DFT_CH8_Problem Session 2

a. x_0[n] = 1 n=0 ; 0 n – 1,2…9 Unit Impulse Result is ALL FREQUENCIES – This is why the impulse response is so important.

P-8.1 **DSP First 2e**

(a) The DTFT of $x_0[n]$ is $X_0(e^{j\hat{\omega}}) = 1$ for all $\hat{\omega}$. Therefore, $X_0[k] = X_0(e^{j(2\pi k/10)}) = 1$ for $k = 0, 1, \ldots, 9$. In MATLAB

```
>> X0=fft([1,zeros(1,9)])
X0 = 1     1     1     1     1     1     1     1     1     1
```

P-8.3b 4-Point pulse with N = 12: This result starts from the positive frequency values. F_max is at location N/2 .

b.

(b) This can be looked up in Table 8-1. It is the third entry with $L = 4$ and $N = 12$ so

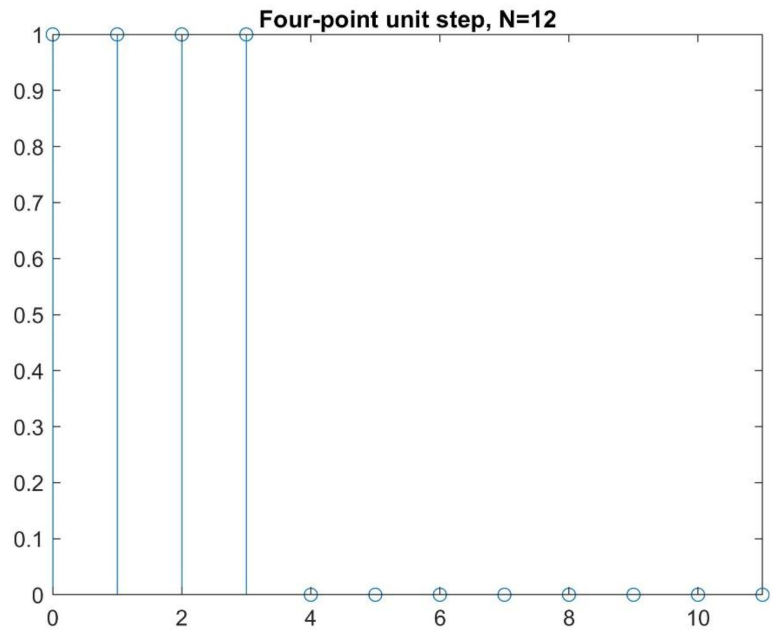
$$Y_1[k] = \frac{\sin(\pi k/3)}{\sin(\pi k/12)} e^{-j\pi k/4}$$

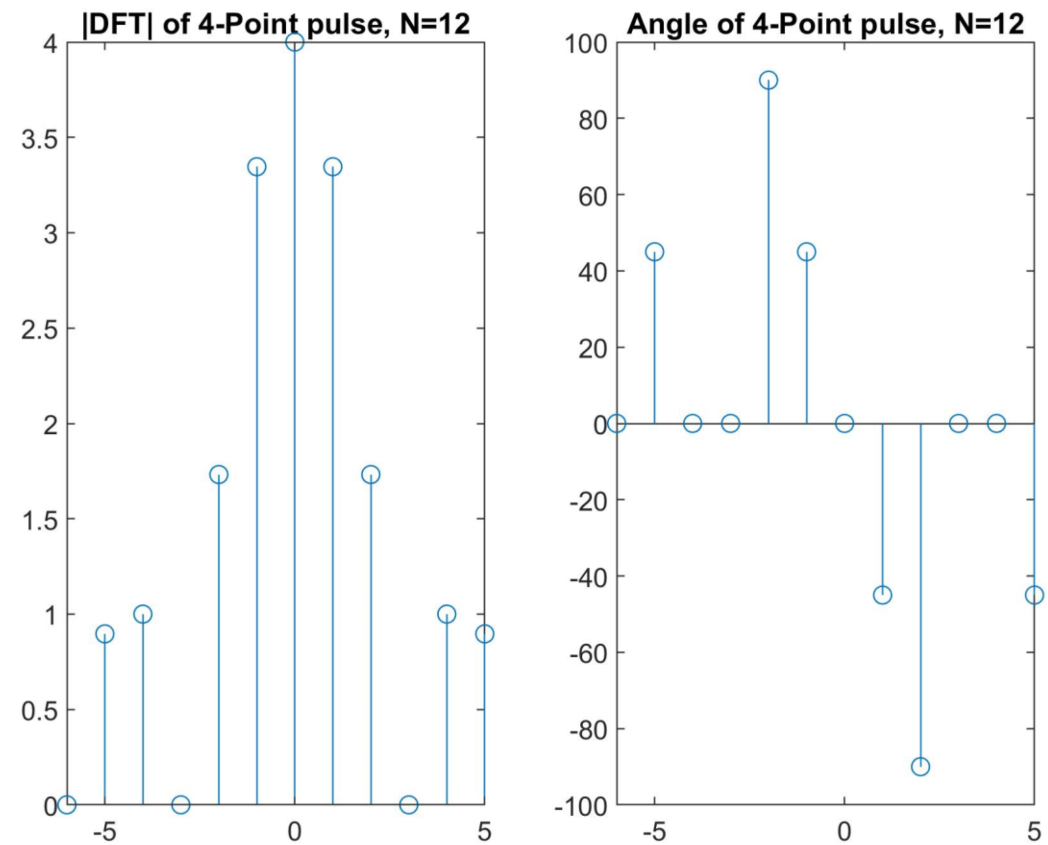
MATLAB verification:

```
>> Y1 = fft([ones(1,4),zeros(1,8)])
Y1 =  4.0000 + 0.0000i   2.3660 - 2.3660i   0.0000 - 1.7321i
      0.0000 + 0.0000i   1.0000 + 0.0000i   0.6340 - 0.6340i
      0.0000 + 0.0000i   0.6340 + 0.6340i   1.0000 + 0.0000i
      0.0000 + 0.0000i   0.0000 + 1.7321i   2.3660 + 2.3660i
```

```
%% DFT of Exercise 8_3 DSPF
%
clc,clear all, clf
L=4;  % 1's
N=12; % Number of Points
n=[0:1:11]  % Start at zero
x=([ones(1,4),zeros(1,8)])
figure(1)
stem(n,x),xlim([0,N-1]),title('Four-point unit step, N=12')
%% Spectrum
FFT1= fft(x,N)
FFT1shift=fftshift(FFT1)
FFTmag=abs(FFT1shift)
FFTangle=(180/pi)*angle(FFT1shift)
f=linspace(-N/2,N/2-1,N)
figure(2),subplot(1,2,1)
stem(f,FFTmag), title('|DFT| of 4-Point pulse, N=12')
```

```
subplot(1,2,2)
stem(f,FFTangle),,title('Angle of 4-Point pulse, N=12')
```





|DFT| of 4-Point pulse, N=12
4
3.5
3
2.5
2
1.5
1
0.5
0
-5
0
5
Angle of 4-Point pulse, N=12
100
80
60
40
20
0
-20
-40
-60
-80
-100
-5
0
5

From TLH Chapter 11

MATLAB Script

```
Example 11.2
function [FT,FTmag,FTang] = clfftf(ft,N,Ts)
% CALL: [FT,FTmag,FTang] = clfftf(ft,N,Ts) Compute the DFT
%  approximation of the Fourier Transform
% Inputs:
% ft Sampled function of time f(nTs)
% N Number of sample points
% Ts Sample interval in seconds
% Outputs:
% FT Approximate Fourier transform using DFT
% FTmag Magnitude of spectrum
% FTang Phase in degrees
% Determine the two-sided spectrum
FT1=Ts*(fft(ft,N));            % Scale to approximate FT
FT=fftshift(FT1);              % Shift 0 to center
%
% Compute the magnitude and phase for the frequency values
%    in hertz fs=1/(N*Ts); fmax=1/(2*Ts)
%
FTmag=abs(FT);                 % Magnitude
FTang=(180/pi)*angle(FT);      % Phase in degrees
```

MATLAB Script

```
Example 11.3
% EX11_3.M  Compute and plot the DFT of f(t)=exp(-t)
%  Creates f(t) sampled each Ts seconds for T seconds
%  Input:  N   -number of points input
%          T   -Period of signal
%      t0 =0   -start of time points
%  Calls clfftf to compute DFT
N=input('Number of points N= ')  % Sample N points
T=input('Period of signal T= ')
```

```
Ts=T/N;                             % Sampling interval
% Form the vector of time points and f(n*Ts)
t0=0;                               % Start of signal
ts=(t0:Ts:Ts*(N-1));                % Compute N points
ft=exp(-ts);
% Determine the spectrum
[Fft,Ffmag,Ffang]=clfftf(ft,N,Ts);
% Compute the frequency values in hertz fs=1/(N*Ts); fmax=1/(2*Ts)
%
fs=1/(N*Ts);                        % Frequency spacing
f=fs*linspace(-N/2,N/2-1,N);        % N points in frequency
% Plot Fexact and DFT result
w=2*pi*f;
Fexact=1./(sqrt(1+w.^2));           % Magnitude
Thetaex=-(180/pi)*atan(w);          % Angle in degrees
clf
subplot(2,1,1),plot(f,Fexact(1:N),'--',f,Ffmag(1:N));
title(['FT and DFT of exp(-t), N=',num2str(N), ' T= ',num2str(T),' sec'])
xlabel('Frequency in hertz')
ylabel('FT and DFT')
legend('FT','DFT')
subplot(2,1,2),plot(f,Thetaex(1:N),'--',f,Ffang(1:N));
xlabel('Frequency in hertz')
ylabel('Phase FT and DFT')
legend('FT','DFT')
```

Let’s Look at the Values

$L = 4 \quad N = 12$

$X[k] =$

$x[0]\, e^{-j\phi k} + x[1]\, e^{-j \frac{2\pi}{12} k \cdot 1}$

$+ x[2]\, e^{-j \frac{2\pi}{12} \cdot 2k} + x[3]\, e^{-j \frac{2\pi}{12} \cdot 3k}$

$+ \phi + 0 + 0 + \cdots 0$ 12 terms

$X[0] = x[0] + x[1]e^{-j\phi} + x[2]\, e^{-j\phi} + x[4]e^{-j\phi}$

$+ 0 + \cdots 0$

$= 1 + 1 + 1 + 1 + 0 + 0 \cdots$

$= 4 + j\phi$

$\vdots$

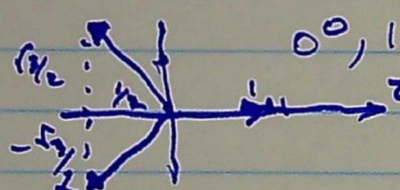
$X[4] = x[0] + x[1]\, e^{-j \frac{\pi}{6} \cdot 4 \cdot 1} + x[2]\, e^{-j \frac{\pi}{6} \cdot 4 \cdot 2}$

$+ x[3]\, e^{-j \frac{\pi}{6} \cdot 3 \cdot 4} + 0 \cdots$

$= \overset{=1}{x[0]} + 1\left[-\tfrac{1}{2} + j\tfrac{\sqrt{3}}{2}\right] + 1\left[-\tfrac{1}{2} - j\tfrac{\sqrt{3}}{2}\right] + 1 \cdot 1$

$0°, 120°, 240°, 360°$

VECTORS



$= +1$ ✓

This does the DFT without shifting the index.

Let’s See how MATLAB does.

FFT1shift =

0.0000 + 0.0000i   X[-6]

0.6340 + 0.6340i

1.0000 + 0.0000i

0.0000 + 0.0000i

0.0000 + 1.7321i

2.3660 + 2.3660i   X[-1]

4.0000 + 0.0000i   X[0]   This would be the dc value

2.3660 - 2.3660i   X[1]

0.0000 - 1.7321

0.0000 + 0.0000i

1.0000 + 0.0000i

0.6340 - 0.6340i   X[5]   would be f_max with fftshift

Since this is a real-valued sequence in x(t) we expect

X[0] and X[N/2} are real – not complex,

X[-k] = X*[k]