EE257/GP258 **Lab/Homework #5**
Scientific Data Processing Handout #15
Due: Thurs., Feb. 14

Problem 1 – FFTs of simple functions.

Compute the FFT of a simple triangle function with widths of 16, 32, and 64 points using a transform length of 1024. Hint: this will be a bit tricky to get the length properly defined, but experiment until you get it right. Compare the result of each transform to the analytic result of a sinc^2 function and quantify the error in your transform calculation. Consider the error in both the real and imaginary parts of the difference function.

Problem 2 – More simple FFTs.

Repeat problem using a 2-D FFT calculation. Plot a difference image between the function you calculate using the FFT and the analytic solution. Note the brightness scale of the difference image in your writeup.

Now, repeat the first part of this problem but with rect() functions of the same width as the triangle functions.

Problem 3 – Fourier domain filters.

Download the image file from the website entitled 'lab5prob3.dat'. This is a byte format file with a line length of 1200. Compute its spectrum and save as an image.

Filter the image using triangle functions of various widths centered at zero and choose one of the filtered images that represents, to you, a good tradeoff between image noise and resolution. Create an image of your selection and turn in as a tiff image.

Problem 4 – Image restoration.

If we can degrade an image's resolution by depressing certain Fourier components, in some cases we can increase the apparent resolution of the image by enhancing some Fourier components. This is called image restoration. Download the image file from the website entitled 'lab5prob4.dat'. In this file, the data are in floating point (real) format, and the line length is 1024. Examine this 'image', it represents an image obtained from a system with a very broad impulse response.

Compute the spectrum of the image. Estimate the bandwidth (in cycles per sample point) of the main lobe of the spectrum – the meaning of this will be apparent if you examine the spectrum. Assuming that the impulse response of the system was a triangle function

of half-width 16, weight the spectrum appropriately to restore the weighting of the Fourier components as well as you can, and inverse transform to obtain the image in restored form.

Problem 5 – Parallel FFT calculations.

In this exercise we will examine a 'coherent' image, that is, an image illuminated with very narrowband light such as a laser. We will use the same photograph as in the previous problem but will simulate it as if it were obtained under laser illumination instead of the sun.

Download the file from the website entitled 'lab5prob5.dat'. This is a complex data file in floating point format, consisting of 2048 lines by 2048 samples. Create an image from the magnitude of each data point, you will end up with a real image of size 2048x2048. Create a byte format version of the image and examine it, and hand in with your writeup. It should appear very "speckly", hence the term "speckle" to describe coherent images.

You can reduce the speckle by averaging successive pixels in the image. Replace every other point in the original magnitude by the average of the 2x2 points at that location, in other words, average 4 points (2 across and 2 down) for each out point to result in a 1024x1024 image. This image should look less speckly. Hand in.

Now, return to the original complex 2048x2048 image. Compute its 2-D Fourier transform and plot the spectrum as a 2-D image. Plot the magnitude of the spectrum, in other words create a 2048x2048 real image consisting of the magnitude of each Fourier coefficient. Hand in. Parallelize this code and see what speed enhancement you can obtain for the Fourier transform calculations.

Finally, compute the average spectrum in both the across and down directions and plot for your result to hand in. Average the magnitudes of the spectra, not the complex values. This is called incoherent averaging of the spectra.

You can use the CEES cluster for any of these problems, and make sure you do the speed comparison using that system for Problem 5.