Homework 5

Mats Thijssen

November 5, 2017

1 Introduction

The Fourier Transform decomposes a function of time into the frequencies that make up the time signal. In other words, a Fourier Transform (FT) takes us to the Fourier space conjugate to our original space, which is often easier to analyze. In particular, for a time dependent signal saturated by noise, in can be incredibly difficult to filter out the actual physics, while in Fourier space it is trivial to filter our particular frequencies creating noise. This is the process applied to the gravitational wave signals detected by LIGO, as the physical signal is way weaker than resonances in the observational equipment and even quantum effects is the materials used.

2 Theory

Two algorithms are implemented to compute Fourier Transforms: The Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT), and both their inverses. The derivation of the FT stems from the fact that every function can be written as a sum of sines and cosines, or equivalently as exponentials as follows:

$$f(x) = \sum_{k=-\inf}^{\inf} \gamma_k \times exp\left(\frac{i2\pi kx}{L}\right)$$

What we are generally interested in are the Fourier coefficients, γ_k, given by:

$$\gamma_k = \frac{1}{L}\int_0^L f(x) \times exp\left(\frac{-i2\pi kx}{L}\right)$$

This relation is easily verified by substituting in the expression for f(x) above. On a computer however, we cannot simply perform the above integral. We instead approximate it by, say, the trapezoidal rule:

$$\gamma_k = \frac{1}{L}\frac{L}{N}\left[\frac{1}{2}f(0) + \frac{1}{2}f(L) + \sum_{n=1}^{N-1} f(x_n) \times exp\left(\frac{-i2\pi kx}{L}\right)\right]$$

for sample points $x_n = nL/N$. By noting that f(x) is periodic (if it isn't, we make it so by forcefully repeating the section of the function we are interested in). Thus, f(0)=f(L) and we get:

$$\gamma_k = \frac{1}{N}\left[\sum_{n=0}^{N-1} y_n \times exp\left(\frac{-i2\pi kn}{N}\right)\right]$$

This is the Discrete Fourier Transform. ($y_n \equiv f(x_n)$) In the following, only the part inside the square brackets will be referred to as the FT (c_k). The inverse is given as with the continuous case by:

$$y_n = \frac{1}{N}\left[\sum_{n=0}^{N-1} c_k \times exp\left(\frac{i2\pi kn}{N}\right)\right]$$

The Fast Fourier Transform scheme used in this analysis is the radix-2-Cooley-Tukey method. It follows from noticing that the DFT can be split into even and odd terms as follows. Here, and

throughout the analysis, N is assumed to be a power of 2.
Let n=2r, then:

$$E_k = \sum_{r=0}^{N/2-1} y_{2r} \times exp\left(\frac{-i2\pi k(2r)}{N}\right) = \sum_{r=0}^{N/2-1} y_{2r} \times exp\left(\frac{-i2\pi k(r)}{N/2}\right)$$

which is just a DFT with N/2 terms. Similarly:

$$O_k = \sum_{r=0}^{N/2-1} y_{2r+1} \times exp\left(\frac{-i2\pi k(2r+1)}{N}\right) = exp\left(-\frac{i2\pi k}{N}\right) \sum_{r=0}^{N/2-1} y_{2r} \times exp\left(\frac{-i2\pi k(r)}{N/2}\right)$$

Which is again a DFT with N/2 terms, with a factor up front. Thus we get:

$$c_k = E_k + exp\left(-\frac{i2\pi k}{N}\right) O_k$$

as an equation for the FT, split into even and odd terms. These terms can again be split into even and odd terms recursively. As we see, we only need N/2 terms for r, and as we successively split up every term, halving the computational cost each time, making our scheme go to order (NlogN), compared to O(N^2) for DFT. The algorithm looks as follows:

$$\begin{aligned} &FFT(x):\\ &\quad x_{even} = FFT(x[::2])\\ &\quad x_{odd} = FFT(x[1::2])\\ &\quad factor = exp(-2j * \pi * array(0-N)/N)\\ &\quad return \quad concatenate([x_{even} + factor[:N/2] * x_{odd}, x_{even} + factor[N/2:] * x_{odd}]) \end{aligned}$$

3 Analysis: FT

To test the programmed FT-schemes, a random array of numbers is generated before being run through the FT's and their inverse. Up to rounding error, both algorithms return the original array, and thus the methods are deemed successful.
A more important quality of the various schemes is their computational efficiency, measured by the time they take to execute. A log-log plot of the computational time for differently sized random arrays is shown below for the DFT, FFT, and NumPy's FFT algorithm.

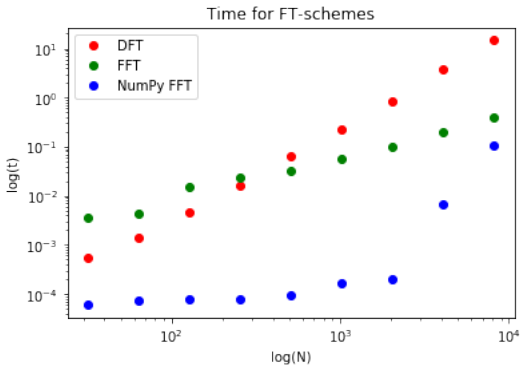

Figure 1: FFT eclipses DFT for larger arrays. NumPy is clearly optimized, but not a lot better than FFT for the largest sizes.

As expected, the FFT is way better than the DFT for arrays of large size. For small arrays, the DFT is better, as the splitting of the arrays is not worth it and ends up taking longer than "brute-forcing" through the FT. NumPy's algorithm contains several clever schemes to exploit symmetries

and other things to compute the FT very fast. However, for larger array-sizes, it is not a lot better than our FFT. The sample size plotted ranges from 2^3 to 2^13, where the DFT already takes 14 seconds. In the LIGO analysis, we are dealing with sizes of 2^17 and so clearly the DFT would be way to slow.

4 Analysis: Gravitational Waves

To analyze the GW signal obtained by LIGO regarding the BH merger, we will use the FFT explained above to filter out the noise and hopefully see the signal. An initial plot of the strain (h(t)), basically the change in length of the interferometer arms, is shown below.

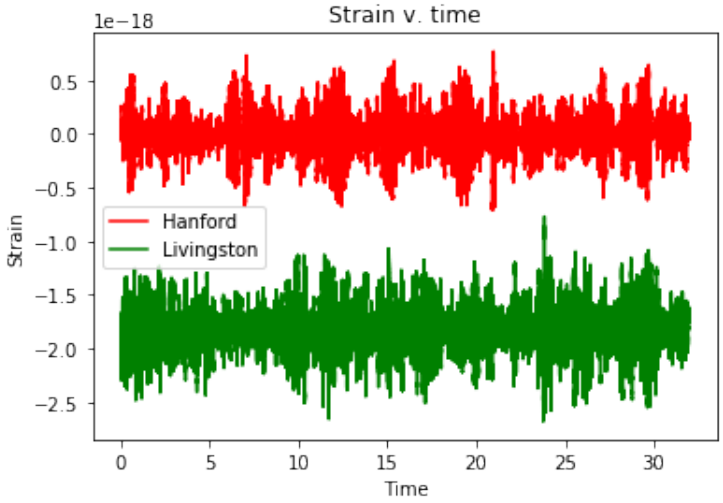

Figure 2: Unfiltered Strain at both detector locations

Clearly, the GW signal is not possible to see. In fact, the waves can only produce a detectable strain of at most order 10^{-21} on earth, while the signal plotted is almost order 10^{-18}.
To filter away the noise, we go into Fourier Space to see what frequencies are contributing an excess amount. To see this, a periodogram is plotted, $P_h h = |H(f)|^2$. This is an estimate of the spectral density of the signal, i.e. the distribution of power into frequency components composing that signal. Frequency components with power spikes are clearly noise.

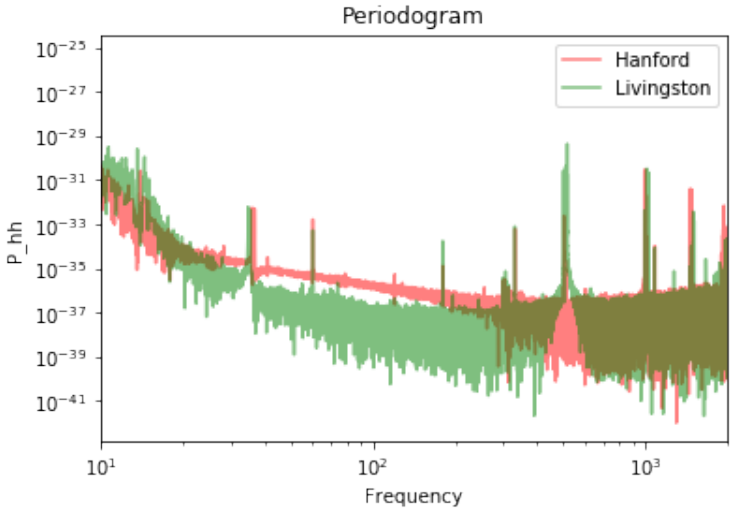

Figure 3: Periodogram of both locations from 10-2000Hz

This plot was made using the FFT to find the periodogram. Using the DFT would've taken 1 hour. This is found by timing it for multiple array sizes and noting that the time increases by a factor of 3.9 for every doubling of the array-size.
As can be seen, there is clearly a lot of spikes corresponding to noise, and a lot of power for f<30Hz as well. To filter out the noise, we equip ourselves with two transfer functions that can be applied to the data in Fourier Space. They are as follows:

$$H_{step}(f) = \frac{1}{1 + (f/f_0)^{2n}}$$
$$H_{gauss}(f) = 1 - exp\left(-\frac{(f - f_0)^2}{2\sigma^2}\right)$$

The first function act as a step function and can be used to severely dampen data above some threshold (f_0). The second function is a straight line but with a gaussian "well" at f_0, thus it can be used to damped a spike at a value f_0. Both functions have their "width" controlled by n and σ, respectively. Choosing these parameters is trial and error.
Because LIGO is most sensitive between 35-350Hz, the step function is used (n=8) to cut out values outside this range. We end up with this:

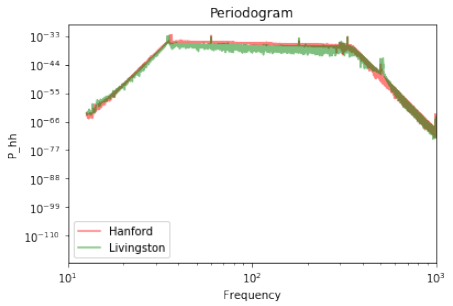

Figure 4: Peridogram after reducing to 35-350Hz

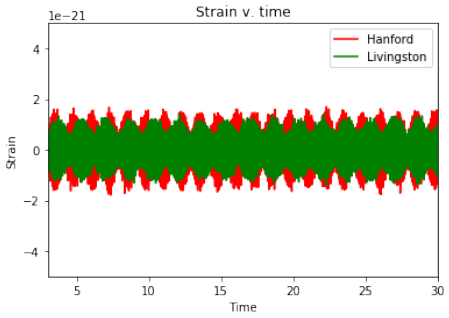

Figure 5: Strain after reducing to 35-350Hz

From the periodogram it's clearly visible that values outside the range are severely dampened. In the strain-plot, we are almost down to order 10^{21}, but still there is noise that prevents us from seeing the signal. Zooming in on the range 35-350Hz on the periodogram, it is easily seen what spectral lines we need to dampen:

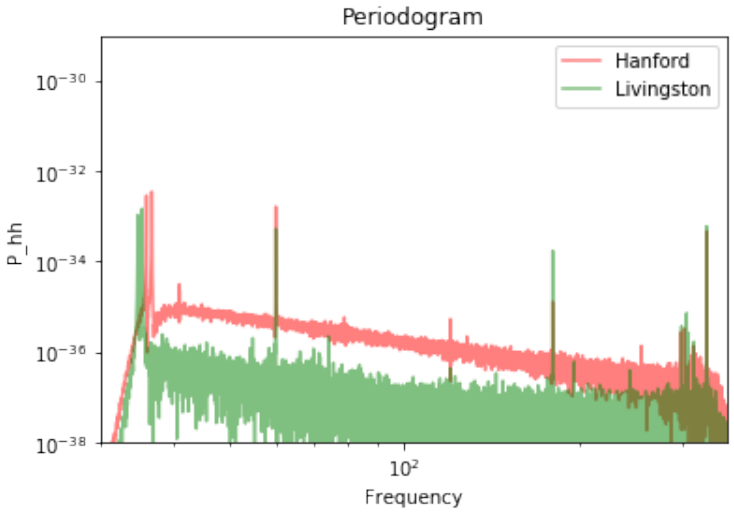

Figure 6: Periodogram after cut, 30-360Hz

The noise is clearly visible. By applying a series of Gaussian filters by trial and error, eventually

one ends up at this:

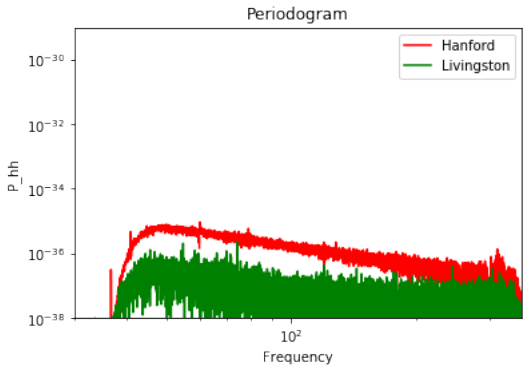

Figure 7: Periodogram after damping noise

Now this looks a lot better! The strain now looks like:

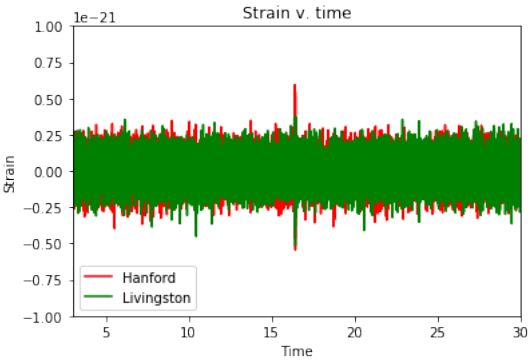

Figure 8: Periodogram after damping noise

Note that the y-axis has changed and we are on the order we want to be. And hey(!), there seems to be a spike around 16seconds, let's zoom in on that:

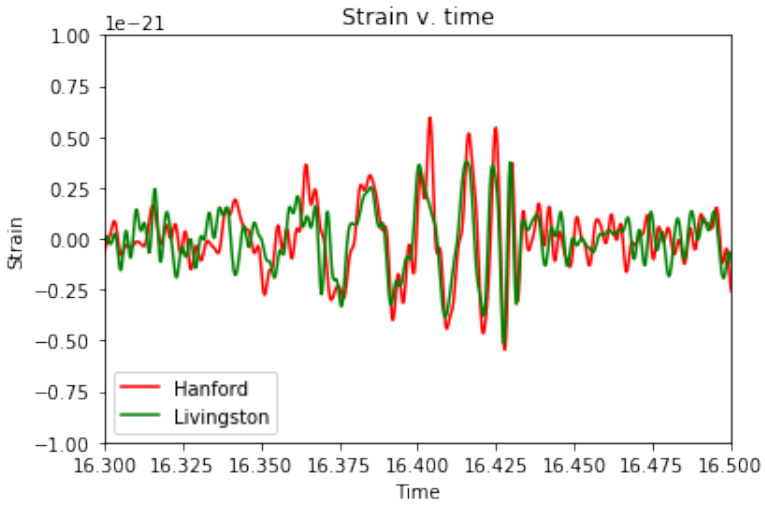

Figure 9: Periodogram after damping noise

Wow! This looks exactly what we expect a gravitational wave to look like! Inspiral, merger and ringdown are clearly visible. Beautiful. Note that the Livingston signal has been inverted and

shifted by 7ms to account for the relative position of the two detectors. To have a valid signal, detection has to occur within the propagation time of 10ms, and they clearly do here!

5 Conclusion

Time signals can be hard to analyze, while frequency components of the same signal can be clear as day. The Fourier Transform is an amazing tool letting us go between different spaces, allowing us to analyze complex signals in a simple frequency domain. The technique was demonstrated here applied to the LIGO detection of gravitational waves, and with very little effort one of the biggest discoveries of modern times was observed.