Measuring Hubble's Constant Using Sodium D Redshift and Gravitational Waves

Jacob Burgos, Jackson Eddy, & Christopher Ewasiuk[1]

[1]*Department of Physics,*
17 February 2021

In this report, we conducted an analysis of the measurements of a binary-star system's electromagnetic radiation and gravitational wave signature in order to calculate the Hubble Constant $H_0 = \frac{v}{D}$. The goal of this experiment is to show how data set manipulation can be used to extract a range of valuable information from measured values. In this experiment, we calculated the Hubble Constant to be $H = 79.2 \pm 5.7$ km/s/Mpc which falls just outside of the range of the accepted values of the constant: $H_0 \approx 70$ km/s/Mpc. We will also go over improvements to our methods which could decrease uncertainty and serve as a valuable guide for other future projects.

INTRODUCTION

The Hubble Constant, H_0, is an extremely important cosmological constant which indicates the rate at which the universe is expanding. It is defined as:

$$H_0 = \frac{v}{D} \tag{1}$$

where v is the recession velocity, the velocity at which two cosmological bodies move away from each other, and D is the distance between the two bodies.

The recession velocity v can be calculated by determining the redshift in radiation emitted by our binary-star system. The redshift is a perceived increase in radiation wavelength when two bodies are moving away from each other. The equation to calculate velocity from redshift is as follows:

$$v = c\Big(\frac{\lambda' - \lambda_{Na}}{\lambda_{Na}}\Big) \tag{2}$$

Where λ_{Na} is Sodium D's wavelength in a rest frame and λ' is Sodium D's redshift wavelength.

In order to calculate the distance between earth and our binary-star system, we analyzed data on the shear caused by gravitational waves on an unknown object. This can be calculated using an instrument known as an interferometer. An interferometer works by merging two sources of light to create an interference patter. If there is a disturbance within the fabric of space-time, matter will compress or stretch and these beams will appear out of phase. When the beams are out of phase, we can measure how much the beams are out of phase and calculate a variable known as shear (h). Gravitational waves cause an object to very slightly oscillate in size, and the frequency of these oscillations as well as the amplitude of the shear can be used to calculate the distance between Earth and our binary-star system using the following equation.

$$D = \frac{4c}{|h|}\frac{5}{96\pi^2}\frac{\dot{f}}{f} \tag{3}$$

In this equation, $|h|$ is the amplitude of shear, f is the frequency of the shear and $\dot{f}$ is the derivative of the frequency function.

METHODS

All of the data used in this report was provided without a method of acquisition, so this report will focus solely on how we analyzed the given data. All findings and visuals came from MatLab and MatLab's Curve Fitting tool.

Our method for calculating the recession velocity included analyzing a specific absorption wavelength range and corresponding flux for a distant binary-star system. Our goal for this analysis was to find the Red-shift in Sodium D. Sodium D is one of the Fraunhofer Lines, which are a set of dark absorption lines in the solar spectrum caused by absorption by solar elements in a star's atmosphere. Along with the data given to us, we were also notified that the non-Red-shifted absorption wavelength for Sodium D was 5896 $\mathring{A}$. Since the two star system that we are focusing on is moving away from us, we can safely assume that our absorption spectrum will be Red-shifted, with the specific wavelengths longer than their rest frame counterparts. Given the non-shifted wavelength for Sodium D, we are able to look for dramatic decreases in flux (due to the spectral absorption of Sodium D in the solar atmosphere) for wavelengths greater than 5986 $\mathring{A}$ within our given data set. With both the non-shifted and Red-shifted wavelengths for Sodium D, we can use Eq. 2 to calculate the recessional velocity of our binary-star system.

The calculation for the proper distance D, Eq. 3, required that we analyze the specific frequencies at which shear (h) fluctuates. Note that *shearing* in this case refers to the percent deformation of a distribution of particles due to gravitational waves, where a shear of 0 indicates no deformation, while a shear of 1 indicates the complete squashing of a distribution of particles. These waves come from the pushing and pulling that the stars perform on space itself as they spiral inward, so we should hope to find hints of periodic motion in the data, shown in Figure 2. To create a spectrograph of our data for h, we broke the entirety of our 232,609 data points into 377 segments (bins) such that within each segment the shear oscillated at a single frequency. We then preformed a Fast Fourier Transform on each bin to gather each individual frequency. We were then able to create a plot of frequency vs. time and, using Matlab's Curve Fitting tool, were able to create a continuous function for frequency as a function of time. We could then take the derivative of this function and use Eq. 3 to calculate the proper distance for each of our 677 bins. We can then take the average of our calculations for D and our value for recessional velocity v to find a value for the Hubble Constant using Eq. 1.

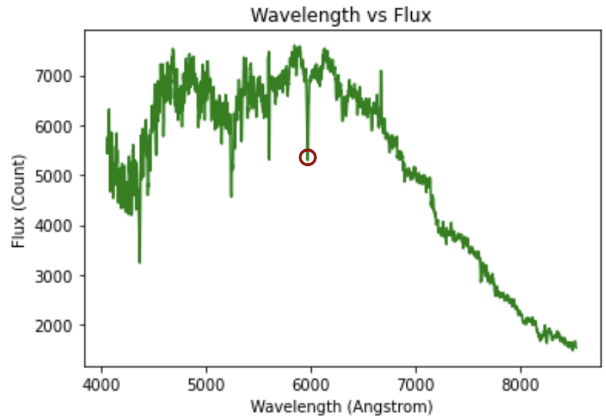

FIG. 1: *Absorption spectrum from the dual-star system with sodium D trough circled.*

Wavelength (angstroms)	Flux (count)
5968.59	5430.175
5970.13	5314.060
5971.68	**5293.947**
5973.22	5553.791
5974.76	5630.310

TABLE I: *A table showing the raw wavelength vs. flux data around the minimum (Sodium D Red-shifted wavelength).*

ANALYSIS/RESULTS

The first data set we analyzed was a flux count over an unspecified period of the solar spectrum emitted by a distant two star system. Our goal for our analysis was to find the Red-shift in Sodium D. In the rest frame, the sodium D emission line is at $\lambda_{Na} = 5896\mathring{A}$, so we can limit our search to troughs that are around and greater than that value. Here we've included a plot, Figure 1, of this data with a circle indicating the trough closest to our expected wavelength.

We deduced from this plot and the given data that the Red-shifted wavelength of Sodium D was $\lambda' = 5971 \pm 2\mathring{A}$. We reached an uncertainty of $\delta\lambda' = \pm 2\mathring{A}$ by pinpointing the range of wavelengths at which λ' was a minimum and including within our uncertainty the 3 wavelengths straddling this point. Table 1 includes the midpoint ($5971.68\mathring{A}$) as well as the surrounding data. Using Eq. 2, we were able to calculate our binary-star system's recession velocity to be:

$$v = (3.8 \pm 0.1) * 10^3 \text{ km/s}$$

We used propagation of errors to calculate our error in recession velocity:

$$\delta v = \frac{dv}{d\lambda'}\delta\lambda' = c\frac{\delta\lambda'}{\lambda} = 101 \text{ km/s}$$

The second set of data we analyzed describes the gravitational wave shearing ratio as a function of time, $h(t)$.

Looking at the MatLab plot of shear vs. time over a large time-range, Figure 2, it is impossible to see this periodic motion. However, on small time scales the periodic nature of shear becomes much more evident, shown in Figure 3.

In order to find the frequency of this oscillation as a function of time, we split the shear vs. time data into many small arrays of equal length (617×1) such that their plots resemble the roughly sinusoidal plot from above. The length 617 was chosen because it is

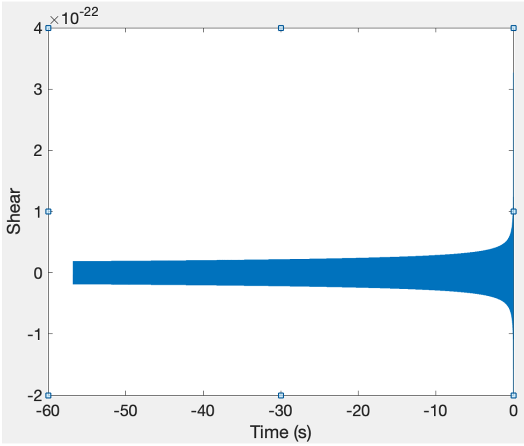

FIG. 2: *The full range of gravitational shear over the total measurement time.*

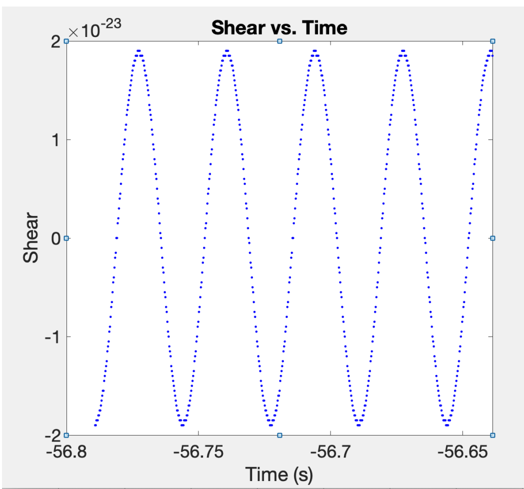

FIG. 3: *An example of one of our small arrays exhibiting periodic motion. Our data for shear showed nearly sinusoidal shear pattern for a small range of time.*

the largest prime factor of the length of our original shear data (232,609). The exact code used to create these arrays is:

```
b = 617;
n = numel(h);
c = mat2cell(h,diff([0:b:n-1,n]));
z = cellfun(@median,c);
```

Now, with hundreds of subsets of $h(t)$, we used the `meanfreq` function on each subset to find the mean frequency of that section, and assign the output to a new array, `freq`, effectively acting as our frequency as a function of time, $f(t)$. In MatLab's command window this would look like:

```
freq = [];
maxi = [];
for i = 1:377
    mf = meanfreq(c{i,1},1/0.00025);
    freq = [freq ; mf];

    maxh = max(c{i,1});
    maxi = [maxi ; maxh];
end
```

Our array fitting method allowed us to determine the frequency over time, however Eq. 3 requires the derivative of frequency as well. To calculate $\dot{f}(t)$, we generated an equation for $f(t)$ which we assumed took the form of a solution to Eq. 3:

$$D \propto \frac{\dot{f}(t)}{f^3(t)}$$

When solved, we find solutions of the form:

$$f(t) = (a_2 - a_1 t)^{-1/2}.$$

where all a_i terms are constants. It's worth noting that $|h|$ in Eq. 3 is certainly not a constant in time, as seen in Figure 2. However, relative to the other variables in the equation, $f(t)$ and $\dot{f}(t)$, the shear magnitude is roughly constant at earlier times, so we will treat it as such while attempting to find our best-fit function.

When attempting to tune this function to $f(t)$, MatLab gives an unsatisfactory solution with a correlation coefficient of $R^2 = 0.881$, but when we allow the best-fit function to include an additional constant term outside of the radical,

$$f(t) = a_3 + (a_2 - a_1 t)^{-1/2} \tag{4}$$

we see a significantly improved best-fit line, shown in Figure 4, and a satisfactory correlation coefficient of $R^2 = 0.985$, .

We are not including uncertainty in this value from the shear versus time data, however MatLab's curve fitting software kindly provides us with an uncertainty range for the coefficients in the best-fit function which we will treat as uncertainty is frequency:

$$a_1 = (7.755 \pm 0.317) \times 10^{-5} \text{ sec}$$

$$a_2 = (3.804 \pm 0.122) \times 10^{-6} \text{ sec}^2$$

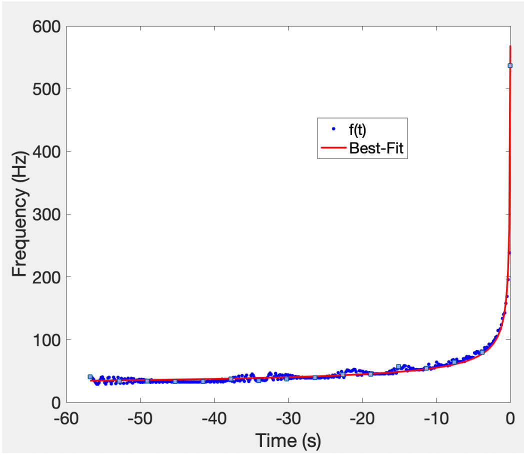

FIG. 4: *Our best-fit function (red) with our measured frequency data (blue)*

$$a_3 = 19.51 \pm 0.71 \text{ Hz}$$

To propagate the uncertainty in our coefficients into our frequency, we use the formula:

$$\delta f = \sqrt{\sum_i \left(\frac{\partial f}{\partial a_i}\delta a_i\right)^2} = 4.198 \text{ Hz} \tag{5}$$

again where a_i are MatLab's predicted coefficients. Thus, we have an uncertainty in frequency $\delta f = 4.20$ Hz.

All we must do to compute $\dot{f}(t)$ is feed the best-fit function $f(t)$ into MatLab's symbolic differentiator `diff(f(t),t)` which gives us back a symbolic function corresponding to the time derivative of $f(t)$. In order to find discrete arrays whose values match those of $f(t)$ and $\dot{f}(t)$, we can define new arrays as such:

```
tnew = linspace(min(t),max(t),377)
fitfreq = f(tnew)
dfitfreq = diff(f(tnew),t)
```

Now we have equal length arrays representing the raw frequency versus time data, the projected frequency versus time data, and the projected time derivative of frequency versus time data. The last term needed to compute the separation distance is the shear magnitude $|h|$ which we previously brushed under the rug as a constant. To fill an array with the shear magnitudes, all we will do is use the same function from earlier that assigns the mean frequency of a chunk of data to a new array, except now we'll be looking for the maximum value of $h(t)$ in that range. The script for this is already included in the frequency-finding script from earlier, where the shear amplitude array is named `maxi` which corresponds to $|h|(t)$.

Now that we have arrays of data for each variable appearing in the formula for separation distance, we can compute Eq. 3 and plot the output for each small time increment. Notice that, despite looking for a fixed distance, there is a range of D at various times shown in Figure 5: Though worrisome at first,

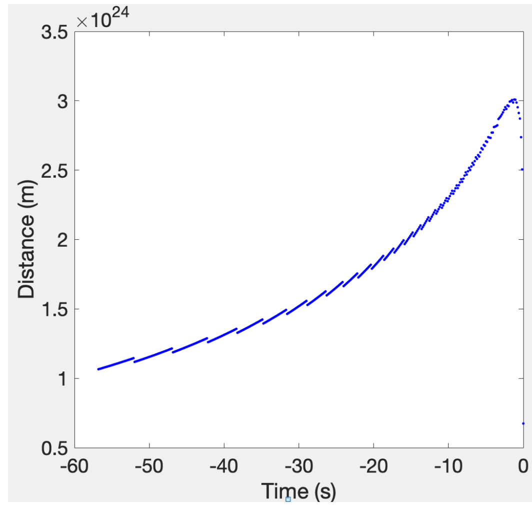

FIG. 5: *Graph of our separation distance which appears to fluctuate with time*

we see that the range of values of D is actually much smaller than the range of values found is $f(t)$, $\dot{f}(t)$, and $|h|(t)$ indicating that our D value is closer to a constant than Figure 5 might have you believe. This graph is not simply a zero-slope line at the correct value of D because of our modified best-fit equation, and specifically the constant a_3 that we introduced to strengthen our correlation.

To provide a single value for D, we will find the average value of D for early times, when Eq. 3 is most accurate, from $-59 \leq t \leq -30$ sec. Doing this, we find a separation distance of $D = (1.5 \pm 0.4) * 10^{24}$ m $= 48.0 \pm 12.8$ Mpc. The uncertainty here comes from $\delta f = 4.198$ Hz, so we can find the uncertainty in D as:

$$\delta D = \sqrt{\left(\frac{\partial D}{\partial f}\delta f\right)^2} = 4.402 * 10^{23} \text{ m}$$

Once we have values and uncertainties for velocity and distance, we can compute Hubble's Constant from Eq. 1:

$$H = \frac{v}{D} = \frac{(3.8 \pm 0.1) * 10^3 \text{ km/s}}{48.0 \pm 12.8 \text{ Mpc}} \approx 79.2 \text{ km/s/Mpc}$$

The uncertainty is propagated from uncertainty in velocity and distance as is shown below:

$$\delta H = \frac{v}{D}\sqrt{\left(\frac{\delta v}{v}\right)^2 + \left(\frac{\delta D}{D}\right)^2} = 5.68 \text{ km/s/Mpc}$$

Thus, we report Hubble's Constant to be

$$\boxed{H = 79.2 \pm 5.7 \text{ km/s/Mpc}}$$

DISCUSSION

The Hubble constant is one of the most important numbers in cosmology because it is needed to measure the size and age of the universe. The accepted value of the Hubble Constant is $H_0 \approx 70$ km/s/Mpc which falls just outside the uncertainty range of our Hubble Constant, $H_0 = 79.2 \pm 5.7$ km/s/Mpc.

The analysis methods discussed in this report are useful for analyzing semi-harmonic functions over large time scales. However, these methods could be improved. Notice from Figure 3 that our arrays of shear contained around 4 wavelengths. It is possible to calculate the frequency and magnitude of our shear with only 1 wavelength, meaning that the total number of values within each array could decrease by a factor of 4. This would, in turn, increase the number of data points in f and $|h|$ by a factor of 4 as well. The reason we did not attempt this here is because smaller prime factors of our total data set led to arrays containing a fraction of a total wavelength. MatLab's `meanfreq` function could not function properly with data sets of this size, so we provided the best analysis we could without dropping any data.

Another way our data analysis methods could have been improved is by developing an ansatz for f which included the time varying factor $|h|$. While our assumption that $|h|$ was relatively constant over the period of our measurements gave a nice line of best fit, the fact remains that $|h|$ was changing, however slightly, over our measurement period. By taking into account $|h|(t)$, our f and $\dot{f}$ values would have been more accurate and result in lower uncertainties.

BIBLIOGRAPHY

[1] What is the Hubble Constant, Adam Mann, livescience.com/hubble-constant.html