

Digital SIgnal Processing Lab

Experiment NO: 01



**Submitted By** : Md.Habibur Rahman

**Roll :** 40

Experiment Name:

**Consider a continuous time Signal , x(t)=sin(2\*pi\*1000\*t)+1/2\*sin(2 \*pi\*2000\*t+3\*pi/4),**

**Determine sampled signal x(n\*Ts)=x(n) , using The Matlab taking n=8 samples at sampling rate,**

**Fs=8000Hz (sample/second).The Sample signal , x(n)=x(n\*Ts)=sin(2\*pi\*1000\*n\*Ts)+1/2\*sin(2\*pi\*2000\*n\*Ts+3\*pi/4)  
where Ts=1/Fs=1/8000 sec is in the sampling period**.

Objective:

The main purpose of this experiment is to transform the given signal from time domain to frequency domain.

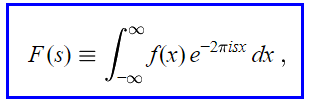
Theory:

Fourier Transform:

The Fourier transform is important in mathematics, engineering, and the physical sciences.  Its discrete counterpart, the Discrete Fourier Transform (DFT), which is normally computed using the so-called Fast Fourier Transform (FFT), has revolutionized modern society, as it is ubiquitous in digital electronics and signal processing.  Radio astronomers are particularly avid users of Fourier transforms because Fourier transforms are key components in data processing (e.g., periodicity searches) and instruments (e.g., antennas, receivers, spectrometers), and they are the cornerstores of interferometry and aperture synthesis.

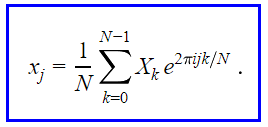
The Fourier transform is a reversible, linear transform with many important properties.  For any function *f*(*x*) (which in astronomy is usually real-valued, but *f*(*x*) may be complex), the Fourier transform can be denoted *F*(*s*), where the product of *x* and *s* is dimensionless.  Often *x* is a measure of time *t* (i.e., the *time-domain* signal) and so *s* corresponds to inverse time, or frequency https://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char17.png(i.e., the *frequency-domain* signal).

The ***Fourier transform*** is defined by



DFT:

The continuous Fourier transform converts a time-domain signal of infinite duration into a continuous spectrum composed of an infinite number of sinusoids.  In astronomical observations we deal with signals that are discretely sampled, usually at constant intervals, and of finite duration or periodic.  For such data, only a finite number of sinusoids is needed and the ***Discrete Fourier Transform*** (DFT) is appropriate.  For almost every Fourier transform theorem or property, there is a related theorem or property for the DFT.  The DFT of *N* uniformly sampled data points *xj* (where *j*=0https://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char3B.pnghttps://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char3A.pnghttps://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char3A.pnghttps://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char3A.pnghttps://www.cv.nrao.edu/course/astr534/jsMath/fonts/cmmi10/alpha/173/char3B.png*N*−1) and its inverse are defined by



Fast Fourier Transform(FFT):

The Fast Fourier Transform does not refer to a new or different type of Fourier transform. It refers to a very efficient algorithm for computing the DFT .The time taken to evaluate a DFT on a computer depends principally principally on the number of multiplications multiplications involved involved. DFT needs N2 multiplications. FFT only needs Nlog2(N) . The central insight which leads to this algorithm is the realization realization that a discrete discrete Fourier Fourier transform transform of a sequence sequence of N points can be written in terms of two discrete Fourier transforms of length N/2 .Thus if N is a power of two, it is possible possible to recursively recursively apply this decomposition until we are left with discrete Fourier transforms of single points.

**For Example:**

MATLAB Code:

x=[1 0 0 1];

y=fft(x);

subplot(2,1,1)

stem(abs(y),'k')

xlabel('m')

ylabel('X(m)')

title('Absolute Value Of DFT Sequence')

subplot(2,1,2)

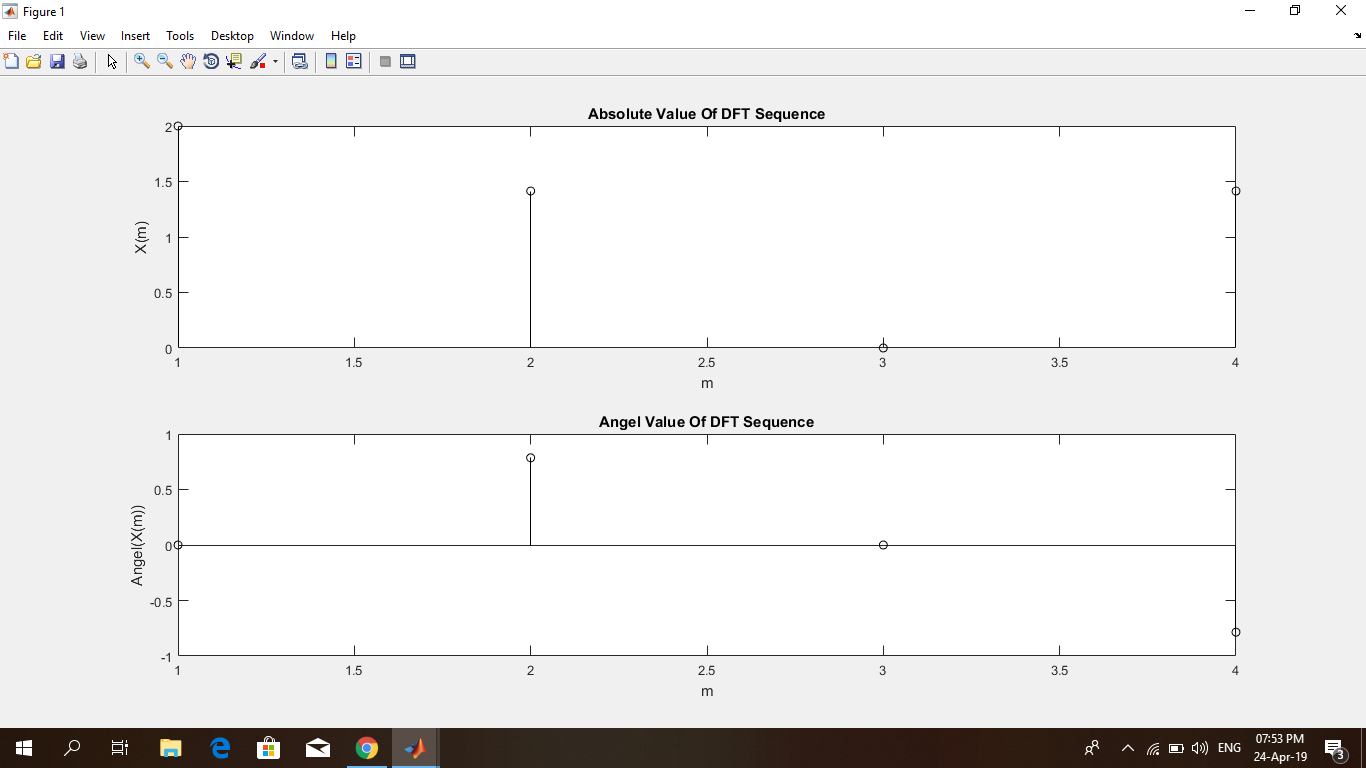
stem(angle(y),'k')

xlabel('m')

ylabel('Angel(X(m))')

title('Angel Value Of DFT Sequence')

Figure:



Rectangular Pulse:

MATLAB Code:

fs=500;

t=-1:1/fs:1;

x=rectpuls(t,0.12)

subplot(2,2,1)

plot(t,x)

grid on

y=fft(x);

y=fftshift(y);

subplot(2,2,2)

plot(abs(y))

grid on

x=rectpuls(t,0.02);

subplot(2,2,3)

plot(t,x)

grid on

y=fft(x);

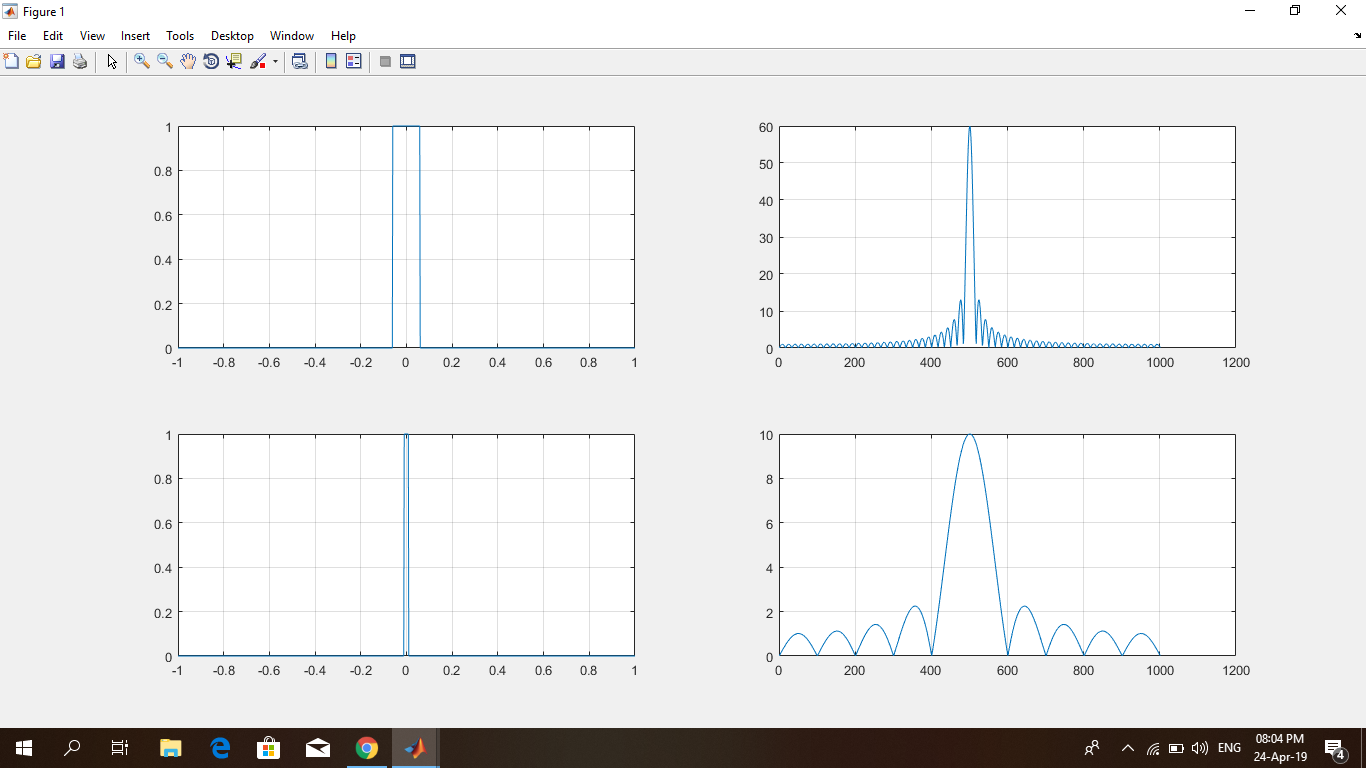
y=fftshift(y);

subplot(2,2,4)

plot(abs(y))

grid on

Figure:

****

DTFT On a 2D Rectangular Pulse:

MATLAB Code:

x=zeros(32);

x(12:17)=ones(6,1);

subplot(2,2,1)

title('2D Rectangular Pulse')

mesh(x)

x=fft(x);

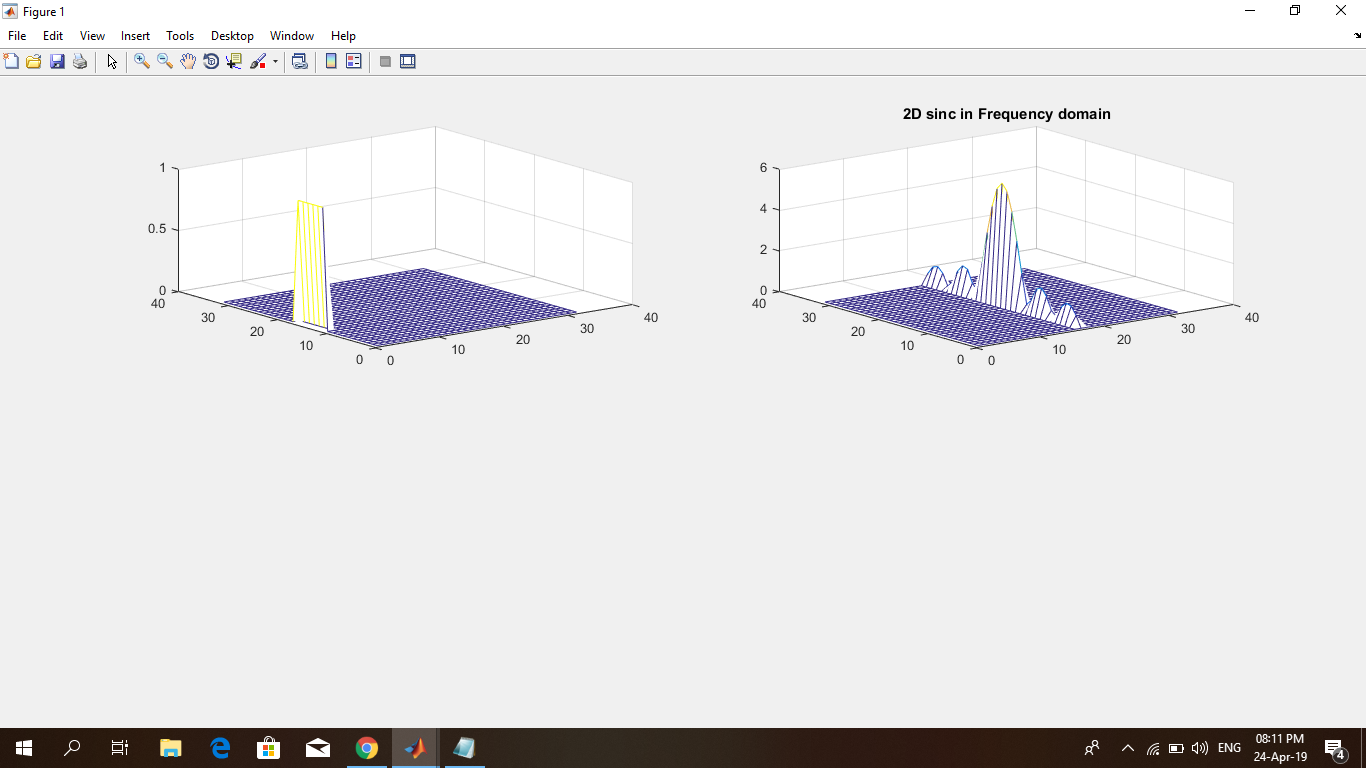
x=fftshift(x);

subplot(2,2,2)

mesh(abs(x))

title('2D sinc in Frequency domain')

Figure:

****

DTFT On a 3D Rectangular Pulse:

MATLAB Code:

x=zeros(32);

x(12:17,12:17)=ones(6);

subplot(2,2,3)

title('2D Rectangular Pulse')

mesh(x)

x=fft(x);

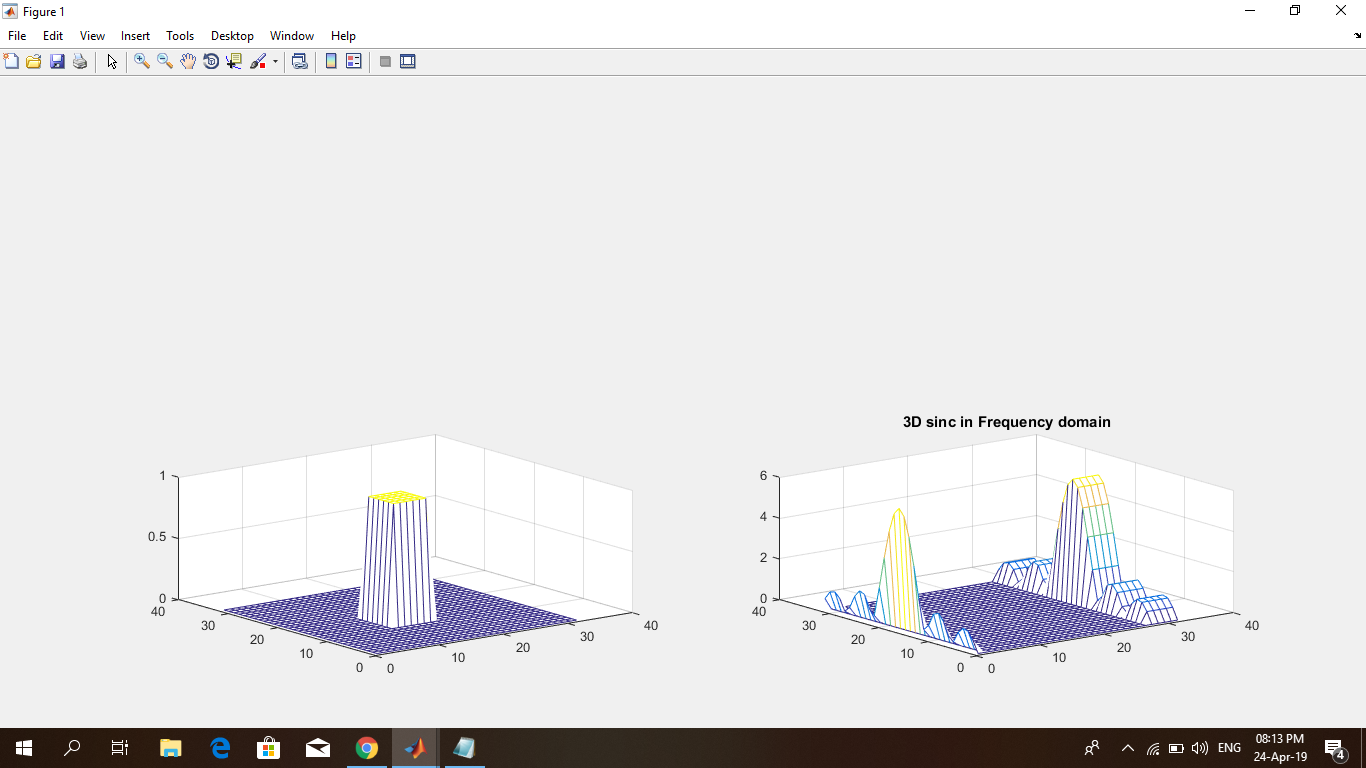
x=fftshift(x);

subplot(2,2,4)

mesh(abs(x))

title('3D sinc in Frequency domain')

Figure:

****

**Discussion:**

If one attempts to use the wrong transform for a given signal, he or she may well succeed in getting an answer, since the transforms have similar properties. But this usually leads to some rather unnatural signal modeling which can sometimes be deceptive. We shall see that the transforms themselves are not difficult to understand and to use. But problems such as sampling, aliasing, and the relation of the fast Fourier transform to the true Fourier transform generally involve signals from separate classes, and here one must be careful..