Analysis of the Hubble Constant and the Age of the Universe

•

Jakub Markiewicz

INTRODUCTION

Using multiple documents that contain information about cepheids in different galaxies, my task was to analyze the data and come to a conclusion about the age of the universe. This was achieved through four main steps which I will go into in more detail. I have used already known relations such as the relationship between the parallax of a star, and its relative distance away from Earth, as well as the relationship between the distance modulus, relative distance, and the extinction magnitude of a galaxy, and the cepheid period-luminosity relation. With all this in use, I managed to output a value of 13.773 billion years as the age of the universe, which matches quite nicely with secondary data *[1]*.

STAGE 1; Cepheid Period-Luminosity Relation

Using a datafile on 10 cepheids, the end goal for this task was to achieve a linear relationship between the absolute magnitude, M (the luminosity), and the logarithm of the pulsation period, logP. The equation for these two variables is given by:

$$\mathbf{M = \alpha \log P + \beta}$$

We are forced to estimate the absolute magnitude M, as we are given only the 'apparent' luminosity magnitude, m. This value differs from the absolute luminosity because for two identical light sources (with identical power), the one that is further away will appear to glow fainter than the closer one. When observing stars, we can only measure this apparent magnitude and are left to estimate the absolute magnitude mathematically ourselves.

The way to obtain absolute magnitude from apparent magnitude requires the use of another relation:

$$\mathbf{m - M = 5\log d - 5 + A}$$

As mentioned before, this is the relation between the distance modulus (m – M), relative distance, d, and the extinction magnitude, A (this is how much matter is obstructing our view to the object, i.e., gas clouds for newly forming stars). Along with the apparent magnitude, we are also provided the parallax, p, and the extinction for each object. Extracting the parallax values and substituting them into the relation between relative distance and said parallax will provide us the needed d for this process.

$$\mathbf{d = 1000/p}$$

At this point I also decided to begin propagating the given errors on the parallax (also provided in the data file). This step simply included taking the percentage error of the parallax on every one of the 10 objects, and then multiplying it back by the distance. This gave the error on the relative distance d.

Now with the errors sorted and the distance obtained, we can use the previous equation and substitute extinction A, logd, and apparent magnitude m, to get an estimate value for M, our absolute magnitude. First, we can rearrange the equation solely to give values for M:

$$\mathbf{M = m - 5logd + 5 - A}$$

Of course, it is important to remember to take the logarithm of the distance rather than plugging the distance itself into the equation. Here, we must also propagate the errors once more onto M.

My initial step was to find how the error propagates from d onto 5logd. For this I have used a pdf by Oregon University *[2]* for the step from d onto logd:

Error on logd = 0.434(error on distance/distance)

I then applied this method in my code. For then finding the error on 5logd the method is simple – I did this by taking the percentage error on logd values, and then multiplying back onto 5logd. The only remaining error to propagate onto M is the extinction error (the values on m are small enough to be negligible, and so are not provided in the data file). This can be done through the addition law of error propagation:

Error on (5logd – A) = $\sqrt{(Error\ on\ 5logd)^2 + (Error\ on\ A)^2}$

All our errors have now been propagated and we have simultaneously obtained M, the absolute magnitude. Since the period has been provided in the data file, we are only a few steps away from being able to plot our values.

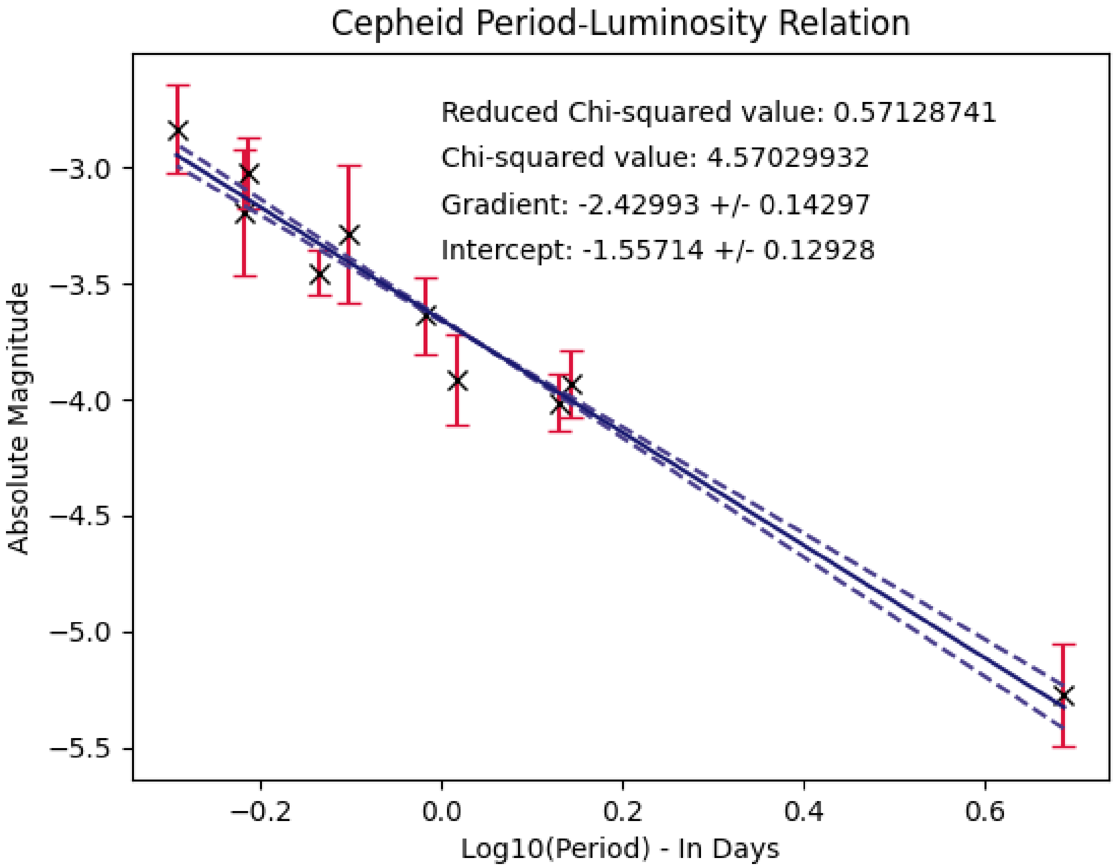
$$\mathbf{M = \alpha \log P + \beta}$$

The equation above requires the period to be in its logarithm form, therefore we use the NumPy python module to easily convert our values from the period column into what we need. There are no errors on the period pulsation values, and so there will be no errors on the x-axis.

Because we will be trying to fit a line of best fit for this linear model, we must 'uncorrelate' the gradient and intercept. Unless we do this, if we perform our chi square test and find something unreasonable, once we try to adjust either our gradient or our intercept one at a time, the other will be affected – depleting our best efforts to find a better fit. An easy way to do this is to take the mean of our x values (in this case, logP) and then shift all the original x values by that amount. As a result, what we plot will be a bit different from the equation above:

$$\mathbf{M = \alpha\ (\log P - mean\ of\ logP) + \beta}$$

The mean of logP was found by using another NumPy function. We now have all the values, and we are ready to plot.



In this graph showing the cepheid period-luminosity relation, we have plotted the log of the period on the x axis and the absolute magnitude on the y axis. The solid blue line represents the line of best fit for the points plotted, while the dashed blue lines are the lines of 'worst' best fit, as they consider the errors on the gradient and the intercept. As the reader can see, the errors on the gradient and intercept are not significant as such that they greatly deviate from the original line of best fit – this shows that our model is quite good. Our χ^2 value further implies a good model.

$$\chi^2 = \sum \frac{(Y_{data\ points} - Y_{best\ fit\ model})^2}{{Y_{\sigma}}^2}$$

The Chi-square test (χ^2) tells us how accurately our model defines the data by considering the sum of the squares of the difference between the data points, and the line used to best define them. If the spaces between the regression line and our points is not ideal, our chi square will get larger the worse it is. Considering two parameters, the acceptable χ^2 for this model is anywhere between 4 and 12, and our value falls close to the lower boundary. When we perform a reduced chi-square test using Y$\boldsymbol{\sigma}$ = 1, we obtain a value of 1. Since our minimum chi square is within the range of 1 +/- $\sqrt{2/(degrees\ of\ freedom)}$ (from 0.5 – 1.5, degrees of freedom = 8) This confirms that our fit, and our errors, are within the acceptable range.

STAGE 2; Obtaining the Distance Modulus

Now that we have successfully worked out the gradient and intercept for the cepheid period-luminosity relation, we can use it for data of many other cepheids to obtain the distance modulus.

The distance modulus is given by:

$$\boldsymbol{\mu = m - M}$$

This is simply the difference between the apparent and absolute magnitudes. To obtain this value I first had to extract data for the logarithm of the pulsation period and apparent magnitude values from 8 new files, each one showing information about a selected number of cepheids in 8 galaxies.

In order to get the distance modulus, we first must find the absolute magnitude using:

$$\boldsymbol{M = \alpha \log P + \beta}$$

Since we have already obtained a general gradient $\boldsymbol{\alpha}$ and a general intercept $\boldsymbol{\beta}$ from the first stage, all there is to do is to run a function which substitutes the data from the 8 galaxies i.e., the function will be called 8 times to process all the numbers. That is exactly what we do, and we obtain an absolute magnitude for every single cepheid in each galaxy. The function method works great, as it still separates the information by the galaxies in which they are in – allowing us to extend this function for further analysis.

Meanwhile however, we must propagate our errors once more. The errors included on the gradient and the intercept from stage 1 should give us a good estimate for the error on M. So, let us propagate this accordingly. We should once more use the sum error propagation law:

Error on M/ $\boldsymbol{\mu}$ = $\sqrt{(\mathrm{logP}\ *\ \mathrm{gradient_{error}})^2 + (\mathrm{intercept_{error}})^2}$

As there is no error on logP, the error only will consider the gradient and the intercept.

Now that we both M and its error, we can move onto finding the distance modulus. This step only involves a subtraction between m and M, and hence is also included in the mentioned function so that each cepheid will have a different distance modulus value, separated by the galaxy its in.

The error on M is also the error on $\boldsymbol{\mu}$ (distance modulus) because the error on m can be ignored. As a result, using the sum law will simply result on the error on M once again.

All the values for the distance moduli are printed inside the attached code.

STAGE 3; Converting $\boldsymbol{\mu}$ into Relative Distance

Now that we have the distance modulus, we can go onto converting this value into the relative distance. We require the relative distance (to earth), d, because our end goal is to plot the Hubble Constant graph; given by the relation:

$$\mathbf{v_{rec}} = \mathbf{H_0 D_{gal}}$$

Where $\mathbf{v_{rec}}$ **=** recessional velocity, $\mathbf{H_0}$ **=** Hubble constant, $\mathbf{D_{gal}}$ **=** d = relative distance (in megaparsecs).

Converting our moduli into distance requires the use of the equation mentioned before in stage 1:

$$\mathbf{m - M = 5logd - 5 + A}$$

Except that, as the reader might remember, m – M is equal to **μ**, therefore this equation looks like this:

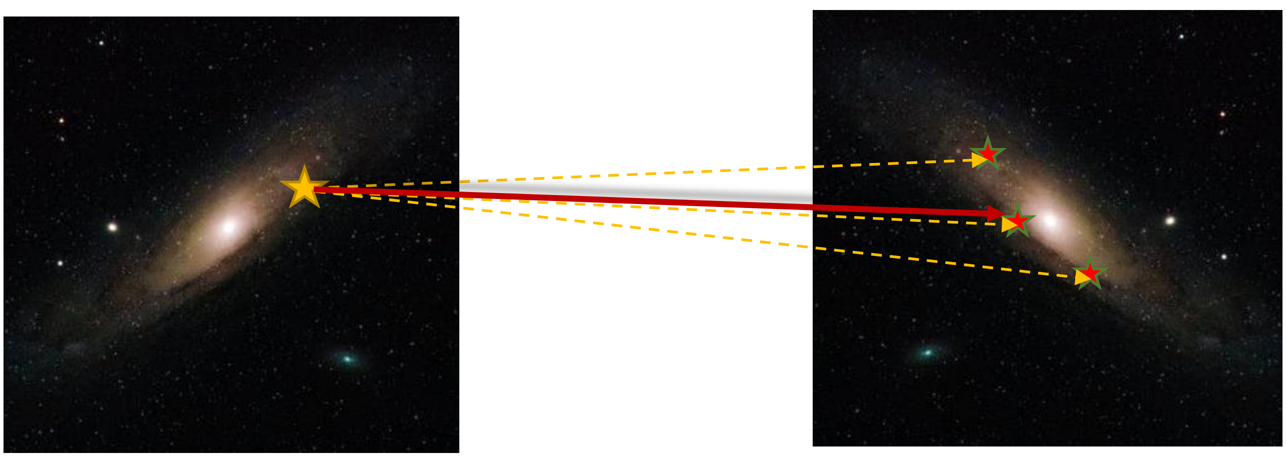
$$\boldsymbol{\mu} \mathbf{= 5logd - 5 + A}$$

We can extend our function from stage 2 to process all the data individually once more and output a relative distance for each cepheid. For this to work, we must add another variable for the extinction values for each of the 8 galaxies, included in a 9th data file that holds information about the recessional velocity and extinction as stated. We must substitute all our numbers into the equation which has been rearranged for d:

$$\mathbf{logd} = \frac{\mu + \mathbf{5 - A}}{\mathbf{5}}$$

$$\boldsymbol{d} = \mathbf{10}^{\frac{\mu + \mathbf{5 - A}}{\mathbf{5}}}$$

Once I have obtained these numbers, I once again used NumPy in order to take the mean average of every relative distance from earth to the cepheids for a given galaxy, in order to find the mean distance from us to the galaxy.



Stock free images from https://www.pexels.com (red line shows how finding mean distance for the stars in each galaxy obtains the mean distance to the galaxy itself)

As we have obtained the values for d, we should now go back and consider what our errors will be. We will use our errors on **μ,** to propagate onto d. The error on the distance will use the base 10 antilog law, which can be found in the pdf referenced before.

Distance error = 0.2*(*error on* **μ** * 2.303 * *distance*)

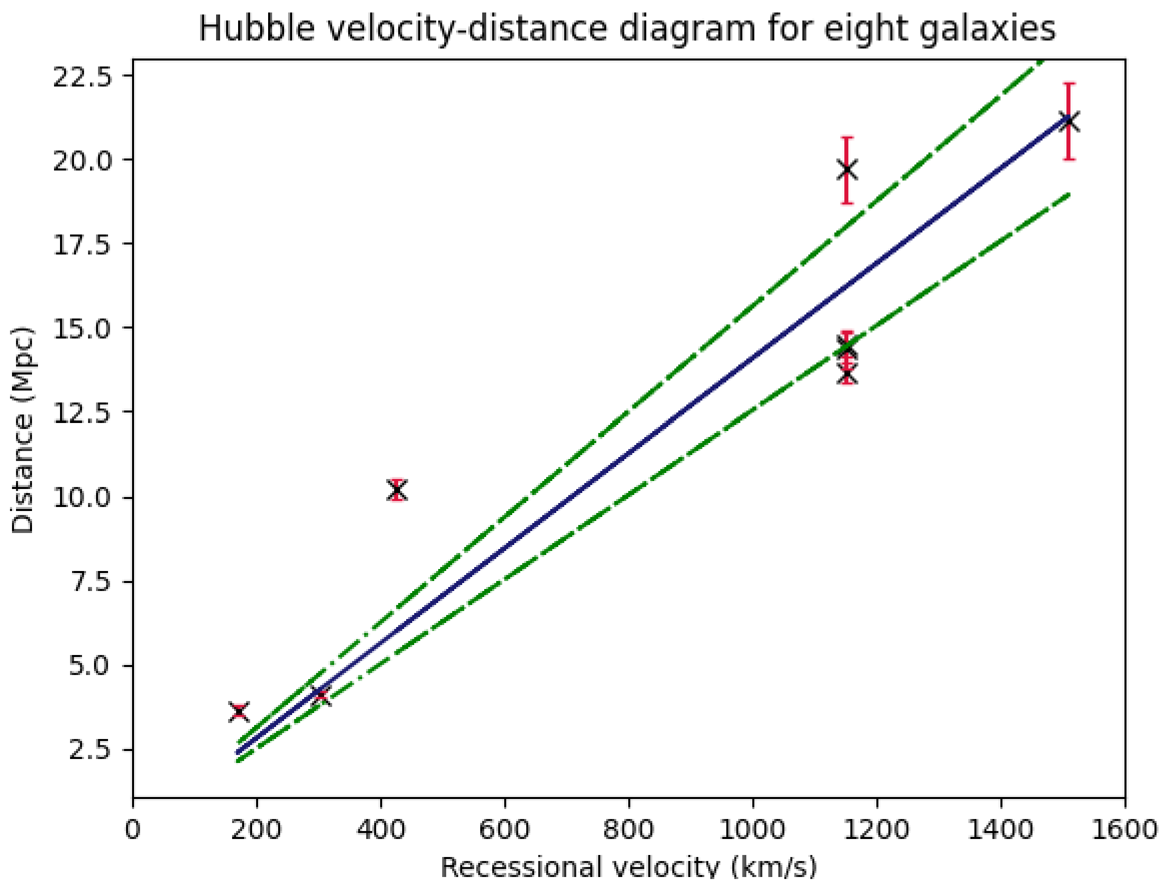
This is the error on the distance of each cepheid in each galaxy.

I then used the stats SciPy module to find the standard error on the mean of the relative distances for the eight galaxies. This is the error that I have further on used in plotting. All our errors have now been sorted out. We are ready to plot.

STAGE 4; Plotting the Hubble Constant

Before we plot our Hubble Constant graph, we must convert our eight distances into a more manageable degree. We can convert our distances in parsecs to mega parsecs by simply diving by a 10^6. We will call these distances Mpc (megaparsecs).

Once we have made sure we have extracted our recession velocity values we can plot.



For the equation mentioned before

$$\mathbf{v_{rec} = H_0 D_{gal}}$$

I have adjusted the axes so that the x-axis contains the data for the recession velocity and the y-axis contains our Mpc values. This means the gradient is adjusted to $1/\mathbf{H_0}$. This will come in handy later.

The blue solid line once again represents the line of best fit, while the green dotted lines show the lines of 'worst' best fit. In accordance with the model, the intercept of the model lines has been set to 0 – the equation shows a directly proportional relationship with a zero intercept. The increase in errors as the distance increases (and as so as recession velocity increases also) is shown by the deviating green lines away from the solid. The lines deviate the faster the galaxy is moving. Admittedly, the lack of a large sample size for different galaxies is something that hurts this model the most. Only two of the eight galaxies make it onto the line of best fit, and the rest deviate quite largely. This con is what will cause quite a large error in our result for the age of the universe.

Nevertheless, performing a reduced χ^2test gives us a value of 1 – meaning our model is sufficient for the data we have used and analyzed; with 7 degrees of freedom, our original chi-square reduces from 412 to 58, and this is about as good as we can get it by adjusting the gradient manually. We can extract the error on the gradient by using the 1st position in the covariance matrix obtained from using curve_fit from the SciPy.optimize module. We can now use our data to find our Hubble constant, the age of the universe, and our associated error on it.

Since our gradient = 1/$\mathbf{H_0}$**,** then the $\mathbf{H_0}$ **=** 1/gradient. This is our Hubble Constant, 71.01.

The error on the gradient follows the same rearranging. Our Hubble error comes out as 9.14 – although this quite a large confidence interval we must once again consider our sample size. Was our sample size a good representation of the total population of galaxies in the universe? If we had access to the entire population, would our values be any different? The obvious answer to these questions is absolutely, and therefore this error value is acceptable for this data analysis I believe.

Our age of the universe is the gradient of our graph: 0.014080882783231972 Mpc/kms^{-1}

Yes, this value is not very nice visually. Therefore, some conversion to have this number in billions of years is going to be useful.

To do this we must first convert our answer back from megaparsecs to parsecs, we can do this by multiplying by 10^6. To convert from parsecs to kilometers, we can multiply by a factor of 3.086e+13.

The kilometers cancel, and we have our answer in seconds. To convert seconds to years, we must divide by 3.154e+7 (this is the number of seconds in a year).

Finally, we can divide by 10^9 to find out answer in 'giga' years, or billion years.

The full equation will be:

$$\text{Age in billion years} = \frac{(gradient \times 10^6)(3.086\times 10^{13})}{(3.154\times 10^7)(10^9)}$$

Our value for the age of the universe: 13.777 +/- 1.7734 billion years.

Evidently, our large error on the Hubble constant translated to a large error on our age of the universe. However, despite our largest uncertainty originating from a small sample size, our value falls right on figures from secondary data (reference 1), in which livescience.com makes a report that the most precise value is around 13.77 albeit with a much smaller error of only 40 million years.

REFERENCES

1 -> https://www.livescience.com/universe-expansion-atacama-hubble-constant-measurement.html#:~:text=Ancient%20light%20from%20the%20Big,or%20take%2040%20million%20years.

2 -> https://sites.science.oregonstate.edu/~gablek/CH361/Propagation.html