Week 1 & 2:

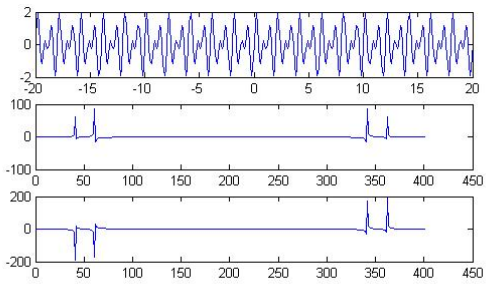
-Introduction to Matlab and the Fast Fourier Transform (FFT)

*Matlab, FFT, noise elimination in signal processing

*Discrete-time Fourier Transform in Fortran

Project 1:

a) Write a code to look at the real and imaginary parts of the FFT of a sum signal of two sinusoids of different frequencies; eliminate the high frequency portion of the spectrum, then show plots of the inverse transforms (IFFT) of the real parts and the imaginary parts. Explain what was the difference versus doing the inverse transform on the direct transform without separating the real and imaginary parts.

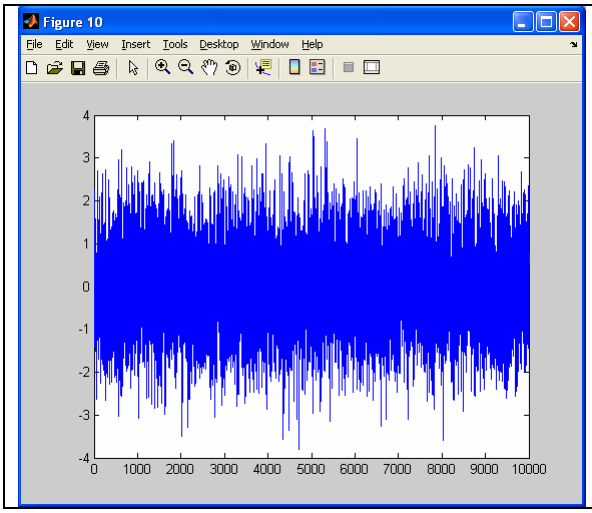


*How to make the filter or how to eliminate higher freq. portion of the spectrum? –Answer:Create an array of ones and zeros, being the zeros for elimination: filter=[111111 0000000000000]; or low=ones(1,nlow); high=zeros(1,nhigh); filter=[low high];
ones(1,nlow): is a 1D array with nlow elements

b) Use a low-pass filter on the spectrum of a signal with white noise added (SNR=2) to recover the original signal without noise. Repeat for SNR=1. Show with plots what did you obtain; is it important to have a minimum number of points of the signal?

*What is the frequency of the signal? –Answer: a sinusoid of any frequency

*What is the noise, how do I generate that? –Answer: a random signal; generate with Matlab using "randn"




```
yn=randn(1,10000);
>> mean(yn)

ans =

    0.0065

>> figure(10), plot(yn)
>>
```

*Signal amplitude needs to be twice the noise amplitude

c) Write a Fortran code to do the discrete-time Fourier Transform of a sum of two sinusoids of different frequencies. Get the output file from the UNIX machine and show the plots using Matlab.

$$\hat{F}(\omega) = \int_{-\infty}^{\infty} dt\, e^{-j\omega t} f(t) \quad \rightarrow \quad \hat{F}[k] = \sum_{n=1}^{N} e^{-jk\,\Delta\omega\, n} f[n] = \sum_{n=1}^{N} e^{-jk\frac{2\pi}{N}n} f[n]$$

$\omega = k(\Delta\omega)$; k=frequency index; Δω is the frequency increment 2π/N

In the Fortran code we will start with N values for f[n], and will do the summations to obtain N values for F[k], which will be stored in a file to be downloaded to a PC for plotting using Matlab

* How do we choose N? Use trial and error at this point, similar to how many points were needed to get a good noise elimination in part b).

*How do you store an output variable into a file in Fortran? Use:

```
      open(10, file='dft.prb')
      do 40 i=1,N
      write(10, *) DFTR(i)
40    continue
      close(10)
```

*Do I need to declare a variable in Fortran at the beginning of the code?

Yes, also indicate the dimension of the (array) variable, in our case x[n] and F[k] are array variables of dimension N. For example for N=1000 use:

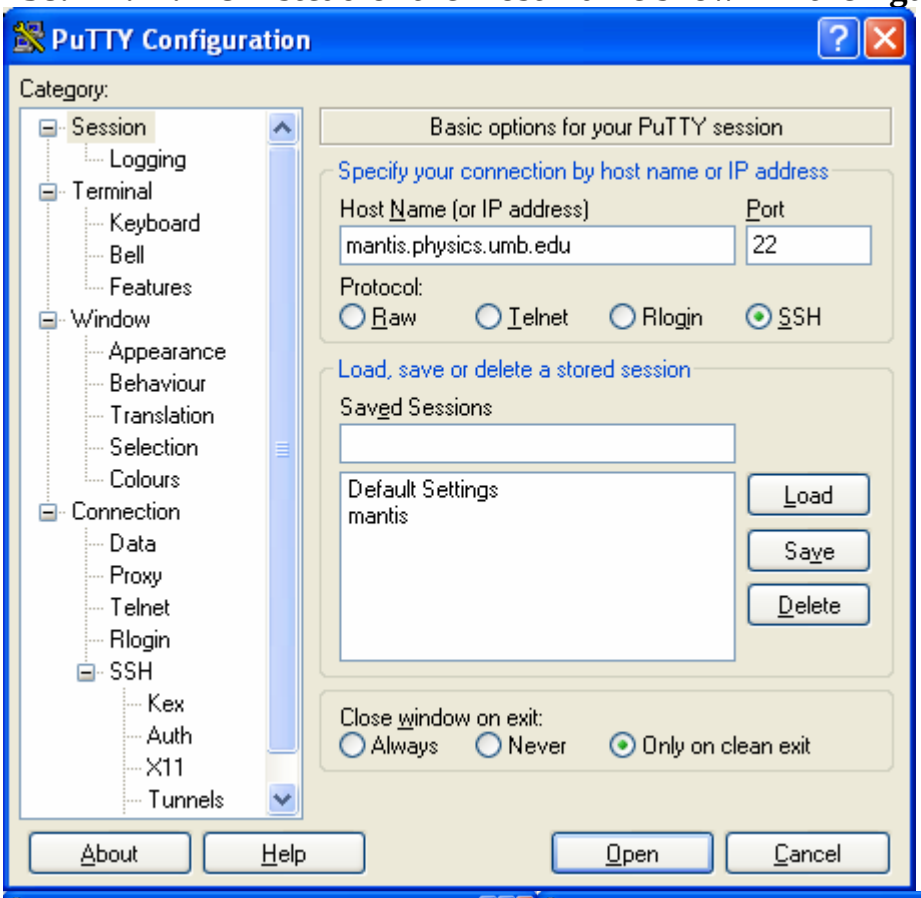
```
      real DFTR(1000), DFTI(1000), X(1000)
```

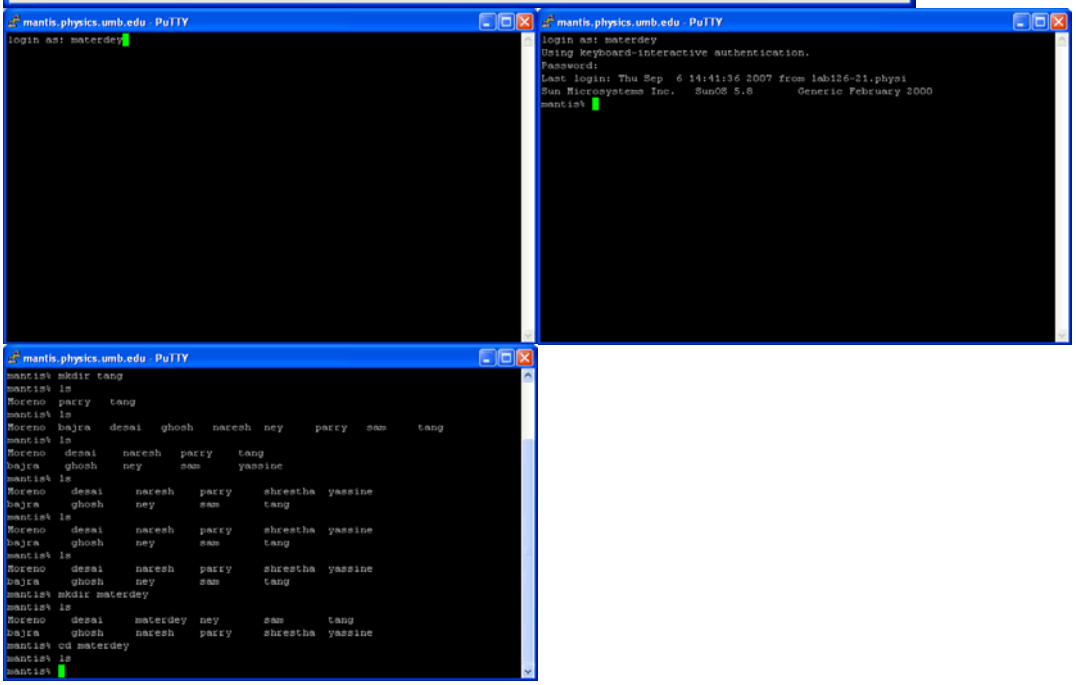
Fortran standards:
http://www.faculty.umb.edu/tomas_materdey/697f05/files/sylfa05.html
Fortran code example:
http://www.faculty.umb.edu/tomas_materdey/697f05/files/sylfa05.html

Skeleton for this code is

```
*     This code calculates the Discrete-Time Fourier Transform of a signal X[n]
*     Physics 640, Sept 11, 2008

      program dft
*     Variable declaration
      *     parameter definition

*     array initialization

*     signal generation and discrete Fourier transform

*     storing output variable into a file

      end
```

You can also use SSH instead of PuTTy. Either way use IP address 158.121.22.148 instead of the Host Name shown in the figure below.





[-Save text files (using Notepad) with extension ".f"

-Send it up to the Unix account into your folder, using PSFTP (under All Programs/Putty) then log in to “materdey” and do “cd yourfolder”. Do “lcd” within PSFTP to change local folder (find what is the local folder by doing ‘!dir’ within PSFTP; go up one level by doing “lcd ..”) to where you stored your file.f. Use “put file.f” to send file.
-Compile it using “g77 file.f” (excutable file will be called by default a.out)]
Repeat [] if there is errors
-Run by typing in “a.out”
-Bring output (1D array) back to Matlab for plotting: by doing “get dft.prb’
-Start Matlab, do ‘yf=load(‘dft.prb’)’; plot(yf)