1P	2.43	2.43	0.0	0.0			
A1As	2.16	2.16	0.0	0.0			3.5
A1Sb	1.6		2.69×10^{-4}	2.788	(c)	0.4	
GaSb	0.81		3.329×10^{-4}	-27.6622	(c)	0.24	3.65
InSb	0.235		2.817×10^{-4}	90.0003	0.014	0.4	4.06
ZnS	3.8	3.8	0.0	0.0	0.4		4.59
ZnSe	2.58	2.58	0.0	0.0	0.1	0.6	
ZnTe	2.28		0.0	0.0	0.1	0.6	4.09
CdS	2.48	2.48	0.0	0.0	0.21	0.8	4.18
CdSe	1.74	1.74	0.0	0.0	0.13	0.45	4.5
CdTe	1.5	1.5	0.0	0.0	0.14	0.37	
HgS	2.5	2.5	0.0	0.0			4.28
HgSe							
HgTe							
PbS	0.37	0.37	0.0	0.0	0.25	0.25	

Table B-31 Band Parameters for Miscellaneous Semiconductors							
Material	**Eg(0)eV**	**Eg(300)eV**	**a**	**b**	**m_c**	**m_v**	**χeV**
PbSe	0.26	0.26	0.0	0.0	0.33	0.34	
PbTe	0.29	0.29	0.0	0.0	0.17	0.20	4.6
SnTe	0.18	0.18	0.0	0.0			
ScN	2.15	2.15	0.0	0.0			
BeTe	2.57	2.57	0.0	0.0			

Notes

- $Nc300 = 5.0\times10^{18}$
- $Nv300 = 1.8\times10^{19}$
- $m_c(X) = 0.39$

  $m_c(G) = 0.09$

  $Nc = Nc(X) + Nc(G)$
- $m_c(G) = 0.047$

  $m_c(L) = 0.36$

  $Nc = Nc(G) + Nc(L)$

Table B-32 Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
Ge	16.0
Diamond	5.5
6H-SiC	9.66
4H-SiC	9.7
3C-SiC	9.72
AlP	9.8
AlAs	12.0
AlSb	11.0
GaSb	15.7
InSb	18.0
ZnS	8.3
ZnSe	8.1

Table B-32 Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
CdS	10
CdSe	10.6
CdTe	9.4
HgS	
HgSe	25.0
HgTe	20.
PbS	170.0
PbSe	250.0
PbTe	412.0
SnTe	1770.0
ScN	
BeTe	
ZnO	2.0

Table B-33 Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cm/s)**
Ge	3900.0(a)	1900.0(b)		
Diamond	500.0	300.0	2.0×10^{7}	
6H-SiC	330.0	300.0	2.0×10^{7}	1×10^{7}
4H-SiC	460	124	2.2×10^{7}	1×10^{7}
3C-SiC	1000.0	50.0	2.0×10^{7}	1×10^{7}
AlP	80.0			
AlAs	1000.0	100.0		
AlSb	200.0	550.0		
GaSb	4000.0	1400.0		
InSb	7800.0	750.0		
ZnS	165.0	5.0		
ZnSe	100.0	16		

Table B-33 Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cm/s)**
CdS	340.0	50.0		
CdSe	800.0			
CdTe	1050.0	100.0		
HgS				
HgSe	5500.0			
HgTe	22000.0	100.0		
PbS	600.0	700.0		
PbSe	1020.0	930.0		
PbTe	6000.0	4000.0		
SnTe				
ScN				
BeTe				

B.11 Insulators

The following tables show the default material parameters for insulator materials. As noted in the Section B.2 “Semiconductors, Insulators, and Conductors”, the only parameter required for electrical simulation in insulator materials is the dielectric constant. Thermal and optical properties are required in Giga and Luminous respectively.

Table B-34 Default Static Dielectric Constants of Insulators

Material	Dielectric Constant
Vacuum	1.0
Air	1.0
Ambient	1.0
Oxide	3.9
Si02	3.9
BSG	3.9
BPSG	3.9
Nitride	7.5
SiN	7.5
Si3N4	7.5
OxyNitride	7.5
Sapphire	12.0
Al2O3	9.3
HfO_2	22.0
$HfSiO_4$	12.0
SnO_2	9.0
Ta_2O_5	26
TiO_2	80
ZrO_2	22
La_2O_3	27

Table B-35 Default Thermal Parameters for Insulators at 300K		
Material	**Thermal Capacity ($J/cm^3/K$)**	**Thermal Conductivity (W/cm/K)**
Vacuum	0.0	0.0
Air	1.0	0.026×10^{-2}
Ambient	1.0	0.026×10^{-2}
Oxide	3.066	0.014
Si02	3.066	0.014
BSG	3.066	0.014
BPSG	3.066	0.014
Nitride	0.585	0.185
SiN	0.585	0.185
Si3N4	0.585	0.185
OxyNitride	0.585	0.185
Sapphire	3.14	0.4191
Al2O3	3.14	0.29
HfO2		
HfSiO2		

B.12 Metals/Conductors

Table B-36 shows the default values for resistivities of metals and the temperature coefficients of resistivity [see also 344].

Table B-36 Resistivity and Temperature Coefficient of Metals

Material	Resistivity (μΩ·cm)	Temperature Coefficient (μΩ·cm/K)
Aluminum	2.6540	0.00429
Gold	2.35	0.004
Silver	1.59	0.0041
Tungsten	5.65	
Titanium	42.0	
Platinum	10.6	0.003927
Palladium	10.8	0.00377
Cobalt	6.24	0.00604
Molybdinum	5.2	
Lead	20.648	0.00336
Iron	9.71	0.00651
Tantalum	12.45	0.00383
Copper	6.24	0.00604
Tin	11.0	0.0047
Nickel	6.84	0.0069

B.13 Optical Properties

The default values for complex index of refraction in Luminous are interpolated from the tables in [63, 124, 180, 237, 242, 365]. Rather than print the tables here, the ranges of optical wavelengths for each material are listed in Table B-37.

Table B-37 Wavelength Ranges for Default Complex Index of Refraction

Material	Temperature (K)	Composition Fraction	Wavelengths (microns)
AlAs	300	NA	0.2213 - 50.0
Alq3	300	NA	0.3 - 0.8
CdS	300	NA	0.2 - 1.6
CIGS	300	YES	0.27 - 1.65
GaAs	300	NA	0.0 - 0.9814
GaN	300	NA	0.372 - 1.167
InSb	300	NA	0.2296 - 6.5
InP	300	NA	0.1689 - 0.975
ITO	300	NA	0.3 - 0.8
NPB	300	NA	0.3 - 0.8
PPV	300	NA	0.3 - 0.8
Poly	300	NA	0.1181 - 18.33
SiO2	300	NA	0.1145 - 1.7614
Silicon	300	NA	0.0103 - 2.0
SnTe	300	NA	0.2 - 1.6
Zn0	300	NA	.349 - .898

Table B-38 show the default parameters for the Sellmeier refractive index model (see Equations 3-688) for various materials.

Table B-38 Default Parameter Values [246] for the Sellmeier Refractive Index Model

Parameter	S0SELL	S1SELL	L1SELL	S2SELL	L2SELL
AlAs	2.616	5.56711	0.2907	0.49252	10.0
AlN	3.14	1.3786	0.1715	3.861	15.0333
GaAs	3.5	7.4969	0.4082	1.9347	37.17526
GaN	3.6	1.75	0.256	4.1	17.86057

Table B-38 Default Parameter Values [246] for the Sellmeier Refractive Index Model					
GaP	3.096	5.99865	0.30725	0.83878	17.32051
GaSb	13.1	0.75464	1.2677	0.68245	10.0
Ge	9.282	6.7288	0.66412	0.21307	62.20932
InAs	11.1	0.71	2.551	2.75	45.6618
InP	7.255	2.316	0.6263	2.756	32.93934
InSb	15.4	0.10425	7.15437	3.47475	44.72136
Si	3.129	8.54297	0.33671	0.00528	38.72983

Table B-39 shows the default parameters for the Adachi refractive index model [1] (see Equations 3-689) for various materials.

Table B-39 Default Parameter Values [246] for the Adachi Refractive Index Model			
Parameter	**AADACHI**	**BADACHI**	**DADACHI**
GaAs	6.3	9.4	0.34
InP	8.4	6.6	0.11
AlAs	25.3	-0.8	0.28
GaSb	4.05	12.66	0.82
AlSB	59.68	-9.53	0.65
InAs	5.14	10.15	0.38
GaP	22.25	0.9	0.08
AlP	24.1	-2.0	0.07
InSb	7.91	13.07	0.81

B.13.1 SOPRA Database

The SOPRA database is a collection of complex refractive index for various materials sometimes as a function of temperature and composition. To access this data, you should specify the file name from Table B-40 using the SOPRA parameter of the **MATERIAL** statement. The SOPRA database is courtesy of SOPRA, a thin film metrology company located world wide (www.sopra-sa.com). The ranges are given in their specified units. Atlas will automatically handle unit conversions.

Table B-40 Contents of SOPRA Database

File	Description	Default
7059.nk	Glass 7059	x
Ag.nk	Silver (0.6eV - 6.6eV)	x
Againp.all	$(Al_xGa_{1-x})_{0.5}In_{0.5}P$ vs composition fraction x	x
Againp0.nk	$Ga_{0.5}In_{0.5}P$ (0.225µm - 1µm)	x
Againp1.nk	$(Al_{0.1}Ga_{0.9})_{0.5}In_{0.5}P$ (0.225µm - 1.0µm)	x
Againp3.nk	$(Al_{0.3}Ga_{0.7})_{0.5}In_{0.5}P$ (0.225µm - 1.0µm)	x
Againp6.nk	$(Al_{0.6}Ga_{0.4})_{0.5}In_{0.5}P$ (0.225µm - 1.0µm)	x
Againp7.nk	$(Al_{0.7}Ga_{0.3})_{0.5}In_{0.5}P$ (0.225µm - 1.0µm)	x
Againp10.nk	$Al_{0.5}In_{0.5}P$ (0.225µm - 1.0µm)	x
Al.nk	Aluminum (0.5eV - 6.75eV)	
Al2o3.nk	Al_2O_3 (Sapphire))(0.25µm - 0.9µm)	x
Al2o3p.nk	Al_2O_3 (Palik) (0.5eV - 6eV)	
Alas.nk	AlAs (0.5eV - 5.6eV)	
Alas.all	AlAs vs temperature from 28K to 626K	x
Alas028t.nk	AlAs 28K (1.255eV - 4.495eV)	x
Alas052t.nk	AlAs 52K (1.255eV - 4.495eV)	x
Alas072t.nk	AlAs 72K (1.255eV - 4.495eV)	x
Alas098t.nk	AlAs 98K (1.255eV - 4.495eV)	x
Alas125t.nk	AlAs 125K (1.255eV - 4.495eV)	x
Alas152t.nk	AlAs 152K (1.255eV - 4.495eV)	x
Alas178t.nk	AlAs 178K (1.255eV - 4.495eV)	x
Alas204t.nk	AlAs 204K (1.255eV - 4.495eV)	x
Alas228t.nk	AlAs 228K (1.255eV - 4.495eV)	x

Table B-40 Contents of SOPRA Database		
Alas305t.nk	AlAs 305K (1.255eV - 4.495eV)	x
Alas331t.nk	AlAs 331K (1.255eV - 4.495eV)	x
Alas361t.nk	AlAs 361K (1.255eV - 4.495eV)	x
Alas390t.nk	AlAs 390K (1.255eV - 4.495eV)	x
Alas421t.nk	AlAs 421K (1.255eV - 4.495eV)	x
Alas445t.nk	AlAs 445K (1.255eV - 4.495eV)	x
Alas469t.nk	AlAs 469K (1.255eV - 4.495eV)	x
Alas499t.nk	AlAs 499K (1.255eV - 4.495eV)	x
Alas527t.nk	AlAs 527K (1.255eV - 4.495eV)	x
Alas552t.nk	AlAs 552K (1.255eV - 4.495eV)	x
Alas578t.nk	AlAs 578K (1.255eV - 4.495eV)	x
Alas602t.nk	AlAs 602K (1.255eV - 4.495eV)	x
Alas626t.nk	AlAs 626K (1.255eV - 4.495eV)	x
Alcu.nk	Aluminum Copper (0.25µm - 0.9µm)	x
Algaas.all	$Al_xGa_{1-x}As$ vs composition fraction x	x
Algaas0.nk	GaAs (0.5eV - 6eV)	x
Algaas1.nk	$Al_{.1}Ga_{.9}As$ (1.5eV - 6eV)	x
Algaas2.nk	$Al_{.2}Ga_{.8}As$ (1.5eV - 6eV)	x
Algaas3.nk	$Al_{.3}Ga_{.7}As$ (1.5eV - 6eV)	x
Algaas4.nk	$Al_{.4}Ga_{.6}As$ (1.5eV - 6eV)	x
Algaas5.nk	$Al_{.5}Ga_{.5}As$ (1.5eV - 6eV)	x
Algaas6.nk	$Al_{.6}Ga_{.4}As$ (1.5eV - 6eV)	x
Algaas7.nk	$Al_{.7}Ga_{.3}As$ (1.5eV - 6eV)	x
Algaas8.nk	$Al_{.8}Ga_{.2}As$ (1.5eV - 6eV)	x
Algaas9.nk	$Al_{.9}Ga_{.1}As$ (1.5eV - 6eV)	x
Algaas10.nk	AlAs (0.5eV - 5.6eV)	x
Alon.nk	Aluminum Oxynitride (0.5eV - 3.1eV)	
Alsb.nk	AlSb (0.5eV - 5.8eV)	x
Alsi.nk	AlSi (0.25µm - 0.8µm)	x

Table B-40 Contents of SOPRA Database		
Alsiti.nk	Aluminum Silicon Titanium (0.25µm - 0.8µm)	
Asi.nk	Aluminum Silicon (0.25µm - 0.8µm)	
Au.nk	Gold (0.25µm - 0.8µm)	
Baf2.nk	BaF_2 (0.2µm - 2µm)	
Bk7.nk	Glass BK7 (1eV - 7eV)	
Bk7_abs.nk	(1eV - 6.5eV)	
Caf2.nk	CaF_2 (1eV - 6.5eV)	
Carbam.nk	Amorphous Carbon (0.3µm - 0.84µm)	
Ccl4.nk	HexaChloromethane (0.25µm - 0.9µm)	
Cdse.nk	CdSe (0.5µm - 6.5µm)	x
Cdte.nk	CdTe (0.5µm - 6.5µm)	x
Co.nk	Co (0.25µm - 0.9µm)	x
Co_2.nk	Cobalt Hexagonal (Thin Flim) (0.65eV - 6.6eV)	
Cor7059.nk	Glass Corning 7059 (0.44µm - 0.65µm)	
Cosi2-4.nk	$CoSi_2$ (1.0eV - 4.8eV)	
Cr.nk	Cr (1.0eV - 6.0eV)	
Cr3si.nk	$CrSi_3$ (0.1eV - 5.9eV)	
Cr5si3.nk	Cr_5Si_3 (0.1eV - 5.9eV)	
Crsi2el2.nk	Chromimum Silicide (11.0eV - 23.1eV)	
Cu.nk	Cu (0.5eV - 6.5eV)	x
Cu2o.nk	Cu_2O (0.5eV - 4.2eV)	
Cuo.nk	CuO (0.5eV - 4.2eV)	
Diam.nk	Diamond (0.5eV - 5.8eV)	x
Fesi2el1.nk	Iron Silicide (4.5eV - 9eV)	
Fesi2el2.nk	Iron Silicide (11.0eV - 23.1eV)	
Fesi2epi.nk	Iron Silicide (0.2eV - 5.0eV)	
Gaas.nk	GaAs (0.5eV - 6eV)	
Gaas100.nk	GaAs 100K	
Gaas111.nk	GaAs 111K (0.234µm -0.84µm)	

Table B-40 Contents of SOPRA Database		
GaAsTemp.ALL	GaAs vs temperature from 31K to 634K	x
Gaas031t.nk	GaAs 31K (1.24eV - 4.495eV)	x
Gaas041t.nk	GaAs 41K (1.24eV - 4.495eV)	x
Gaas060t.nk	GaAs 60K (1.24eV - 4.495eV)	x
Gaas081t.nk	GaAs 81K (1.24eV - 4.495eV)	x
Gaas103t.nk	GaAs 103K (1.24eV - 4.495eV)	x
Gaas126t.nk	GaAs 126K (1.24eV - 4.495eV)	x
Gaas150t.nk	GaAs 150K (1.24eV - 4.495eV)	x
Gaas175t.nk	GaAs 175K (1.24eV - 4.495eV)	x
Gaas199t.nk	GaAs 199K (1.24eV - 4.495eV)	x
Gaas224t.nk	GaAs 224K (1.24eV - 4.495eV)	x
Gaas249t.nk	GaAs 249K (1.24eV - 4.495eV)	x
Gaas273t.nk	GaAs 273K (1.24eV - 4.495eV)	x
Gaas297t.nk	GaAs 297K (1.24eV - 4.495eV)	x
Gaas320t.nk	GaAs 320K (1.24eV - 4.495eV)	x
Gaas344t.nk	GaAs 344K (1.24eV - 4.495eV)	x
Gaas367t.nk	GaAs 367K (1.24eV - 4.495eV)	x
Gaas391t.nk	GaAs 391K (1.24eV - 4.495eV)	x
Gaas415t.nk	GaAs 415K (1.24eV - 4.495eV)	x
Gaas443t.nk	GaAs 443K (1.24eV - 4.495eV)	x
Gaas465t.nk	GaAs 465K (1.24eV - 4.495eV)	x
Gaas488t.nk	GaAs 488K (1.24eV - 4.495eV)	x
Gaas515t.nk	GaAs 515K (1.24eV - 4.495eV)	x
Gaas546t.nk	GaAs 546K (1.24eV - 4.495eV)	x
Gaas579t.nk	GaAs 579K (1.24eV - 4.495eV)	x
Gaas603t.nk	GaAs 603K (1.24eV - 4.495eV)	x
Gaas634t.nk	GaAs 634K (1.24eV - 4.495eV)	x
Gaaso.nk	Gallium Arsenide Oxide (1.0eV - 6.0eV)	
Gaasox.nk	Gallium Arsenide Oxide (1.0eV - 6.0eV)	

Table B-40 Contents of SOPRA Database		
Gan-uv.nk	GaN (0.25µm - 0.8µm)	
Gan01.nk	(1eV - 5eV)	
Gan02.nk	(0.2404368µm - 0.8µm)	
Gan03.nk	(0.25µm - 0.8µm)	
Gan60.nk	(0.3µm - 0.8µm)	
Gan70.nk	(0.3µm - 0.8µm)	
Gap.nk	GaP (0.5eV - 6.0eV)	x
Gap100.nk	GaP (100)	
Gapox.nk	Gallium Phosphide Oxide (1.6eV - 6.0eV)	
Gasb.nk	GaSb (1.25eV - 6.0eV)	x
Gasbox.nk	Gallium Antimonide Oxide (1.6eV - 6.0eV)	
Ge.nk	Ge (0.5eV - 6.0eV)	
Ge100.nk	Ge (100)	
H2o.nk	H_2O (1.5eV - 6.0eV)	
Hfo2.nk	H_FO_2 (0.25µm - 0.9µm)	
Hfsi2.nk	$HFSi_2$ (0.05µm - 9.5µm)	
Hgcdte.all	$Hg_{1-x}Cd_xTe$ as a function of composition fraction x	x
Hgcdte0.nk	HgTe (0.2µm - 0.9µm)	x
Hgcdte2.nk	$Hg_{.8}Cd_{.2}Te$ (0.2µm - 0.9µm)	x
Hgcdte3.nk	$Hg_{.71}Cd_{.29}Te$ (0.2µm - 0.9µm)	x
Inas.nk	InAs (1eV - 6.0eV)	x
Inasox.nk	Indium Arsenide Oxide (1eV - 6eV)	
Ingaas.nk	$In_{0.53}Ga_{0.41}As$ (1.5eV - 6.0eV)	
Ingasb.all	$In_{1-x}Ga_xSb$ as a function of composition fraction x	x
Ingasb0.nk	InSb (0.225µm - 0.825µm)	x
Ingasb1.nk	$In_{.9}Ga_{.1}Sb$ (0.225µm - 0.825µm)	x
Ingasb3.nk	$In_{.7}Ga_{.3}Sb$ (0.225µm - 0.825µm)	x
Ingasb5.nk	$In_{.5}Ga_{.5}Sb$ (0.225µm - 0.825µm)	x
Ingasb7.nk	$In_{.3}Ga_{.7}Sb$ (0.225µm - 0.825µm)	x

Table B-40 Contents of SOPRA Database		
Ingasb9.nk	$In_{.1}Ga_{.9}Sb$ (0.225µm - 0.825µm)	x
Ingasb10.nk	GaSb (0.225µm - 0.825µm)	x
Inp.nk	InP (0.5eV - 6.0eV)	x
Inpox.nk	Indium Phosphide Oxide (1.6eV - 6.0eV)	
Insb.nk	InSb (1.0eV - 6.0eV)	x
Insbox.nk	Indium Antimonide Oxide (1.6eV - 6.0eV)	
Ir.nk	Ir (0.5eV - 6.5eV)	
Ir3si5e.nk	Indium Silicide (4.1eV - 8.2eV)	
Ir3si5p.nk	Indium Silicide (0.6eV - 55eV)	
Ito2.nk	Indium Tin Oxide (0.25µm - 0.85µm)	x
Kcl.nk	KCl (0.5eV - 6.0eV)	
Lasf9.nk	Glass LASF9	
Li.nk	Li	
Lif.nk	LiF (10.5eV - 6.75eV)	
Mgf2.nk	MgF_2 (0.25µm - 0.9µm)	
Mo.nk	Mo (0.5eV - 6.5eV)	x
Mosi2-a.nk	Molybdenuin Silicide (Parallel) (1eV - 4.4eV)	
Mosi2-b.nk	Molybdenuin Silicide (Perpendicular) (0.05eV - 6eV)	
Nbsi-a.nk	Niobuim Silicide (Parallel) (0.05eV - 6eV)	
Nbsi-b.nk	Niobuim Silicide (Perpendicular) (0.05eV - 6eV)	
Ni.nk	Ni (0.05eV - 6.0eV)	x
Ni2si.nk	Nickel Silicide (0.1eV - 5.8eV)	
Ni3si.nk	Nickel Silicide (0.1eV - 5.8eV)	
Nisi.nk	Nickel Silicide (0.5eV - 5.8eV)	x
Os.nk	Os (0.5eV - 6.5eV)	
P_sias.nk	Silicon Arsenide	
P_siud.nk	Polysilicon (heavy As doping)	
Pbs.nk	PbS (0.05eV - 6.2eV)	x
Pbse.nk	PbSe (1.0eV - 6.0eV)	x

Table B-40 Contents of SOPRA Database		
Pd.nk	Pd (0.5eV - 6.5eV)	x
Pd2si-a.nk	Palladium Silicide (Parallel) (0.1eV - 7eV)	
Pd2si-b.nk	Palladium Silicide (Perpendicular) (0.1eV - 7eV)	
Pt.nk	Pt (0.5eV - 6.0eV)	x
Resi1-75.nk	$Si_{.75}Ge_{.25}$ (0.1eV - 12eV)	x
Resige.all	Si_xGe_{1-x} as a function of composition fraction x	x
Resige0.nk	Ge (0.5eV - 7.6eV)	x
Resige1.nk	Si (0.5eV - 6eV)	x
Resige22.nk	$Si_{.22}Ge_{.78}$ (1.7eV - 5.6eV)	x
Resige39.nk	$Si_{.39}Ge_{.61}$ (1.7eV - 5.6eV)	x
Resige51.nk	$Si_{.51}Ge_{.49}$ (1.7eV - 5.6eV)	x
Resige64.nk	$Si_{.64}Ge_{.36}$ (1.7eV - 5.6eV)	x
Resige75.nk	$Si_{.75}Ge_{.25}$ (1.7eV - 5.6eV)	x
Resige83.nk	$Si_{.83}Ge_{.27}$ (1.7eV - 5.6eV)	x
Resige91.nk	$Si_{.91}Ge_{.09}$ (1.7eV - 5.6eV)	x
Rh.nk	Rh (0.5eV - 6.0eV)	
Ringas.all	$In_xGa_{1-x}As$ as a function of composition fraction x	x
Ringas0.nk	GaAs (0.5eV - 6.0eV)	x
Ringas10.nk	$In_{.1}Ga_{.9}As$ (0.24eV - 0.84eV)	x
Ringas20.nk	$In_{.2}Ga_{.8}As$ (1.3eV - 5.1eV)	x
Ringas24.nk	$In_{.24}Ga_{.78}As$ (1.5eV - 5.0eV)	x
Sf11.nk	Glass SF11	
Si100_2.nk	Si (100) (0.23µm - 0.84µm)	
Si110.nk	Si (110) (0.23µm - 0.84µm)	
Si111.nk	Si (111) (0.23µm - 0.84µm)	
Si11ge89.nk	$Si_{0.11}Ge_{.89}$	
Si20ge80.nk	$Si_{0.2}Ge_{0.8}$	
Si28ge72.nk	$Si_{0.88}Ge_{0.72}$	
Si3n4.nk	Si_3N_4 (1.0eV - 8.0eV)	x

Table B-40 Contents of SOPRA Database		
Si65ge35.nk	$Si_{0.35}Ge_{0.35}$	
Si85ge15.nk	$Si_{0.85}Ge_{0.15}$	
Si98ge02.nk	$Si_{0.98}Ge_{0.02}$	
Siam1.nk	Amorphous Si (0.6eV - 6.0eV)	
Siam2.nk	Amorphous Si (1.0eV - 6.0eV)	
Sic.nk	SiC (0.19µm - 1.25µm)	x
Sicr-t.all	Si (0-450C) (1.5eV - 4.7eV)	x
Sicr-t02.nk	Si (20C) (1.5eV - 4.7eV)	x
Sicr-t10.nk	Si (100C) (1.5eV - 4.7eV)	x
Sicr-t15.nk	Si (150C) (1.5eV - 4.7eV)	x
Sicr-t20.nk	Si (200C) (1.5eV - 4.7eV)	x
Sicr-t25.nk	Si (250C) (1.5eV - 4.7eV)	x
Sicr-t30.nk	Si (300C) (1.5eV - 4.7eV)	x
Sicr-t35.nk	Si (350C) (1.5eV - 4.7eV)	x
Sicr-t40.nk	Si (400C) (1.5eV - 4.7eV)	x
Sicr-t45.nk	Si (450C) (1.5eV - 4.7eV)	x
Sicrir.nk	(4.959408×10^{-2}eV - 1.241092eV)	
Sige_ge.nk	SiGe on Ge	
Sige_si.nk	SiGe on Si	
Singas.all	InGaAs (Strained)	
Singas0.nk	GaAs (Strained) (0.24µm - 0.84µm)	
Singas10.nk	$In_{0.1}Ga_{0.9}$AS (Strained) (0.24µm - 0.84µm)	
Singas20.nk	$In_{0.2}Ga_{0.8}$AS (Strained) (0.24µm - 0.84µm)	
Singas24.nk	$In_{0.24}Ga_{0.76}$AS (Strained) (0.24µm - 0.84µm)	
Sio.nk	SiO (0.6eV - 5.9eV)	
Sio2.nk	SiO2 (0.6eV - 7.6eV)	x
Sio2ir.nk	SiO2 (0.0495941eV - 1.386316eV)	
Sion0.nk	SiliconOxynitride (10% N) (0.25µm - 0.8µm)	
Sion20.nk	SiliconOxynitride (20% N) (0.25µm - 0.8µm)	

Table B-40 Contents of SOPRA Database		
Sion40.nk	SiliconOxynitride (40% N) (0.25µm - 0.8µm)	
Sion60.nk	SiliconOxynitride (60% N) (0.25µm - 0.8µm)	
Sion80.nk	SiliconOxynitride (80% N) (0.25µm - 0.8µm)	
Siop.nk	SiO (0.5eV - 6.5eV)	
Sipoly.nk	Polysilicon	x
Sipoly10.nk	(0.5eV - 6.5eV)	
Sipoly20.nk	(0.5eV - 6.5eV)	
Sipoly30.nk	(0.5eV - 6.5eV)	
Sipoly40.nk	(0.5eV - 6.5eV)	
Sipoly50.nk	(0.5eV - 6.5eV)	
Sipoly60.nk	(0.5eV - 6.5eV)	
Sipoly70.nk	(0.5eV - 6.5eV)	
Sipoly80.nk	(0.5eV - 6.5eV)	
Sipoly90.nk	(0.5eV - 6.5eV)	
Sipore.nk	Porous Silicon (1.0eV - 5.9eV)	
Stsg.all	strained Si_xGe_{1-x} verses composition fraction x	
Stsg0.nk	Ge(strained) (2.0eV - 4.5eV)	
Stsg064.nk	$Si_{.064}Ge_{.936}$(strained) (2.0eV - 4.5eV)	
Stsg123.nk	$Si_{.123}Ge_{.877}$(strained) (2.0eV - 4.5eV)	
Stsg169.nk	$Si_{.169}Ge_{.831}$(strained) (2.0eV - 4.5eV)	
Stsg229.nk	$Si_{.229}Ge_{.771}$(strained) (2.0eV - 4.5eV)	
Ta.nk	Ta (0.18µm - 2.48µm)	
Taox1.nk	Tantalum Oxide (0.3µm - 0.85µm)	
Taox2.nk	Tantalum Oxide (0.25µm - 0.85µm)	
Tasi2-a.nk	Tantalum Silicide (Parallel) (0.05eV - 10eV)	
Tasi2-b.nk	Tantalum Silicide (Perpendicular) (0.05eV - 10eV)	
Thf4.nk	ThF_4 (0.65eV - 5.6eV)	
Ti.nk	Ti (1.0eV - 6.0eV)	x
Tini.nk	TiN (0.25µm - 0.9µm)	

Table B-40 Contents of SOPRA Database		
Tio2.nk	TiO_2 (0.18µm - 1.5µm)	
Tio2b.nk	TiO_2 (0.22µm - 2.42µm)	
Tisi-a.nk	TiS_i (0.1eV - 19.5eV)	
V.nk	V (0.5eV - 6.0eV)	
Vsi2-a.nk	VSi_2 (Parallel) (0.05eV - 4.9eV)	
Vsi2-b.nk	VSi_2 (Perpendicular) (0.05eV - 4.9eV)	
W.nk	W (0.05eV - 6.0eV)	
Wsi2-a.nk	WSi_2 (Parallel) (0.1eV - 7.5eV)	
Wsi2-b.nk	WSi_2 (Perpendicular) (0.1eV - 7.5eV)	
Y2o3.nk	Y_2O_3 (0.5eV - 6.0eV)	
Zncdte.all	$Zn_{1-x}Cd_xTe$ as a function of composition fraction x	
Zncdte0.nk	ZnTe (1.1eV - 5.4eV)	
Zncdte1.nk	$Zn_{.9}Cd_{.1}Te$ (1.1eV - 5.4eV)	
Zncdte10.nk	CdTe (1.1eV - 5.4eV)	
Zncdte3.nk	$Zn_{.7}Cd_{.3}Te$ (1.1eV - 5.4eV)	
Zncdte5.nk	$Zn_{.5}Cd_{.5}Te$ (1.1eV - 5.4eV)	
Zncdte7.nk	$Zn_{.3}Cd_{.7}Te$ (1.1eV - 5.4eV)	
Zncdte9.nk	$Zn_{.1}Cd_{.9}Te$ (1.1eV - 5.4eV)	
Znscub.nk	ZnS (Cubic) (1.0eV - 6.0eV)	
Znse.nk	ZnSe (1.0eV - 6.0eV)	x
Znsete.all	$ZnSe_xTe_{1-x}$ as a function of composition fraction x	
Znsete0.nk	ZnTe	x
Znsete1.nk	$ZnSe_{.1}Te_{.9}$ (0.7eV - 6.3eV)	
Znsete10.nk	ZnSe (0.7eV - 6.3eV)	
Znsete3.nk	$ZnSe_{.3}Te_{.7}$ (0.7eV - 6.3eV)	
Znsete5.nk	$ZnSe_{.5}Te_{.5}$ (0.7eV - 6.3eV)	
Znsete7.nk	$ZnSe_{.7}Te_{.3}$ (0.7eV - 6.3eV)	
Znsete9.nk	$ZnSe_{.9}Te_{.1}$ (0.7eV - 6.3eV)	
Zro2.nk	ZrO_2	

Table B-40 Contents of SOPRA Database		
Zrsi2.nk	$ZrSi_2$	
gan-tit.NK		
sin_bf5.NK		

Note: You can add the INDEX.CHECK parameter to the **SOLVE** statement to list the values of real and imaginary index being used in each solution. The indices are only printed when you perform the ray trace at the first reference to a given beam on the **SOLVE** statement.

B.13.2 Silvaco Spectrum Library

Table B-41 Spectrum Files in the Silvaco Spectra Library			
Name	**Samples**	**Reference**	**Description**
Alq3.spc	20	[181]	Alq3 PL spectrum
Alq3_0.spc	25	[182]	Alq3 PL spectrum
Alq3_1.spc	13	[19]	Alq3 PL spectrum
Alq3_3.spc	29	[216]	Alq3 PL spectrum
Alq3_4p2rubrene.spc	32	[216]	Alq3 doped with 4.2% rubrene PL spectrum
Alq3_C545T.spc	21	[181]	Alq3 doped with C545T PL spectrum
Alq3_PtOEP.spc	18	[19]	Alq3 doped with 8% PtOEP PL spectrum
BAlq3.spc	12	[314]	Balq PL spectrum
BCHA-PPV.spc	19	[154]	BCHA-PPV PL spectrum
BDOO-PF.spc	23	[154]	BDOO-PF PL spectrum
BEH-PPV.spc	25	[154]	BEH-PPV PL spectrum
BuEH-PPV.spc	20	[154]	BuEH-PPV PL spectrum
BuEH-PPV_MEH-PPV_1.spc	18	[154]	BuEH-PPV:MEH-PPV copolymer(x=0.1) PL spectrum
BuEH-PPV_MEH-PPV_7.spc	18	[154]	BuEH-PPV:MEH-PPV copolymer(x=0.7) PL spectrum
BuEH-PPV_MEH-PPV_9.spc	17	[154]	BuEH-PPV:MEH-PPV copolymer(x=0.9) PL spectrum

Table B-41 Spectrum Files in the Silvaco Spectra Library			
BuEH-PPV_MEH-PPV_95.spc	20	[154]	BuEH-PPV:MEH-PPV copolymer(x=0.95) PL spectrum
BuEH-PPV_MEH-PPV_975.spc	17	[154]	BuEH-PPV:MEH-PPV copolymer(x=0.975) PL spectrum
CN-PPP.spc	17	[154]	CN-PPP PL spectrum
DCM2Alq3.spc	18	[19]	Alq3 doped with 2% DCM2 PL spectrum
fluorescent.spc	92	[269]	GE Triphosphorus fluorescent SP30 bulb spectrum
HEH-PF.spc	26	[154]	HEH-PF PL spectrum
hgvapor.spc	51	[208]	Example mercury vapor lamp spectrum
incondescent.spc	42	[269]	GE daylight incondescent bulb spectrum
Irppy3.spc	15	[314]	Ir(ppy)3 PL spectrum
Irppy_CBP.spc	30	[314]	Ir(ppy)3 in CBP EL spectrum
Mars_30.spc	46	[79]	Mars solar spectrum 30 degrees from zenith
Mars_60.spc	36	[79]	Mars solar spectrum 60 degrees from zenith
MEH-PPV.spc	21	[154]	MEH-PPV PL spectrum
PPV.spc	25	[208]	PPV PL spectrum
PPV_0.spc	28	[63]	PPV PL spectrum
TPD.spc	30	[216]	TPD PL spectrum
TPD_3p5rubrene	29	[216]	TPD Doped with 3.5 % rubrene PL spectrum

B.14 User Defined Materials

The current version of Atlas doesn't directly support user-defined materials. A simple workaround can be done using the already existing user specifications. This workaround is based on the use of an already existing material name and modifying the material parameters as appropriate.

In Atlas, material names are defined to give you a reasonable set of default material parameters. You can override any of these defaults using the **MATERIAL**, **IMPACT**, **MODELS**, and **MOBILITY** statements.

The key to defining new materials is to choose a material name defined in Atlas. Then, modify the material parameters of that material to match the user material. Here, you should choose a material that has default parameter values that might best match the user material, while making sure to choose a material that is not already in the user device. Then, associate this material name with the device regions where the new material is present. To do this, either specify the chosen material name on the appropriate **REGION** statements (when the device is defined in the Atlas syntax) or choose the material name from the materials menu when defining the region in DevEdit. Next, modify the material statements using **MATERIAL**, **IMPACT**, **MOBILITY**, and **MODELS** statements. When doing this, the MATERIAL parameter of the given statement should be assigned to the chosen material name.

For materials with variations in composition fraction, choose a defined material with X and/or Y composition fractions (i.e., a ternary or quaternary material). You may also find it convenient to use C interpreter functions to define the material parameters as a function of composition.

The C interpreter functions that are useful for this approach are: F.MUNSAT, F.MUPSAT, F.BANDCOMP, F.VSATN, F.VSATP, F.RECOMB, F.INDEX, F.BGN, F.CONMUN, F.CONMUP, F.COPT, F.TAUN, F.TAUP, F.GAUN, and F.GAUP.

When defining new materials, there is a minimum set of parameters that should be defined. This set includes bandgap (EG300), electron and hole density of states (NC300 and NV300), dielectric permittivity (PERMITIVITY), and electron and hole mobilities (MUN and MUP). For bipolar devices, certain recombination parameters should also be defined such as: lifetimes (TAUN and TAUP), radiative recombination rates (COPT), and Auger coefficients (AUGN and AUGP).

For devices with variations in material composition, certain band-edge alignment parameters should also be defined: either electron affinity (AFFINITY) or edge alignment (ALIGN). See Section 6.1.3 "Temperature Dependence of Electron Affinity" if you're using F.BANDCOMP to specify a lattice temperature dependent AFFINITY. If impact ionization is considered, the impact ionization coefficients should also be defined.

As an example, consider the case where a device is simulating with an AlInGaP region. In Table B-1, we see that this material system is not defined in Atlas. We then choose a material that's defined in Atlas, which has default material parameters that best approximate the material parameters of the new material. In this case, we choose InGaAsP since, at least for this example, we feel that this material is closest to the AlInGaP. Next, we must specify InGaAsP as the material of the region(s) that is/are composed of AlInGaP. This can be done either on the REGION statement if the structure is defined in Atlas syntax or from the material menu when the region is defined in DevEdit.

Supposing that we're satisfied with the default values of the parameters from the minimum set discussed above and that we're principally concerned with the recombination and heat flow parameters defaults, the following section of the input deck illustrates how these parameter defaults can be modified:

```
# new material AlInGaP
MATERIAL MATERIAL=InGaAsP
# SRH
MATERIAL  MATERIAL=InGaAsP TAUN0=1.1e-9 TAUP0=2.3e-8
# Auger
MATERIAL MATERIAL=InGaAsP AUGN=5.8e-30 AUGP=1.1e-31
# Optical
material material=InGaAsP COPT=1.7e-30
# Thermoconductivity
MATERIAL MATERIAL=InGaAsP TC.A=2.49
# Heat capacity
MATERIAL  MATERIAL=InGaAsP HC.A=1.9
```

Appendix C
RF and Small Signal AC Parameter Extraction

When performing small signal analysis, you can choose to capture various RF analysis parameters [330] to the log file along with the default capacitance and conductance values. These include *Y, Z, S, ABCD,* and *H* parameters and certain gain parameters including *GUmax* (unilateral power gain), *GTmax* (maximum unilateral transducer power gain), *k* (stability factor), and *h21* (current gain in dB). In all cases, these are calculated as two port parameters.

To include any of these parameters, set the appropriate parameters of the LOG statement (see Section 22.30 “LOG”).

All of the RF analysis parameters are calculated from the small signal capacitance and conductance values provided during small signal analysis (see Sections 2.9.3 “Small-Signal AC Solutions”, 2.10.3 “Parameter Extraction In DeckBuild”, and 21.10 “Small Signal and Large Signal Analysis”).

An intermediate step in calculation of most of the RF parameters is the calculation of the *Y* parameters. Equations C-1 through C-8 describe the relationship between small signal capacitances and conductances and the complex *Y* parameters.

$$Re(Y_{11}) = G_{11} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-1}$$

$$Im(Y_{11}) = 2\pi f C_{11} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-2}$$

$$Re(Y_{12}) = G_{12} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-3}$$

$$Im(Y_{12}) = 2\pi f C_{12} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-4}$$

$$Re(Y_{21}) = G_{21} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-5}$$

$$Im(Y_{21}) = 2\pi f C_{21} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-6}$$

$$Re(Y_{22}) = G_{22} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-7}$$

$$Im(Y_{22}) = 2\pi f C_{22} \cdot \mathtt{IMPEDANCE} \cdot \mathtt{WIDTH} \quad \text{C-8}$$

Here, IMPEDANCE is the matching impedance in ohms specified on the LOG statement. WIDTH is the device width in microns defined on the MESH statement. *f* is the frequency in Hz. C_{nm} is the small signal capacitance between ports *n* and *m*. *Gnm* is the small signal conductance between ports *n* and *m*. *Ynm* is the complex *Y* parameter between ports *n* and *m*. *Re()* indicates the real part of the argument. *Im( )* indicates the imaginary part of the argument.

The complex *S* parameters are calculated from the complex *Y* parameters as given in Equations C-9 through C-13.

$$\Delta_1 = \frac{1}{(1 + Y_{11})(1 + Y_{22}) - Y_{12}Y_{21}} \quad \text{C-9}$$

$$S_{11} = [(1 - Y_{11})(1 + Y_{22}) + Y_{12}Y_{21}]\Delta_1 \quad \text{C-10}$$

$$S_{12} - 2Y_{12}\Delta_1 \quad \text{C-11}$$

$$S_{21} - 2Y_{21}\Delta_1 \quad \text{C-12}$$

$$S_{22} = [(1 + Y_{11})(1 - Y_{22}) + Y_{12}Y_{21}]\Delta_1 \quad \text{C-13}$$

During post-processing, you can choose to apply lumped resistances and inductances to the calculation of the complex *S* parameters. In the first step, Atlas is programmed to convert the *S* parameters into complex *Z* parameters.

Equations C-14 through C-18 describe the conversion from *S* parameters to *Z* parameters.

$$\Delta_2 = \frac{1}{(1 - S_{11})(1 - S_{22}) - S_{12}S_{21}} \quad \text{C-14}$$

$$Z_{11} = [(1 + S_{11})(1 - S_{22}) + S_{12}S_{21}]\Delta_2 \quad \text{C-15}$$

$$Z_{12} = 2S_{12}\Delta_2 \quad \text{C-16}$$

$$Z_{21} = 2S_{21}\Delta_2 \quad \text{C-17}$$

$$Z_{22} = [(1 - S_{11})(1 + S_{22}) + S_{12}S_{21}]\Delta_2 \quad \text{C-18}$$

Once you have *Z* parameters, you can apply the lumped resistances and inductances as described in Equations C-19 through C-26.

$$Re(Z'_{11}) = Re(Z_{11}) + (\mathtt{RIN} + \mathtt{RCOM})/\mathtt{IMPEDANCE} \quad \text{C-19}$$

$$I_m(Z'_{11}) = I_m(Z_{11}) + 2\pi f(\mathtt{LIN} + \mathtt{LCOM})/\mathtt{IMPEDANCE} \quad \text{C-20}$$

$$Re(Z'_{12}) = Re(Z_{12}) + \mathtt{RCOM}/\mathtt{IMPEDANCE} \quad \text{C-21}$$

$$I_m(Z'_{12}) = I_m(Z_{12}) + 2\pi f\mathtt{LCOM}/\mathtt{IMPEDANCE} \quad \text{C-22}$$

$$Re(Z'_{21}) = Re(Z_{21}) + \mathtt{RCOM}/\mathtt{IMPEDANCE} \quad \text{C-23}$$

$$I_m(Z'_{21}) = I_m(Z_{21}) + 2\pi f\mathtt{LCOM}/\mathtt{IMPEDANCE} \quad \text{C-24}$$

$$Re(Z_{22}^{'}) = Re(Z_{22}) + (\texttt{ROUT} + \texttt{RCOM})/\texttt{IMPEDANCE} \quad \text{C-25}$$

$$I_m(Z_{22}^{'}) = I_m(Z_{22}) + 2\pi f(\texttt{LOUT} + \texttt{LCOM})/\texttt{IMPEDANCE} \quad \text{C-26}$$

Here, RCOM, LCOM, RIN, LIN, ROUT and LOUT are the common, input and output resistances and inductances defined on the LOG statement.

Once you get the updated Z parameters, you can convert them back into S parameters as described in Equations C-27 through C-31.

$$\Delta_3 = \frac{1}{(1+Z_{11})(1+Z_{22}) - Z_{12}Z_{21}} \quad \text{C-27}$$

$$S_{11} = [(Z_{11}-1)(Z_{22}+1) - Z_{12}Z_{21}]\Delta_3 \quad \text{C-28}$$

$$S_{12} = 2Z_{12}\Delta_3 \quad \text{C-29}$$

$$S_{21} = 2Z_{21}\Delta_3 \quad \text{C-30}$$

$$S_{22} = [(Z_{11}+1)(Z_{22}-1) - Z_{12}Z_{21}]\Delta_3 \quad \text{C-31}$$

When not involving the application of lumped resistance and inductance to the S parameters, the Z parameters are calculated directly from the Y parameters as described by Equations C-32 through C-36.

$$\Delta_4 = \frac{1}{(Y_{11}Y_{22} - Y_{12}Y_{21})} \quad \text{C-32}$$

$$Z_{11} = Y_{22}\Delta_4 \quad \text{C-33}$$

$$Z_{12} = -Y_{12}\Delta_4 \quad \text{C-34}$$

$$Z_{21} = -Y_{21}\Delta_4 \quad \text{C-35}$$

$$Z_{22} = Y_{11}\Delta_4 \quad \text{C-36}$$

The H parameters can also be calculated directly from the Y parameters as described in Equations C-37 through C-40.

$$H_{11} = 1/Y_{11} \quad \text{C-37}$$

$$H_{11} = -Y_{12}/Y_{11} \quad \text{C-38}$$

$$H_{21} = Y_{21}/Y_{11} \quad \text{C-39}$$

$$H_{22} = (Y_{11}Y_{22} - Y_{12}Y_{21})/Y_{11} \quad \text{C-40}$$

The ABCD parameters are also directly calculated from the *Y* parameters as described by Equations C-41 through C-44.

$$A = -Y_{22}/Y_{21} \quad \text{C-41}$$

$$B = -1/Y_{21} \quad \text{C-42}$$

$$C = -(Y_{11}Y_{22} - Y_{12}Y_{21})/Y_{21} \quad \text{C-43}$$

$$D = -Y_{11}/Y_{21} \quad \text{C-44}$$

The gain parameters *k*, *GUmax*, *GTmax* and h_{21} are calculated from the *S* parameters as given by Equations C-45 through C-50.

$$\Delta_5 = S_{11}S_{22} - S_{12}S_{21} \quad \text{C-45}$$

$$k = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + \Delta_5^2}{2|S_{12}||S_{21}|} \quad \text{C-46}$$

$$GUmax = \frac{0.5|S_{21}/S_{12} - 1|^2}{k|S_{21}/S_{12}| - Re(S_{21}/S_{12})} \quad \text{C-47}$$

$$GTmax = \frac{|S_{21}|^2}{(1 - |S_{11}|^2)(1 - |S_{22}|^2)} \quad \text{C-48}$$

$$\Delta_6 = \frac{-2|S_{21}|^2}{(1 - S_{11})(1 + S_{22}) + S_{12}S_{21}} \quad \text{C-49}$$

$$h_{21} = 20\log_{10}(|\Delta_6|) \quad \text{C-50}$$

Appendix D
MC Device Files

D.1 The Default Configuration File: defaults.in

```
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
# This is the default config file for the current version of moca.
# All parameters that make sense are assigned reasonable default
values
# (e.g., temp = 300K  cut=(1,0,0)  etc).
# All others are set to illegal values (such as -1 or 0).
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@

#------------------------------------------------------------------
--------
# General simulation parameters
# If maxwell=yes and restart=yes, positions are read from the
# the restart file, but momentums are given a Maxwellian
distribution.
#------------------------------------------------------------------
--------

algo mode=-1 carrier=e dt=2e-15 iter=0 trans=0 restart=no
seed=4212345 \
  maxwell=no

reflect mindist=-1E0

debug npick=0

output scatoutfile="scat.out" summaryoutfile="summary.out" \
  difflogfile="diff.log" currentlogfile="current.log" \
                                    difframplogfile="diff_ramp.log"
currentramplogfile="current_ramp.log" \
  solstrfile="sol.str" \
  tstep=-1 xstep=(1,1,1) estep=50 restart=-1 wpstep=pinf \
  init=no hydro=no hist=yes qeta=no outfiles=no

#------------------------------------------------------------------
--------
```

```
# dz [=] cm.  If dz > 0.0, device width is fixed to this value,
# otherwise, if conc is set device width is set based on conc
# otherwise, width is set to achieve N particles using initial
guess
# for particle concentration
#-----------------------------------------------------------------
--------
particle n=0 nnorm=0 conc=0.0 dz=-1.0


# The threshold energy for DoS-dominated phonon scattering is fixed
at
# 1.5eV, i.e., sufficiently above the L valley to account properly
for
# XL scattering but still more or less within the validity of the
k.p
# nonparabolic approximation for the first valley.
#
# bf eth=1.70, where non-parabolic dos and full-band dos do not
differ
# bf eth=0.63, where non-parabolic dos and full-band dos do not
differ
# bf eth=0.57, to get agreement with average over measured ii-
coeff. (Kane)
# bf eth=0.42, to get agreement with lowest measured ii-coeff.
(Cartier)
phon eth=0.42 model=0 file=”dos_c.in”


#-----------------------------------------------------------------
--------
# Interpolation of energy from k, using bandstructure tables
#
# 0 = most accurate cubic interpolation
# For strain model (when using “mmat type=1”):
#    Include the quad correction for energy and velocity WITHOUT
USING
#   interpolated fitting parameters for energy/velocity as
#   a function of mole fraction.
#
# 1 = less accurate linear interpolation
# For strain model (when using “mmat type=1”):
```

```
#   Include the quad correction for energy and velocity USING
#   interpolated fitting parameters for energy/velocity as
#   a function of mole fraction.
#
# 2 = more accurate linear interpolation
# For strain model (when using “mmat type=1”):
#   Include the quad correction for energy and velocity USING
#   interpolated fitting parameters for energy/velocity as
#   a function of mole fraction.
#
#-----------------------------------------------------------------
--------

bands interp=0 n=2 nk=40 file=”eder40_2.in”
# The setttings for the n, nk, and file parameters were moved
# from the bands2d.elec.in file to here.
#  n=(2,3)
#  nk=(40,40)
#  interp=(0,0)
#  file=(“eder40_2.in”,”eder40_3v.in”)

qbands interp=0 n=1 nk=100 file=”qbandtest”
# The settings for the n, nk, and file parameters were moved
# from the bands2d.elec.in file to here.
# The file parameter on the qbands statement is not presently used.

#-----------------------------------------------------------------
--------
# Bulk, 1-D, and 2-D simulation parameters
#-----------------------------------------------------------------
--------



mat cut=(1.0,0.0,0.0) temp=300.0

bulk field=(0.0,0.0) doping=0.0 conc=0.0 memory=no mmfrac=0.0
```

```
oned   mode=1   size=(0.0,0.0,0.0)   nx=(100,0,0)   conc=(0.0,0.0)
reflect1d=(no,no) vbias=0.0



enhan mode=0 tstep=-1 estep=100 xstep=(0,0,0) \

   damp=0.0  lothre=1e-20  hithre=1e20  relthre=0.05  ratio=2.0  n=10
nfrac=0.95


#-------------------------------------------------------------------
--------

#traj: If mode is not 0, then particles will be initialized to be
uniformly

#          distributed  in  the  BOUND  box.   If  mode=2,  detailed
information

#        will be stored about particle trajectories.  If mode=1,
detailed

#      will not be stored.  If ZEROSCAT is not 0, all scattering

#      in the simulation will be disabled.

#-------------------------------------------------------------------
--------

traj        mode=0        zeroscat=no        bound=(0.0,0.0,0.0,0.0)
boundp=(0.0,0.0,0.0,0.0)


tunnel mode=0


excit energy=0.0 shape=0 width=0.0 rate=0.0 alpha=(0.0,0.0,0.0)


#-------------------------------------------------------------------
--------

#cutoff: maximum cutoff for doping if complete ionization is used.

#-------------------------------------------------------------------
--------

poisson   tstep=5   iter=100   tol=(2e-4,1e-6)   ioniz=no   start=1
freeze=pinf \

  linpoi=no cutoff=1e25 ipoly=no hideplasmons=no


#-------------------------------------------------------------------
--------

#negfield: If yes, getfield, will produce negative of the field
calculated
```

```
#             from the potential.  Useful in a hole frozen field
simulation,
#          where potential is read in from an electron simulation.
#-------------------------------------------------------------------
--------
field maxfield=(-1.0,-1.0) ramp=(-1,-1) negfield=no usebarr=no

quantum restart=no option=(no,no) decay=(100e-15,2000e-15) \
  minconc=(1e10,1e10) maxqv=-1.0 maxqf=-1.0 strength=1.0 \
  q1d=(-1,-1,-1) freeze=pinf sigma=8e-8 radius=3 ladders=(1,0,0) \
  energy=no ttemp=no

import type=8 volt=false nconc=false pconc=false dop=false \
  avgvolt=false avgnconc=false avgpconc=false \
  mesh=false regions=false \
  dipsh=no adddop=no bound=(0,0,0,0) \
  boundp=(0.0,0.0,0.0,0.0) \
  voltfile="volt.in" concfile="conc.in" dopfile="dop.in" \
  gridfile="grid_filename" solfile="sol_filename" \
  verfile="ver_filename" trifile="tri_filename" \
  sdfile="sd_filename" retrofile="retro_filename" \
  strfile="str_filename"

#-------------------------------------------------------------------
--------
# Physical models
#
# Phonon-scattering parameters from Tang and Hess, JAP 54:5139
# bf threshold in bands.elec.in is set so that xl vanishes
# this fairly reproduces the drift velocity versus field charac-
# teristics at 300 and 77K.
#-------------------------------------------------------------------
--------

xxf temp=(220.0,550.0,685.0) defo=(3e7,2e8,2e8)

xxg temp=(140.0,215.0,720.0) defo=(5e7,8e7,11e8)
```

```
xl temp=(672.0,634.0,480.0,197.0) defo=(2e8,2e8,2e8,2e8)


ac defo=9.0


#-------------------------------------------------------------------
--------
# Parameters for all four impact-ionization models
# (default=Cartier, then Cartier et al, APL 62(25):3339).
# model=Cartier
# final=no:          Sets all 3 particles after ii event to zero
energy.
# weight=yes:        Doubles the weight of the primary impacting
particle.
# sec=yes:        Generates a secondary particle equal in weight to
#                  primary.
# csec=yes:       Generates a complementary secondary particle for
slave
#                  process.
# randsec=yes:    Secondary particles are randomly placed in rspos
region.
# rspos=(min max): If randsec=yes, specifies minimum and maximum
radii (cm)
#                 of a ring-shaped region in which secondaries are
placed.
#                  The ring is centered about the location of ii.
#-------------------------------------------------------------------
--------


impact          model=cartier          param=(6.25e10,3e12,6.8e14)
eth=(1.2,1.8,3.45) \
  final=yes weight=no sec=no csec=no start=1000 randsec=no \
  rspos=(0.0,0.0)


#-------------------------------------------------------------------
--------
# default scale = 1
# bf scale = 1.9 to get agreement with the experimental doping
dependent
# low-field mobility of majority carriers
```

```
#-------------------------------------------------------------------
--------
ridley tstep=-1 scale=1.9


#-------------------------------------------------------------------
--------
# Parameters for analytic surface scattering model
# For more info, see
# J. Appl. Phys., vol. 79, Jan 1996, p. 911
#
# In some calibration tests, reducing SEXPL to 11.0e-10 reduced
rolloff
# in mobility at high fields and gave a better fit to the universal
# mobility curves from Tagaki: Trans. on ED, 41, december 1992, p.
2357
#-------------------------------------------------------------------
--------


ssparam      sexpl=22.0e-10      sexpd=1.78e-10      ssfield=-1.0
file=”ssphonon.dat” \
        surfphon=(0.0,0.0)
# surfphon=(7e10,0.34)
# Values above were used in a bulk example.


#-------------------------------------------------------------------
--------
# Extra crossing estimators (energy distributions as a function of
space)
# are calculated for the crossest region (ignored for crossreg = 0)
#-------------------------------------------------------------------
--------


crossest crossreg=0


#-------------------------------------------------------------------
--------
# Pipe sets up interprocess communication which lets an electron
and a hole
# simulation exchange secondary particles created by impact
ionization.
```

```
# If mode = master, the simulation proceeds as usual except that
secondaries
# are exchanged.  If mode=slave, the particles move in fields which
are
# received from the master.  The communcation occurs in the pipe
files
# designated by msfile and smfile.  These files must be specified
exactly
# the same in both the master and slave command files.  If initnlin
= yes,
# non-linear poisson is solved once at the beginning of the
simulation to
# determine the initial concentration of slave particles.  If
initnlin=no,
# the concentration is set by charge neurality.  initnlin must also
be
# specified the same in slave and master command files.
#-----------------------------------------------------------------
--------

bipolar comm=none msfile="msfile" smfile="smfile" nslave=0
nfracii=0.5 initnlin=yes


#-----------------------------------------------------------------
--------
#INJECT block controls injection from contact
#
#  dn: fraction of mesh to inject in normal direction
#  dp: fraction of mesh to inject in parallel direction
#    hemimax: if no, inject carriers on a full maxwellian
distribution
#  overinj: Maintain overinj fraction more carriers at interface
than
#           required by charge neutrality
#-----------------------------------------------------------------
--------
inject dn=0.0 dp=1.0 hemimax=no overinj=1e4 tstep=-1


#-----------------------------------------------------------------
--------
# The treatment of broadening is a bit incomplete and inconsistent.
```

```
# Also, bulk simulations do not appear to be sensitive to cutoff,
so,
# play it safe and turn off broadening. (cutoff = 0.0)
#-----------------------------------------------------------------
--------
final cutoff=0.0 n=178152 nk=100 format=0 file="eord100_2.in"
# The setting for n, nk, format, and file parameter were moved
# from # bands2d.elec.in to here.
#  NK= (100,100)
#  N=(178152,267228)
#  FORMAT=(0,0)
#  FILE=("eord100_2.in","eord100_3v.in")

#-----------------------------------------------------------------
--------
# Fullband numerical scattering rates
#-----------------------------------------------------------------
--------

escat type=analytic fbproc=7 fbband=4 anproc=24 anband=1 \
  file="scatfl_c_uni.in" \
  phncplfile="phcpl_c.in" \
  phnfacfile="phon_fac_c_uni_300k.in" \
  plotfile="scateng_c.in"

hscat type=analytic fbproc=7 fbband=3 anproc=24 anband=1 \
  file="scatfl_v.in" \
  phncplfile="phcpl_v.in" \
  phnfacfile="phon_fac_v.in" \
  plotfile="scateng_v.in"

#-----------------------------------------------------------------
--------
# Multimaterial options
#-----------------------------------------------------------------
--------

mmat type=no
```

```
#-------------------------------------------------------------------
--------
# Multimaterial for electrons for strained silicon
#-------------------------------------------------------------------
--------

ssielec band="eder40_c_ssi_interp.in" \
  rate="rates_ssi_c_45_uni.in" phncpl="coupl_ssi_c.in" \
  latt="latts_ssi_c.in"

units length=um

#-------------------------------------------------------------------
--------
# Material definition for 2-D simulation
#-------------------------------------------------------------------
--------

#Note: Hard-coded associations exist (in mat.*) for materials
#       with n=1, 2, 3, 4, 5, 11, and 12.  These should not be
changed
#      to relate to different materials than their names below.
matdef n=1 name="Air"   eps= 1.0 barrier=0.0  rough=0.0 \
       egap=1.1245 affinity=4.17
matdef n=2 name="Si"    eps=11.7 barrier=0.0  rough=0.0 \
       egap=1.1245 affinity=4.17
matdef n=3 name="Poly"  eps=11.7 barrier=0.0  rough=0.0 \
       egap=1.1245 affinity=4.17
# For SiO2, use barrier=4.0 eV instead of barrier=3.15 for hole
simulation.
# (For example, when simulating a p-MOSFET with "ALGO CARRIER=H".)
#matdef n=4 name="SiO2"  eps= 3.9 barrier=3.15 rough=0.10 \
#       egap=1.1245 affinity=4.17
matdef n=4 name="SiO2"  eps= 3.9 barrier=3.15 rough=0.25 \
       egap=8.9000 affinity=1.02
matdef n=5 name="Si3N4" eps= 7.5 barrier=3.15 rough=0.0 \
       egap=1.1245 affinity=1.02
```

```
matdef n=6 name="Al"    eps= 1.0 barrier=0.69 rough=0.0 \
       egap=1.1245 affinity=3.48
matdef n=7 name="Pt"    eps= 1.0 barrier=0.85 rough=0.0 \
       egap=1.1245 affinity=3.32
matdef n=8 name="W"     eps= 1.0 barrier=0.65 rough=0.0 \
       egap=1.1245 affinity=3.52

#use barrier=0.34? for hole simulation
matdef n=9 name="Au"    eps= 1.0 barrier=0.79 rough=0.0 \
       egap=1.1245 affinity=3.38

# I will leave the two definitions below commented for now.
# Note that these already exist in the hard codes mentioned above.
#matdef n=11 name="SSi"  eps=11.7 barrier=0.0  rough=0.0 \
#       egap=1.1245 affinity=4.17
#matdef n=12 name="SiGe" eps=11.7 barrier=0.0  rough=0.0 \
#       egap=1.1245 affinity=4.17
```

D.2 File Formats

This appendix provides a description of input and output file formats used by MC Device. By default, MC Device writes `diff.log` files for bulk simulations and `current.log` and `sol.str` files for 2D simulations. Section D.2.1 “Silvaco Output Files” describes these file formats.

Many output files are written using the UIUC MOCA file formats. These output files use the suffix `.out` are only written when the `OUTFILES` parameter on the **`OUTPUT`** statement is set to `1|TRUE|YES`. By default, the `OUTFILES` parameter in the **`OUTPUT`** section is set to 0 and these output files are suppressed. In addition to the `OUTFILES` parameter, some `*.out` files have additional boolean control parameters (see Section 20.3.4 “Commonly-Used Statements”). Sections D.2.2 “General Output Files” through Section D.2.6 “Miscellaneous Input And Output Files” describe the UIUC MOCA output files.

Some input files use UIUC MOCA file formats. These input files use the suffix `.in` and have the same format as the corresponding UIUC MOCA output (`.out`) file.

Many files written in the UIUC MOCA file format consist of three columns. The first two columns are the x and y positions (in cm) and the third column is data field associated with that file in cgs units. You can see from the x and y values that the rows (all values of *y*) are scanned for each column (each value of *x* with y fixed) and then for each row (each value of y). Note that data (x and y values and data field) may be written at the nodes of the mesh (for potential and MC carrier concentration) or at the element centers (for the doping). Quantities are averaged over iterations greater than or equal to `ISTART` (or time>=0). All output files are in ASCII format.

D.2.1 Silvaco Output Files

`*diff*.log` files with this extended suffix contain several time-dependent time-averaged quantities for a bulk simulation. They are as follows:

- Time = [s].
- Average Position = [cm].
- Variance of Position = [cm^2].
- Diffusion Coefficient = [cm^2/s].
- Diffusion-Derived Mobility = [$cm^2/V/s$].
- Drift-derived Mobility = [$cm^2/V/s$].

Here is the format of the columns:

```
Time     <X(t)>     <X(t)^2>-<X>^2     Diff_Coeff(t)     Mobility_D(t)
Mobility_E(t)
```

You can change this filename with the `DIFFLOGFILE` parameter on the **`OUTPUT`** statement (see Section 20.3.4 “Commonly-Used Statements”). We recommend you use the extended file suffix `diff.log` (as shown in the examples) to keep this filename consistent with this documentation.

Note: The first asterisk in the file name above respresents the base file name. For example, `mcdeviceex01_diff.log`. The second asterisk in the file name above represents a possible ramp step. For example, `mcdeviceex01_diff_field=1000.log`.

`*diff_ramp.log` files with this extended suffix contain several time-averaged quantities for a bulk simulation for different bias voltages, which typically represent the steady-state solution for each bias voltage. The quantities are the same as those for `*diff*.log` but exclude the Time. Instead, they include the Electric Field = [V/cm].

You can change this filename with the `DIFFRAMPLOGFILE` parameter on the **`OUTPUT`** statement (see Section 20.3.4 “Commonly-Used Statements”). We recommend you use the extended file suffix `diff.log` (as shown in the examples) to keep this filename consistent with this documentation.

`*current*.log` files with this extended suffix contain time-dependent time-averaged and space-averaged currents over `CREGION`s defined in the input file. They are as follows:

- Time = [s].
- Current = [A/cm]. The sign convension is given at the end of the current regions section. See “Current Regions” on page 972.
- Velocity = [cm/s].
- Concentration-weighted force = [V/cm].
- Mobility = [$cm^2/V/s$].

Each current, velocity, force, and mobility has first an x-directed, then a y-directed column denoted by `dim=1` and `dim=2` respectively. Each current, velocity, force, and mobility is also written for each current region and denoted by `creg=1` up to `creg=n`, where `n` is the number of current regions.

Here is the format of the columns:

```
Time   Current(t)   Velocity(t)   Force(t)   Mobility(t)
```

You can change this filename with the `CURRENTLOGFILE` parameter on the **`OUTPUT`** statement (see Section 20.3.4 “Commonly-Used Statements”). We recommend you use the extended file suffix `current.log` (as shown in the examples) to keep this filename consistent with this documentation.

Note: The first asterisk in the file name above respresents the base file name. For example, `mcdeviceex02_current.log`. The second asterisk in the file name above represents a possible ramp step. For example, `mcdeviceex02_current_vdrain=1.log`.

`*current_ramp.log` files with this extended suffix contain several time-averaged and space-averaged currents over **`CREGION`**s defined in the input file for different bias voltages, which typically represent the steady-state solution for each bias voltage. The quantities are the same as those for `*current*`.log but exclude the Time. Instead, they include the contact voltage as `vcontact_name = [V]`, where `contact_name` represents the name of the contacted for which the voltage is ramped. For example, `vdrain`.

You can change this filename with the CURRENTRAMPLOGFILE parameter on the **OUTPUT** statement (see Section 20.3.4 “Commonly-Used Statements”). We recommend you use the extended file suffix current_ramp.log (as shown in the examples) to keep this filename consistent with this documentation.

***sol*.str** files contain quantities for a 2D simulation. They are as follows:

- Arsenic = [cm^{-3}] (when DOP=0 on IMPORT or IONIZ=1 on POISSON)
- Band gap = [eV]
- Boron = [cm^{-3}] (when DOP=0 on IMPORT or IONIZ=1 on POISSON)
- Conduction Band DOS = [cm^{-3}]
- Conudction Band Energy = [eV]
- e- Energy = [eV] (when CARRIER=E on ALGO)
- e- Velocity X = [cm/s]
- e- Velocity Y = [cm/s]
- E Field X = [V/cm]
- E Field Y = [V/cm]
- (EHP) Generation Rate = [s^{-1}]
- Electron Conc = [cm-3]
- Electron QFL = [eV]
- h+ Energy = [eV] (when CARRIER=H on ALGO)
- h+ Velocity X = [cm/s]
- h+ Velocity Y = [cm/s]
- Hole Conc = [cm-3]
- Hole QFL = [eV]
- Impact (Ionization) Gen(eration) Rate = [cm^{-3}/s]
- Indium = [cm^{-3}] (when DOP=0 on IMPORT or IONIZ=1 on POISSON)
- Instant Electron Conc = [cm^{-3}]
- Instant Hole Conc = [cm^{-3}
- Instant Potential = [V]
- Net Doping = -Boron + Phosphorus - Indium + Arsenic = [cm^{-3}]
- Phosphorus = [cm^{-3}] (when DOP=0 on IMPORT or IONIZ=1 on POISSON)
- Potential = [V]
- Relative Permittivity = [-]
- Total Doping = B + P + In + As = [cm^{-3}] (when DOP=0 on IMPORT or IONIZ=1 on POISSON)
- Valence Band DOS = [cm^{-3}]
- Valence Band Energy = [eV]

All quantitites above beginning with “Average” are time averages from time=0. These quantities are also Monte Carlo carrier-ensemble averages from the MC carriers, which resided in the rectangular box centered about each mesh node.

You can change this filename with the SOLSTRFILE parameter on the **OUTPUT** statement (see Section 20.3.4 "Commonly-Used Statements"). We recommend you use the extended file suffix sol.str (as shown in the examples) to keep this filename consistent with this documentation.

Note: The first asterisk in the file name above respresents the base file name. For example, mcdeviceex02_sol.str. The second asterisk in the file name above represents a possible ramp step. For example, mcdeviceex02_sol_vdrain=1.str.

D.2.2 General Output Files

diff.out contains several time-dependent time-averaged quantities for a bulk simulation. They are as follows:

- Time = [s].
- Average Position = [cm].
- Variance of Position = [cm^2].
- Diffusion Coefficient = [cm^2/s].
- Diffusion-Derived Mobility = [$cm^2/V/s$].
- Drift-derived Mobility = [$cm^2/V/s$].

Here is the format of the columns:

```
Time    <X(t)>    <X(t)^2>-<X>^2      Diff_Coeff(t)      Mobility_D(t)
Mobility_E(t)
```

sol.out contains several time-averaged quantities for a 2D simulation. They are as follows:

- X/Y = [cm].
- Potential = [V].
- Force = [V/cm].
- Particle concentration = [cm^{-3}].
- Velocity = [cm/s].
- Energy = [eV].
- Impact Ionization Rate = [cm^{-3}/s].
- Electron-Hole Pair (EHP) generation rate = [s^{-1}].

Force is the electric field for holes and its negative for electrons. Force and velocity each have two components: the first is x-directed, the second is y-directed.

Here is the format of the columns:

```
X    Y    Potential    Force    Concentration    Velocity    Energy
Generation
```

xhist.out, **yhist.out** contain the energy distribution along slices in the X/Y directions. They are as follows:

- Energy = [eV].
- X/Y slice = [cm].
- Distribution = [$cm^{-3}eV^{-1}$].

Here is the format of the columns:

```
Energy   Position  Distribution
```

volt.out contains the instantaneous electric potential in V. You can import the contents of this file by copying it to another file (i.e., volt.in) and using an **IMPORT** statement. Importing the contents of this file may be useful when restarting a simulation or when starting with a better initial guess. The electric potential is written at each node in the mesh. Here is the format of the columns:

```
X   Y   Potential
```

avgvolt.out contains the time-averaged (for time>=0) electric potential in V. You can import the contents of this file by copying it to another file (i.e., volt.in) and using an IMPORT statement. Importing the contents of this file may be useful when performing simulations with a fixed electric potential (fixed field). The format of this file is the same as volt.out.

init*volt.out contains the instantaneous electric potential in V. Here * varies from 1 to 5 at different points during initialization. These files can be helpful to understand problems associated with initialization. The format of this file is the same as volt.out.

conc.out contains the instantaneous MC carrier concentration in cm^{-3}. You can import the contents of this file by copying it to another file (i.e., conc.in) and using an **IMPORT** statement. Importing the contents of this file may be useful when restarting a simulation or when starting with a better initial guess. The MC carrier concentration is written at each node in the mesh. Here is the format of the columns:

```
X   Y   Concentration
```

avgconc.out contains the time-averaged (for time>=0) MC carrier concentration in cm^{-3}. You can import the contents of this file by copying it to another file (i.e., conc.in) and using an **IMPORT** statement. Importing the contents of this file may be useful when starting with a better initial guess. The format of this file is the same as conc.out.

init*conc.out contains the instantaneous MC carrier concentration in cm^{-3}. Here * varies from 1 to 5 at different points during initialization. These files can be helpful to understand problems associated with initialization. The format of this file is the same as conc.out.

dop.out contains the doping in the device. They are as follows:

- X/Y = [cm].
- Doping = [cm^{-3}].
- Total = -B + P - In + As.

Here is the format of the columns:

```
X   Y   Total   B   P   In   As
```

status.out contains the running statistics of every particle in the ensemble. The first two columns are the X and Y coordinates in [cm]. The third column is the band index of the particle. The next three columns are the components of k. The seventh column is the Monte Carlo particle weight. The last two columns are the total energy and quantum corrected energy in [eV].

Here is the format of the columns:

```
X   Y   band_num   kx   ky   kz   weight   Etot   Eq
```

xaxis.out, **yaxis.out** contain the coordinates of the grid lines in the *X* and *Y* directions. The last column is the half-way point between grid point i and i+1. Coordinate = [cm].

Here is the format of the columns:

```
grid_point    X    X(1/2)
```

runstats.out contains the running statistics of particles in the simulation. Here are the following statistics for every 5000 iterations:

- number of particles injected
- number of scattering events which occurred
- number of reflections during that iteration

Here is the format of the columns.

```
iteration    particles    scatterings    reflections
```

***summary.out** files with this extended suffix contain all MC Device input related to an MC solve. You can change this filename with the SUMMARYOUTFILE parameter on the **OUTPUT** statement. We recommend you use the extended file suffix summary.out (as shown in the examples) to keep this filename consistent with this documentation. The *summary.out file will contain a summary of the input used for the last solve statement in your input file.

D.2.3 Current Output Files

current.out contains the time-averaged and space-averaged over “cregion” boxes defined in the command file. They are as follows:

- Cregion = #.
- Samples = #.
- Time = [s].
- Current = [A/cm].
- Velocity = [cm/s].
- Concentration-weighted force = [V/cm].
- Mobility = [cm^2/V/s].

Each current, velocity, force and mobility has first an x-directed then a y-directed column. Here is the format of the columns:

```
Cregion   Samples  Time   Current   Velocity   Force   Mobility
```

To extract this into a file, which covers only a single cregion, you can use a statement such as the following.

```
 awk  '{if ($1 == 1) print $3, $4, $5, $6, $7, $8}' current.out >
cbox1.out
```

creg.out contains physical properties of each current estimator region. n is the cregion number. z is the normalized width of the device. x and y are the length and depth of the cregion. Here is the format of the columns:

```
cbox  n   z   x   y
```

isub.out shows the impact ionization currents after the transient period has elapsed. The first column is the iteration number, the next column is the impact ionization current, and the last column is generation current. Here is the format of the columns:

```
iteration   II_current   generation_current
```

isub2.out contains the averaged impact ionization currents. The first line is the device width used. The second line has the impact ionization current. The second quantity is generation current.

D.2.4 Scattering Output Files

impact.out contains the running impact ionization statistics. The first column is the iteration during which the event occurred. The second column is the weight of the particle before impact ionization. The next two columns are the x and y coordinates of the particle before scattering in [cm]. The last two columns are the energies of the particle before and after the event in [eV]. Here is the format of the columns:

```
iter   weight   X   Y   Eb   Ea
```

scatmesh.out contains several time averaged scattering quantities. First, X/Y [cm]. The remaining columns are scattering rates.

***scat.out** files with this extended suffix contain the scattering rate table. See "Using Scattering Parameters" on page 1007 for more information. You can change this filename with the SCATOUTFILE parameter on the **OUTPUT** statement. We recommend you use the extended file suffix scat.out to keep this filename consistent with this documentation.

sec.out contains the information about impact ionization secondary particles. The file contains the iteration number, weight of the particle, X/Y in [cm], and the last column is energy of the secondary particle in [eV].

```
iteration   weight   X   Y   energy
```

serest.out contains the scattering estimator output. The first column is the number of the scattering process and the remaining columns are the scattering rates in each of the different defined **SEREGION**s. The last two scattering processes are inelastic and elastic empirical scattering respectively. For a description on the remaining processes, see Section 20.7.5 "Phonon Scattering".

```
process_num   rate1   rate2   ...   rateN
```

ssrate.out contains surface scattering estimator output. The first column is the number of the ssregion region. The second column is the surface scattering iteration number. The third column is a factor. The last column is the region-dependent scattering rate.

```
region_num   iteration   ???   rate
```

surfphon.out contains surface phonon prefactor.

trackscat.out contains tracking information on the scattering in a specified tsregion. The first line is the total scattering rate in the chosen region. The remaining lines are the scattering from each process.

D.2.5 Quantum Output Files

calcsp2moca.out contains the transformation between the MC Device solution grid and the Schrödinger grid. This file is only output if the quantum correction is turned on. The first column is the MC Device grid point. The second column is the quantum correction grid point that falls closest to the MC Device grid point. The third column is the relative distance of the SP grid point between MC grid point i and i+1. The last two columns are the real-space values of the quantum correction grid and the MC Device grid.

Here is the format of the columns:

```
MC_grid   SP_grid   R   SP_dist   MC_dist
```

etabarr.out, etabarr1.out show the Eta barrier for each mesh area. The Eta barrier for each mesh area is taken to be the average of the eta barriers of the materials on each of the 4 sides of the mesh. The file is written as

```
for y=ymin:ymax {
   for x=xmin:xmax {
      write etabarr
   }
}
```

spgrid.out contains the definition of Schrödinger-Poisson grid. It is the same as with xaxis.out. The first column is the grid point index. The second column is the real-space coordinate in [cm].

```
grid_point   X
```

qatten.out contains quantum correction attenuation factor through the whole device traversing it one y-grid line at a time.

qcap.out contains quantum solution for simulation of a MOS capacitor with the same bounds as in qregion. The first two columns are the X and Y coordinates respectively, in Å. The next two columns are carrier concentration and energy respectively. Here is the format of the columns:

```
X   Y   Concentration   Energy
```

qhist.out contains quantum correction histogram. The first column is the energy bin in [eV] and the second is the energy histogram of particles in the quantum region.

qrestart.out shows the transient quantum correction particle status. The quantum Eta for each grid point in the device structure and traversed y-grid line by line.

qsol.out contains quantum correction throughout the entire device in [eV]. The results are written for each increasing y-grid line.

qsol.trans.out has exactly the same format as qsol.out but is the transient solution.

qstatus.out contains the quantum correction particle status. The quantum Eta for each grid point in the device structure, traversed y-grid line by line.

qvolt.out contains quantum corrected quantum potential. This may be used as an input by copying it to qvolt.in and using the **IMPORT** statement.

ttemp.out contains transverse temperature in [K] for the transverse grid lines. First column is the grid line number. The third column is the actual transverse temperature. Columns 2 and 4 are fitting parameters for the Marquardt fit method. Here is the format of the columns:

```
Grid_Line    Param1    Transverse_Temp    Param2
```

D.2.6 Miscellaneous Input And Output Files

inj_2.in contains parameters where you tell MC Device how you want to simulate a Schottky barrier device. bar_poisson must equal the value of the silicide barrier height set in your input command (*.in) file. hqflreg is typically set to the substrate contact region. It is best not to modify any of the parameters relating to the Airy functions.

balance.out contains the variance reduction balancing grid. The file contains the size of the balancing grid and the number of energy bins.

oxide0x.out contains the oxide injection currents for the various current estimator regions.

crosshist.out, **xcross.out**, **ycross.out** contains tracked energies of particles that change regions.

fieldeff.out contains the effective electric field in a surface scattering region. The field is written along the x grid points only. The field is given in [V/cm].

geom.out contains the boundaries [cm] of all regions in the order they were loaded or defined in the input file. Unlike the *summary.out file (see Section D.2.2 "General Output Files"), which only contains region definitions in the input file, geom.out contains all region definitions including those obtained by importing the structure.

mat.out contains the location of the cell centers [cm], the relative permittivity [-], the band gap [eV], the density of states in the conduction band [cm^{-3}], the density of states in the valence band [cm^{-3}], and the quasi-Fermi potential [V]. The quasi-Fermi potential is used to calculate the density of non-MC carriers. It is also used to calculate the ionized doping concentration when using incomplete ionization.

mesh.out contains the node location [cm], mesh areas ([cm^2] for 2D) inside cells of type MC (with a negative average area where it would be 0), the Poisson boundary conditions [V], and the Poisson boundary conditions types where 0=Dirichlet, 1=Neumann, 2=Interior.

D.2.7 Band Structure Input Files

Three types of band structure files are used in building the data files used in the MC simulation: band structure (used for high-resolution energy calculations), gradients, and eigenstates. Figure D-1 shows the formats. The files are located in the `$MCDEVICEDIR` directory and are named `eder40_*.in`.

a.

k_x	k_y	k_z	E_0	E_1	...	E_{N_b-1}
.						
.						
.						

b.

k_x	k_y	k_z	$\frac{\partial E_0}{\partial x}$	...	$\frac{\partial E_{N_b-1}}{\partial x}$	...	$\frac{\partial E_0}{\partial z}$	...	$\frac{\partial E_{N_b-1}}{\partial z}$
.									
.									
.									

c.

k_x	k_y	k_z	E_0	...	E_{N_b-1}
c_0^0	c_0^1	...			$c_0^{N_c}$
.					
.					
.					

Figure D-1: Formats of the band structure files: a. energies, b. gradients, c. eigenstates.

The final state is chosen based on final energy. To make this task efficient, the bandstructure tables are reordered according to energy. Figure D-2 shows the format. The files are in the `$MCDEVICEDIR` directory and are named `reord100_*.in`.

k_x	k_y	k_z	E	n
.				
.				
.				

Figure D-2: Format of the final-state table.

There is also a density of states table for electrons and holes. These files are located in the same directory as the bandstructure files and are named `dos_c.in` for the conduction band and `dos_v.in` for the valence band. The format of these files is a single column of probabilities of occupation discretized over 100 pts.

D.3 Examples

This appendix contains several example input files for MC Device. These include Section D.3.1 “Bulk n-type Silicon: mcdeviceex01”, Section D.3.2 “A 25-nm n-MOSFET: mcdeviceex02”, Section D.3.3 “A 25-nm n-MOSFET with Quantum Correction: mcdeviceex03”, Section D.3.4 “A 25-nm p-MOSFET: mcdeviceex04”, and Section D.3.5 “An imported 25-nm n-MOSFET: mcdeviceex05”. The example input files below and the sample output and TonyPlot set files (for each output file) may be accessed with DeckBuild (see Section 20.3.6 “Accessing The Examples”).

D.3.1 Bulk n-type Silicon: mcdeviceex01

```
go atlas
mcdevice
# Bulk simulation for electron transport in silicon at STP
algo mode=0 carrier=e iter=10000 dt=1e-15
# bulk simulation
bulk field=(5e2,5e2)
#bulk field=(1e3,1e3)
# write output
output tstep=1000 \
  difflogfile    =”mcdeviceex01_diff.log” \
  difframplogfile=”mcdeviceex01_diff_ramp.log” \
  summaryoutfile =”mcdeviceex01_summary.out”
# Set the number of particles (i.e. electrons).
particle n=10000

# Do a single solve at field=(5e2,5e2) V/cm.
solve
tonyplot mcdeviceex01_diff.log -set mcdeviceex01_diff_log.set

# Do a 1-step field ramp from field=5e2 V/cm to field=1e3 V/cm.
# with uniform spacing on a linear scale.
#solve fstep=5e2 ffinal=1e3
#tonyplot              mcdeviceex01_diff_ramp.log              -set
mcdeviceex01_diff_ramp_log.set

# Do a 3-step field ramp from field=1e3 V/cm to field=1e6 V/cm.
# with uniform spacing on a logarithmic scale.
#bulk field=(1e3,1e3)
```

```
#solve nsteps=3 ffinal=1e6 uselog=1
#tonyplot              mcdeviceex01_diff_ramp.log              -set
mcdeviceex01_diff_ramp_log.set


# Do a second solve at field=(1e3,1e3) V/cm.
#output difflogfile="mcdeviceex01_diff2.log"
#bulk field=(1e3,1e3)
#solve
#tonyplot mcdeviceex01_diff2.log -set mcdeviceex01_diff_log.set


quit
```

D.3.2 A 25-nm n-MOSFET: mcdeviceex02

```
go atlas
mcdevice
# 25-nm MOSFET definition
# see http://www-mtl.mit.edu/researchgroups/Well/
# MIT's "well-tempered" MOSFET
# Reduce from iter=200000 to iter=2500 to make a short test.
# Note that this test will not have fully reached staty-state at
iter=2500.
algo mode=2 carrier=e iter=2500 dt=0.1e-15
poisson tstep=5
#Turn off enhancement for now!
#enhan   mode=2   tstep=20   estep=300   xstep=(1,1,0)   hithre=10.0
nfrac=0.8
# save output at regular intervals
# (-1 will default to a reasonable value)
output tstep=500 init=1 \
  currentlogfile    ="mcdeviceex02_current.log" \
  currentramplogfile="mcdeviceex02_current_ramp.log" \
  solstrfile        ="mcdeviceex02_sol.str" \
  summaryoutfile    ="mcdeviceex02_summary.out"
# Set the maximum number of particles (i.e. electrons).
particle n=40000
# X Mesh Lines
xmesh node=1   loc=0.0000
xmesh node=45  loc=0.0225 ratio=1.00
xmesh node=145 loc=0.0725 ratio=1.00
xmesh node=189 loc=0.0950 ratio=1.00
# Y Mesh Lines
ymesh node=1  loc=-0.0615
ymesh node=6  loc=-0.0015 ratio=0.400
ymesh node=11 loc= 0.0000 ratio=0.800
ymesh node=90 loc= 0.0800 ratio=1.035
# Simulation Box
region n=1 mat=SiO2 type=out     boundp=(0.0,0.095,-0.0615,0.08)
# Substrate
```

```
region n=2 mat=Si   type=mc      boundp=(0.0,0.095,0.0,0.08)
# Gate oxide
region n=3 mat=SiO2 type=block   boundp=(0.0,0.095,-0.0015,0.0)
# Source contact
region n=4 mat=Si   type=contact boundp=(0.0,0.0026,0.0026,0.0073)
\
       name=”source”
# Drain contact
region        n=5        mat=Si                        type=contact
boundp=(0.0924,0.095,0.0026,0.0073) \
       name=”drain” usefermi=1
# Poly gate
region  n=6  mat=Poly  type=contact  boundp=(0.0225,0.0725,-0.0615,-
0.0015) \
       name=”gate”
# Substrate contact
region n=7 mat=Si   type=contact boundp=(0.0,0.095,0.0771,0.08) \
       name=”substrate”
# calculate current in cregions
cregion boundp=(0.0275,0.0675,0.0,0.08)
cregion boundp=(0.0325,0.0475,0.0,0.080)
cregion boundp=(0.0475,0.0625,0.0,0.08)
## background
doping dopant=B conc=1e15 boundp=(0.0000,0.0950,0.0,0.0800)
## source
doping dopant=As conc=2e20 \
  boundp=(0.0000,0.0293,0.0,      0.0001) \
  char =(0.0040,0.0040,0.0001,0.0170)
## source halo
doping dopant=B conc=1e19 \
  boundp=(0.0000,0.0293,0.0179,0.0181) \
  char =(0.0182,0.0182,0.0160,0.0160)
## drain
doping dopant=As conc=2e20 \
  boundp=(0.0657,0.0950,0.0,      0.0001) \
  char =(0.0040,0.0040,0.0001,0.0170)
## drain halo
```

```
doping dopant=B conc=1e19 \
  boundp=(0.0657,0.0950,0.0179,0.0181) \
  char =(0.0182,0.0182,0.0160,0.0160)
## gate
doping dopant=As conc=5e20 \
  boundp=(0.0225,0.0725,-0.0615,-0.0015)
#Uncomment to use surface scattering in this region
#ssregion boundp=(0.0153,0.0355,0.0,0.0006)
#Make rough=0.0 for the Si-SiO2 interface when using ssREGION
#matdef n=4 name="SiO2" eps=3.9 barrier=3.15 rough=0.0

# Do a single solve at vgate=vdrain=1.0 V.
solve vgate=1.0 vdrain=1.0
tonyplot               mcdeviceex02_current.log               -set
mcdeviceex02_current_log.set
tonyplot mcdeviceex02_sol.str -set mcdeviceex02_sol_str.set

# Do a 3-step voltage ramp from vdrain=3.0 V to vdrain=0.0 V.
#solve vgate=1.0 vdrain=3.0 name="drain" vfinal=0.0 vstep=-1

# Do a 3-step voltage ramp from vdrain=0 V to vdrain=3 V.
#solve vgate=1.0 vdrain=0.0 name="drain" vfinal=3.0 vstep=1.0

# Do a longer 5-step voltage ramp from vdrain=5.0 V to vdrain=0.0
V.
#algo iter=7500
#solve vgate=1.0 vdrain=5.0 name="drain" vfinal=0.0 vstep=-1

# Do a second solve at Vgate=vdrain=2.0 V.
#output currentlogfile="mcdeviceex02_current2.log" \
#       solstrfile    ="mcdeviceex02_sol2.str"
#solve vgate=2.0 vdrain=2.0
#tonyplot              mcdeviceex02_current2.log              -set
mcdeviceex02_current_log.set
#tonyplot mcdeviceex02_sol2.str -set mcdeviceex02_sol_str.set

quit
```

D.3.3 A 25-nm *n*-MOSFET with Quantum Correction: mcdeviceex03

```
go atlas
mcdevice
# This n-MOSFET example with quantum correction
# was created by adding input associated with the quantum
# correction model to the n-MOSFET example (mcdeviceex02).
# Please see comments in mcdeviceex02.in for a description
# of the key parameter settings.
algo mode=2 carrier=e iter=2500 dt=0.1e-15
poisson tstep=2
output restart=-1 outfiles=no init=1 \
  currentlogfile    =”mcdeviceex03_current.log” \
  currentramplogfile=”mcdeviceex03_current_ramp.log” \
  solstrfile        =”mcdeviceex03_sol.str” \
  summaryoutfile    =”mcdeviceex03_summary.out” \
  tstep=1000
particle n=60000
# X Mesh Lines
xmesh node=1   loc=0.0000
xmesh node=76  loc=0.0225 ratio=1.00
xmesh node=243 loc=0.0725 ratio=1.00
xmesh node=318 loc=0.0950 ratio=1.00
# Y Mesh Lines
ymesh node=1  loc=-0.0615
ymesh node=6  loc=-0.0015 ratio=0.4000
ymesh node=9  loc=-0.0008 ratio=0.4000
ymesh node=10 loc= 0.0000 ratio=0.8000
ymesh node=15 loc= 0.0005 ratio=1.0000
ymesh node=89 loc= 0.0800 ratio=1.0508
# Simulation Box
region n=1 mat=SiO2 type=out     boundp=(0.0,0.095,-0.0615,0.08)
# Substrate
region n=2 mat=Si   type=mc      boundp=(0.0,0.095,0.0,0.08)
# Gate oxide
region n=3 mat=SiO2 type=block   boundp=(0.0,0.095,-0.0015,0.0)
# Source contact
```

```
region n=4 mat=Si   type=contact boundp=(0.0,0.0072,0.0033,0.0135)
\
       name=”source”
# Drain contact
region        n=5        mat=Si                        type=contact
boundp=(0.0878,0.095,0.0033,0.0135) \
       name=”drain” usefermi=1
# Poly gate
region  n=6  mat=Poly  type=contact  boundp=(0.0225,0.0725,-0.0615,-
0.0015) \
       name=”gate”
# Substrate contact
region n=7 mat=Si type=contact   boundp=(0.0,0.095,0.0761,0.08) \
       name=”substrate”
# calculate current in cregions
cregion boundp=(0.0285,0.0665,0.0,0.08)
cregion boundp=(0.0357,0.0506,0.0,0.08)
cregion boundp=(0.0447,0.0596,0.0,0.08)
# use quantum correction
qregion     mode=schroedinger     dir=y     tstep=4000     xstep=1
boundp=(0.0,0.095,-0.0008,0.0026)
# Regions for quantum correction
schrregion n=1 mat=SiO2 boundp=(-0.0015,0.0000)
schrregion n=2 mat=Si   boundp=( 0.0000,0.0300)
#schroedinger mesh
schrmesh node=1  loc=-0.0015
schrmesh node=11 loc= 0.0000 ratio=0.84
schrmesh node=41 loc= 0.0050 ratio=1.069
schrmesh node=71 loc= 0.0300 ratio=1.043
#does nothing, set boundaries equal to device boundaries
ebregion boundp=(0.0,0.095,-0.0615,0.0178)
#surface scattering region
#ssregion boundp=(0.0225,0.0725,-0.0008,0.0014)
#Make rough = 0.0 for interface when using ssregion
#matdef N=4 name=”SiO2” eps=3.9 barrier=3.15 rough=0.0
## background
doping dopant=B conc=1e15 \
```

```
  boundp=(0.0000,0.0950,0.0,0.0800)
## source
doping dopant=As conc=2e20 \
  boundp=(0.0000,0.0293,0.0,      0.0001) \
  char =(0.0040,0.0040,0.0001,0.0170)
## source halo
doping dopant=B conc=2.5e19 \
  boundp=(0.0000,0.0293,0.0179,0.0181) \
  char =(0.0182,0.0182,0.0160,0.0160)
## drain
doping dopant=As conc=2e20 \
  boundp=(0.0657,0.0950,0.0,      0.0001) \
  char =(0.0040,0.0040,0.0001,0.0170)
## drain halo
doping dopant=B conc=2.5e19 \
  boundp=(0.0657,0.0950,0.0179,0.0181) \
  char =(0.0182,0.0182,0.0160,0.0160)
## gate
doping dopant=As conc=2e20 \
  boundp=(0.0225,0.0725,-0.0615,-0.0015)

# Do a single solve for vgate=vdrain=1.0 V.
solve vgate=1.0 vdrain=1.0
tonyplot               mcdeviceex03_current.log               -set
mcdeviceex03_current_log.set
tonyplot mcdeviceex03_sol.str -set mcdeviceex03_sol_str.set

# Do a 1-step voltage ramp from vdrain=1.0 V to vdrain=2.0 V.
#solve vgate=1.0 vdrain=1.0 name=”drain” vfinal=2.0 vstep=1.0

# Do a second solve at Vgate=Vdrain=2.0 V.
#output currentlogfile=”mcdeviceex03_current2.log” \
#       solstrfile    =”mcdeviceex03_sol2.str”
#solve vgate=2.0 vdrain=2.0
#tonyplot              mcdeviceex03_current2.log              -set
mcdeviceex03_current_log.set
#tonyplot mcdeviceex03_sol2.str -set mcdeviceex03_sol_str.set
```

```
quit
```

D.3.4 A 25-nm *p*-MOSFET: mcdeviceex04

```
go atlas
mcdevice
# This p-MOSFET example was created by changing the type
# of all dopants and the polarity of all applied biases for the
# the n-MOSFET example (mcdeviceex02).  Please see comments
# in mcdeviceex02.in for a description of key parameter settings.
algo mode=2 carrier=h iter=2500 dt=0.1e-15
poisson tstep=5 linpoi=no
output outfiles=no tstep=500 \
  currentlogfile    =”mcdeviceex04_current.log” \
  currentramplogfile=”mcdeviceex04_current_ramp.log” \
  solstrfile        =”mcdeviceex04_sol.str” \
  summaryoutfile    =”mcdeviceex04_summary.out”
particle n=40000
# X Mesh Lines
xmesh node=1   loc=0.0000
xmesh node=45  loc=0.0225 ratio=1.00
xmesh node=145 loc=0.0725 ratio=1.00
xmesh node=189 loc=0.0950 ratio=1.00
# Y Mesh Lines
ymesh node=1   loc=-0.0615
ymesh node=6   loc=-0.0015 ratio=0.400
ymesh node=11  loc =0.0000 ratio=0.800
ymesh node=90  loc =0.0800 ratio=1.035
# Simulation Box
region n=1 mat=SiO2 type=out     boundp=(0.0,0.095,-0.0615,0.08)
# Substrate
region n=2 mat=Si   type=mc      boundp=(0.0,0.095,0.0,0.08)
# Gate oxide
region n=3 mat=SiO2 type=block   boundp=(0.0,0.095,-0.0015,0.0)
# Source contact
region n=4 mat=Si    type=contact boundp=(0.0,0.0026,0.0026,0.0073)
\
  name=”source”
# drain contact
```

```
region        n=5        mat=Si                          type=contact
boundp=(0.0924,0.095,0.0026,0.0073) \
  name="drain" usefermi=1
# Poly gate
region  n=6  mat=Poly  type=contact  boundp=(0.0225,0.0725,-0.0615,-
0.0015) \
  name="gate"
# Substrate contact
region n=7 mat=Si  type=contact  boundp=(0.0,0.095,0.0771,0.08) \
  name="substrate"
# calculate current in cregions
cregion boundp=(0.0275,0.0675,0.0,0.08)
cregion boundp=(0.0325,0.0475,0.0,0.080)
cregion boundp=(0.0475,0.0625,0.0,0.08)
## background
doping dopant=As conc=1e15 \
  boundp=(0.0000,0.0950,0.0,0.0800)
## source
doping dopant=B conc=2e20 \
  boundp=(0.0000,0.0293,0.0,      0.0001) \
  char  =(0.0040,0.0040,0.0001,0.0170)
## source halo
doping dopant=As conc=1e19 \
  boundp=(0.0000,0.0293,0.0179,0.0181) \
  char  =(0.0182,0.0182,0.0160,0.0160)
## drain
doping dopant=B conc=2e20 \
  boundp=(0.0657,0.0950,0.0,      0.0001) \
  char  =(0.0040,0.0040,0.0001,0.0170)
## drain halo
doping dopant=As conc=1e19 \
  boundp=(0.0657,0.0950,0.0179,0.0181) \
  char =(0.0182,0.0182,0.0160,0.0160)
## gate
doping dopant=B conc=5e20 \
  boundp=(0.0225,0.0725,-0.0615,-0.0015)
#Uncomment to use surface scattering in this region
```

```
##ssregion bound=(31,71,11,14)
#ssregion boundp=(0.0153,0.0355,0.0e+0,0.0006)
#Make rough=0.0 for the Si-SiO2 interface when using ssregion
#matdef n=4 name="SiO2" eps=3.9 barrier=3.15 rough=0.0
# These settings are for MC holes in unstrained silicon.
phon eth=1.5 model=0 file="dos_v.in"
bands n=3 nk=40 file="eder40_3v_so.in"
final n=267228 nk=100 format=0 file="eord100_3v_so.in"

# Do a singe solve at Vgate=vdrain=-1.0 V.
solve vgate=-1.0 vdrain=-1.0
tonyplot               mcdeviceex04_current.log               -set
mcdeviceex04_current_log.set
tonyplot mcdeviceex04_sol.str -set mcdeviceex04_sol_str.set

# Do a 1-step voltage ramp from vdrain=-1 V to vdrain=-2 V.
#solve vgate=-1.0 vdrain=-1.0 name="drain" vfinal=-2.0 vstep=-1.0

# Do a second solve at Vgate=vdrain=-2.0 V.
#output currentlogfile="mcdeviceex04_current2.log" \
#       solstrfile    ="mcdeviceex04_sol2.str"
#solve vgate=-2.0 vdrain=-2.0
#tonyplot              mcdeviceex04_current2.log              -set
mcdeviceex04_current_log.set
#tonyplot mcdeviceex04_sol2.str -set mcdeviceex04_sol_str.set

quit
```

D.3.5 An imported 25-nm *n*-MOSFET: mcdeviceex05

```
go atlas
mcdevice
algo     mode=2 iter=2500 dt=0.1e-15
import   mesh=true regions=true dop=true \
         strfile="mcdeviceex05_atlas.str"
particle n=40000
region   name="source"    type=contact
region   name="drain"     type=contact usefermi=1
region   name="gate"      type=contact
region   name="substrate" type=contact
cregion  boundp=(0.0275,0.0675,0.0,0.08)
cregion  boundp=(0.0325,0.0475,0.0,0.08)
cregion  boundp=(0.0475,0.0625,0.0,0.08)
output   tstep=500 \
         currentlogfile    ="mcdeviceex05_current.log" \
         currentramplogfile="mcdeviceex05_current_ramp.log" \
         solstrfile        ="mcdeviceex05_sol.str" \
         summaryoutfile    ="mcdeviceex05_summary.out"

# Do a single solve at Vgate=vdrain=1.0 V.
solve    vgate=1.0 vdrain=1.0
tonyplot               mcdeviceex05_current.log               -set
mcdeviceex05_current_log.set
tonyplot mcdeviceex05_sol.str -set mcdeviceex05_sol_str.set

# Do a 1-step voltage ramp from vdrain=1.0 V to vdrain=2.0 V.
#solve   vgate=1.0 vdrain=1.0 name="drain" vfinal=2.0 vstep=1.0

# Do a second solve at Vgate=Vdrain=2.0 V.
#output   currentlogfile="mcdeviceex05_current2.log" \
#         solstrfile    ="mcdeviceex05_sol2.str"
#solve    vgate=2.0 vdrain=2.0
#tonyplot              mcdeviceex05_current2.log              -set
mcdeviceex05_current_log.set
#tonyplot mcdeviceex05_sol2.str -set mcdeviceex05_sol_str.set
```

```
quit
```

1. Adachi, S. *Physical Properties of III-V Semiconductor Compounds InP, InAs, GaAs, GaP, InGaAs, and InGaAsP.* New York: John Wiley and Sons, 1992.
2. Adachi, S., "Band gaps and refractive indices of AlGaAsSb, GaInAsSb, and InPAsSb: Key properties for a variety of the 2-4 um optoelectronic device applicatios", *J. Appl. Phys.*, Vol. 61, No. 10, 15 (May 1987): 4896-4876.
3. Adamsone A.I., and B.S. Polsky, "3D Numerical Simulation of Transient Processes in Semiconductor Devices", *COMPEL—The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* Vol. 10, No. 3 (1991): 129-139.
4. Akira Hiroki, S. Odinaka et. al., "A Mobility Model for Submicrometer MOSFET Simulations, including Hot-Carrier Induced Device Degradation", *IEEE Trans. Electron Devices* Vol. 35 (1988): 1487-1493.
5. Alamo, J. del, S. Swirhun and R.M. Swanson, "Simultaneous measuring of hole lifetime, hole mobility and bandgap narrowing in heavily doped n-type silicon", *IEDM Technical Digest*, (December 1985): 290-293.
6. Albrecht, J.D. et al, "Electron transport characteristics of GaN for high temperature device modeling", *J. Appl. Phys.* Vol. 83 (1998): 4777-4781.
7. Allegretto, W., A.Nathan and Henry Baltes, "Numerical Analysis of Magnetic Field Sensitive Bipolar Devices", *IEEE Trans. on Computer-Aided Design*, Vol. 10, No. 4 (1991): 501-511.
8. Ambacher, O. et. al., "Two Dimensional Electron Gases Induced by Spontaneous and Piezoelectric Polarization in Undoped and Doped AlGaN/GaN Heterosturctures", *J. Appl. Phys.* Vol. 87, No. 1, (1 Jan. 2000): 334-344.
9. Anderson, D.G., "Iterative Procedure for Nonlinear Integral Equations", *ACM Journal* Vol. 12, No. 4 (n.d.): 547-560.
10. Andrews, J.M., and M.P. Lepselter, "Reverse Current-Voltage Characteristics of Metal-Silicide Schottky Diodes", *Solid-State Electronics* 13 (1970): 1011-1023.
11. Antogneti, P., and G. Massobrio. *Semiconductor Device Modeling With SPICE*. McGraw-Hill Book Company, 1987.
12. Apanovich Y. et. al, "Numerical Simulation of Submicrometer Devices, Including Coupled Non-local Transport and Non-isothermal Effects," *IEEE Trans. Electron Devices* Vol. 42, No. 5 (1995): 890-898.
13. Apanovich Y. et. al, "Stready-State and Transient Analysis of Submircon Devices Using Energy Balance and Simplified Hydrodyamic Models," *IEEE Trans. Comp. Aided Design of Integrated Circuits and Systems* Vol. 13, No. 6 (June 1994): 702-710.
14. Arkhipov, V.I., P. Heremans, E.V. Emelianova, G.J. Adriaenssens, H. Bassler, "Charge carrier mobility in doping disordered organic semiconductors", *Journal of Non-Crystalline Solids*, Vol. 338-340 (2004): 603-606.
15. Arkhipov, V.I., P. Heremans, E.V. Emelianova, G.J. Adriaenssens, H. Bassler, "Charge carrier mobility in doped semiconducting polymers", *Applied Physics Letters*, Vol. 82, No. 19 (May 2003): 3245-3247.

16. Arora, N.D., J.R. Hauser, and D.J. Roulston, “Electron and Hole Mobilities in Silicon as a function of Concentration and Temperature”, *IEEE Trans. Electron Devices* Vol. 29 (1982): 292-295.
17. Baccarani, G., M. Rudan, R. Guerrieri, and P. Ciampolini, “Physical Models for Numerical Device Simulation”, *European School of Device Modeling*, University of Bologna, 1991.
18. Baldo et. al., "Very high-efficiency green organic light-emitting based on electrophosphorescence", Appl. Phys. Lett., V. 75: 4-6.
19. Baldo and O'Brien, "Excitonic singlet-triplet ratio in a semiconducting organic thin film", *Phys. Rev. B.*, Vol. 60, No. 20 (November 1999): 14422-14428.
20. Bank, R.E., and A.H. Sherman, “A Refinement Algorithm and Dynamic Data Structure for Finite Element Meshes”, University of Texas at Austin Technical Report CS TR-159/ CNA Tr-166, October 1980.
21. Bank, R., W.M. Coughran, W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith, “Transient Simulation of Silicon Devices and Circuits”, *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 1992-2007.
22. Baraff G.A., “Distribution Functions and Ionisation Rates for Hot Electrons in Semiconductors”, *Appl. Phys. Rev.* 128 (1962): 2507-2517.
23. Barnes, J.J., R.J. Lomax, and G.I. Haddad, “Finite-element Simulation of GaAs MESFET’s with Lateral Doping Profiles and Sub-micron Gates”, *IEEE Trans. Electron Devices* Vol. 23 (September 1976): 1042-1048.
24. Beattie, A.R., and A.M. White, “An Analytical Approximation with a Wide Range of Applicability for Electron Initiated Auger Transitions in Narrow-gap Semiconductors.” *J. Appl. Phys.* Vol. 79, No. 12 (1996): 802-813.
25. Behnen, E., “Quantitative examination of the thermoelectric power of n-type Si in the phonon drag regime”, *J. Appl. Phys.*, Vol 87, (Jan 1990): 287-292.
26. Beinstman P. et al. “Comparision of Optical VCSEL Models on the Simulation at Oxide-Confined Devices.” *IEEE J. of Quantum Electron* 37 (2001): 1618-1631.
27. Bennett, H.S., and C.L. Wilson, “Statistical comparisons of data on band-gap narrowing in heavily doped Silicon: Electrical and Optical measurements”, *J. Appl. Phys.* Vol 55, No. 10, (1984): 3582-3587.
28. Bernardini, F., and V. Fiorentini, "Spontaneous Polarization and Piezoelectric Constants of III-V Nitrides", *Phys. Rev.* B Vol. 56, No. 16, 15 (Oct. 1997): R10024-R10027.
29. Berenger, J.P., "A Perfectly Matched Layer for the Absorption of Electromatic Waves", *Journal of Computational Physics*, Vol. 114 (1994): 185-200.
30. Bescond, M., N. Cavassilas and M. Lanoo, "Effective-mass Approach for n-type Semiconductor Nanowire MOSFETs Arbitrarily Oriented", *Nanotechnology*, 18 (2007): 255201.
31. Bir, G.L., and G.E. Pikus, *Symmetry and Strain-Induced Effects in Semiconductors*, New York, Wiley, 1974.
32. Blom, P.W.M., M.J.M. de Jong, and S. Breedijk, "Temperature Dependent Electron-Hole Recombination in Polymer Light-Emitting Diodes", *Applied Physics Letters* Vol. 71, No. 7 (August 1997): 930-932.
33. Blotekjaer, K., *IEEE Trans. Electron Devices* Vol. 17 (1970): 38.

34. Bonani, F., and G. Ghione. *Noise in Semiconductor Devices - Modeling and Simulation*. Berlin Heidelberg: Springer- Verlag (2001): 24-26.
35. Bouhassoune, M., S.L.M.van Mensfoort, P.A.Bobbert, R.Coehoorn, "Carrier-density and field-dependent charge-carrier mobility in organic semiconductors with correlated Gaussian disorder", *Organic Electronics* 10 (2009): 437-445.
36. Born, M., and E. Wolf. *Principles of Optics*. 6th ed., Cambridge Press, 1980.
37. Boyd, R.W. *Nonlinear Optics*. New York: Academic Press, 1992.
38. Bradley A. "Foreman. Envelope-function formalism for electrons in abrupt heterostructures with material-dependent basis functions," *Phys. Rev. B*, 54 (July 1996): 1909–1921.
39. Broido, D. A and L. J. Sham, "Eective masses of holes at gaas-algaas heterojunctions," *Phys. Rev. B*, 31 (January 1985): 888–892.
40. Brower, K. L., "Dissociation kinetics of hydrogen-passivated (111) Si-SiO_2 interface defects", *Phys. Rev. B*, Vol. 42 (1990): 3444-3453.
41. Brunstrom C., and C. Svensson, "ESR studies of thermally oxidized silicon wafers", *Solid State Commun.*, Vol 37 (1981): 399-404.
42. Burt, M.G., "An exact formulation of the envelope function method for the determination of electronic states in semiconductor microstructures," Semiconductor Science and Technology, 3 (1988):739.
43. Burt, M.G., "Fundamentals of envelope function theory for electronic states and photonic modes in nanostructures," *Journal of Physics: Condensed Matter*, 11 (9)(1999):53.
44. Buturla, E.M., and P.E. Cotrell, "Simulation of Semiconductor Transport Using Coupled and Decoupled Solution Techniques", *Solid State Electronics* 23, No. 4 (1980): 331-334.
45. Canali, C., C. Jacoboni, F. Nava, G. Ottaviani and A. Algerigi-Quaranta, "Electron Drift Velocity in Silicon", *Phys. Rev. B*, Vol. 12 (1975): 2265-2284.
46. Canali, C., G.Magni, R. Minder and G. Ottaviani, "Electron and Hole drift velocity measurements in Silicon and their empirical relation to electric field and temperature", *IEEE Trans. Electron Devices ED-22* (1975): 1045-1047.
47. Carre, B.A., "The Determination of the Optimum Acceleration for Successive Over-Relaxation", *Computer Journal* Vol. 4, Issue 1 (1961): 73-79.
48. Caughey, D.M., and R.E. Thomas. "Carrier Mobilities in Silicon Empirically Related to Doping and Field." Proc. IEEE 55, (1967): 2192-2193.
49. Cassi C., and B. Ricco, "An Analytical Model of the Energy Distribution of Hot Electrons", *IEEE Trans. Electron Devices* Vol. 37 (1990): 1514-1521.
50. Chakrabarti, P., A. Gawarika, V. Mehta, D. Garg, "Effect of TAT on the performance of homojunction Mid-Infrared photodetectors based on InAsSb", *J. Microwaves and Optoelectronics*, Vol. 5, No. 1 (2006): 1-14.
51. Chao, C. and S.L. Chuang, "The spin-orbit coupling effects on the valence band structure of strained semiconductor quantum wells", *Phys. Rev. B*, Vol. 46 (1992): 4110-4122.
52. Chen, S., G. Wang, "High-field properties of carrier transport in bulk wurtzite GaN: A Monte Carlo perspective", *J. Appl. Phys.*, Vol. 103 (2008): 023703-1 - 023703-6.

53. Chen Z., K. Hess, J. Lee, J. W. Lyding, E.Rosenbaum, I. Kizilyalli, S. Chetlur and R. Huang, “On the mechanism for interface trap generation in MOS transistors due to Channel Hot carrier stressing”, *IEEE Electron Device Letters*,Vol. 21, No. 1 (Jan 2000): 24-26.

54. Chinn, S., P. Zory, and A. Reisinger, "A Model for GRIN-SCH-SQW Diode Lasers", *IEEE J. Quantum Electron.* Vol. 24, No. 11, November 1988.

55. Choo, S.C., “Theory of a Forward-Biased Diffused Junction P-L-N Rectifier -- Part 1: Exact Numerical Solutions”, *IEEE Transactions on Electron Devices*, ED-19, 8 (1972): 954-966.

56. Chuang S.L., "Optical Gain of Strained Wurtzite GaN Quantum-Well Lasers", *IEEE J. Quantum Electron.* Vol. 32, No. 10 (Oct. 1996): 1791-1800.

57. Chuang, S.L., and C.S. Chang, "k*p Method for Straind Wurtzite Semiconductors", *Phys. Rev.* B Vol. 54, No. 4, 15 (July 1996): 2491-2504.

58. Chuang, S.L., and C.S. Chang, "A Band-Structure Model of Strained Quantum-Well Wurtzite Semiconductors", *Semicond. Sci. Technol.*, No. 12 (Nov. 1996): 252-262.

59. Shun Lien Chuang. *Physics of Optoelectronic Devices*. John Wiley and Sons, New York, 1995.

60. Chung, Steve S., et. al., "A novel leakage current separation technique in a direct tunnelling regime Gate Oxide SONOS memory cell", IEDM 2003 Technical Digest, (2003): 26.6.1-26.6.4.

61. Choi, Sangmoo et al, "Improved metal-oxide-nitride-oxide-silicon-type flash device with high-k dielectrics for blocking layer", *J. Appl. Phys.* , Vol 94, No. 8, (2003): 5408-5410.

62. Chynoweth A.G., “Ionisation Rates for Electrons and Holes in Silicon”, *Phys. Rev.* 109 (1958): 1537-1540.

63. Cimrova, V, and D. Nehr, “Microcavity effects in single-layer light-emitting device based on poly (p-phenylen vinylene)”, *J. Appl. Phys.* Vol. 79, No. 6 (March 15, 1996): 3299-3306.

64. Clugston, D.A. and P.A. Basore, Proceedings of the 26th IEEE Photovoltaic Specialists Conference, Anaheim, CA, 1997.

65. Cohen, M.L., and T. K. Bergstresser, *Phys. Rev.* Vol. 141 (1966): 789-796.

66. Concannon, A., F. Piccinini, A. Mathewson, and C. Lombardi. “The Numerical Simulation of Substrate and Gate Currents in MOS and EPROMs.” IEEE Proc. IEDM (1995): 289-292.

67. Crofton, J., and S. Sriram, "Reverse Leakage Current Calculations for SiC Schottky Contacts", *IEEE Trans. Electron Devices* Vol. 43, No. 12: 2305-2307.

68. Crowell, C.R., and S.M. Sze, “Current Transport in Metal-Semiconductor Barriers”, *Solid State Electronics* 9 (1966): 1035-1048.

69. Crowell, C.R., and S.M. Sze, “Temperature Dependence of Avalanche Multiplication in Semiconductors”, *Applied Physics Letters* 9 (1966): 242-244.

70. Curtis, Jr., O. L. and J. R. Srour, “The Multiple-Trapping Model and Hole Transport in SiO_2,” *Journal of Applied Physics*, Vol. 48, No. 9 (Sept. 1977): 3819–3828.

71. Darwish, M. et al, “An Improved Electron and Hole Mobility Model for General Purpose Device Simulation”, *IEEE Trans. Electron Devices* Vol. 44, No. 9 (Sept. 1997): 1529-1537.

72. Davydov, Y., et. al., "Band Gap of Hexagonal InN and InGaN Alloys", *Phys. Stat. Sol.* (B), Vol. 234, No. 3 (2002): 787-795.

73. De Mari, A., “An Accurate Numerical One-dimensional Solution of the P-N Junction Under Arbitrary Transient Conditions”, *Solid-State Electronics* 11 (1968): 1021-1053.

74. DiMaria, D.J., “Defect generation in Field-Effect transistors under channel-hot-electron stress”, *J. Appl. Phys.*, Vol. 87, No. 12 (2000): 8707-8715.

75. De Salvo, Barbara, *et al*., “Experimental and Theoretical Investigation of nano-Crystal and Nitride-Trap Memory Devices”, *IEEE Transactions on Electron Devices*, Vol 48, No. 8, (2001): 1789-1799.

76. Ding, Meng, et al, “Silicon Field Emission Arrays With Atomically Sharp Tips”, *IEEE Transactions on Electron Devices*, Vol 49 (2002): 2333-2341.

77. Dorkel, J., and Ph. Leturcq, “Carrier Mobilities in Silicon Semi-Empirically Related to Temperature Doping and Injection Level”, *Solid-State Electronics* 24 (1981): 821-825.

78. Dziewior J. and W. Schmid, “Auger Coefficient for Highly Doped and Highly Excited Silicon”, *Appl. Phys. Lett.* Vol. 31 (1977): 346-348.

79. Edmondson et.al., "Simulation of the Mars Surface Solar Spectra for Optimized Performacnce of Triple-Junction Solar Cells". http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/37746/1/05-2732.pdf. Date Accessed: June 12, 2013.

80. Egley, J.L., and D., Chidambarrao, “Strain Effects on Device Characteristics: Implementation in Drift-Diffusion Simulators”, *Solid-State Electronics*, Vol. 36, No. 12, (1993): 1653-1664.

81. Eisenstat, S.C., M.C. Gursky, M.H. Schultz, A.H. Sherman, Computer Science Department, Yale Univ. Tech. Rep. 112, 1977.

82. Eissfeller, T. and P. Vogl., “Real-space multiband envelope-function approach without spurious solutions,” *Phys. Rev. B*, 84 (Nov. 2011):195122.

83. Engl, W.L., and H.K. Dirks, *Models of Physical Parameters in an Introduction to the Numerical Analysis of Semiconductor Devices and Integrated Circuits*, ed. J.J.H. Miller (Dublin: Boole Press, 1981).

84. Erdelyi, M., et.al. “Generation of diffraction-free beams for applications in optical microlithography”, *J. Vac. Sci. Technol. B*, Vol. 15, No. 2 (Mar/Apr 1997): 287-292.

85. Eriksson, J., N. Rorsman, and H. Zirath, "4H-Silicon Carbied Schottky Barrier Diodes for Microwave Applications", *IEEE Trans. on Microwave Theory and Techniques* Vol. 51, No. 3 (2003): 796-804.

86. Farahmand, M. et. al., "Monte Carlo Simulation of Electron Transport in the III-Nitride Wurtzite Phase Materials System: Binaries and Terniaries", *IEEE Trans. Electron Devices* Vol. 48, No. 3 (Mar. 2001): 535-542.

87. Feigna C., and Venturi F., “Simple and Efficient Modelling of EPROM Writing”, *IEEE Trans. Electron Devices* Vol. 38 (1991): 603-610.

88. Fischetti, V., S.E. Laux, and E. Crabb, “Understanding hot-electron transport in silicon devices: Is there a shortcut?”, *J. Appl. Phys.*, Vol. 78 (1995): 1050.

89. Foldyna, M., K. Postave, J. Bouchala, and J. Pistora, "Model dielectric functional of amorphous materials including Urbach tail", *Proc. SPIE*, Vol. 5445 (June 16, 2004): 301.

90. Foreman, Bradley A., “Eective-mass hamiltonian and boundary conditions for the valence bands of semiconductor microstructures,” *Phys. Rev. B*, 48 (August 1993):4964–4967.

91. Foreman, Bradley A., “Envelope-function formalism for electrons in abrupt heterostructures with material-dependent basis functions,” *Phys. Rev. B*, 54 (July 1996): 1909–1921.
92. Fossum, J.G. and D.S. Lee, “A Physical Model for the Dependence of Carrier Lifetime on Doping Density in Nondegenerate Silicon”, *Solid State Electronics* Vol. 25 (1982): 741-747.
93. Fossum, J.G., R.P. Mertens, D.S. Lee and J.F. Nijs, “Carrier recombination and lifetime in highly doped Silicon”, Solid State Electronics Vol. 26, No 8 (1983): 569 -576.
94. Fujii, K., Y. Tanaka, K. Honda, H. Tsutsu, H. Koseki, and S. Hotta. “Process Techniques of 15 inch Full Color and High Resolution a-Si TFT LCD.” 5th Int. MicroProcess Conf., Kawasaki, Japan, 1992.
95. Furnemont, A. et. al., "Model for electron redistribution in Silicon Nitride", ESSDERC 2006 (2006):447-450.
96. Furnemont, A. et. al., "A consistent model for the SANOS programming operation", 22nd IEEE Non-volatile Semiconductor Memory Workshop, (2007): 96-97.
97. Gear, C.W. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.
98. Geballe T.H., and T.W. Hull, “Seebeck Effect in silicon”, *Phys. Rev. 98* (1955): 940-947.
99. Gill, W.D., "Drift Mobilities in Amorphous Charge-Transfer Complexes of Trinitrofluorenone and Poly-n-vinylcarbazole", *J. Appl. Phys.* Vol. 55, No. 12 (1972): 5033.
100. Gloeckler, M.A., “Device Physics of Cu(In, Ga)Se2 Thin-Film Solar Cells” (Ph.D diss., Colorado State Univeristy, Summer 2005).
101. Gnudi, A., D. Ventura, G.Baccarani and F.Odeh, “Two-dimensional MOSFET simulation by means of a multi-dimensional spherical harmonics expansion of the Boltzmann transport equation”, Solid State Electronics, Vol. 36 , No. 4 (1993): 575-581.
102. Goliber, T.E. and J.H. Perlstein, "Analysis of Photogeneration in a Doped Polymer System in Terms of a Kinetic Model for Electric-Field-Assisted Dissociation of Charge-Transfer States", J. Chem. Phys., Vol. 90, No. 9, 1 (May 1984): 4162-4167.
103. Gonzalez, B., V.Palankovski, H. Kosina, A. Hernandez, and S. Selberherr, “An energy relaxation time model for device simulation”, *Solid-State Electronics*, Vol. 43 (1999): 1791-1795.
104. Gossens, R.J. G., S. Beebe, Z. Yu and R.W. Dutton, “An Automatic Biasing Scheme for Tracing Arbitrarily Shaped I-V curves”, *IEEE Trans. Computer-Aided Design* Vol. 13, No. 3 (1994): 310-317.
105. Grasser, T. et al., “The Time-Dependent Defect Spectroscopy (TDDS) for the Characterization of Bias Temperature Instability”, *Proceedings of the International Reliability Physics Symposium (IRPS)*, 2A.2.1-2A.2.10, (2010).
106. Grant, W.N., “Electron and Hole Ionization Rates in Epitaxial Silicon at High Electric Fields”, *Solid-State Electronics* 16 (1973): 1189-1203.
107. Grasser T., B.Kaczer, W.Goes, Th. Aichinger, Ph. Hehenberger, and M. Nelhiebel, "A two-stage model for NBTI", 33-43, Proc. 47th International Reliability Physics Symposium, Montreal .
108. Grove, A.S. *Physics and Technology of Semiconductor Devices*. Wiley, 1967.

109. Guerin, C., V. Huard and A. Bravaix, “General framework about defect creation at the Si/SiO2 interface”, *J.Appl.Phys.*, Vol. 105 (2009): 114513.

110. Gundlach, K. H., *Solid-State Electronics* Vol. 9 (1966): 949.

111. Hack, B.M. and J.G. Shaw. "Numerical Simulations of Amorphous and Polycrystalline Silicon Thin-Film Transistors." Extended Abstracts 22nd International Conference on Solid-State Devices and Materials, Sendai, Japan (1990): 999-1002.

112. Hacker R., and A. Hangleiter, “Intrinsic upper limits of the carrier lifetime in Silicon”, *J. Appl. Physics*, Vol. 75 (1994): 7570-7572.

113. Hall, R.N., “Electron Hole Recombination in Germanium”, *Phys. Rev.* 87 (1952): 387.

114. Hansch, W., Th. Vogelsang, R. Kirchner, and M. Orlowski, “Carrier Transport Near the Si/SiO2 Interface of a MOSFET”, *Solid-State Elec.* Vol. 32, No. 10 (1989): 839-849.

115. Hansch, W. et al., “A New Self-Consistent Modeling Approach to Investigating MOSFET Degradation”, *IEEE Trans. Electron Devices* Vol. 11 (1990): 362.

116. Hareland, S., S. Jallepalli, G. Chindalore, W. Shih, A. Tasch, and C. Maziar, “A Simple Model for Quantum Mechanical Effects in Hole Inversion Layers in Silicon PMOS Devices”, *IEEE Trans. Elec. Dev.*, Vol. 44, No. 7 (July 1997): 1172-1173.

117. Hareland, S., M. Manassian, S. Jallepalli, H. Wang, G. Chindalore, A. Tasch, and C. Maziar, “Computationally Efficient Models for Quantization Effects in MOS Electron and Hole Accumulation Layers”, *IEEE Trans. Elec. Dev.*, Vol. 45, No. 7 (July 1998): 1487-1493.

118. Harper, J.G., H.E. Matthews, and R. H. Bube, “Photothermoelectric Effects in Semiconductors: n- and p-type Silicon”, *J. Appl. Phys.*, Vol 41 (Feb 1970): 765 -770.

119. Harvey, J.E., “Fourier treatment of near-field scalar diffraction theory”. *Am. J. Phys.* Vol. 47, No. 11 (1979): 974.

120. Hatakeyama, T., J. Nishio, C. Ota, and T. Shinohe, “Physical Modeling and Scaling Properties of 4H-SiC Power Devices”, Proc. SISPAD, Tokyo (2005): 171-174.

121. Hatta, M., S.F. Wan, N.Soin, J.F.Zhang, “The effect of gate Oxide Thickness and Drain Bias on NBTI Degradation in 45 nm PMOS”, Proc. ICSE2010 (2010): 210-213.

122. He, S.S, and V.L. Shannon, “Hydrogen diffusion and redistribution in PECVD Si-rich silicon nitride during rapid thermal anneal”, 4th International conference on Solid-state and Integrated Circuit Technology (1995): 269-271.

123. He, Yie, “Discretization method for Current Continuity Equation Containing Magnetic Field”, *IEEE Transactions on Electron devices*, Vol. 39, No. 4 (1994): 820-824.

124. Heavens, O.S. *Optical Properties of Thin Solid Films*. Dover Publishing, 1991.

125. Heiman, F.P. and G. Warfield, “The effects of Oxide Traps on the MOS Capacitance”, *IEEE Trans. Electron Devices*, Vol. 12, No. 4, (April 1965): 167-178.

126. Herring, C., “Theory of the Thermoelctric Power of Semiconductors” *Phys. Rev.*, Vol 96, (Dec. 1954): 1163-1187.

127. Hockney, R.W., and J.W. Eastwood, *Computer Simulation Using Particles*, New York: McGraw Hill, 1981.

128. Hong, S-M., A-T. Pham, C. Jungemann, *Deterministic Solvers for the Boltzmann transport equation*, Springer-Verlag, (2011).

129. Horowitz, G., "Organic Field-Effect Transistors", *Advanced Materials* Vol. 10, No.5 (1998): 365-377.

130. Huang, Y.L., Y.Ma, R.Job and A.G.Ulyashin, “Hydrogen diffusion at moderate temperatures in p-type Czochralski Silicon”, *J. Appl. Phys.* Vol. 96, No. 12 (2004): 7080-7086.

131. Huldt, L., N. G. Nilsson and K.G.Svantesson, “The temperature dependence of band-to-band Auger recombination in Silicon”, *Applied Physics Letters*, Vol. 35 (1979): 776-777.

132. Hurkx, G.A.M, D.B.M. Klaassen, and M.P.G. Knuvers, “A New Recombination Model for Device Simulation Including Tunneling,” *IEEE Trans. Electron Devices* Vol. 39 (Feb. 1992): 331-338.

133. Hurkx, G.A.M., D.B.M. Klaasen, M.P.G. Knuvers, and F.G. O’Hara, “A New Recombination Model Describing Heavy-Doping Effects and Low Temperature Behaviour”, *IEDM Technical Digest* (1989): 307-310.

134. Hurkx, G.A.M., H.C. de Graaf, W.J. Klosterman et. al. “A Novel Compact Model Description of Reverse Biase Diode Characteristics including Tunneling.” ESSDERC Proc. (1990): 49-52.

135. Iannaccone, G., G. Curatola, and G. Fiori. “Effective Bohm Quantum Potential for device simulation based on drift-diffusion and energy transport.” SISPAD 2004.

136. Ielmini, D., M. Manigrasso, F. Gattel amd M. G. Valentini, “A new NBTI Model Based on Hole Trapping and Structural Relaxation in MOS Dielectrics”, *IEEE Trans. on Electron Devices*, Vol. 56, No.9 (2009): 1943-1952.

137. Ielmini, D., A.S.Spinelli, M.A.Rigamonti, A.L.Lacaita, “Modelling of SILC based on Electron and Hole Tunneling-part 1: Transient Effects”, *IEEE Trans. on Electron Devices*, Vol 47, No 6 (June 2000: 1258-1265.

138. Ielmini, D., A.S.Spinelli, M.A.Rigamonti, A.L.Lacaita, “Modelling of SILC based on Electron and Hole Tunneling-part 2: Steady State”, *IEEE Trans. on Electron Devices*, Vol 47, No 6 (June 2000): 1266-1272.

139. Ieong, M., Solomon, P., Laux, S., Wong, H., and Chidambarro, D., "Comparison of Raised and Schottky Source/Drain MOSFETs Using a Novel Tunneling Contact Model", *Proceedings of IEDM (*1998): 733-736.

140. Ioffe Physico-Technical Institute. “New Semiconductor Materials Characteristics and Properties”. http://www.ioffe.ru/SVA/NSM/introduction.html (accessed January 7, 2009).

141. Ishikawa, T. and J. Bowers, "Band Lineup and In-Plane Effective Mass of InGaAsP or InGaAlAs on InP Strained-Layer Quantum Well", *IEEE J. Quantum Electron.* Vol. 30, No. 2 (Feb. 1994): 562-570.

142. Ismail, K., “Effect of dislocation in strained Si/SiGe on electron mobility,” J. Vac. Sci. Technol. B, Vol. 14, No.4, 1996: 2776-2779.

143. Jacoboni, C., and L. Reggiani, “The Monte Carlo Method for the Solution of Charge Transport in Semiconductors with Applications to Covalent Materials”, *Reviews of Modern Physics*, Vol. 55 (1983): 645-705.

144. Jaeger, R.C and F. H. Gaensslen, “Simulation of Impurity Freezeout through Numerical Solution of Poisson’s Equations and Application to MOS Device Behavior”, *IEEE Trans. Electron Devices* Vol. 27 (May 1980): 914-920.

145. Juska, G., K. Genevicius, N. Nekrasas, G. Silauzys, and R. Osterbacka, “Two dimensional Langevin recombination in regioregular poly(3-hexylthiophene)”, *Applied Physics Letters* Vol. 95 (2009): 013303-1..3.

146. Joyce, W.B., and R.W. Dixon, “Analytic Approximation for the Fermi Energy of an ideal Fermi Gas”, *Appl. Phys Lett.* 31 (1977): 354-356.

147. Kahen, K.B., “Two-Dimensional Simulation of Laser Diodes in Steady State”, *IEEE J. Quantum Electron.* Vol. 24, No. 4, April 1988.

148. Kane, E.O., “Zener tunneling in Semiconductors”, *J. Phys. Chem. Solids*, Vol. 12, (1959): 181-188.

149. Katayama, K. and T. Toyabe, “A New Hot Carrier Simulation Method Based on Full 3D Hydrodynamic Equations”, *IEDM Technical Digest* (1989): 135.

150. Keeney, S., F. Piccini, M. Morelli, A. Mathewson et. al., “Complete Transient Simulation of Flash EEPROM Devices”, *IEDM Technical Digest* (1990): 201-204.

151. Kemp, A.M., M. Meunier, and C.G. Tannous, "Simulations of the Amorphous Silicon Static Induction Transistor", *Solid-State Elect.* Vol. 32, No. 2 (1989): 149-157.

152. Kernighan, Brian, W. and Dennis M. Ritchie, “The C Programming Language”, Prentice-Hall, 1978.

153. Kerr, M.J., and A., Cuevas, "General Parameterization of Auger Recombination in Crystalline Silicon", *Jour. of Appl. Phys.*, Vol. 91, No. 4 (2002): 2473-2480.

154. Kim, T., W. Want, and Q. Li, "Advancement in materials for energy-saving lighting devices", *Front. Chem. Sci. Eng.* V6 (2012):13-26.

155. Kimura M. et. al., “Development of Poly-Si TFT Models for Device Simulation: In-Plane Trap Model and Thermionic Emission Model”, (Proceedings of Asia Display/IDW '01, 2001), 423.

156. Kimura M., R. Nozawa, S. Inoue, T. Shimoda, B.O-K. Lui, S.W-B. Tam, P. Migliorato, “Extraction of Trap States at the Oxide-Silicon Interface and Grain Boundary for Polycrystalline Silicon Thin-Film Transistors”, *Jpn J. Appl. Phys.*, Vol. 40 (2001): 5227-5236.

157. Klaassen, D.B.M., *Physical Modeling Bipolar Device Simulation”, In: Simulation of Semiconductor Devices and Processes* Vol. 4, ed. W. Fichtner and D. Aemmer (Harting-Gorre, 1991), 23-43.

158. Klaassen, D.B.M., “A Unified Mobility Model for Device Simulation- I. Model Equations and Concentration Dependence”, *Solid-State Elect.* Vol. 35, No. 7 (1992): 953-959.

159. Klaassen, D.B.M., “A Unified Mobility Model for Device Simulation - II. Temperature Dependence of Carrier Mobility and Lifetime”, *Solid-State Elect.* Vol. 35, No. 7 (1992): 961-967.

160. Klaassen, D.B.M., J.W. Slotboom, and H.C. De Graaff, “Unified Apparent Bandgap Narrowing in n- and p- type Silicon”, *Solid-State Elect.* Vol. 35, No. 2 (1992):125-129.

161. Klausmeier-Brown, M., M. Lundstrom, M. Melloch, “The Effects of Heavy Impurity Doping on AlGaAs/GaAs Bipolar Transistors”, *IEEE Trans. Electron Devices* Vol. 36, No. 10 (1989): 2146-2155.

162. Krc, J. et.al. "Optical Modeling and Simulation of Thin-Film Cu(In,Ga)Se2 Solar Cells", NUSOD (2006): 33-34.

163. Krc, J. Zeman, M. Smole, F. and Topic, M. "Optical Modeling of a-Si:H Solar Cells Deposited on Textured Glass/SnO2 Substrates", *J. Appl. Phys.*, Vol. 92. No. 2 (Feb. 2002): 749-755.

164. Krieger, G., and R.M. Swanson, “Fowler-Nordheim electron tunneling in thin Si-SiO[sub 2]-Al structures,” *J. Appl. Phys.*, Vol. 52 (1981): 5710.

165. Krishnan, A. T., C. Chancellor, S. Chakravarthi, P. E. Nicollian, V. Reddy,A. Varghese, R. B. Khamankar, S. Krishnan, “Material Dependence of Hydrogen Diffusion: Implications for NBTI Degradation”, *IEDM Technical Digest* (December 5, 2005): 691-695.

166. Kufluoglu H., and M. A. Alam, “A generalized Reaction-Diffusion Model with explicit H-H2 Dynamics for NBTI Degradation”, *IEEE Trans. Electron Devices*, Vol. 54, No. 5, (2007): 1101-1107.

167. Kumagai, M., S.L. Chuang, and H. Ando, “Analytical Solutions of the Block-diagonalized Hamiltonian for Strained Wurtzite Semiconductors”, *Phys. Rev.* B Vol. 57, No. 24, 15 (June 1998): 15303-15314.

168. Kuzmicz, W., “Ionization of Impurities In Silicon”, *Solid-State Electronics*, Vol. 29, No. 12 (1986): 1223-1227.

169. Lackner, T., “Avalanche multiplication in semiconductors: A modification of Chynoweth's law”, *Solid-State Electronics* Vol. 34 (1991): 33-42.

170. Lades, M., *Modeling and simulation of wide bandgap semiconductor devices: 4H/6H-Sic*, Ph.D Dissertation, T.U. Munich, 2000.

171. Lades M. and G. Wachutka. “Extended Anisotropic Mobility Model Applied to 4H/6H-SiC Devices.” Proc. IEEE SISPAD (1997): 169-171.

172. Lambert, J. *Computational Methods in Ordinary Differential Equations*. Wiley, 1973.

173. Lang, D.V., “Measurement of the band gap of GexSi1-X/Si strained-layer heterostructures,” *Appl. Phys. Lett.*, Vol. 47, No. 12, (1985): 1333-1335.

174. Larez, C, Bellabarba, C. and Rincon, C., “Allow composition and temperature dependence of the fundamental absorption edge in CuGaxIn1-xSe2”, *Appl. Pys. Lett*. Vol 65, No. 26 (Sept 1994): 1650-1652.

175. Laux, S.E., “Techniques for Small-Signal Analysis of Semiconductor Devices”, *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 2028-2037.

176. Laux, S.E., and M.V. Fischetti, “Monte Carlo Simulation of Submicron Si *n*-MOSFETs at 77 and 300”, *IEEE Electron Device Letters*, Vol. 9, No. 9 (September 1988): 467-469.

177. Laux, S.E., and M.V. Fischetti, “Numerical aspects and implementation of the DAMOCLES Monte Carlo device simulation program” from *Monte Carlo Device Simulation: Full Band and Beyond*, Dordrecht: K.Hess, Ed. Kluwer Academic Publishers, 1991.

178. Law, M.E. et. al., Self-Consistent Model of Minority-Carrier Lifetime, Diffusion Length, and Mobility, *IEEE Electron Device Letters* Vol. 12, No. 8, 1991.

179. Lee, C.C., M.Y. Chang, P.T. Huang and S.W. Liu, “Electrical and optical simulation of organic light-emitting devices with fluorescent dopant in the emitting layer”, *J. Appl Physics*, Vol. 101 (2007): 114501.

180. Lee, C., Chang, M., Jong, Y., Huang, T., Chu, C. and Chang, Y., "Numerical Simulation of Electrical and Optical Characteristics of Multilayer Organic Light-Emitting Devices", *Japanese Journal of Appl. Phys.*, Vol. 43, No. 11A (2004): 7560-7565.

181. Lee et. al., "Electrical and optical simulation of organic light-emitting devices with fluorescent dopant in the emitting layer", *J. Appl. Phys.* Vol. 101 (2007): 114501-1-114501-11.

182. Lee et.al., "Numerical Simulation of Electrical and Optical Characteristics of Multilayer Organic Light-Emitting Devices", *Jap. J. App. Phys.*, Vol. 43, No. 11a (2004): 7560-7565.

183. Li, Z., K. Dzurko, A. Delage, and S. McAlister, "A Self-Consistent Two-Dimensional Model of Quantum-Well Semiconductor Lasers: Optimization of a GRIN-SCH SQW Laser Structure", *IEEE J. Quantum Electron.* Vol. 28, No. 4 (April 1992): 792-802.

184. Lindefelt U., "Equations for Electrical and Electrothermal Simulation of Anisotropic Semiconductors", *J. Appl. Phys.* 76 (1994): 4164-4167.

185. Lindefelt U., "Doping-induced band edge displacements and band gap narrowing in 3C-, 4H-, 6H-SiC and Si", *J.Appl. Phys.*, Vol. 84, No. 5, (1998): 2628-2637.

186. Littlejohn, M.J., J.R. Hauser, and T.H. Glisson, "Velocity-field Characteristics of GaAs with $\Gamma_6^e - L_6^e - X_6^e$ Conduction Band Ordering", *J. Appl. Phys.* 48, No. 11 (November 1977): 4587-4590.

187. Lok, C. Lew Yan Voon and Morten Willatzen. *The kp Method Electronic Properties of Semiconductors*, Springer, New York (2009).

188. Lombardi et al, "A Physically Based Mobility Model for Numerical Simulation of Non-Planar Devices", *IEEE Trans. on CAD* (Nov. 1988): 1164.

189. Lui O.K.B., and P. Migliorato, "A New Generation-Recombination Model For Device Simulation Including The Poole-Frenkel Effect And Phonon-Assisted Tunneling", *Solid-State Electronics* Vol. 41, No. 4 (1997): 575-583.

190. Lundstrom, M., Fundamentals of Carrier Transport, 2nd ed., Cambridge University Press, 2000.

191. Lundstrom, Ingemar and Christer Svensson, "Tunneling to traps in insulators", *J. Appl. Phys.*, Vol.43. No. 12 (1972): 5045-5047.

192. Lundstrom M., and R. Shuelke, "Numerical Analysis of Heterostructure Semiconductor Devices", *IEEE Trans. Electron Devices* Vol. 30 (1983): 1151-1159.

193. Luo, J.K., H.Thomas, D.V.Morgan and D.Westwood, "Transport properties of GaAs layers grown by molecular beam epitaxy at low temperature and the effects of annealing", *J. Appl. Phys*., Vol. 79, No. 7 , (Apr 1996), pp. 3622-3629.

194. Ma, T. P. and Paul V. Dressendorfer, *Ionizing Radiation Effects in MOS Devices and Circuits*, John Wiley and Sons, 1989.

195. Mahapatra, S., P. Bharath Kumar and M. A. Alam, "Investigation and Modeling of Interface and Bulk Trap Generation During Negative Bias Temperature Instability of p-MOSFETS", *IEEE Transactions on Electron Devices*, Vol. 51, No. 9 (2004): 1371-1379.

196. Manku, T., and A. Nathan, "Electron Drift Mobility Model for Devices Based on Unstrained and Coherently Strained $Si_{1-x}Ge_x$ Grown on <001> Silicon Substrate", *IEEE Transaction on Electron Devices* Vol. 39, No. 9 (September 1992): 2082-2089.

197. Marques, M., Teles, L., Scolfaro, L., and Leite, J., "Lattice parameter and energy band gap of cubic $Al_xGa_yIn_{1-x-y}N$ quaternary alloys", *Appl. Phys. Lett.*, Vol. 83, No. 5, 4 (Aug. 2003): 890-892.

198. Mars, P. "Temperature Dependence of Avalanche Breakdown Voltage in p-n Junctions", *International Journal of Electronics* Vol. 32, No. 1 (1963): 23-27.

199. Martinez, O., et. al., "InGaP Layers Grown on Different GaAs Surfaces for High Efficiency Solar Cells", Mater. Res. Soc. Symp. Vol. 1167, Materials Research Society, (2009): Paper 1167-O03-04.

200. Masetti G., M.Severi, and S.Solmi, “Modeling of Carrier Mobility Against Carrier Concentration in Arsenic , Phosphorous and Boron doped Silicon”, *IEEE Trans. Elec. Dev. ED-30*, (1983): 764-769.

201. Maslov, A. V. and Cun.-zheng Ning, (2007), “GaN Nanowire Lasers”, *Nitride Semiconductor Devices: Principles and Simulation*, ed J. Piprek, (Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2007), Ch 21.

202. Matsuzawa, K., K. Uchida, and A. Nishiyama, "A Unified Simulation of Schottky and Ohmic Contacts", *IEEE Trans. Electron Devices*, Vol. 47, No. 1 (Jan. 2000): 103-108.

203. Mayaram, K., “CODECS: A Mixed-Level Circuit and Device Simulator”, Memo No. UCB/ERL M88/71, University of California, Berkeley, CA, November 1988.

204. Mayaram, K., and D.O. Pederson, “Coupling Algorithms for Mixed-Level Circuit and Device Simulation”, *IEEE Transactions on Computer-Aided Design* Vol. 11, No. 8 (August 1992): 1003-1012.

205. Mclintock, R., J. Pau, K. Minder, C. Bayram, P. Kung, and M. Razeghi, "Hole-initiated multiplication in back-illuminated GaN avalanche photodiodes", *Appl. Phys. Lett.*, Vol. 90 (2007): 14112-1-14112-3.

206. McIntyre, R., “On the Avalanche Initiation Probability of Avalanche Diodes Above the Breakdown Voltage”, *IEEE Trans. Elec. Dev.*, V. ED-20, N. 7l (July 1973): 637-641.

207. McPherson, J. W., R. B. Khamankar and A. Shanware, “Complementary model for intrinsic time-dependent dielectric breakdown”, *J. Appl. Phys.*, Vol. 88, No. 9 (2000): 5351-5359.

208. Mercury vapor lamp. http://www.olympusmicro.com/primer/lightandcolor/lightsourcesintro.html. Date Accessed: June 12, 2013.

209. Meinerzhagen, B., K. Bach, I. Bozk, and W.L. Eugl. “A New Highly Efficient Nonlinear Relaxation Scheme for Hydrodynamic MOS Simulations.” Proc. NUPAD IV (1992): 91.

210. Meinerzhagen, B., and W.L. Engl, “The Influence of Thermal Equilibrium Approximation on the Accuracy of Classical Two-Dimensional Numerical Modeling of Silicon Submicrometer MOS Transistors”, *IEEE Trans. Electron Devices* Vol. 35, No. 5 (1988): 689-697.

211. Meftah, A.F., A.M. Meftah and A. Merazga, “A theoretical study of light induced defect creation, annealing and photoconductivity degradation in a-Si:H”, *J. Phys.: Condens. Matter*, Vol.16 (2004): 3107-3116.

212. M Hoffmann and ZG Soos. “Optical absorption spectra of the Holstein molecular crystal for weak and intermediate electronic coupling”. *Physical Review B*, 66(2):024305, 2002.

213. Milanowski, R. J., M. P. Pagey, L. W. Massengill, R. D. Schrimpf, M. E. Wood, B. W. Oxford, R. J. Graves, K. F. Galloway, C. J. Nicklaw, and E. P. Kelley, “TCAD-Assisted Analysis of Back-Channel Leakage in Irradiated Mesa SOI nMOSFETs,” *IEEE Transactions on Nuclear Science*, Vol. 45, No. 6 (Dec. 1998): 2593–2599.

214. Miller et. al., “Device modeling of ferroelectric capacitors”, *J. Appl. Phys.* 68, 12, 15 Dec. 1990.

215. Muller, J-M., et al., Handbook of Floating-Point Arithmetic, Birkhäuser, 2009.

216. Murata, H., C. Merritt, Z. Kafafi, "Emission Mechanism in Rubrene-Doped Molecular Organic Light-Emitting Diodes: Direct Carrier Recombination at Luminesent Centers", *IEEE J. of Sel. Top. in Quantum Elec.* V. 4, N. 1 (Jan/Feb. 1998): 119-124.

217. Muth, J., et. al., "Absorption Coefficient, Energy Gap, Exciton Binding Energy, and Recombination Lifetime of GaN Obtained from Transmission Measurements", *Appl. Phys. Lett.* Vol. 71 (1997): 2572-2574.

218. Nakagawa, A., and H. Ohashi, “A Study on GTO Turn-off Failure Mechanism — A Time-and Temperature-Dependent 1D Model Analysis”, *IEEE Trans. Electron Devices* Vol. 31, No. 3 (1984): 273-279.

219. Nemanich, R.J., et al, “Electron emission properties of crystalline diamond and III-Nitride surfaces”, *Applied Surface Science*, (1998): 694-703.

220. Nenashev, A.V., F. Jansson, S.D. Baranovskii, R. Osterbacka, A.V. Dvurechenskii, and F. Gebhard , "Role of diffusion in two-dimensional bimolecular recombination", Applied Physics Letters Vol. 96, (2010): 213304-1..3.

221. Neugroschel, A., J. S. Wang and F. A. Lindholm, “Evidence for excess carrier storage in electron-hole plasma in Silicon transistors”, *IEEE Electron Device Letters*, Vol. 6, No. 5, (1985): 253-255.

222. Ning, T. H. , “High-field capture of electrons by Coulomb-attractive centres in Silicon dioxide”, *J. Appl. Phys.*, Vol. 47, No. 7 (July 1976): 3203-3208.

223. Nomenclature of Inorganic Chemistry,” *Journal of the American Chemical Society* No. 82 (1960): 5525.

224. Oh, C., and M. Escuti, "Time-somain analysis of periodic anisotropic media at oblique incidence: an efficient FDTD implementation", *Optics Express*, Vol. 14, No. 24, (Nov. 27, 2006).

225. Ohtoshi, T., K. Yamaguchi, C. Nagaoka, T. Uda, Y. Murayama, and N. Chionone, “A Two-Dimensional Device Simulator of Semiconductor Lasers”, *Solid-State Electronics* Vol. 30, No.6 (1987): 627-638.

226. Okuto, Y., and R. Crowell, “Threshold energy effects on avalanche breakdown Voltage in semiconductor junctions”, *Solid-State Electronics* Vol. 18 (1975): 161-168.

227. Ozgur, U., et. al., "A comprehensive review of ZnO materials and devices", *J. Appl. Phys.*, Vol. 98 (2005): 041301-1-041301-103.

228. Oguzman, I., et. al., "Theory of Hole Initiated Impact Ionization in Bulk Zincblende and Wurtzite GaN", *J. Appl. Phys.* V. 81, No. 12 (1997): 7827-7834.

229. Oguzman, I., E. Bellotti, and K. Brennan, "Theory of hole initiated impact ionization in bulk zincblende and wurtzite GaN", *J. Appl. Phys*, B. 81 (June 15, 1997): 7827-7834.

230. Paasch G., and S.Scheinert, "Charge carrier density of organics with Gaussian density of states: Analytical approximation for the Gauss-Fermi integral", *J. Appl. Phys*, Vol 107, 104501 (2010): 1-4

231. Pacelli, A., “On the Modeling of Quantization and Hot Carrier Effects in Scaled MOSFETs and Other Modern Devices.” Dissertation, Polytechnic Institute of Milan, 1997.

232. Pacelli, A., W. Duncan, and U. Ravaioli. “A multiplication scheme with variable weights for Ensemble Monte Carlo simulation of hot-electron tails”, Proc. IX Conference on Hot Carriers in Semiconductors, J.P. Leburton, K. Hess, and U. Ravaioli, Eds., Plenum Press (July 31-August 4, 1995):407-410.

233. Pacelli, A., and U. Ravaioli, “Analysis of variance-reduction schemes for Ensemble Monte Carlo simulation of semiconductor devices,” *Solid-State Electronics*, Vol. 4, Issue 4 (April 1997): 599-605.

234. Pagey, M. P., R.D. Schrimpf, K. F. Galloway, C. J. Nicklaw, S. Ikeda and S.Kamohara, “A hydrogen-transport-based interface-trap-generation model for Hot-carrier reliability prediction”, *IEEE Electron Device Letters*, Vol. 22, No. 6 (2001): 290-292

235. Palankovski, V., “Simulation of Heterojunction Bipolar Transistors”, Ph.D Dissertation, T.U. Vienna, (2000).

236. Palankovski, V., Schultheis, R. and Selberherr, S., "Simulation of Power Heterojunction Bipolar Transistors on Gallium Arsenide", *IEEE Trans. on Elec. Dev.*, Vol. 48, No. 6, 6 (June 2001): 1264-1269.

237. Palik, E.D., Ed. *Handbook of Optical Constants of Solids*. New York: Academic Press Inc., 1985.

238. Passler, R., *Phys.Stat.Solidi* (b) 216 (1999): 975.

239. Pasveer, W.F., et. al., “Unified Description of Charge-Carrier Mobilities in Disordered Semiconducting Polymers”, *Phys. Rev. Lett.*, Vol 94 (2005): 206601.

240. Patil, M. B., "New Discretization Scheme for Two-Dimensional Semiconductor Device Simulation on Triangular Grid", *IEEE Trans. on CAD of Int. Cir. and Sys.* Vol. 17., No. 11 (Nov. 1998): 1160-1165.

241. Patrin, A. and M. Tarsik, "Optical-Apsorption Spectrum of Silicon Containing Internal Elastic Stresses", *Journal of Applied Spectroscopy* Vol. 65, No. 4 (July 1998): 598- 603.

242. Paulson, P., R. Birkmire, and W. Shafarman, "Optical Characterization of CuIn1-xGaxSe2 Alloy Thin Films by Spectroscopic Ellipsometry", *Journal of Appl. Phys.*, V. 94, No. 2 (2003): 879-888.

243. Pearson, G.L., and J. Bardeen, “Electrical Properties of Pure Silicon and Silicon Alloys Containing Boron and Phosphorus”, *Phys. Rev.*, Vol. 75, No. 5 (Mar. 1949): 865-883.

244. Pinto M.R., Conor S. Rafferty, and Robert W. Dutton, “PISCES2 - Poisson and Continuity Equation Solver”, Stanford Electronics Laboratory Technical Report, Stanford University, September 1984.

245. Pipinys P., and V. Lapeika, “Temperature dependence of reverse-bias leakage current in GaN Schottky diodes as a consequence of phonon-assisted tunneling,” *J.Appl. Phys.* 99 (2006): 093709.

246. Piprek, J., *Semiconductor Optoelectronic Devices: Introduction to Physics and Simulation*. UCSB: Academic Press (2003): 22.

247. Piprek, J., Peng, T., Qui, G., and Olowolafe, J., "Energy Gap Bowing and Refractive Index Spectrum of AlInN and AlGaInN". 1997 IEEE International Symposium on Compound Semiconductors (September 8-11, 1997): 227-229.

248. Poindexter, E.H., “MOS interface states: overview and physicochemical perspective”, *Semicond. Sci. Technol.*, Vol 4. (1989): 961-969.

249. Polsky, B., and J. Rimshans, “Half-Implicit Scheme for Numerical Simulation of Transient Processes in Semiconductor Devices”, *Solid-State Electronics* Vol. 29 (1986): 321-328.

250. Powell, M.J., and S.C. Deane, “Improved defect-pool model for charged defects in amorphous silicon”, *Phy. Rev. B*, Vol. 48, No. 15 (1993): 10815-10827.

251. Powell, M.J., and S.C. Deane, “Defect-pool model and the hydrogen density of states in hydrogenated amorphous silicon”, Phy. Rev. B, Vol. 53, No. 15 (1996): 10121-10132.

252. Price, C.H., “Two Dimensional Numerical Simulation of Semiconductor Devices”, Ph.D. Dissertation, Stanford University, May 1982.

253. Price, P.J., and J. M. Radcliffe, "IBM Journal of Research and Development" 364, October 1959.

254. Prinz E.J., et. al, "An embedded 90 nm SONOS Flash EEPROM Utilizing Hot Electron Injection Programming and 2-sided Hot Hole Injection Erase", *IEDM Digest*, (2002): 927-930.

255. Rafferty, C.S., M.R. Pinto, and R.W. Dutton, "Iterative Methods in Semiconductor Device Simulation", *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 2018-2027.

256. Rajkanan, K. R., Singh, and J. Shewchun, "Absorption Coefficient of Silicon for Solar Cell Calculations", *Sol. St. Elec.*, Vol. 22, (September 1979): 793-795.

257. Rakic, A.D., and M.L. Majewski. *Cavity and Mirror Design for Vertical-Cavity Surface-Emitting Lasers", Vertical-Cavity Surface-Emitting Laser Devices.* Springer Series in Photonics, 2003.

258. Reed, M. L., and J. D. Plummer, "Chemistry of Si-SiO2 interface trap annealing", *J.Appl.Phys.*, Vol. 63, No. 12 (June 1988): 5776-5793.

259. Reggiani, S., M. Valdinoci, L. Colalongo, and G. Baccarani, " A unified analytical model for bulk and surface mobility in S n- and p-channel MOSFETs", ESSDERC, 1999.

260. Reggiani, S., G.Barone, S.Poli, E.Gnani, A.Gnudi, G.Baccarani, M-Y.,Chuang, W.Tian, R.Wise, "TCAD Simulation of Hot-Carrier and Thermal Degradation in STI-LDMOS Transistors", *IEEE Trans. Elec. Dev*. Vol. 60, No.2 (2013): 691-698.

261. Renewable Resource Data Center (RReDC). "Solar Spectra: Air Mass Zero". http://rredc.nrel.gov/solar/spectra/am0/ (accessed January 16, 2009).

262. Renewable Resource Data Center (RReDC). "Reference Solar Spectral Irradiance: Air Mass 1.5". http://rredc.nrel.gov/solar/spectra/am1.5/ (accessed January 16, 2009).

263. Rhoderick, E.H. and R.H. Williams. *Metal-Semiconductor Contacts*, 2nd ed. Oxford Science Publications, 1988.

264. Riccobene, C., K. Gartner, G. wachutka, H. Baltes and W. Fichtner, "First Three-Dimensional Numerical Analysis of Magnetic Vector Probe", Proceeding of IEDM94 (1994): 727-730.

265. Riccobene, C., G.Wachutka, J.Burgler, H.Baltes, "Operating principle of Dual Collector Magnetotransistors studied by Two-Dimensional Simulation", *IEEE Trans. Electron Devices*, Vol. 41, No. 7 (1994): 1136-1148.

266. Richter A. et al, "Improved quantitative description of Auger Recombination in crystalline silicon", *Physical Review B, Vol 86*, (2012):165202.

267. Roblin, P., A. Samman, and S. Bibyk, "Simulation of Hot-Electron Trapping and Aging of n-MOSFET's", *IEEE Trans. Electron Devices* Vol. 35 (1988): 2229.

268. Robertson, J., "High Dielectrric Constant Oxides", *Eur. Phys. J. Appl. Phys.* Vol. 28 (2004): 265-291.

269. Roper, L., and M. Rosenzweig, *Electromagnetic Spectra of Light Bulbs*. http://www.roperld.com/science/electromagneticspectraoflightbulbs.htm. Date Accessed: June 12, 2013.

270. Rose, D.H., and R.E. Bank, "Global Approximate Newton Methods", *Numerische Mathematik* 37 (1981): 279-295.

271. Roulston, D.J., N.D. Arora, and S.G. Chamberlain, “Modeling and Measurement of Minority-Carrier Lifetime versus Doping in Diffused Layers of n ±p Silicon Diodes”, *IEEE Trans. Electron Devices* Vol. 29 (Feb. 1982): 284-291.

272. Ruhstaller, B., S.A. Carter, S. Barth, H. Riel, W. Riess, J.C. Scott, "Transient and Steady-State Behavior of Space Charges in Multilayer Organic Light-Emitting Diodes", *J. Appl. Phys.* Vol. 89, No. 8 (2001): 4575-4586.

273. Rzepa, G. et al., “Physical Modeling of NBTI: From Individual Defects to Devices”, *Proceedings of the 20th International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 81-84, (2014).

274. Rzepa, G. et al., “Complete Extraction of Defect Bands Responsible for Instabilities in n and pFinFETs”, *2016 Symposium on VLSI Technology Digest of Technical Papers*, 208-209, (2016).

275. Sangiorgi, E., C.S. Rafferty, M.R. Pinto, and R.W. Dutton, “Non-planar Schottky Device Analysis and Applications”, (Proc. International Conference on Simulation of Semiconductor Devices and Processes, Swansea, U.K.), July 1984.

276. Scharfetter, D.L., and H.K. Gummel, “Large-Signal Analysis of a Silicon Read Diode Oscillator”, *IEEE Trans. Electron Devices* Vol. 16 (January 1969): 64-77.

277. Schenk, A., and G. Heiser, “Modeling and simulation of tunneling through ultra-thin gate dielectrics”, *J.Appl.Phys.*Vol. 81 (1997): 7900-7908.

278. Schenk, A., “A model for the field and temperature dependence of SRH lifetimes in Silicon”, *Solid-State Electronics*, Vol. 35 (1992): 1585-1596.

279. Schenk, A., “Finite-temperature full random-phase approximation model of band-gap narrowing for silicon device simulation”, *J.Appl. Phys*., Vol. 84, No. 7, (1998): 3684-3694.

280. Schenk, A., “Rigorous Theory and Simplified Model of the Band-to-Band Tunneling in Silicon”, *Solid State Electronics* Vol. 36 (1993): 19-34.

281. Schwarz, S.A., and S.E. Russe, “Semi-Empirical Equations for Electron Velocity in Silicon: Part II — MOS Inversion Layer”, *IEEE Trans. Electron Devices* Vol. 30, No. 12 (1983): 1634-1639.

282. S-C Lee and Andreas Wacker. “Nonequilibrium Green's function theory for transport and gain properties of quantum cascade structures”. *Physical Review B*, 66(24):245314, 2002.

283. Scott, J.C., and G.G.Malliaras, “Charge injection and recombination at the metal-organic interface”, *Chemical Physics Letters*, Vol. 299 (1999): 115-119.

284. Scott, J.C., S. Karg, and S.A. Carter, "Bipolar Charge and Current Distributions in Organic Light-Emitting Diodes", *J. Appl. Phys.* Vol. 82, No. 3 (August 1997): 1454-1460.

285. Seki, S., T. Yamanaka, and K. Yokoyama, “Two-Dimensional Analysis of Current Blocking Mechanism in InP Buried Heterostructure Lasers”, *J. Appl. Phys.* Vol. 71, No. 7 (April 1992): 3572-3578.

286. Selberherr, S. *Analysis and Simulation of Semiconductor Devices*. Wien, New York: Springer-Verlag, 1984.

287. Selberherr, S., “Process and Device Modeling for VLSI”, *Microelectron. Reliab.* 24 No. 2 (1984): 225-257.

288. Shaw, John G., and Michael Hack, "An Analytical Model For Calculating Trapped Charge in Amorphous Silicon", *J. Appl. Phys.* Vol. 64, No. 9 (1988): 4562-4566.

289. Shen J., and J. Yang, "Physical Mechanisms in Double-Carrier Trap-Charge Limited Transport Processes in Organic Electroluminescent Devices: A Numerical Study", *J. Appl. Phys.* Vol. 83, No. 12 (June 1998): 7706-7714.

290. Shih. I.B., and Z. Mi, "CIS, CIG, and Related Materials for Photovoltaic Applications" (Photovoltaic Workshop, MeMaster University, Feb. 19, 2009).

291. Shin, H., A.F. Tasch, C.M. Maziar, and S.K. Banerjee, "A New Approach to Verify and Derive a Transverse-field-dependent Mobility Model for Electrons in MOS Inversion Layers", *IEEE Trans. Electron Devices* Vol. 36, No. 6 (1989): 1117-1123.

292. Shin, H., G.M. Yeric, A.F. Tasch, and C.M. Maziar. *Physically-based Models for Effective Mobility and Local-field Mobility of Electrons in MOS Inversion Layers* (to be published).

293. Shirahata M., H. Kusano, N. Kotani, S. Kusanoki, and Y. Akasaka, "A Mobility Model Including the Screening Effect in MOS Inversion Layer", *IEEE Trans. Computer-Aided Design* Vol. 11, No. 9 (Sept. 1992): 1114-1119.

294. Shockley W., and W.T. Read, "Statistics of the Recombination of Holes and Electrons", *Phys. Rev.* 87 (1952): 835-842.

295. Simmons, J.G., and G.W. Taylor , *Phys. Rev.* B, 4 (1971): 502.

296. Slotboom, J.W., "The PN Product in Silicon", *Solid State Electronics* 20 (1977): 279-283.

297. Slotboom, J.W. and H.C. De Graaf, "Measurements of Bandgap Narrowing in Silicon Bipolar Transistors", *Solid State Electronics* Vol. 19 (1976): 857-862.

298. Sotoodeh, M., Khalid, A., and Rezazadeh, A., "Empirical low-field mobility model for III-V compounds applicable in device simulation codes", *J. Appl. Phys.*, Vol. 87, No. 6, 15 (March 2000): 2890-2900.

299. Starkov, I., S. Tyaginov, H. Enichlmair, J.Cervenka, C.Jungemann, S. Carniello, J.M.Park, H.Ceric and T.Grasser, "Hot-carrier degradation caused interface state profile - Simulation versus experiment", *J.Vac. Sci. Technol B*, Vol. 29 (2011): 01AB09-1/8.

300. Staudigel, J., M.Stossel, F.Steuber, J.Simmerer, "A quantitative numerical model of multilayer vapor-deposited organic light emitting diodes", *Journal of Applied Physics* 86 (1999): 3895-3910.

301. Stern, F., and W. E. Howard, "Properties of Semiconductor Surface Inversion Layers in the Electric Quantum Limit", Phys. Rev. , 163 (1967): 816

302. STR Group. Modeling of Crystal Growth and Devices. http://www.str-soft.com/products (accessed on May 25, 2010).

303. Stratton, R., *Phys. Rev.*, 126, 6 (1962): 2002.

304. Stratton, R., "Semiconductor Current-Flow Equations (Diffusion and Degeneracy)", *IEEE Trans. Electron Devices* Vol. 19, No. 12 (1972): 1288-1292.

305. Sutherland, J., and F. Hauser, "A Computer Analysis of Heterojunction and Graded Composition Solar Cells", *IEEE Trans. Electron Devices* Vol. 24 (1977): 363-373.

306. Swirhun, S., Y.H. Kawark and R.M. Swanson, "Simultaneous measuring of hole lifetime, hole mobility and bandgap narrowing in heavily doped p-type silicon", *IEDM Technical Digest*, (December 1986): 24-27.

307. Sze, S.M. "Physics of Semiconductor Devices". Wiley, 1981.

308. Taflove, A. and S.C., Hagness, *Computational Electrodynamics The Finite-Difference Time-Domain Method*, Third Edition, Artech House, Boston and London, 2005.

309. Takahashi Y., K. Kunihiro, and Y. Ohno, “Two-Dimensional Cyclic Bias Device Simulator and Its Applications to GaAs HJFET Pulse Pattern Effect Analysis,” *IEICE Journal C*, June 1999.

310. Takayama, T., “Low-Noise and High-Power GaAlAs Laser Diode with a New Real Refractive Index Guided Structure,” *Japan J. Appl. Phys.* Vol. 34, No. 7A(1999): 3533-3542.

311. Tam, S., P-K Ko, and C. Hu, “Lucky-electron Model of Channel Hot-electron Injection in MOSFET’s”, *IEEE Trans. Electron Devices* Vol. 31, No. 9, September 1984.

312. Tanimoto, H. et.al., “Modeling of Electron Mobility Degradiation for HfSiON MISFETs”, Proc. of SISPAD (2006): 47-50.

313. Tan, G., N. Bewtra, K. Lee, Xu, and J.M., “A Two-Dimensional Nonisothermal Finite Element Simulation of Laser Diodes”, *IEEE J. of Q.E.* Vol. 29, No.3 (March 1993): 822-835.

314. Tanaka, et. al., "Forster and Dexter energy-transfer processes in fluorescent BAlq thin films doped with phosphorescent Ir(ppy)3 molecules", *J. Appl. Phys.*, Vol. 99 (April 2006): 073501-1 - 073501-5.

315. Tut, T., et. al., "Experimental evaluation of impact ionization coefficients in AlxGa1-xN based avalanche photodiodes", *Appl. Phys. Lett*, Vol. 89 (2006): 183524-1-183524-3.

316. Tuttle, B. “Energetics and diffusion of hydrogen in SiO_2”, *Phys. Rev. B.*, Vol. 61 (2000): 4417-4420.

317. Uchida, K., and S. Takagi, “Carrier scattering induced by thickness fluctuation of silicon-on-insulator film in ultrathin-body metal-oxide-semiconductor field-effect transistors”, *Applied Phys. Lett.*, Vol. 82, No. 17, 22 (April 2003): 2916-2918.

318. Uehara, K. and H. Kikuchi, “Generation of Nearly Diffraction-Free Laser Beams”’, *Appl. Phys. B*, Vol 48 (1989): 125-129.

319. *Utmost III User's Manual* and *Utmost III Extraction Manual*. Silvaco, Inc., 2008.

320. Vaillant, F., and D. Jousse, “Recombination at dangling bonds and steady-state photoconductivity in a-Si:H”, *Phys. Rev. B*, Vol. 34, No. 6 (1986): 4088-4098.

321. Valdinoci, M., D. Ventura, M.C. Vecchi, M. Rudan, G. Baccarani, F. Illien, A. Stricker, and L. Zullino, “The Impact-ionization in Silicon at Large Operating Temperature”. SISPAD ‘99, Kyoto, Japan, Sept. 6-8, 1999.

322. Vanheusden, K., W.L. Warren, R.A.B. Devine, D.M. Fleetwood, J.R.Schwank, M.R.Shaneyfelt, P.S.Winokur, and Z.J.Lemnios, “Non-volatile memory device based on mobile protons in SiO2 thin films”, *Nature*, Vol 386 (1997): 587-589.

323. Van Dort, M.J., P.H. Woerlee, and A.J. Walker, “A Simple Model for Quantisation Effects in Heavily-Doped Silicon MOSFETs at Inversion Conditions”, *Solid-State Elec.* Vol. 37, No. 3 (1994): 411-414.

324. Van Herwaarden A.W., and P.M.Sarro, “Thermal Sensors based on the Seebeck effect”, *Sensors and Actuators* Vol.10, (1986): 321-346.

325. Van Overstraeten, R., and H. Deman, “Measurement of the Ionization Rates in Diffused Silicon p-n Junctions”, *Solid-State Electronics* 13 (1970): 583-608.

326. van Mensfoort, S.L.M., and Coehoorn, R, "Effect of Gaussian disorder on the voltage dependence of the current density in sandwich-type devices based on organic semiconductors" Phys. Rev. B, Vol. 78, (2008): 085207.

327. Van Roey, J., J. Van Der Donk, P.E. Lagasse., “Beam-propagation method: analysis and assessment”, *J. Opt. Soc. Am.* Vol. 71, No 7 (1981): 803.

328. Van de Walle, C. G. and B. R. Tuttle, “Microscopic Theory of Hydrogen in Silicon Devices”, *IEEE Trans. Electron Devices*,Vol. 47, No. 10 (Oct 2000): 1779-1785.

329. Varga, R.S. *Matrix Iterative Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1962.

330. Vendelin, G. *Design of Amplifiers and Oscillators by the S-Parameter Method*. John Wiley & Sons, New York, 1982.

331. Ventura, D., A.Gnudi, G.Baccarani, “A deterministic approach to the solution of the BTE in semiconductors”, *Rivista del Nuovo Cimento*, Vol. 18, No. 6 (1995): 1-32.

332. Verlaak, S., Cheyns D., Debucquoy M., Arkhipov, V. and Heremans P., "Numerical Simulation of Tetracene Light-Emitting Transistors: A Detailed Balance of Exciton Processes", *Applied Physics Letters* Vol. 85, No. 12 (Sep 2004): 2405-2407.

333. Visser, T.D., H. Blok, B. Demeulenaere and D. Lenstra, “Confinement factors and Gain in Optical Amplifiers”, *IEEE J. Quantum Electron.*, 33(10), (1997): 1763–1766.

334. V.Uhnevionak et al, “Comprehensive Study of the Electron Scattering Mechanisms in 4H-SiC MOSFETS”, *IEEE Trans. of Electron Devices*,Vol 62, (2015),pp. 2562-2570.

335. Vurgaftman, I., J.R. Meyer, and L. R. Ram-Mohan, "Band Parameters for III-V Compound Semiconductors and their Alloys", *Journal of Applied Physics* Vol. 89, No. 11 (June 2001): 5815-5875.

336. Wachutka, G.K., “Rigorous Thermodynamic Treatment of Heat Generation in Semiconductor Device Modeling”, *IEEE Trans., Computer-Aided Design* Vol. 9, No. 11 (1990): 1141-1149.

337. Wada, M. et al. “A Two-Dimensional Computer Simulation of Hot Carrier Effects in MOSFETs.” Proc. IEDM Tech. Dig. (1981): 223-225.

338. Waltl, M. et al., “A Single-Trap Study of PBTI in SiON nMOS Transistors: Similarities and Differences to the NBTI/pMOS Case”, *Proceedings of the International Reliability Physics Symposium (IRPS)*, XT.18.1-XT.18.5, (2014).

339. Wegener et al., "Metal-insulator-semiconductor transistor as a nonvolatile storage ", IEEE IEDM Abstract, (1967): 58.

340. Walker, D., Zhang, X., Saxler, Z., Kung, P., Xj, J., Razeghi, M., "AlxGa(1-x)N(0<=x,=1)Ultraviolet Photodetectors Grown on Sapphire by Metal-Organic Chemical-Vapor Deposition", *Appl. Phys. Lett.* Vol. 70, No. 8 (1997): 949-951.

341. Wu, J., et. al., "Unusual properties of the fundamental band gap of InN", *Applied Physics Letters*, Vol. 80, No. 21, 27 (May 2002): 3967-3969.

342. Watt, J.T., Ph.D., Thesis, Stanford University, 1989.

343. Watt, J.T., and J.D. Plummer. “Universal Mobility-Field Curves for Electrons and Holes in MOS Inversion Layers.” 1987 Symposium on VLSI Technology, Karuizawa, Japan.

344. Weast, R. Ed. *CRC Handbook of Chemistry and Physics*. CRC Press, 1976.

345. Welser, J., J. L. Hoyt, S. Takagi, J.F. Gibbons, “Strain dependence of the performance enhancement in strained-Si *n*-MOSFETs”, *IEDM Tech. Dig.*, 1994: 337-376.

346. Wenus, J., J. Rutkowski, and A. Rogalski, “Two-Dimensional Analysis of Double-Layer Heterojunction HgCdTe Photodiodes”, *IEEE Trans. Electron Devices* Vol. 48, No.7 (July 2001): 1326-1332.

347. Wenzel, H., and H.J. Wünsche, “The effective frequency method in the analysis of vertical-cavity surface-emitting lasers,” *IEEE J. Quantum Electron* 33 (1997): 1156-1162.

348. Wettstein, A., A. Schenk, and W. Fichtner, “Quantum Device-Simulation with the Density Gradient Model on Unstructured Grids”, *IEEE Trans. Electron Devices* Vol. 48, No. 2 (Feb. 2001): 279-283.

349. White, Marvin H., Dennis A. Adams, and Jiankang Bu, "On the go with SONOS ", *IEEE Circuits and Devices Magazine*, Vol, 16, No. 4,(2000): 22-31.

350. White, W.T. et.al., *IEEE Trans. Electron Devices*. Vol. 37 (1990): 2532.

351. Wilt, D.P. and A. Yariv, “Self-Consistent Static Model of the Double-Heterostructure Laser,” *IEEE J. Quantum Electron.* Vol. QE-17, No. 9 (1981): 1941-1949.

352. Wimp, T. *Sequence Transformation and their Application*. Academic Press, 1981.

353. Winkler, Roland. *Spin Orbit Coupling Effects in Two-Dimensional Electron and Hole Systems*. Springer-Verlag, Berlin, 2003.

354. Winstead, B., “Impact of source-side heterojunctions on MOSFET performance,” M.S. thesis, University of Illinois at Urbana-Champaign, 1998.

355. Winstead, B., “Monte Carlo simulation of silicon devices including quantum correction and strain”, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 2001.

356. Wood, D.M, and Alex Zunger, “Successes and failures of the k p method: A direct assessment for gaas/alas quantum structures,” *Phys. Rev. B*, 53 (March 1996): 7949–7963.

357. Wünsche, H.J., H. Wenzel, U. Bandelow, J. Piprek, H. Gajewski, and J. Rehberg, “2D Modelling of Distributed Feedback Semiconductor Lasers,” *Simulation of Semiconductor Devices and Processes* Vol. 4 (Sept. 1991): 65-70.

358. Wu, C.M., and E.S. Yang, “Carrier Transport Across Heterojunction Interfaces”, *Solid-State Electronics* 22 (1979): 241-248.

359. Xiao, H. et. al., "Growth and characterization of InN on sapphire substrate by FR-MBE", *Journal of Crystal Growth*, Vol. 276 (2005): 401-406.

360. Yamada, M., S. Ogita, M. Yamagishi, and K. Tabata, “Anisotropy and broadening of optical gain in a GaAs/AlGaAs multi-quantum-well laser,” *IEEE J. Quantum Electron.* QE-21 (1985): 640-645.

361. Yamada, T., J.R. Zhou, H. Miyata, and D.K. Ferry, “In-Plane Transport Properties of Si/Si(1-x)Ge(x) Structure and its FET Performance by Computer Simulation”, *IEEE Trans. Electron Devices* Vol. 41 (Sept. 1994): 1513-1522.

362. Yamaguchi, K. “A Mobility Model for Carriers in the MOS Inversion Layer”, *IEEE Trans. Electron Devices* Vol. 30 (1983): 658-663.

363. Yan, R., S. Corzine, L. Coldren, and I. Suemune, "Corrections to the Expression for Gain in GaAs", *IEEE J. Quantum Electron.* Vol. 26, No. 2 (Feb. 1990): 213-216.

364. Yang K., J. East, and G. Haddad, “Numerical Modeling of Abrupt Heterojunctions Using a Thermionic-Field Emission Boundary Condition”, *Solid-State Electronics* Vol. 36, No. 3 (1993): 321-330.

365. Yariv, A. *Optical Electronics*. CBS College Publishing, 1985.

366. Yee, K. S., "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media", *IEEE Trans. on Antennas and Propagation*, V. AP-14, No. 3 (May, 1966): 302-307.

367. Yeo, Y., King, T., and Hu, C., “Metal-dielectric Band Alignment and its Implications for Metal Gate Complementry Metal-Oxide-Semiconductor Technology”, *J. Appl. Phys.*, Vol. 92, No. 12 (Dec. 15, 2002): 7266-7271.

368. Young, G. A. and J. R. Scully, “The diffusion and trapping of hydrogen in high purity aluminum”, *Acta Mater* Vol. 46, No. 18 (1998): 6337-6349.

369. Yu G.C., and S.K. Ken, “Hydrogen Diffusion coefficient of silicon nitride thin films”, *Applied Surface Science*, Vol. 201 (2002): 204-207.

370. Yu, Z., D. Chen, L. So, and R.W. Dutton, “Pisces-2ET, Two Dimensional Device Simulation for Silicon and Heterostructures”, Integrated Circuits Laboratory, Stanford University (1994): 27.

371. Yu, Z., and R.W. Dutton, “SEDAN III-A Generalized Electronic Material Device Analysis Program”, Stanford Electronics Laboratory Technical Report, Stanford University, July 1985.

372. Zabelin, V., Zakheim, D., and Gurevich, S., "Efficiency Improvement of AlGaInN LEDs Advanced by Ray-Tracing Analysis", *IEEE Journal of Quantum Electronics*, Vol. 40, No. 12, (Dec. 2004): 1675-1686.

373. Zaidman, Ernest G., “Simulation of Field Emission Microtriodes”, *IEEE Transactions on Electron Devices*, Vol 40 (1993):1009-1015.

374. Zakhleniuk, N. A., “Nonequilibrium drift-diffusion transport in semiconductors in presence strong inhomogeneous electric fields”, *Applied Physics Letters*, Vol. 89, (2006): 252112.

375. Zamdmer, N., Qing Hu, K. A. McIntosh and S. Verghese, “Increase in response time of LT-GaAs photoconductive switches at high voltage bias”, *Appl. Phys. Lett.*, Vol. 75, No. 15, (October 1999): 2313-2315.

376. Zappa, F., Lovati, P., and Lacaita, A., “Temperature Dependence of Electron and Hole Ionization Coefficients in InP.” (Eighth International Conference Indium Phosphide and Related Materials, IPRM '96, Schwabisch Gmund, Germany April 21-25, 1996), 628-631.

377. Zeman, M., R. van Swaaij, and J. Metselaar, "Optical Modeling of a-Si:H Solar Cells with Rough Interfaces: Effect of Back Contact and Interface Roughness", J. Appl. Phys., Vol. 88, No. 11 (Dec. 2000): 6436-6443.

378. Zhou, B., Butcher, K., Li, X., Tansley, T., "Abstracts of Topical Workshop on III-V Nitrides". TWN '95, Nagoya, Japan, 1995.

379. Zhou, J.R. and D.K. Ferry, “Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations,” *IEEE Trans. Electron Devices* Vol. 39 (Mar. 1992): 473-478.

380. Zhou, J.R. and D.K. Ferry, “Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations - II:Volocity Overshoot”, *IEEE Trans. Electron Devices* Vol. 39 (Aug. 1992): 1793-1796.

381. Zhu, W. and Ma, T., “Temperature Dependence of Channel Mobility in HfO2-Gated NMOSFETs”, *IEEE Elec. Dev. Lett.*, Vol. 25, No. 2, (Feb. 2, 2004): 89-91.

382. Zundel, T., A.Mesli, J.C. Muller and P.Siffert, “Boron neutralization and hydrogen diffusion in silicon subjected to low-energy hydrogen implantation”, *Applied Physics A*, Vol. 48 (1989): 31-40.

383. Zundel, T., and J.Weber, “Trap-limited hydrogen diffusion in boron-doped silicon”, *Phys. Rev. B*, Vol. 46 (1992): 2071-2077.

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X