ASTP 720 Homework Assignment 8

Isabella Cox – igc5972@rit.edu

November 5, 2020

Task 1

Your goal is to calculate the physical properties of the resolved binary system in the simulated data. In order to do so, you should program the Cooley-Tukey Algorithm to perform an FFT. At what point you stop the recursive step is up to you.

The simulated strain data "taken" over the course of two years is plotted in Figure 1. The data contain no instrumental noise, and the only the following signals: (1) double white dwarf (WD) binary background, with $S_h(f) \propto f^{-7/3}$ and (2) a simulated binary.

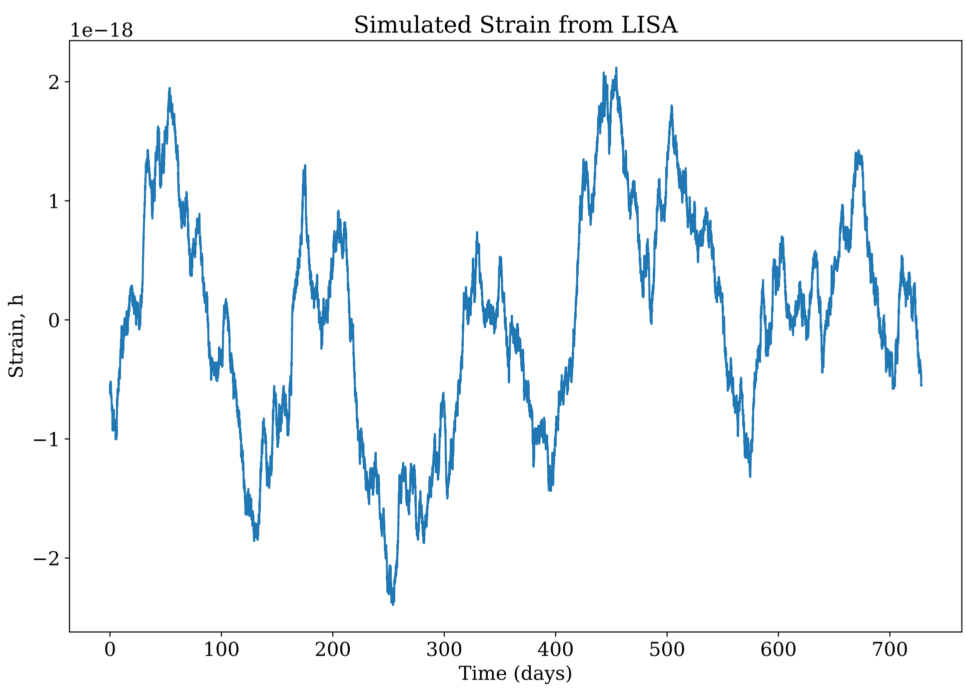

Figure 1: Simulated strain LISA data of a double WD binary background and a simulated binary.

For the simulated binary, we can work out rough expressions for the strain (Equation 1) and frequency (Equation 2) in terms of the system's total mass (M), separation (R), and distance to system D.

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

$$f_{GW} \approx 10 \times 10^{-4} Hz \left(\frac{M}{M\odot}\right)^{1/2} \left(\frac{R}{R\odot}\right)^{-3/2} \tag{2}$$

We should be able to see the spike of the gravitational wave frequency in the amplitude spectrum. To construct such a spectrum, the Cooley-Tucky (CT) algorithm can be used to construct a Fast Fourier Transform (FFT). The CT algorithm works by speeding up computing a FFT process and making it faster than the native 2^N approach. This speed-up works by computing the Direct Fourier Transforms (DFT) on the even and odd terms of the original time series separately.

My code works recursively, by re-calling my FFT code until it either is less than a length of 1, so it returns, or is greater than 1 but less than a user-input length that should be very small, at which point it calls another function I wrote to compute the DFT of the remainder time series. I did get stuck when assigning the FFT values for each time series step, in the sense of keeping the odd and even values straight. I am not sure my code is working correctly, so I switched to using `np.fft.fft` from here on out....

The plot of the amplitude against the frequency can be determined, and is shown in Figure 2. The frequency is computed by multiplying the integers up to the number of data points by the sampling frequency and dividing by the number of points where the sampling frequency is equal to the number of data points divided by the time span of the series.

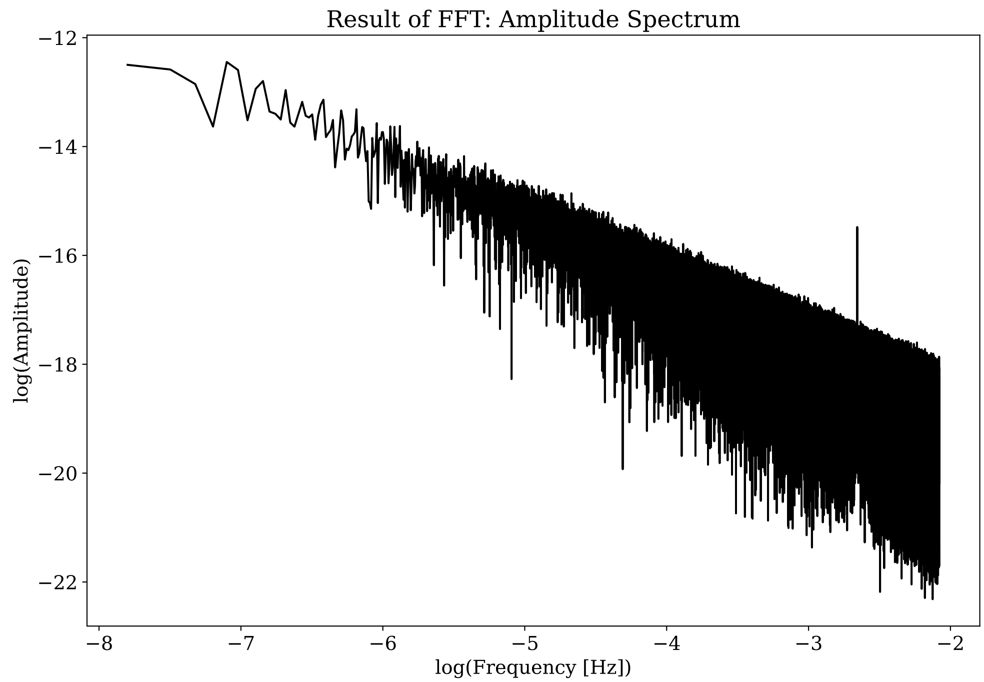

Figure 2: The amplitude of FFT as a function of frequency. Between frequency values of 10E-2 and 10E-3 Hz, you can see the peak.

The frequency and amplitude of the peak can be found by using something like `scipy.signal.find_signal` or what I did was to just find the maximum y-value and the corresponding x-value within a restricted range around where I eye-balled the peak to be. I found the frequency of the peak to be at -2.65 Hz and the amplitude to be -5.07 $\times 10^{-6}$. The amplitude of the signal is found as the height of the spike, divided by the number of data points and multiplied by 2.

Task 2

Once you have the f_{GW} and h measurements from the amplitude spectrum, assume that the system is at the distance of WD 0727+482, a known double WD system determined at a distance of 12 pc. Determine the total mass M and separation R.

So to do this, I would use Equations 1,2, but am having trouble with this. So I plugged the two expressions with two unknowns (D was given) into a system of equations solver on Wolfram, and got negative values for M and R which doesn't make sense....