1 Instructions

Please save your code for the following exercises as an m-file and email the m-file to me (at `m3becker@ucsd.edu`). Download a file called `lab2start.m` (at `http://topex.ucsd.edu/rs/labs2019/lab2/lab2start.m`), which includes a few lines of code to get you started. Please be sure to provide an answer for each and every component of each exercise! (Some have several!)

2 Exercises

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

2) Take the Fourier transform of the function that you made in Exercise 1 using the built-in MATLAB function `fft()`. Plot the real and imaginary parts of the transform output against the wavenumber. Then, 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 function (or vice versa)? Remember to label all of your plots!

Helpful hints: If there are nx points in your series, then the wavenumber should be `k = -nx/2:nx/2-1`. There is a built-in MATLAB function called `fftshift()` that will shift the zero wavenumber to the center of the plot.

Listing 1: Example of `fft()` and `fftshift()`

```
% generate the wavenumbers
k=-nx/2:nx/2-1;
% take the Fourier transform, then shift the wavenumbers
cy=fftshift(fft(y));
```

3) Use the built-in MATLAB function `meshgrid()` to create a 2-D 400×300 array of numbers. Design an interesting function (e.g., the product of two sine functions) using the array values as inputs. Make a grayscale image of this function with `imagesc()`. Add a colorbar and label both axes. Change the values of the array elements [100:110, 50:60] to make something that will be visible in the image. Is this "anomaly" in the correct location? If it is not, try using the function `flipud()`, and describe if/how your result changes.

4) Generate a 1024×1024 array of zeros. Insert a rectangular patch of ones in the array of zeros. We will call this the aperture. Consider the angular resolution of the aperture by taking the Fourier transform of the array and examining its amplitude. How does the angular resolution change as you change the size and the shape of the aperture?

5) Create your own aperture and examine its Fourier transform. Be creative! We will show the results in class and hold a vote to determine the most unusual aperture.