Table of Contents

lab4.m

```
clear
delete(allchild(0));
w = linspace(-pi, pi, 11);
x = sequence([1 4 3 -2 6], -1);
%x = sequence([1 5 2 -1 4 1], -2);
```

Problem #1: Even, Odd

```
test_lab4('even(x)');
test_lab4('odd(x)');
test_lab4('trim(plus(even(x), odd(x)))');
```

```
_____________
even(x): sequence O.K.
Your answer:
z=
    data: [3 -1  2  4  2 -1  3]
  offset: -3
_____________
odd(x): sequence O.K.
Your answer:
z=
    data: [-3  1 -1  0  1 -1  3]
  offset: -3
_____________
trim(plus(even(x), odd(x))): sequence O.K.
Your answer:
z=
    data: [1  4  3 -2  6]
  offset: -1
```

Problem #2: DTFT

```
x = sequence([1 1 1], -1);
test_lab4('dtft(x, w)');
```

```
% Simple impulse Caution! check your answer for this.
% It should be a sequence.
x = sequence(1, 0);
test_lab4('dtft(x, w)');

%x = sequence([1 4 3 -2 6], -1)
x = sequence([1 3 -1 -4 1], -2);
test_lab4('dtft(x, w)');

%x = sequence([1 4 3 -2 6], -1)
x = sequence([1+j 0 1-j], -1);
test_lab4('dtft(x, w)-dtft(conj(flip(x)), w)');
```

```
_____________
dtft(x, w): data O.K.
Your answer:

z =

  Columns 1 through 7

   -1.0000   -0.6180    0.3820    1.6180    2.6180    3.0000    2.6180

  Columns 8 through 11

    1.6180    0.3820   -0.6180   -1.0000

_____________
dtft(x, w): data O.K.
Your answer:

z =

     1     1     1     1     1     1     1     1     1     1     1

_____________
dtft(x, w): data O.K.
Your answer:

z =

  Columns 1 through 4

   2.0000 - 0.0000i   0.4271 - 4.1145i  -2.3090 - 6.6574i  -2.9271 -
 6.6574i

  Columns 5 through 8

  -1.1910 - 4.1145i   0.0000 + 0.0000i  -1.1910 + 4.1145i  -2.9271 +
 6.6574i

  Columns 9 through 11

  -2.3090 + 6.6574i   0.4271 + 4.1145i   2.0000 + 0.0000i
```

```
_____________
dtft(x, w)-dtft(conj(flip(x)), w): data O.K.
Your answer:

z =

     0     0     0     0     0     0     0     0     0     0     0
```

Problem #3: Real and Imaginary

```
x = sequence([1 1 1 1 1], -1);
test_lab4('dtft2(x, w)');

%x = sequence([1 4 3 -2 6], -1);
x = sequence([1 2 2 -1 2 1], -2);
test_lab4('dtft2(x, w)');
```

```
_____________
dtft2(x, w): data O.K.
Your answer:

z =

  struct with fields:

    real: [1×11 double]
    imag: [1×11 double]

_____________
dtft2(x, w): data O.K.
Your answer:

z =

  struct with fields:

    real: [3 2.4271 0.0729 -0.9271 3.4271 7 3.4271 -0.9271 0.0729
 2.4271 3]
    imag: [1×11 double]
```

Problem #4: Magnitude and Phase

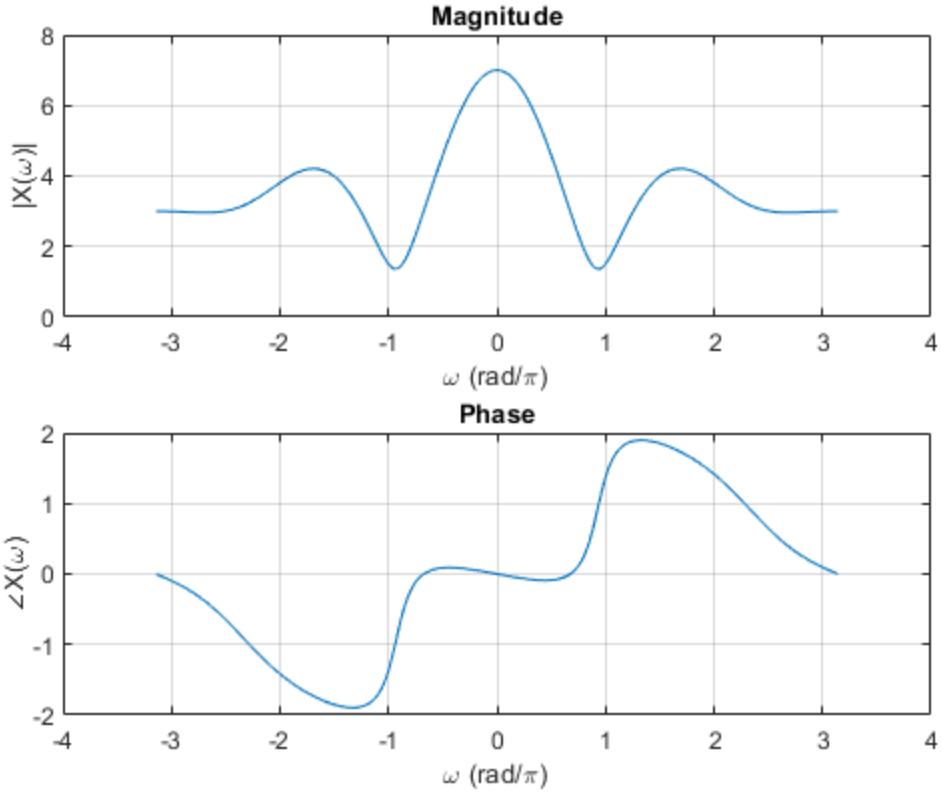
```
test_lab4('mag_phase(dtft2(x, w))');
```

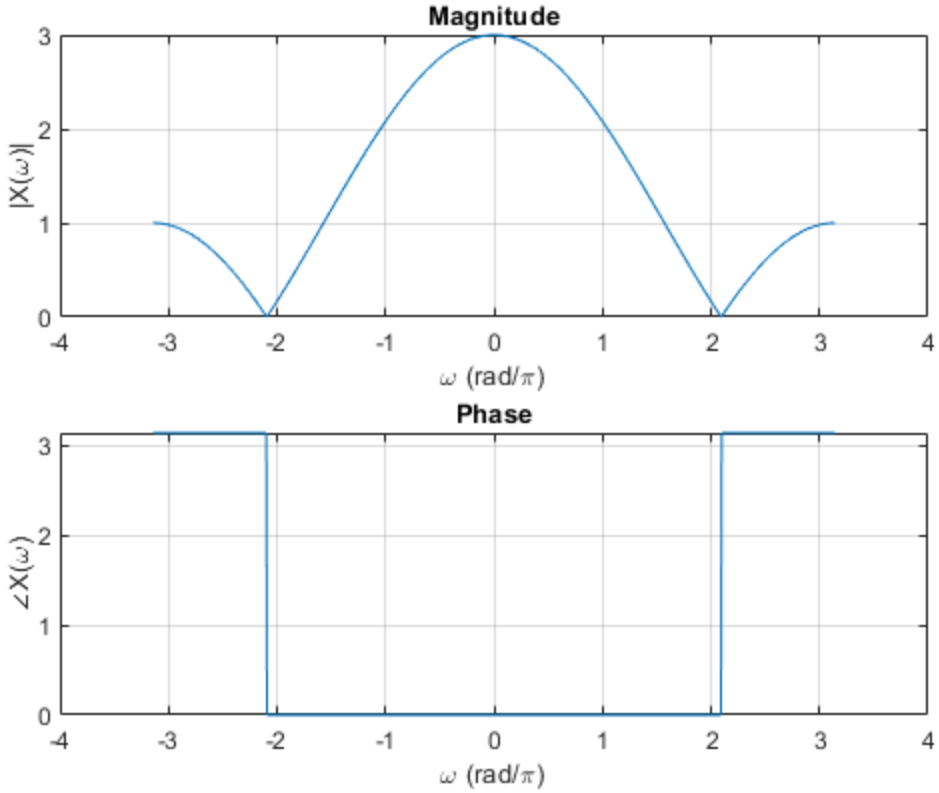
```
_____________
mag_phase(dtft2(x, w)): data O.K.
Your answer:

z =
```

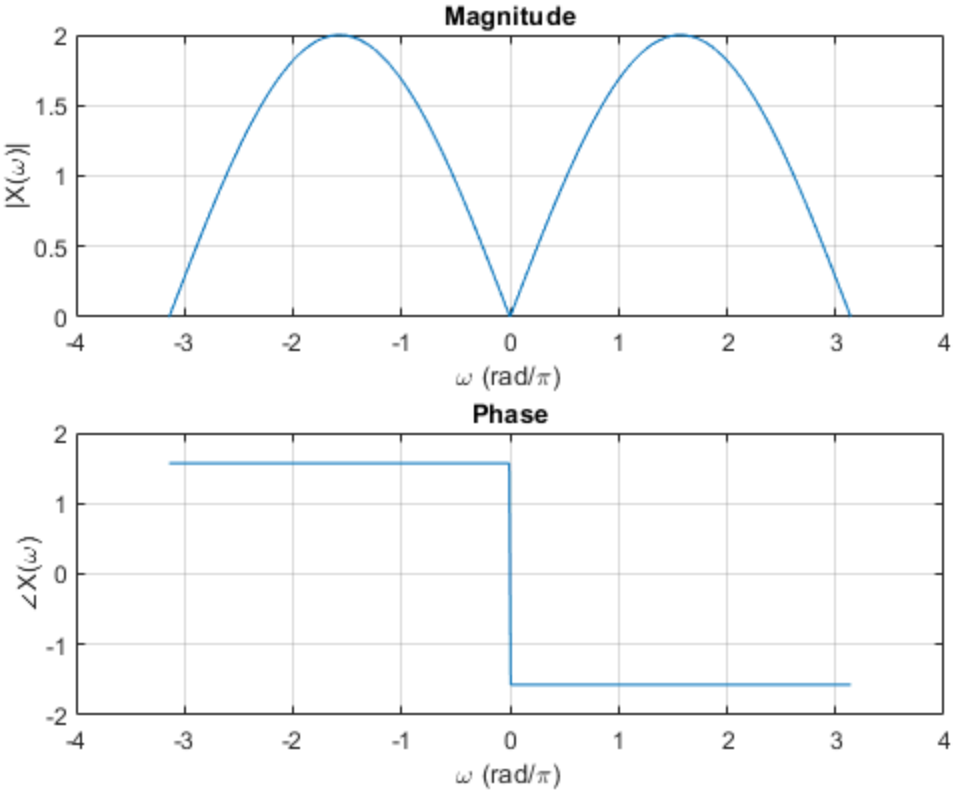
```
  struct with fields:

      mag: [3 3.0000 4.0294 3.0000 3.4299 7 3.4299 3.0000 4.0294
 3.0000 3]
    phase: [1×11 double]
```

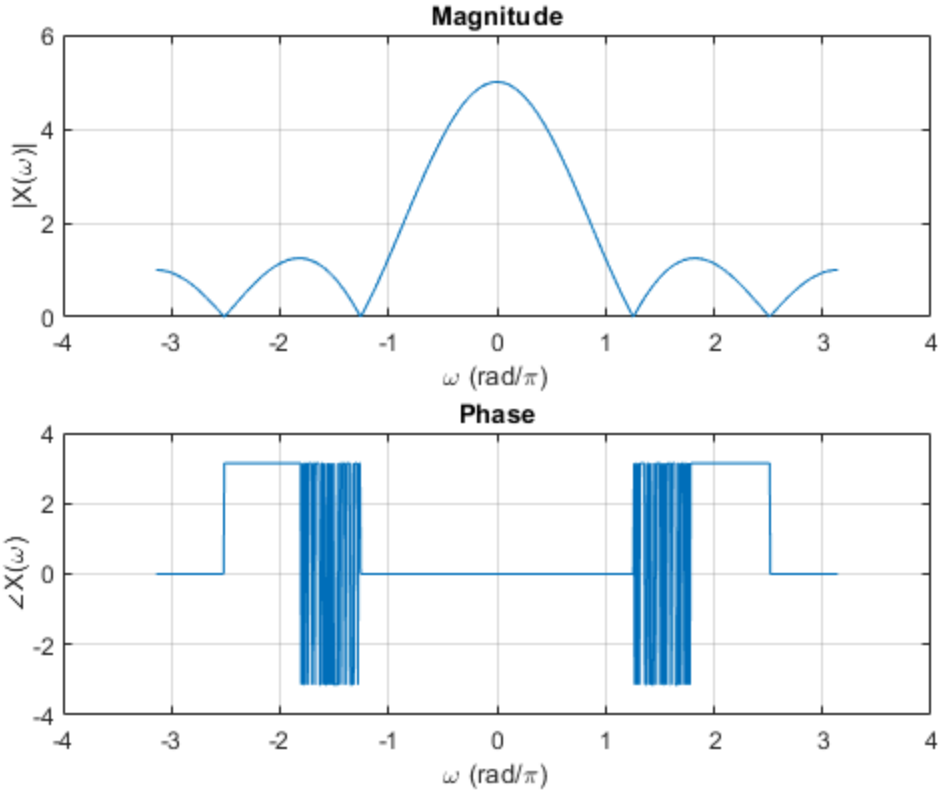
Problem #5 Plotting

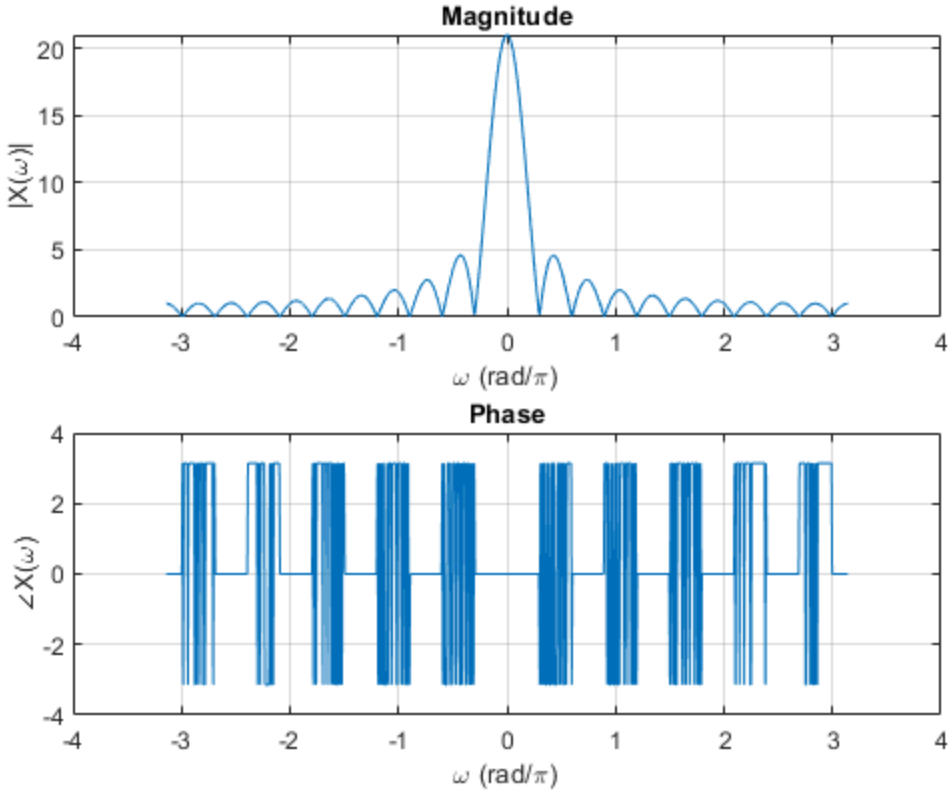
```
w = linspace(-pi, pi, 1001);
plot_magph(x, w);

% This is a purely real and even function.
% What can you say about the phase?
% Specifically why is it either 0 or pi?
x = sequence([1 1 1], -1);
set(figure, 'Color', 'w');
plot_magph(x, w);

% This is a purely real and odd function.
% What can you say about the phase?
% Specifically why is it either +pi/2 or -pi/2?
x = sequence([-1 0 1], -1);
set(figure, 'Color', 'w');
plot_magph(x, w);

% Here are a series of pulse functions.
% What happens to the magnitude of the transform as the pulse gets
 broader?
% You may note that the phase 'chatters' between +pi and -pi at some
 values of w.
% This doesn't look nice and it's confusing. How could you fix this in
 your plot_magph
% program so that the phase doesn't chatter? No biggie if you can't.
% (Hint: it has something to do with a very small imaginary part...).

x = sequence(ones(1, 5), -2);
set(figure, 'Color', 'w');
plot_magph(x, w)

x = sequence(ones(1, 21), -10);
set(figure, 'Color', 'w');
plot_magph(x, w)

x = sequence(ones(1, 101), -50);
set(figure, 'Color', 'w');
plot_magph(x, w)
```

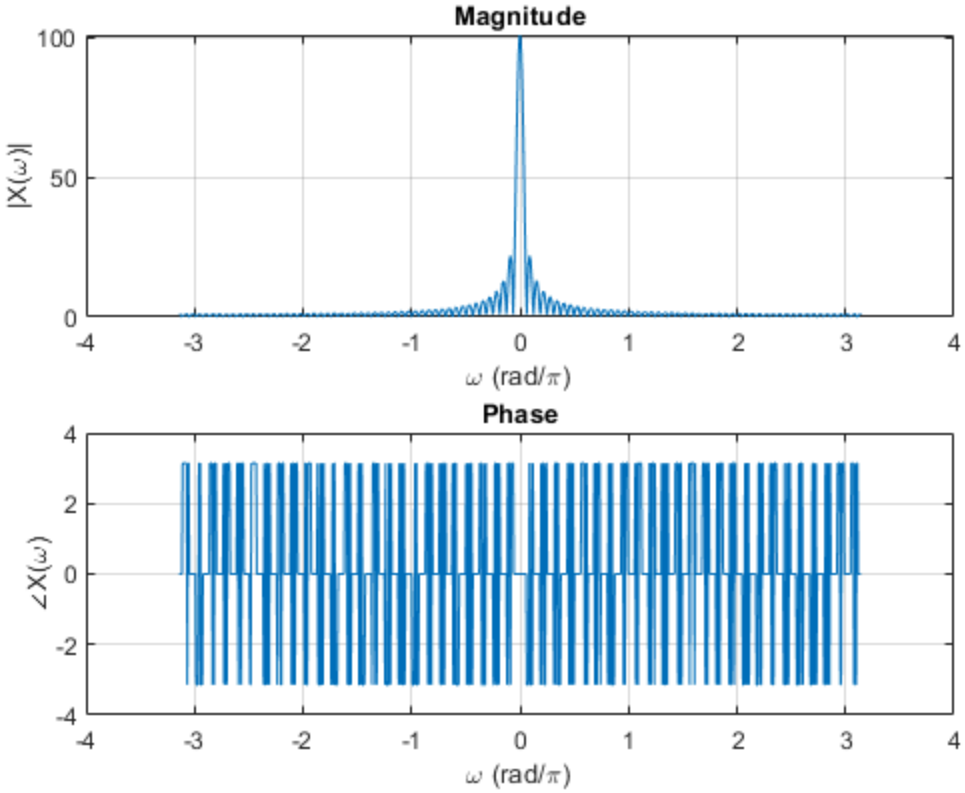

Magnitude
|X(ω)|
ω (rad/π)
Phase
∠X(ω)
ω (rad/π)


Magnitude
|X(ω)|
ω (rad/π)
Phase
∠X(ω)
ω (rad/π)


Magnitude
|X(ω)|
ω (rad/π)
Phase
∠X(ω)
ω (rad/π)


Magnitude
|X(ω)|
ω (rad/π)
Phase
∠X(ω)
ω (rad/π)


Magnitude
20
15
10
5
0
|X(ω)|
-4
-3
-2
-1
0
1
2
3
4
ω (rad/π)
Phase
4
2
0
-2
-4
∠X(ω)
-4
-3
-2
-1
0
1
2
3
4
ω (rad/π)



Print programs

```
disp(' ')
disp('--- dtft.m --------------------------------')
type('dtft')
disp('--- dtft2.m -------------------------------')
type('dtft2')
disp('--- mag_phase.m ---------------------------')
type('mag_phase')
disp('--- plot_magph.m --------------------------')
type('plot_magph')


--- dtft.m --------------------------------

% DTFT
% Periodic result between -pi pi or -2pi 2pi
% Real valued: magnitude response = even function
% Example: x[n] = dirac[n+3]+dirac[n+2]+dirac[n+1]
% X(e^jw) = e^3jw + e^2jw + e^jw = e^2jw(e^jw + 1 + e^-jw) =
 e^2jw(1+2cos(w))
% w is a vector from -pi to pi , not a variable
% A = exp(-j*Q) answer

function y = dtft(x,w)
    lx = length(x.data);
```

```
    n = linspace(x.offset,x.offset+lx-1,lx);
    Q = n'*w;
    y = x.data*exp(-j*Q); % Does this need to be cos(wn) + j sin(wn)
end
--- dtft2.m -------------------------------

% DTFT2 Find real and imaginary parts of DTFT
function y = dtft2(x,w)
    Q = dtft(x,w);
    lx = length(x);
    lw = length(w);
    for i = 1 : lx
        for j = 1 : lw
            Q(i,j);
        end
    end
    y.real = real(Q);
    y.imag = imag(Q);
end
--- mag_phase.m --------------------------

function y = mag_phase(x)
    y.mag = sqrt(power(x.real,2)+power(x.imag,2));
    y.phase = atan2(x.imag,x.real);
end

--- plot_magph.m -------------------------

function plot_magph(x, w)
    z = mag_phase(dtft2(x,w));
    figure();
    subplot(2,1,1);
    plot(w,z.mag);
    grid on;
    title('Magnitude');
    xlabel('\omega (rad/\pi)');
    ylabel('|X(\omega)|');
    hold on;
    subplot(2,1,2);
    plot(w,z.phase);
    grid on;
    title('Phase');
    xlabel('\omega (rad/\pi)');
    ylabel('\angleX(\omega)');
end
```

Published with MATLAB® R2019b