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: [0.5         2.5           2           2           2
 2.5         0.5]
  offset: -3
____________
odd(x): sequence O.K.
Your answer:
z=
    data: [-0.5        -1.5           3           0          -3
  1.5         0.5]
  offset: -3
____________
trim(plus(even(x), odd(x))): sequence O.K.
Your answer:
z=
    data: [1  5  2 -1  4  1]
  offset: -2
```

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 5 2 -1 4 1], -2);
test_lab4('dtft(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   0.6180 - 5.4288i  -2.4721 - 8.0575i  -1.6180 -
 4.5308i

  Columns 5 through 8

   6.4721 + 0.2775i  12.0000 + 0.0000i   6.4721 - 0.2775i  -1.6180 +
 4.5308i

  Columns 9 through 11

  -2.4721 + 8.0575i   0.6180 + 5.4288i   2.0000 + 0.0000i
```

Problem #3: Real and Imaginary

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

%x = sequence([1 4 3 -2 6], -1);
x = sequence([1 5 2 -1 4 1], -2);
test_lab4('dtft2(x, w)');
```

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

z = 

  struct with fields:

    real: [-1 -0.6180 0.3820 1.6180 2.6180 3 2.6180 1.6180 0.3820
 -0.6180 -1]
    imag: [1.2246e-16 7.5687e-17 0 0 0 0 0 0 0 7.5687e-17 1.2246e-16]

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

z = 

  struct with fields:

    real: [2 0.6180 -2.4721 -1.6180 6.4721 12 6.4721 -1.6180 -2.4721
 0.6180 2]
    imag: [1×11 double]
```

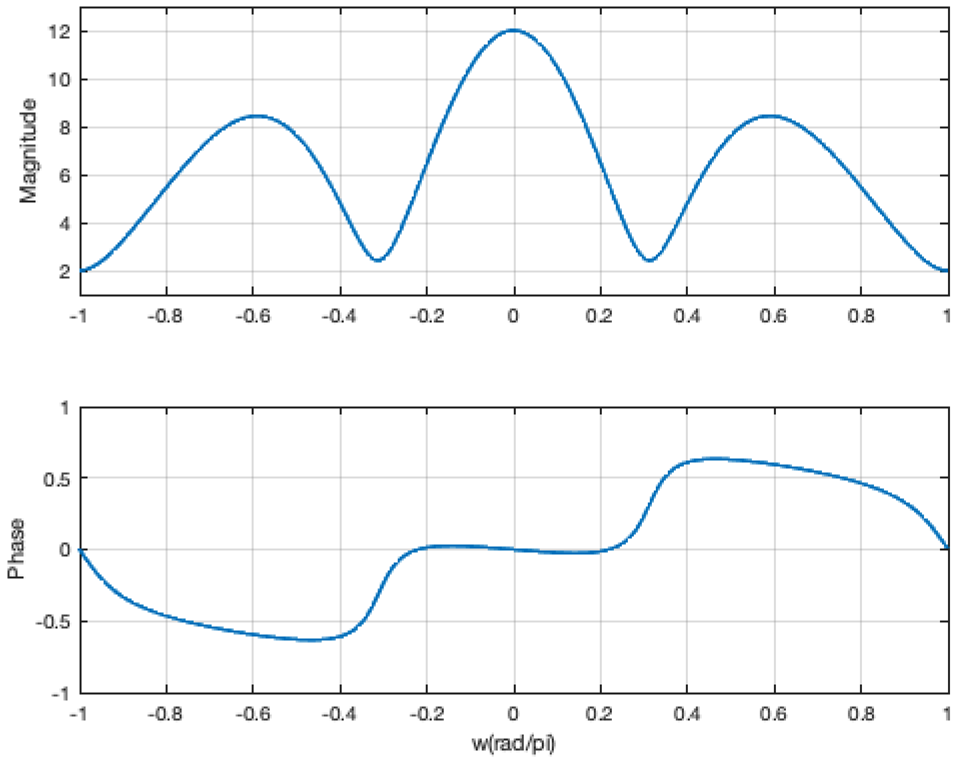
Problem #4: Magnitude and Phase

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

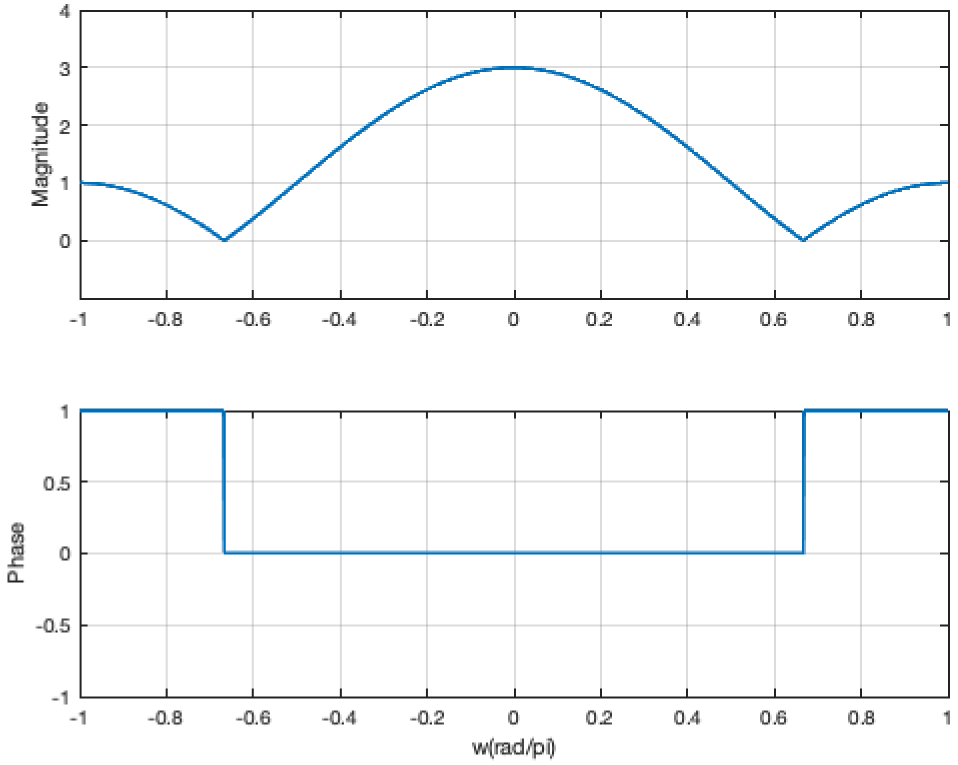
```
____________
mag_phase(dtft2(x, w)): data O.K.
Your answer:

z = 

  struct with fields:

      mag: [2 5.4639 8.4282 4.8110 6.4781 12 6.4781 4.8110 8.4282
 5.4639 2]
    phase: [1×11 double]
```

Problem #5 Plotting

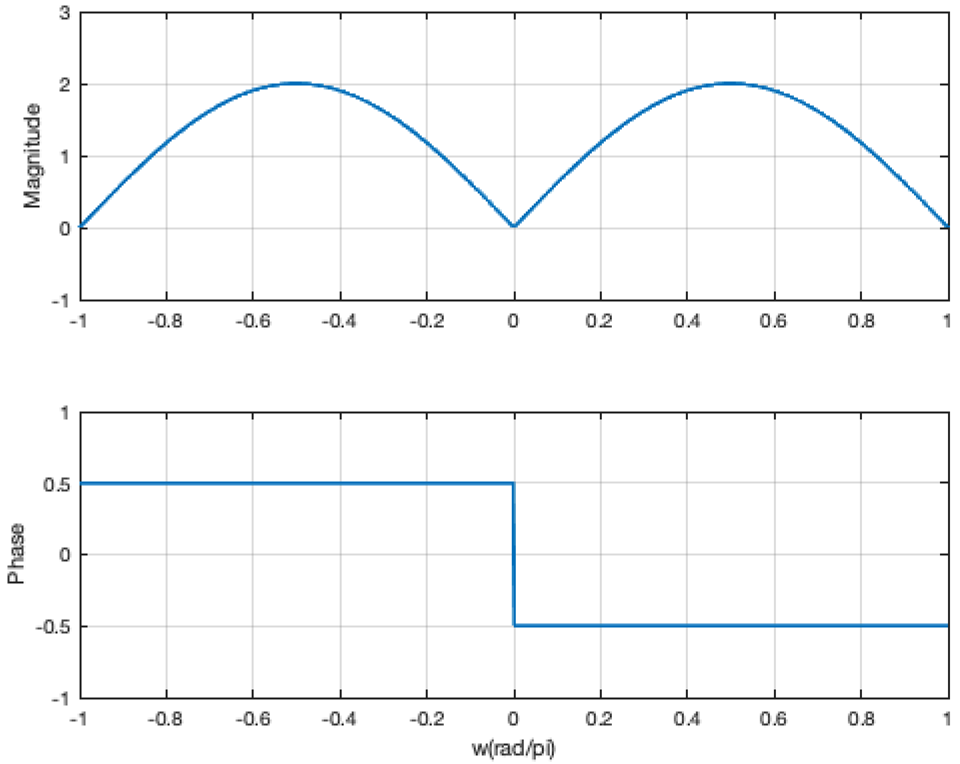
```
w = linspace(-pi, pi, 1001);
plot_magph(x, w);

% 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)
```

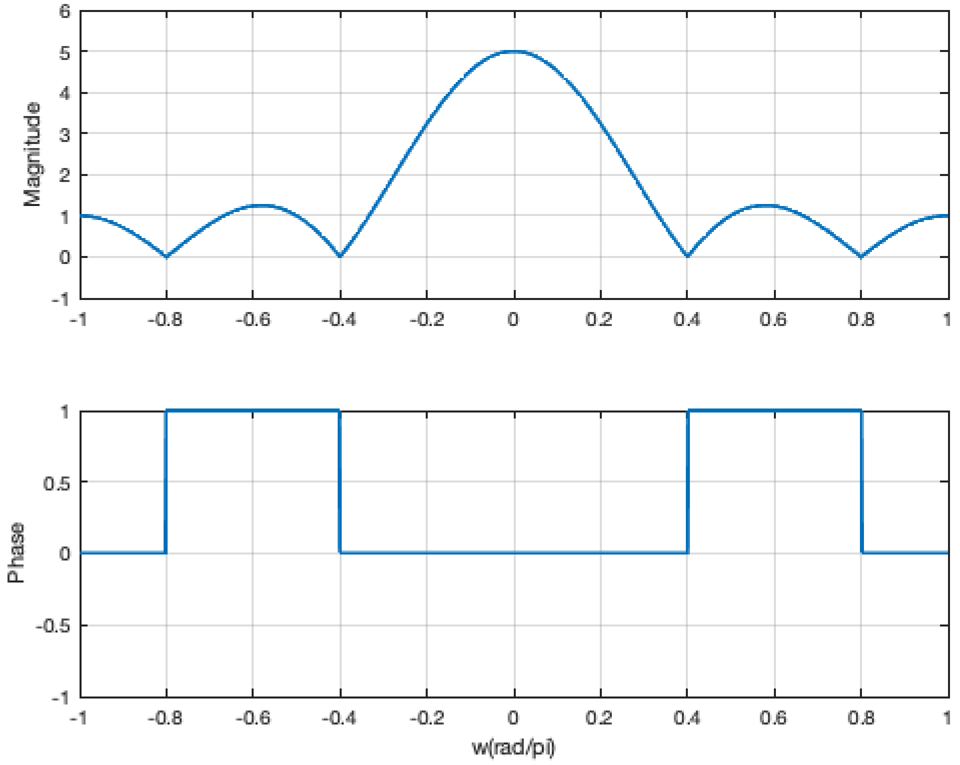


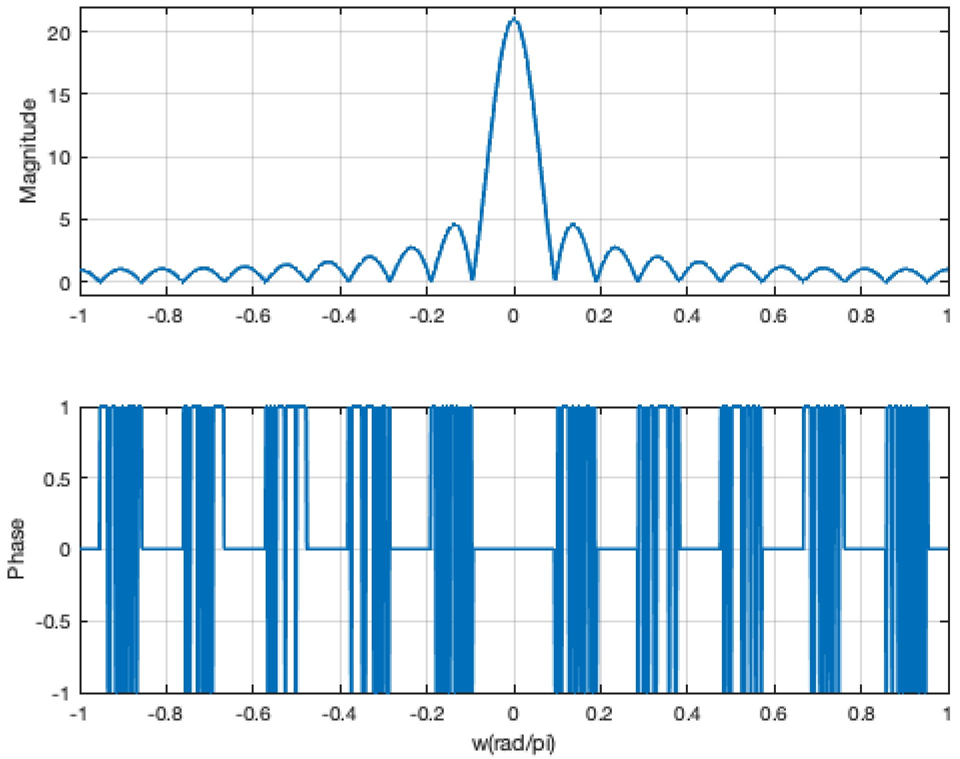
Magnitude
12
10
8
6
4
2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Phase
1
0.5
0
-0.5
-1
w(rad/pi)

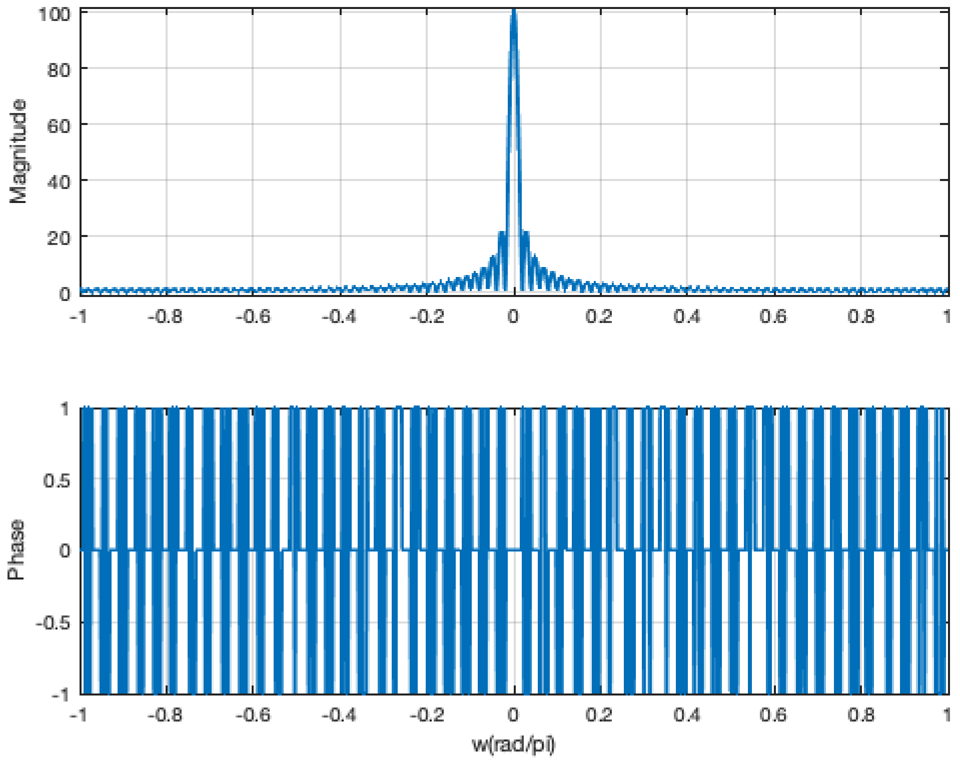


Magnitude
4
3
2
1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Phase
1
0.5
0
-0.5
-1
w(rad/pi)


Magnitude
3
2
1
0
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Phase
1
0.5
0
-0.5
-1
w(rad/pi)


Magnitude
6
5
4
3
2
1
0
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Phase
1
0.5
0
-0.5
-1
w(rad/pi)


Magnitude
20
15
10
5
0
Phase
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
w(rad/pi)


Magnitude
100
80
60
40
20
0
Phase
1
0.5
0
-0.5
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
w(rad/pi)

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

function y = dtft(x, w)

% DTFT Evaluate the DTFT of Matlab sequence object, x, at radial
 frequencies
% given by double array w. Return a double array, y.
n = [x.offset: x.offset + length(x.data)-1]; %sequence of offsets
Q = n'*w; %transpose n to multiply with w
y = x.data*exp(-1i*Q);  %y = (x)e^(-jw*x.offset)
end
--- dtft2.m -------------------------------

function y = dtft2(x,w)
% DTFT2 Evaluate the DTFT of Matlab sequence object, x, at frequencies
% given by array w.
% Return values are a structure with y.real (real part) and y.imag
% (imaginary part)

n = (x.offset:x.offset + length(x.data)-1);
Q = n'*w;
dtft = x.data*exp(-1i*Q);

    for n=1:length(dtft)
        mag = abs(dtft(n)); %magnitude = sqrt[((x.real)*(x.real)) +
 ((x.imag)*(x.imag))]
        ang = angle(dtft(n));
        y.real(n) = mag*cos(ang);
        y.imag(n) = mag*sin(ang);
    end
end


--- mag_phase.m ---------------------------

function y = mag_phase(x)
% MAG_PHASE Input argument is a structure with x.real and x.imag
% Return values are y.mag (magnitude) and y.phase (phase in radians)

mag = sqrt(((x.real).*(x.real))+((x.imag).*(x.imag)));
```

```
P = zeros(1,length(x.real));

    for n=1:length(x.real)
        P(1,n) = atan2(x.imag(n), x.real(n));
        y.mag = mag;
        y.phase = P;
    end
end
--- plot_magph.m -------------------------

function plot_magph(x,w)
% PLOT_MAGPH Plot the magnitude and phase of Matlab sequence object, x
% at frequencies given by array w

z = mag_phase(dtft2(x, w));

subplot(2, 1, 1),
plot(w/pi,z.mag,'linewidth',2);
grid on;
ylabel('Magnitude');
ylim([min(z.mag)-1 max(z.mag)+1]);  % +/- y limits set to magnitudes

subplot(2, 1, 2),
plot(w/pi, z.phase/pi,'linewidth',2);
ylim([-1 1]);   %highest possible values of phase
grid on;
ylabel('Phase');
xlabel('w(rad/pi)');

end
```

Published with MATLAB® R2018b