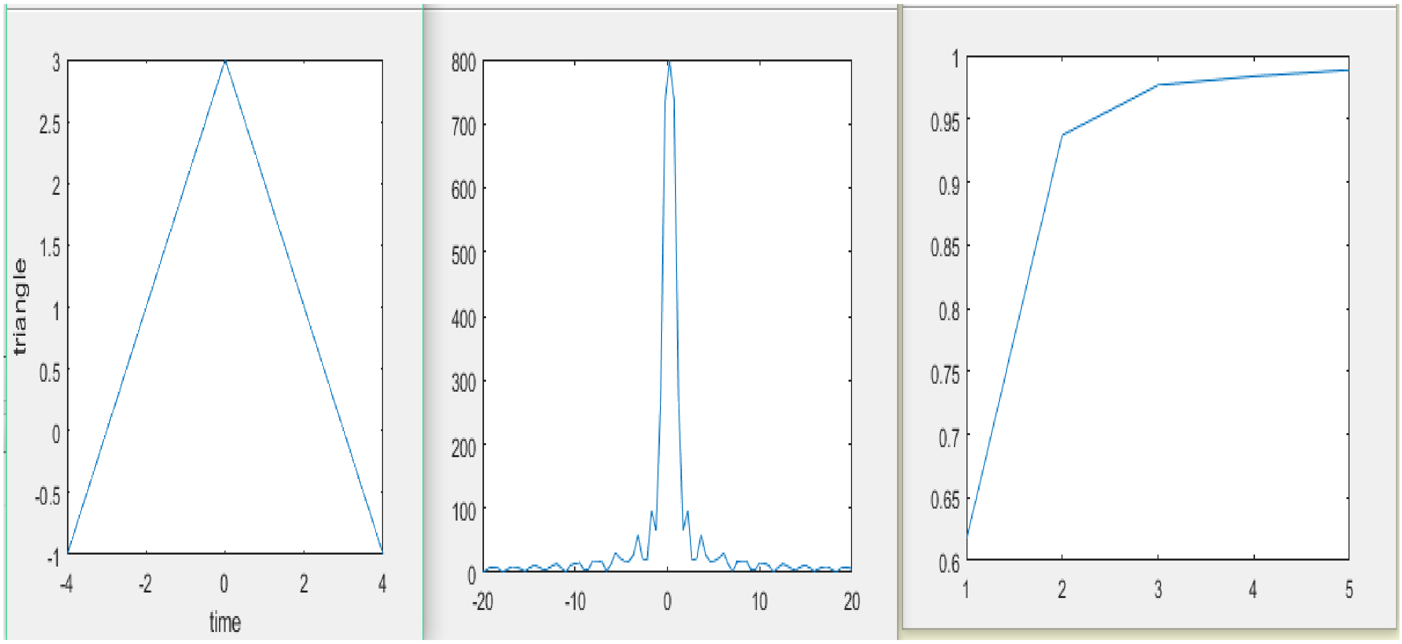
Assignment – 3

Name : Prashant Kumar Mahanta **Rollno. :** 201601066

1 . Using the matlab function fft

```
clc
clear all;
close all;
%%
Fs=500;
To=3;
t=-4:0.01:4;
last=length(t);
triangle=zeros(size(t));
for i=1:last
    if t(i)>=0
        triangle(i)= t(i)*-1+To;
        %T-t     for  0 < t < T
    else
        triangle(i)= t(i)+To;
        %t + T   for  -T < t < 0
    end
end
figure;
plot(t,triangle);
xlabel('time');
ylabel('triangle');
%fourier transform using the matlab function fft
l = length(triangle);
NFFT =2^(nextpow2(l));
ut = abs(fft(triangle,NFFT));
fvec = (Fs/2)*linspace(-1,1,NFFT);
ut = fftshift(ut);
Ene=sum(abs(ut).^2)
%Ploting X(omega) and frequency
figure;
plot(fvec,ut);
xlim([-20 20]);
in=1;
Eg=0;
sf = NFFT/2;
freq = NFFT/2+1
%While loop to check when energy upto 99%
while Eg/Ene <= 0.99
    Eg=sum(abs(ut(sf:freq)).^2);
    freq=freq+1;
    eN(in)=Eg;
    sf = sf-1;
    in = in+1;
end
%Bandwidth = 2* frequency
Bandwidth=2*(freq-1);
display(Bandwidth) % new bandwidth
figure;
```

GRAPH:



frequency = 513
Bandwidth = 1038

Take a time vector t which ranges from - 4 to 4 with interval of 0.01 and T= 3. For negative values of t, assign variable triangle = t + T and for positive values of t, assign variable triangle = T-t. After doing this if we plot triangle and t then we will get a triangle function. After that by using fft( ) function we will get its fourier transformed and stored in some variable, then by using fftshift( ) function we can shift it to origin. Now we can plot X(w) and we know that it is not a bandlimted.

So to bandlimit this we need to consider the signal values only till, where we 99% of the energy. By doing this the signal gets bandlimited and values of the signal will be approximately equal to the original signal.

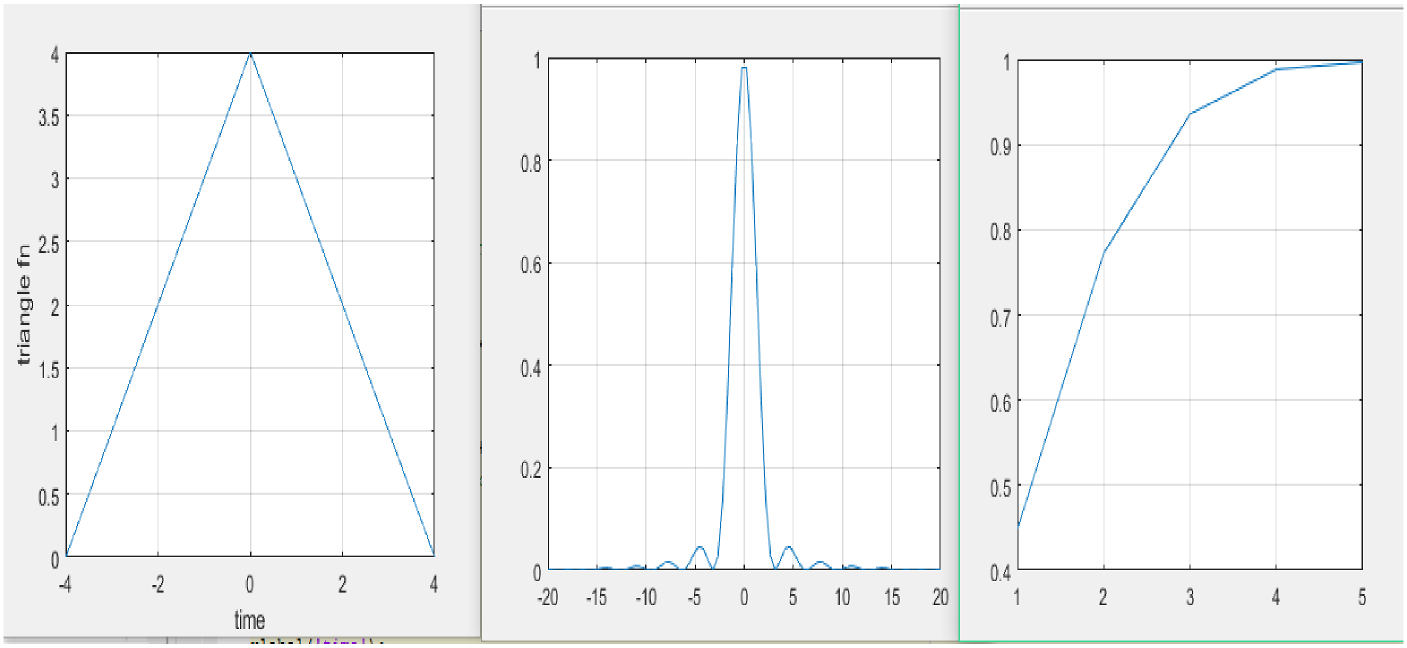
=>By this process we can bandlimit a signal which is originally strictly not bandlimited.

2. Using theoretical definition.

```
clc
clear all;
close all;
%%
Fs=500;
T=3;
t=-4:0.01:4;
last=length(t);
triangle=zeros(size(t));
%Ploting triangle function
for i=1:last
    if t(i)>=0
        triangle(i)= t(i)*-1+T;
        %T-t     for  0 < t < T
    else
        triangle(i)= t(i)+T;
        %t + T   for  -T < t < 0
    end
end
figure;
plot(t,triangle),grid;
xlabel('time');
ylabel('triangle fn');
%fourier transform using theoritical defination
l = length(triangle);
NFFT =2^(nextpow2(l));
fvec = Fs/2*linspace(-1,1,NFFT);
ut=(sin(fvec)./fvec).*(sin(fvec)./fvec);% SINC function
Ene=sum(abs(ut).^2)
%Ploting X(omega) vs frequency
figure;
plot(fvec,ut),grid;
xlim([-20 20]);
in=1;
E=0;
s = NFFT/2;
freq = NFFT/2+1
%While loop to check when energy upto 99%
while E/Ene <= 0.99
    E=sum(abs(ut(s:freq)).^2);
    freq=freq+1;
    En(in)=E;
    in = in+1;
    s = s-1;

end
%Bandwidth = 2* frequency
Bandwidth=2*(freq-1);
display(Bandwidth);% new bandwidth
figure;
plot(En/Ene),grid;
```

GRAPH:



frequency = 513

Bandwidth =1034

Take a time vector t which ranges from - 4 to 4 with interval of 0.01 and T= 3. For negative values of t, assign variable triangle = t + T and for positive values of t, assign variable triangle = T-t. After doing this if we plot triangle and t then we will get a triangle function. After that we need its fourier transformation signal in this process we are using theoretical method. The Fourier transform of triangle(t) is not band limited. Thus, any chosen value of Ts will cause aliasing. The values of the sinc function go fast to zero, so that one could compute an approximate maximum frequency that covers 99% of the energy of the signal.

By doing this the signal gets bandlimited and values of the signal will be approximately equal to the original signal.

=>By this process we can bandlimit a signal which is originally strictly not bandlimited.

<u>Conclusion:</u>

After plotting both the signal we conclude that both the methods i.e., by using fft() function and by using theoretical method we got same result for same values of timevector t (-4 to 4) and T=3 . <u>Frequency</u> = 513 and bandwidth = 1034 in both cases.