ERTH135 LAB 2 – Fourier Transforms Due April 22

Instructions
Please save your code for the following exercises to an m-file and send it to me by email (mwei@ucsd.edu). To help you get started, lab2start.m has a few lines of code.

1) Create a sine (or cosine) function 2048 points long with exactly 32 or 64 full cycles over this interval. Plot this function and label the axes. (You'll recognize this part from Lab 1.)

2) Take the fourier transform of the function that you made in problem 1 using the fft() function in Matlab. Plot the real and imaginary parts of the transform versus wavenumber. Take the inverse fourier transform of your series. Do you get what you started with? What happens if you use an even function instead of an odd one (or vice versa)? Label all your plots.
(Helpful Hint: There is a function fftshift() that will shift the zero wavenumber to the center of the plot. If there are nx points in your series, then the wavenumber should be k=-nx/2:nx/2-1 .)

3) Use Matlab functions to create a 2-D array of numbers 400 x 300 using the meshgrid() function (anything interesting). Make an image of this function using imagesc() in black/white. Supply a colorbar and label the x and y axes. Change the value of the array elements 100:110, 50:60 to make something that will be visible in the image. Is the spot in the correct location? If not try the command flipud().

4) Generate an array of numbers 1024 by 1024 containing zeros. Imbed a rectangular patch of ones in the array of zeros. We'll call this the aperture. Take the fourier

transform and look at the amplitude (angular resolution of aperture). How does the angular resolution change as the size and shape of the aperture are changed?

5) Create your own aperture and look at its fourier transform. Be creative; we'll show the results in class and vote on the most unusual aperture.