

Peter Craig
 ASTP 720
 HW 7

I wrote up a code that can take the discrete fourier transform of a given data set using the Cooley-Tukey algorithm. There is some code that compares the results that I get with my algorithm to the results from the fast fourier transform in numpy. The implementation is not as fast, but it gives the same results. So I plugged the strains given in strains.npy into my fft code, and produced a log-log plot of the results, which is shown below in Figure 1. The feature to note is the spike located between 10^{-2} and 10^{-3} Hz. This spike is the signal of the binary that we are investigating.

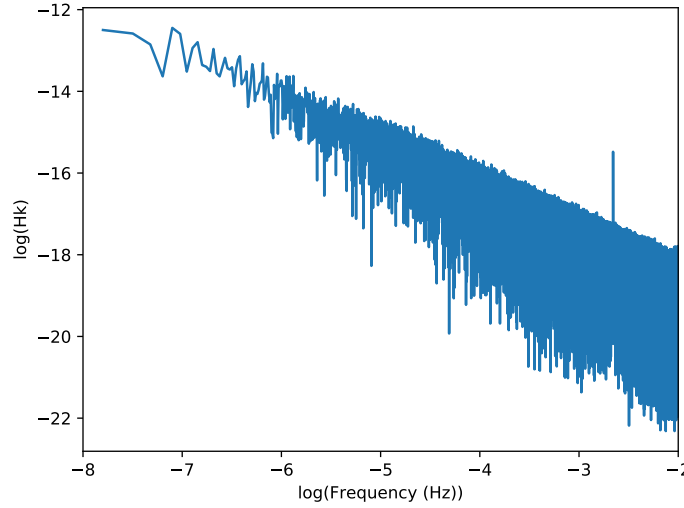


Figure 1: The Fourier transform of the given strain data. This shows the log of the frequency in Hz vs the log of the value of the fourier transform at that frequency.

Next I wanted to compute M and R for our binary system, so we start with the equations for f and h given in the HW and solve for the desired parameters. The given equations are equations 1 and 2.

$$h \approx 2.6 \times 10^{-21} \left(\frac{M}{M_{\odot}} \right)^2 \left(\frac{D}{pc} \right)^{-1} \left(\frac{R}{R_{\odot}} \right)^{-1} \quad (1)$$

$$f \approx 10^{-4} Hz \left(\frac{M}{M_{\odot}} \right)^{\frac{1}{2}} \left(\frac{R}{R_{\odot}} \right)^{-3/2} \quad (2)$$

Now I solve for the mass in the frequency equation, which is shown in equation 3. The plan is to substitute this into the strain equation, and then solve for the radius.

$$\frac{M}{M_{\odot}} \approx \left(\left(\frac{R}{R_{\odot}} \right)^{3/2} \frac{f}{10^{-4} Hz} \right)^2 \quad (3)$$

Equation 4 shows the strain equation with this relationship for the mass plugged in.

$$h \approx 2.6 \times 10^{-21} \left(\left(\frac{R}{R_{\odot}} \right)^{3/2} \frac{f}{10^{-4} Hz} \right)^4 \left(\frac{D}{pc} \right)^{-1} \left(\frac{R}{R_{\odot}} \right)^{-1} \quad (4)$$

Now we are free to combine our factors of R.

$$h \approx 2.6 \times 10^{-21} \left(\frac{f}{10^{-4} Hz} \right)^4 \left(\frac{D}{pc} \right)^{-1} \left(\frac{R}{R_{\odot}} \right)^5 \quad (5)$$

H

Table 1: Table showing the end results of my code

Parameter	Value
h	6.32883
f	0.00220 Hz
R	0.10437 R_{\odot}
M	0.55215 M_{\odot}

The equation below is the result of solving this expression for the radius of our binary. once we have our frequency and strain from the Fourier transform, we can use this to easily find the radius. The next step would then be to plug this back into equation 3, which will give us the mass.

$$\left(\frac{R}{R_{\odot}}\right) \approx \left(\frac{h}{2.6 \times 10^{-21}} \left(\frac{10^{-4} \text{ Hz}}{f}\right)^4 \left(\frac{D}{\text{pc}}\right)\right)^{1/5} \quad (6)$$

The results of this are shown in Table 1. This just includes the frequency and strain that I got from the Fourier transform, and the Radius and Mass determined by the equations above.

The next thing I did was something done as an experiment. The goal was to recover the actual strain signal from the binary as a function of time. So I started out with our data set and took the fast Fourier transform, as before. This time I then took the data in frequency space and ran it through a high and low pass filter. Then I wrote up a bit of code to run inverse Fourier transforms, given that this was easy with the fft code already written. I set the frequency cutoffs at 10^{-2} Hz for the low pass filter and 10^{-3} Hz for the high pass filter. The filtering functions are ultimately just a piecewise function, defined to be 1 between these two frequencies, and 0 elsewhere. Figure 2 shows the original strains plotted as a function of time, and Figure 3 shows the strain after it has been filtered. In Figure 2 the signal is not really visible, and it is definitely there in Figure 3.

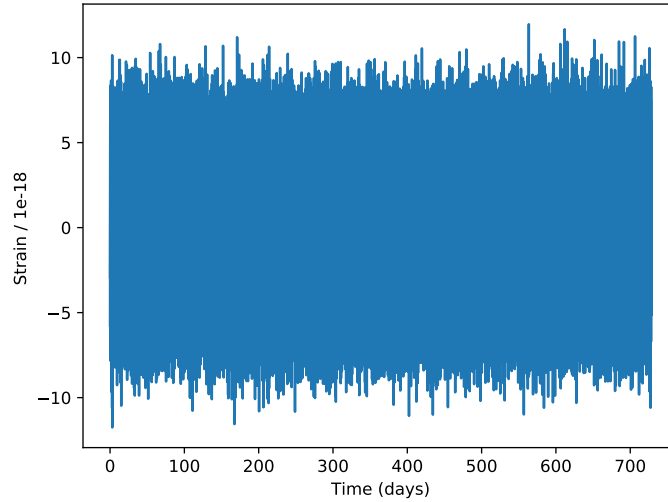


Figure 2: Strain plotted as a function of time in days. This plot simply shows our original data set.

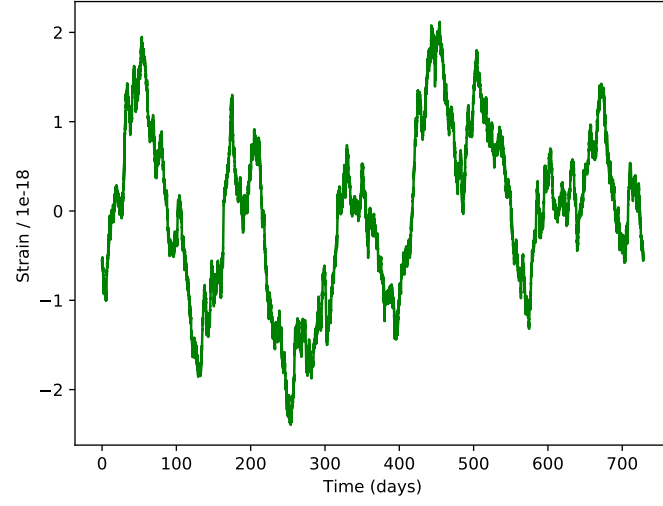


Figure 3: A plot showing our filtered strain signal.