National Tsing Hua University

Department of Electrical Engineering

EE3660 Intro. to Digital Signal Processing, Spring 2020

Homework Assignment #4: Chap7 Solution

Problem 1.

(a) Solution:
The DTFT of $(0.9)^n u[n]$ is:

$$\frac{1}{1-0.9\mathrm{e}^{-\mathrm{j}\omega}}$$

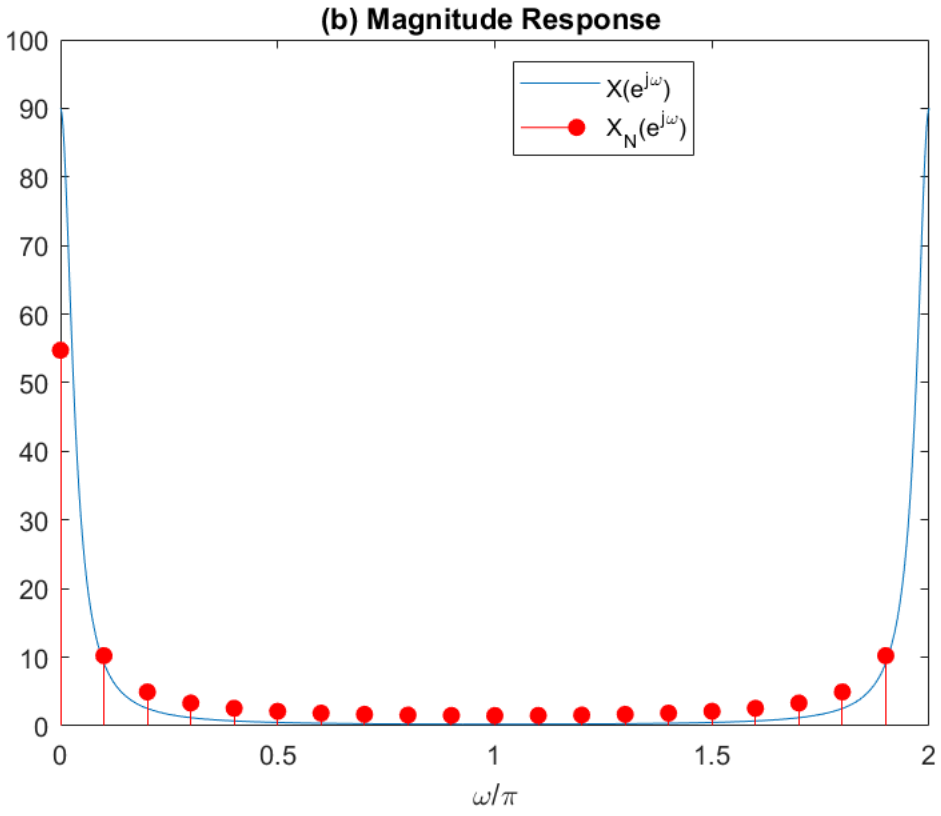
The DTFT of $x[n]$ is:

$$\begin{aligned}\tilde{X}(\mathrm{e}^{\mathrm{j}\omega}) &= (-\mathrm{j})\frac{\mathrm{d}}{\mathrm{d}\omega}\left(\frac{1}{1-0.9\mathrm{e}^{-\mathrm{j}\omega}}\right)\\ &= \frac{0.9\mathrm{e}^{-\mathrm{j}\omega}}{(1-0.9\mathrm{e}^{-\mathrm{j}\omega})^2}\end{aligned}$$

(b)

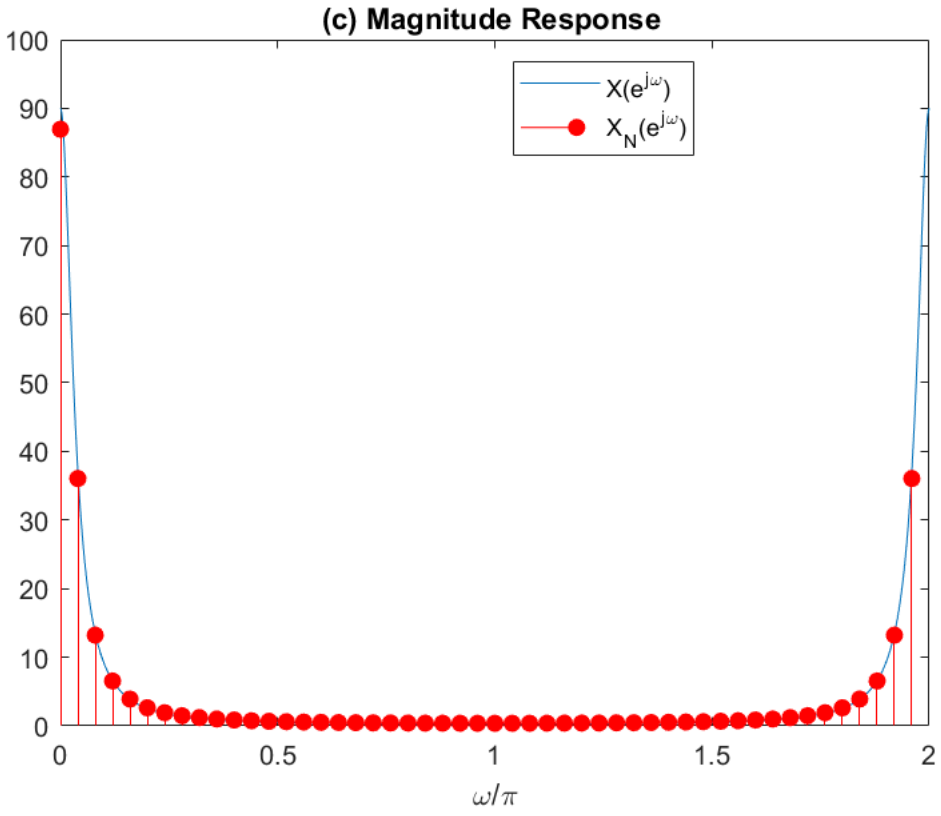
```
close all; clc; clear all;
j = sqrt(-1);
w = linspace(0,2,1000)*pi;
X = 0.9*exp(-j*w)./(1-0.9*exp(-j*w)).^2;

N = 20; % Part (b)
n = 0 : N-1;
x_n = n.*0.9.^n;
X_N = fft(x_n);
w_k = 2/N*(0:N-1);
%Mag response
plot(w/pi,abs(X));
hold on;
stem(w_k,abs(X_N),'filled','color','red');
xlabel('\omega/\pi');
title('(b) Magnitude Response ')
legend('X(e^{j\omega})','X_N(e^{j\omega})','location','best')
hold off;
```



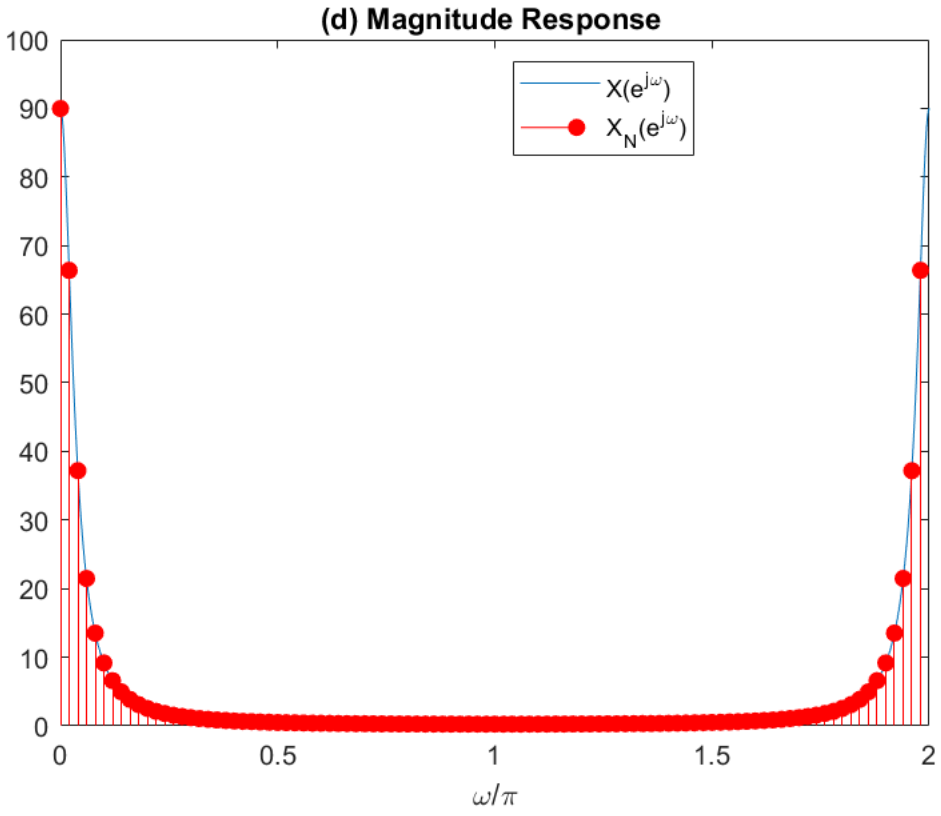
(c)

```
N = 50; % Part (c)
n = 0 : N-1;
x_n = n.*0.9.^n;
X_N = fft(x_n);
w_k = 2/N*(0:N-1);
%Mag response
plot(w/pi,abs(X));
hold on;
stem(w_k,abs(X_N),'filled','color','red');
xlabel('\omega/\pi');
title('(c) Magnitude Response ')
legend('X(e^{j\omega})','X_N(e^{j\omega})','location','best')
hold off;
```



(d)

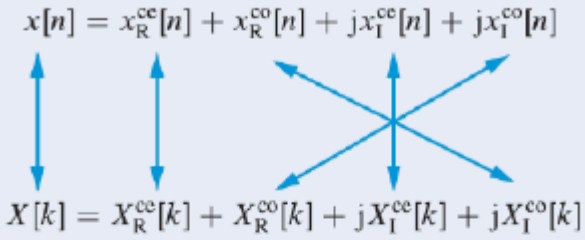
```
N = 100; % Part (d)
n = 0 : N-1;
x_n = n.*0.9.^n;
X_N = fft(x_n);
w_k = 2/N*(0:N-1);
%Mag response
plot(w/pi,abs(X));
hold on;
stem(w_k,abs(X_N),'filled','color','red');
xlabel('\omega/\pi');
title('(d) Magnitude Response ')
legend('X(e^{j\omega})','X_N(e^{j\omega})','location','best')
hold off;
```


(d) Magnitude Response
X(e^jω)
X_N(e^jω)
100
90
80
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
ω/π

Problem 2.

(a)

題目給定$\mathbf{x[n]} = x_1[n] + jx_2[n]$
根據下圖的特性，



N-point sequence	*N*-point DFT
Real	real part is even - imaginary part is odd
Imaginary	real part is odd - imaginary part is even
Real and even	real and even
Real and odd	imaginary and odd
Imaginary and even	imaginary and even
Imaginary and odd	real and odd

$$x_1[n] = x_R^{ce}[n] + x_R^{co}[n] \overset{DFT}{\longleftrightarrow} X_1[k] = X_R^{ce}[k] + jX_I^{co}[k]$$

$$jx_2[n] = jx_I^{ce}[n] + jx_I^{co}[n] \overset{DFT}{\longleftrightarrow} jX_2[k] = X_R^{co}[k] + jX_I^{ce}[k]$$

$$X^{cce}[k] = \frac{1}{2}(X[k] + X^*[\langle -k \rangle_n])$$

$$(\text{根據} X^{ce}[k] = X^{ce}[\langle -k \rangle_n], X^{co}[k] = -X^{co}[\langle -k \rangle_n] \text{和上圖 X[k]}) = X_1[k]$$

$$X^{cco}[k] = \frac{1}{2}(X[k] - X^*[\langle -k \rangle_n])$$

$$(\text{根據} X^{ce}[k] = X^{ce}[\langle -k \rangle_n], X^{co}[k] = -X^{co}[\langle -k \rangle_n] \text{和上圖 X[k]}) = jX_2[k]$$

(b)

```
function [X1 X2] = tworealDFTs(x1, x2)
j = sqrt(-1);
xc = x1 ++ j*x2;
X = fft(xc);
XX = conj([X(1) fliplr(X(2:end))]);
X1 = (X+XX)/2;
X2 = (X-XX)/(2*j);
```

(c)

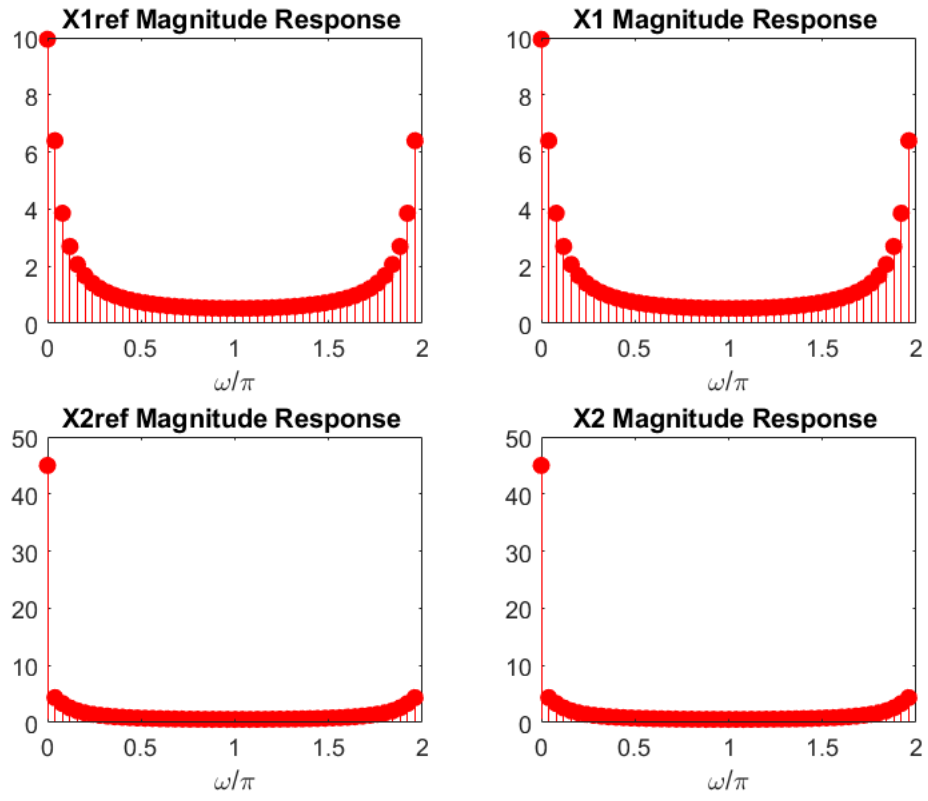
```
% part (c)
close all; clc
n = 0:49;
N = length(n);
x1 = 0.9.^n;
x2 = 1 - 0.8.^n;
[X1, X2] = tworealDFTs(x1, x2);
% Verification
X1_ref = fft(x1);
X2_ref = fft(x2);

% %Magnitude response
w_k = 2/N*(0:N-1);
figure;
subplot(2,2,1);
stem(w_k,abs(X1_ref),'filled','color','red');
xlabel('\omega/\pi');
title('X1ref Magnitude Response ')

subplot(2,2,2);
stem(w_k,abs(X1),'filled','color','red');
xlabel('\omega/\pi');
title('X1 Magnitude Response ')
hold off;

subplot(2,2,3);
stem(w_k,abs(X2_ref),'filled','color','red');
xlabel('\omega/\pi');
title('X2ref Magnitude Response ')

subplot(2,2,4);
stem(w_k,abs(X2),'filled','color','red');
xlabel('\omega/\pi');
title('X2 Magnitude Response ')
hold off;
```



Problem 3.

(a) Solution:

Solving the circular convolution using hand calculation:

$$\begin{bmatrix} 2 & 0 & -1 & 1 & -1 \\ -1 & 2 & 0 & -1 & 1 \\ 1 & -1 & 2 & 0 & -1 \\ -1 & 1 & -1 & 2 & 0 \\ 0 & -1 & 1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \\ 0 \\ 6 \\ 7 \end{bmatrix}$$

(b) (c)

```
close all; clc
xn1 = 1:5;
xn2 = [2 -1 1 -1];
% part (b)
xn = circonv(xn1', [xn2 0]').'
```

```
xn = 1×5
    -2     4     0     6     7
```

```
% part (c)
N = max(length(xn1), length(xn2));
```

```
Xk1 = fft(xn1, N);
Xk2 = fft(xn2, N);
Xk = Xk1.* Xk2;
xn_dft = ifft(Xk) % [-2, 4, 0, 6, 7]
```

```
xn_dft = 1×5
   -2.0000    4.0000    0.0000    6.0000    7.0000
```

Problem 4.

(a)

```
function y = lin2circonv(x,h)
% compute the circular convolution
% thru the results of linear convolution

y1 = conv(x(:)', h(:)');
N1 = length(x);
N2 = length(h);
N = max(N1, N2);
L = N1+N2-1;
nn = -L-1:L-1;
y1 = [zeros(1, L+1), y1];
ll = floor(L/N);
y = zeros(size(nn));
for ii = 0:ll
    y = y + [y1(ii*N+1:end) zeros(1,ii*N)];
end
y = y(L+2: L+N+1);
```

(b)

```
% part (b)
close all; clc
xn = 1:4;
hn = [1 -1 1 -1];
y1 = lin2circonv(xn, hn)
```

```
y1 = 1×4
    -2     2    -2     2
```

```
y2 = circonv(xn', hn').'
```

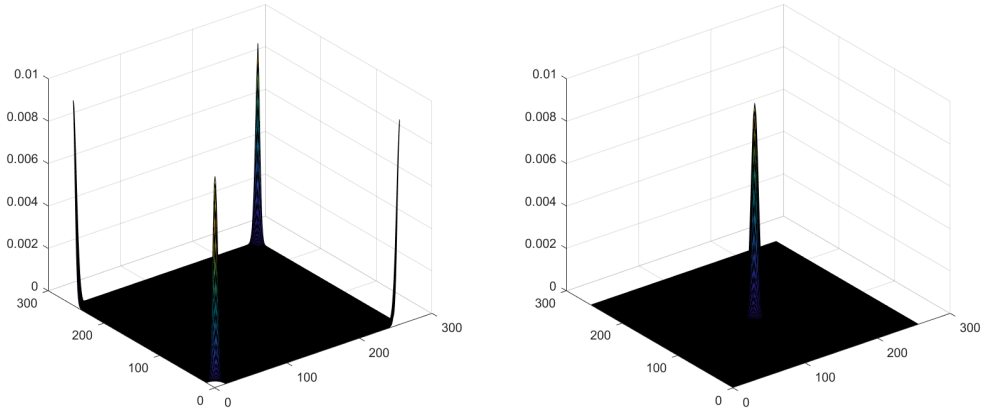
```
y2 = 1×4
    -2     2    -2     2
```

Problem 5.

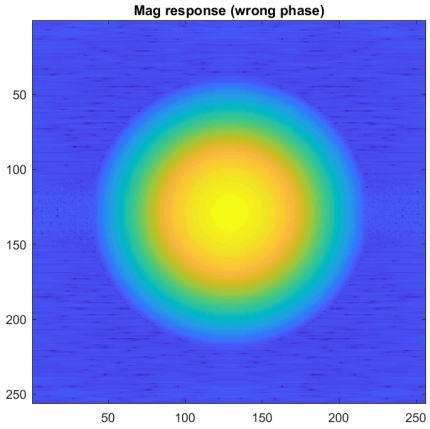
(a)

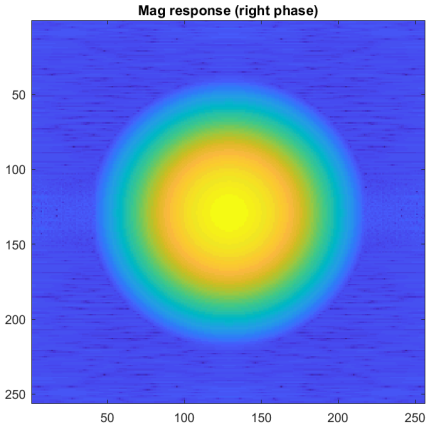
```
m = (-128:1:127);
n = (-128:1:127);
sigma = 4;
%Use matrix multiplication:
h_matrix = 1/(2*pi*sigma^2)*exp(-(m'.^2+n.^2)/(2*sigma^2));
```

```
%Use the property of circular convolution to prevent phase shifting in frequency domain
%Trick method by fftshift function, or you can move their position manually
h_matrix_right_phase = fftshift(h_matrix);
h_matrix_wrong_phase = h_matrix;
fig = figure;
subplot(1,2,1);
surf(h_matrix_right_phase);
subplot(1,2,2);
surf(h_matrix_wrong_phase);
set(fig, 'Position', [0 0 1250 500]);
hold off;
```

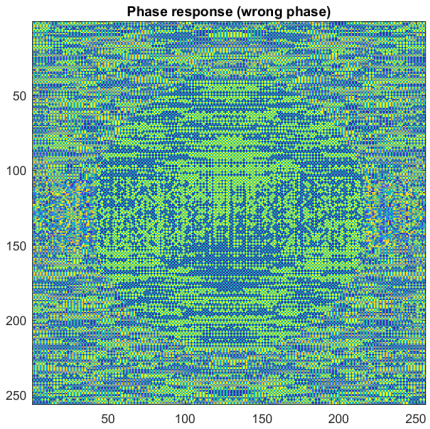


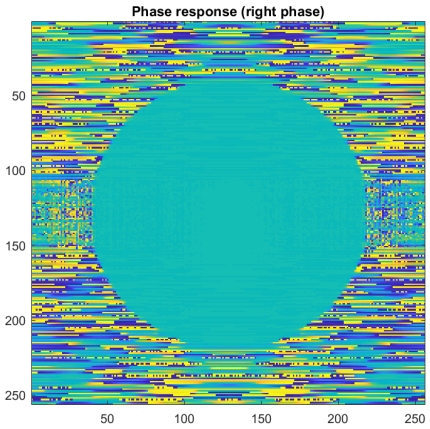

```
H_right_phase = fft2(h_matrix_right_phase);
H_wrong_phase = fft2(h_matrix_wrong_phase);
%mag
H_mag_wrong_phase = 20*log10(abs(H_wrong_phase));
H_mag_right_phase = 20*log10(abs(H_right_phase));
fig = figure;
subplot(1,2,1);
imagesc(fftshift(H_mag_wrong_phase)); title("Mag response (wrong phase)");
subplot(1,2,2)
imagesc(fftshift(H_mag_right_phase)); title("Mag response (right phase)");
set(fig, 'Position', [0 0 1250 500]);
hold off;
```






```
%phase
H_pha_wrong_phase = angle(H_wrong_phase);
H_pha_right_phase = angle(H_right_phase);
fig = figure;
subplot(1,2,1);
imagesc(fftshift(H_pha_wrong_phase)); title("Phase response (wrong phase)");
subplot(1,2,2)
imagesc(fftshift(H_pha_right_phase)); title("Phase response (right phase)");
set(fig, 'Position', [0 0 1250 500]);
hold off;
```

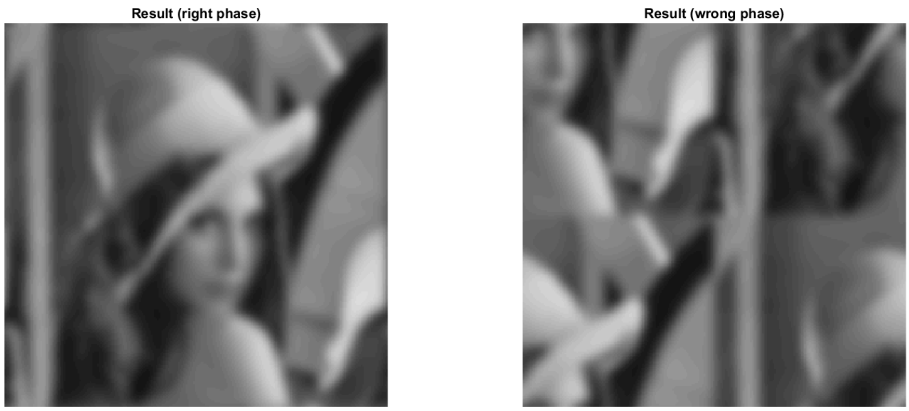




(b)

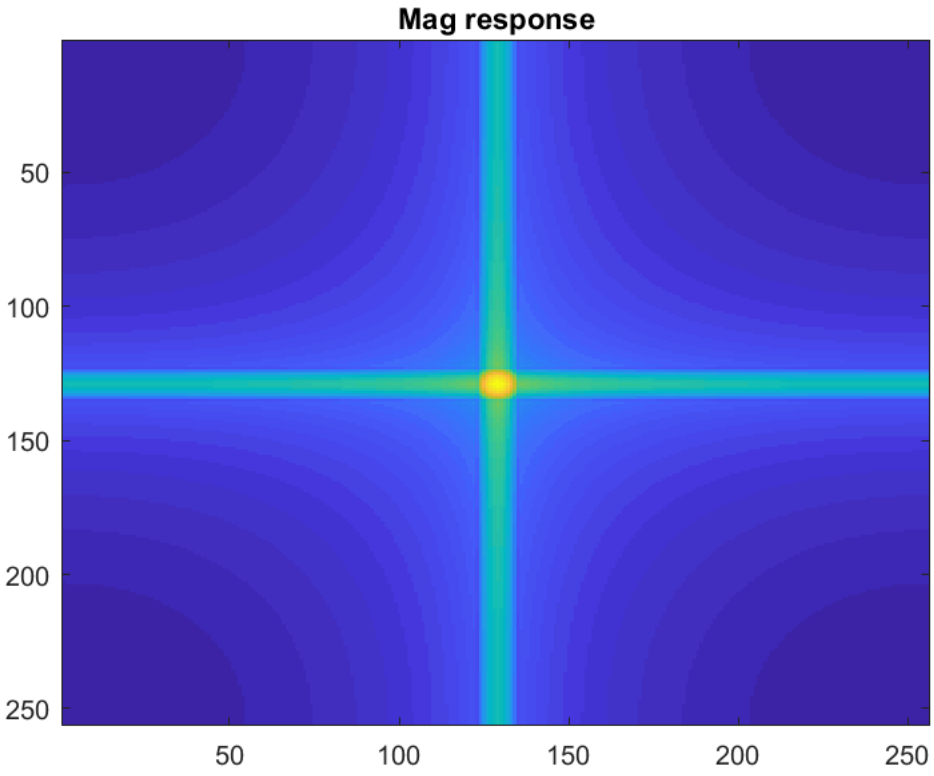
```
%Read Lena image (remeber to change the type "uint8"-> "double" and normalize)
lena = double(imread('lena.jpg'))/255.0;
fig = figure;
imshow(lena);
```

```
%Be careful the phase shifting issue:
%Use H with "right phase"
LENA = fft2(lena);
LENA_result_right_phase = LENA.*H_right_phase;
lena_result_right_phase = ifft2(LENA_result_right_phase);
fig = figure;
subplot(1,2,1);
imshow(lena_result_right_phase); title("Result (right phase)");
%Use H with "wrong phase"
LENA = fft2(lena);
LENA_result_wrong_phase = LENA.*H_wrong_phase;
lena_result_wrong_phase = ifft2(LENA_result_wrong_phase);
subplot(1,2,2);
imshow(lena_result_wrong_phase); title("Result (wrong phase)");
set(fig, 'Position', [0 0 1250 500]);
hold off;
```

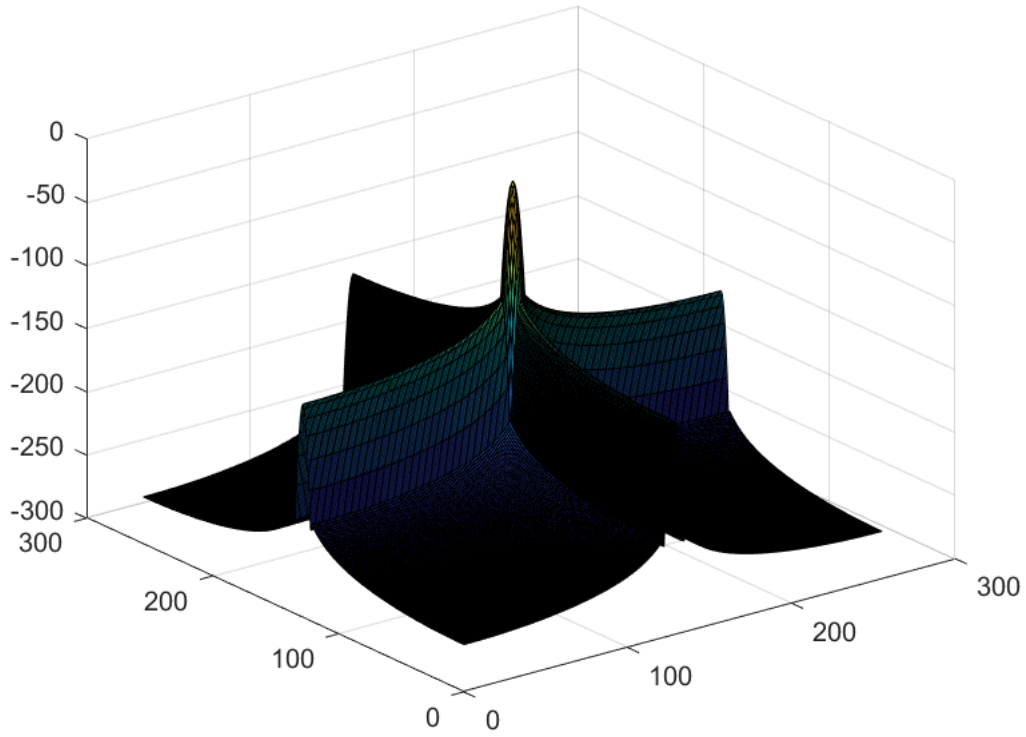


(c)

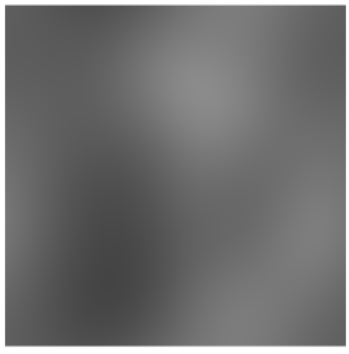
```
sigma = 32;
h_matrix = 1/(2*pi*sigma^2)*exp(-(m'.^2+n.^2)/(2*sigma^2));
h_matrix = fftshift(h_matrix);
H = fft2(h_matrix);
%mag
H_mag = 20*log10(abs(H));
fig = figure;
imagesc(fftshift(H_mag)); title("Mag response");
```




```
figure;
surf(fftshift(H_mag));
```



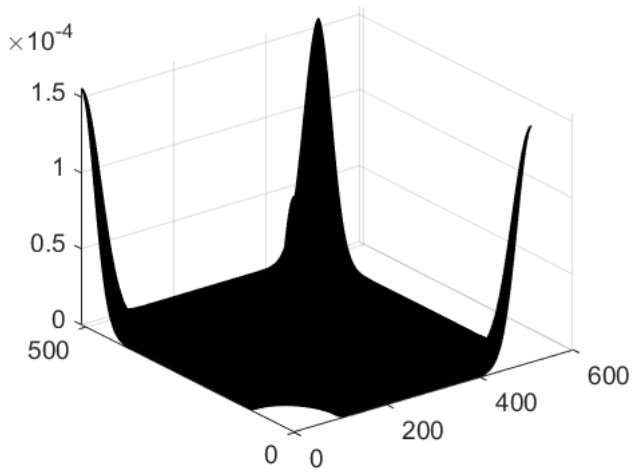

```
LENA_result = LENA.*H;
lena_result = ifft2(LENA_result);
imshow(lena_result);
```



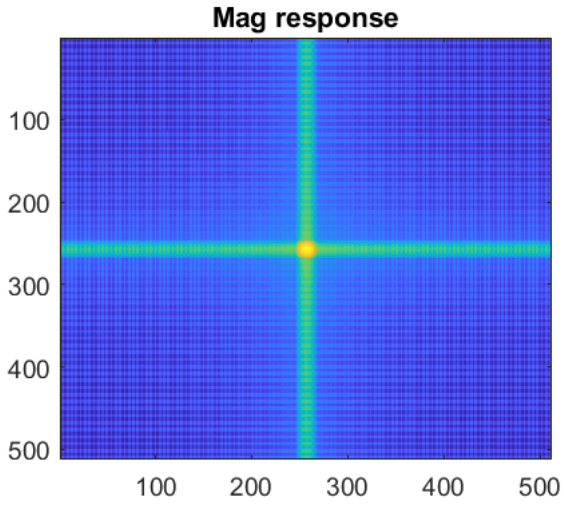
(d)

```
%Filter:
sigma = 32;
```

```
h_matrix = 1/(2*pi*sigma^2)*exp(-(m'.^2+n.^2)/(2*sigma^2));
% Padding before shifting in time domain
h_matrix = padarray(h_matrix, [128, 128], 'both');
h_matrix = fftshift(h_matrix);
surf(h_matrix);
```




```
H = fft2(h_matrix);
%mag
H_mag = 20*log10(abs(H));
imagesc(fftshift(H_mag)); title("Mag response");
```




```
%Lena:
% Padding Lena image
lena_padding = padarray(lena, [128, 128], 'both');
imshow(lena_padding);
```

```
LENA_padding = fft2(lena_padding);
LENA_padding_result = LENA_padding.*H;
lena_padding_result = ifft2(LENA_padding_result);
fig = figure;
subplot(1,2,1);
imshow(lena_result);
subplot(1,2,2);
imshow(lena_padding_result(129:384,129:384));
set(fig, 'Position', [0 0 1250 500]);
hold off
```

(e)

```
%Filter:
sigma = 4;
h_matrix = 1/(2*pi*sigma^2)*exp(-(m'.^2+n.^2)/(2*sigma^2));
h_matrix = fftshift(h_matrix);
H = fft2(h_matrix);
H_high_freq = 1-H;
LENA_high_freq = H_high_freq.*LENA;  %result in (e)
LENA_low_freq = H.*LENA;  %result in (b)
lena_high_freq_result = ifft2(LENA_high_freq);  %result in (e)
lena_low_freq_result = ifft2(LENA_low_freq);  %result in (b)
fig = figure;
subplot(1,3,1);
imshow(lena_high_freq_result); title("Result in (e)");
subplot(1,3,2);
imshow(lena_low_freq_result); title("Result in (b)");
subplot(1,3,3);
imshow(lena_low_freq_result+lena_high_freq_result); title("Result (b) + (e)");
set(fig, 'Position', [0 0 1250 500]);
hold off;
```