

Homework 6

Due: Nov. 9, 2018

Write a **Python** program to perform Fourier transforms (both Fast and not-so-fast) and use it to analyze the LIGO data containing the first binary black hole detection!

1. **Warm-up:** Write **Python** functions in a single file (I suggest `fft.py`) that compute the Discrete Fourier Transform (DFT) and inverse DFT in two ways: the simple (naive) scheme and the FFT. You may assume your incoming data has a length 2^n for some positive integer n . Remember: the key to the FFT is not only the splitting between even and odd terms, but that only half of the even and odd coefficients need be calculated at each step. Details are at:

https://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm.

- Test both your (slow) DFT and FFT on an array of random numbers. By applying the Fourier transform and then the inverse transform you should end up with the same data (up to rounding error).
 - Time the slow DFT, your FFT, and the **numpy** FFT on various size arrays. Show the scaling of the execution time for each scheme on a log-log plot, do they match theoretical predictions? Convince yourself you should never use the slow DFT.
2. **LIGO Data!** Use Fourier techniques to find the merging black hole signal GW150914. Write a **Python** program to read in the LIGO data, do some simple cleaning, and find the signal.

- LIGO has two independent observatories, one in Livingston, LO and the other in Hanford, WA. Both detectors must see a signal within 10ms of each other for it to count! A window of data (32 seconds) around the time of the event is available for download here:

https://losc.ligo.org/s/events/GW150914/H-H1_LOSC_4.V1-1126259446-32.hdf5

https://losc.ligo.org/s/events/GW150914/L-L1_LOSC_4.V1-1126259446-32.hdf5

- These are HDF5 files. As discussed in section, these files are much smaller and quicker to read/write than plain text files. The **h5py** library (included in **Anaconda** and available through **pip**) interacts with these files. A function to do so is:

```
def loadLIGOdata(filename):
    f = h5py.File(filename, "r")
    strain = f['strain/Strain'][...]
    t0 = f['strain/Strain'].attrs['Xstart']
    dt = f['strain/Strain'].attrs['Xspacing']
    t = t0 + dt * np.arange(strain.shape[0])
    f.close()
```

return t, strain

This returns the time (in seconds) and the strain (which has no units). The measurements in this data set are evenly sampled at 4096Hz .

- The strain $h(t)$ is what is measured by LIGO. You can think of it as the fractional change in the length of the 4km interferometer arms: $\Delta L/L$. Plot the strain as a function of time for both detectors. Gravitational waves from astrophysical sources produce a maximum strain on Earth of about 10^{-21} . Can you see a gravitational wave in the data?
- No. You cannot. Most of the strain is “noise” coming from various physical effects in the detector. To find how the noise affects the data, plot the *periodogram* $P_{hh} = |h_k|^2$ of the data for each detector using your FFT. How long would this take using the slow DFT? The periodogram is a crude estimate of the *power spectrum*, which indicates how much power is in each Fourier mode of the data.
- The periodogram shows a lot of power for $f < 30\text{Hz}$ and many spectral lines. These lines correspond to resonances in the LIGO machinery (the cables suspending the mirrors, the 60Hz electrical frequency, etc). These are clearly noise! Filtering out particular modes in Fourier space is not difficult, simply multiply the DFT of your data h_f by a *transfer function* $H(f)$, where $|H(f)| < 1$, then perform the inverse transform to see the filtered data. Here are two very simple filters:

$$H_{\text{step}}(f) = \frac{1}{1 + (f/f_0)^{2n}} , \quad (1)$$

$$H_{\text{gauss}}(f) = 1 - \exp\left(-\frac{(f - f_0)^2}{2\sigma_f^2}\right) . \quad (2)$$

In the above, f_0 is the location of the filter and n or σ_f control the width. What does each filter do?

- LIGO is most sensitive between 35Hz and 350Hz . Use H_{step} with $n \sim 8$ to filter out the modes outside this band. Plot the resulting waveform and periodogram.
- See a signal yet? Use the H_{gauss} filter to remove spectral lines from the data as well. Plot the resulting waveform and periodogram.
- This dataset contains the *first ever* detection of a binary black hole merger! What time does it occur at? Remember: a real signal appears in both data sets with a time delay of no more than 10ms .

Write a report summarizing your work, showing all plots, giving your results, and discussing the questions. Include the report **pdf** and all Python files in the repo.

Bonus: Make an audio file of the cleaned data!