PHYS 391, Homework #5
Due Friday, Dec. 4th

Key Concepts

- Fourier Series
- Discrete Fourier Transform

Reading: Fourier Handout

Homework Problems

1. Find (by integration) an expression for the real Fourier coefficients a_n and b_n for a 'ramp' function $f(x) = x$ over the range $-\pi$ to π. Use python to produce a plot of the Fourier Series expansion for your solution for the first six non-constant terms. Hint: save yourself some work by considering whether the function $f(x)$ is even or odd.

2. Find (by integration) the real Fourier coefficients a_n and b_n for a rectified sine function given by

$$f(x) = \begin{cases} -sin(x), & -\pi < x < 0, \\ +sin(x), & 0 < x < +\pi. \end{cases}$$

   What is the lowest non-constant term in the Fourier series? In other words, what is the fundamental frequency of this rectified sine function? Note, this is the hum you hear when you turn on a guitar amplifier...

3. Write a python function which will produce a 1D array of values sampled at discrete times from the function $A\sin(2\pi\nu t) + B$, where A, B, ν, as well as the sampling frequency ν_s and N (the total number of samples) are input parameters. This is to create "fake data" which can be used in the next problems. Be careful to make sure your discrete time values have a spacing of exactly $\Delta t = 1/\nu_s$. Demonstrate that your function works by producing a plot with $A = 1$, $B = 1$, $\nu = 1$, $\nu_s = 20$ and $N = 100$.

4. Use your creation from the last problem to explore aliasing. Generate a curve with $N = 100$ where $\nu = 9.9$ and $\nu_s = 10$. What is the apparent frequency from the plot and how does this compare to the true frequency ν? Explain qualitatively what is going on here. What minimum sampling frequency ν_s would you need here to avoid aliasing?

5. Write a python function which will plot the amplitude vs. frequency for a time series of data (like the output of problem 3). In addition to your data, this function will need to know the $\Delta\nu = \nu_s/N$ in order to properly plot amplitude vs. frequency from the output of the `fft` function. You can either do this 'by hand' or use some of the `scipy` helper functions like `fftshift` and `fftfreq` to potentially make your life easier. Check that your function works by applying this function to the output of problem 3. Attach the plot of amplitude vs. frequency as well as your code. Compare the amplitudes seen with what you would expect.

6. Take the Fourier transform of the same function with $\nu = 2$ rather than $\nu = 1$. Attach (or just sketch) the amplitude vs. frequency and explain why you see what you see.

7. Do the same thing for $\nu = 2.05$. What is different here? How can we understand this?

8. Redo the last problem but now use $N = 1000$ rather than $N = 100$. Explain in words what happened. What is the difference in the frequency range sampled by this larger value of N? What is the difference in the frequency resolution?

9. Redo problem 7 with $N = 100$ but now apply a Hanning window to the time series before taking the Fourier Transform. Windows can be found in the `scipy.signal` library, and generally return an array of weights (e.g. `w = hann(N)`) which are then just multiplied with your data array to apply the window.