Phys 512 Problem Set 5

Due on github Friday October 29 at 11:59 PM. You may discuss problems, but everyone must write their own code.

1) Write a function that will shift an array by an arbitrary amount using a convolution (yes, I know there are easier ways to do this). The function should take 2 arguments - an array, and an amount by which to shift the array. Plot a gaussian that started in the centre of the array shifted by half the array length, in "gauss_shift.png".

2) The correlation function $f \star g$ is $\int f(x)g(x+y)dx$. Through a proof similar to the one we saw for the convolution theorem, one can show $f \star g = ift(dft(f) * conj(dft(g)))$. Write a routine to take the correlation function of two arrays. Plot the correlation function of a Gaussian with itself.

3) Using the results of part 1 and part 2, write a routine to take the correlation function of a Gaussian (shifted by an arbitrary amount) with itself. How does the correlation function depend on the shift? Does this surprise you?

4) The circulant (wrap-around) nature of the dft can sometimes be problematic. Write a routine to take the convolution of two arrays without any danger of wrapping around. You may wish to add zeros to the end of the input arrays. Call your routine conv_safe(f,g), where f and g are not necessarily the same length. How long is the output array?

5) DFTs work very nicely out of the box when there are an integer number of periods of a wave in the region analyzed. Sadly, when we are dealing with real data, we usually are forced to analyze a finite chunk of data, and there will in general be no particular relation between the frequencies in the data and the interval we're analyzing. We'll look at the effects of this a bit now.

a) Show that:

$$\sum_{x=0}^{N-1} \exp(-2\pi ikx/N) = \frac{1-\exp(-2\pi ik)}{1-\exp(-2\pi ik/N)}$$

It may help to recognize that the sum can be re-written $\sum \alpha^x$ where $\alpha = \exp(-2\pi ik/N)$ so we can treat it as the sum of a geometric series.

b) Show that this approaches N as k approaches zero, and is zero for any integer k that is not a multiple of N.

c) We can use this to analytically write down the DFT of a non-integer sine wave. Pick a non-integer value of k and plot your analytic estimate of the DFT. Show that the FFT agrees (to within machine precision) with your analytic estimate. Normally, we think of the Fourier transform of a pure sine wave to be a delta function. Are we close to that? This phenomenon is usually known as *spectral leakage*.

d) A common tool to get around this is the use of *window functions*. The leakage essentially comes from the fact that we have a sharp jump at the edge of the interval. If we multiply our input data by a function that goes to zero at the edges, this cancels out the jump, and so prevents the leakage from the jumps at the edges. Of course, since we have multiplied by the window in real space, we have convolved by it in Fourier space. One simple window we could use is $0.5 - 0.5\cos(2\pi x/N)$ (there are many, many choices). Show that when we multiply by this window, the spectral leakage for a non-integer period sine wave drops dramatically.

e) As we saw in class, show that the Fourier transform of the window is $[N/2\ -N/4\ 0\ ...\ 0\ -N/4]$ (either numerically or analytically). Use this to show that you can get the windowed Fourier transform by appropriate combinations of each point in the unwindowed Fourier transform and its immediate neighbors (you may need to be careful with signs here, since if you work through the math, some of the transforms need to be inverse FFTs). The choice of suitable windows is as much art as science (and depends on the details of what you're most concerned about), but I hope this gives at flavor of what's going on.

6) a) Finish the calculation we started in class to show that the expected power spectrum of a random walk goes like k^{-2}. As a reminder, we pulled a trick to show that the correlation function of two points separated by δ is proportional to $c - |\delta|$, where c is some large number, and pay attention to those absolute values.

b) Generate a random walk, which you can do with

```
np.cumsum(np.random.randn(n))
```

. Plot the power spectrum of your random walk in rw_ps.png, and show that it indeed has a k^{-2} power spectrum. Feel free to ignore the $k = 0$ mode.

Bonus: Show that if you forgot to window your random walk (which of course you would *never* do), your power spectrum still behaves as expected. Explain why this is (you might want to consider the Fourier transform of a line with non-zero slope). If we generated non-circulant data with a power spectrum k^{α}, for what range of α could we get away without windowing?