Problem Set 4: Preparations for Analyzing Potential Field Data Sets
Fourier Transforms: FFTs in MATLAB with synthetic data

Due in class on Thursday Oct 18th. Turn in a single, commented script (start each question in a new cell, comment your code) and handwritten or typed notes (<2 pages) for the comments requested. For ease of grading, please make the figures exactly as requested.

This problem set is designed to build understanding from scratch in case you have not had much experience with actually doing FFTs. It uses synthetic data so that you know your input signal and can (a) know some of the properties of Fourier transforms by experiment, and (b) see what happens given various choices of sampling and/or the presence of noise.

Part 1: Basic FFTs in MATLAB and showing various properties of FFTs (16 points)

1. (1 point) Write a program to generate 16 cycles of a sine function that has a wavelength of 32 m, data spaced every 1m. Make sure your synthetic data has a total of 512 points. Plot the results in Figure 1 and add labels.

2. (7 points) Take the Fourier transform of the function that you made in (1). Don't generate the wavenumber vector yet. Use fftshift to shift the zero frequency to the center of the spectrum. Plot your spectrum with and without using fftshift to see what fftshift does (don't turn this plot in). Compare your fft with the Fourier transform pairs in the notes. Take the inverse FFT. Do you get what you started with? (Don't forget to undo the fftshift before taking the inverse fft)

    Generate a 2-sided spectrum (normalize the spectrum properly and generate the wavenumbers). Make Figure (2) with the following subplots in it:
    - subplot(2,2,1) – the real part of the Fourier transform
    - subplot(2,2,2) – the imaginary part of the Fourier transform
    - subplot(2,3,3) – the amplitude of the Fourier transform
    - subplot(2,2,4) – the phase (in radians).

    Label your plots and axes including units where applicable

    Generate a 1-sided spectrum (properly normalized with wave#s). In figure 3 plot:
    - subplot(3,1,1) – the amplitude of the Fourier transform
    - subplot(3,1,2) – the input signal (plotted as points) and the recovered (via the ifft) signal (plotted as a line)
    - subplot(3,1,1) – the difference between your input and output signal

3. (4 points) Use your sine function to demonstrate the shift property of the Fourier transform by shifting your sine function by 8 m.
    a. Plot the original function and this shifted sine function
    b. Write down what you expect the Fourier transform of this phase-shifted signal to be using the shift property. Code this up and take the inverse Fourier transform. Check that the resulting function matches what you have plotted in (a)

4. (3 points) Use the sine function to demonstrate the derivative property of the Fourier

transform.

   a. Calculate the derivative of the sine function analytically and plot it.
   b. Use the derivative property of the FFT to write down the form for the FFT of the derivative of the sine function. Code this up, take the inverse Fourier transform of this and check that the resulting function matches what you have plotted in (a).

We could similarly demonstrate the stretch property etc.

Part 2: Sampling and Aliasing at Work (22 points)

5. (6 points) You now have a time series that spans 10 cycles of a sine wave that has a period of 10 seconds and is sampled at a frequency of 2 Hz. Generate the time series and compute the fft. In a new figure plot
   a. subplot(211): The sine function. Label the axes (w/ units where appropriate).
   b. subplot(212): The 1-sided amplitude spectrum, normalized and with the frequency axis computed correctly. Give the numerical values for
      i. the Nyquist frequency?
      ii. the lowest (non-zero) frequency?
      iii. the frequency increment, d*f*?

6. (6 points) *Playing with sampling frequency.* Seeing aliasing at work.
   a. Resample your function at 2-second intervals. In a new figure subplot(311), plot the 1-sided amplitude spectrum.
   b. Do the same thing but sampling your time series at 4-second intervals. Plot the spectrum in subplot(312)
   c. Now at 8 second intervals (it will be easiest in this case to increase the time series to be 104 seconds long). Plot the spectrum in subplot(313)
   d. Explain what you see in (a)-(c).

7. (10 points) *Moving toward a real time series: more than 1 frequency with noise.* Now add a second function: a cosine with period 2.5 times that of your sine wave, and amplitude 4x that of the sine wave. Use a sampling frequency of 1 Hz and let your time series be 100 points.
   a. In a new figure, subplot(211): Plot the sine wave, the cosine wave and their sum (y(t)) versus time
   b. subplot(212): Compute and plot the 1-sided amplitude spectrum for y(t) including the amplitude of the spectrum at the zero frequency. Verify that you get the correct frequencies and amplitudes corresponding to your known input signal.
   c. In a new figure: Now add random noise to your time series, using noise drawn from a uniform distribution on the interval [0,1]. Plot the time series (subplot(221)) and then compute and plot the 1-sided fft including the zero frequency again (subplot(222)).
   d. Now increase the amplitude of your noise level by a factor of 10. Plot the original time series (y(t)), the noise (n(t), and the "real" time series y(t) + n(t) in subplot(223). Plot the 1-sided fft including the zero frequency in subplot(224). What do you see? Are your original frequencies resolved? What has happened at

the zero frequency? How would you deal with this in practice? Try some different things and explain what you have done and what you see.