**Hubble Constant Analysis**

**Data Structures**

The first task was to read the data from the files provided and store it in the program in a useful format.

I created three python classes for this purpose.

*Class name Data filename*

GalaxyData galaxy\_data.dat

MWCepheids MW\_Cepheids.dat

GalCepheids hst\_gal[1-8]\_cepheids.dat

Each of these classes comprises a list dicts of the data. A dict holds a record in key/value pairs where the key is the column name commented out at the head of the file. Additional fields have been added to the records to hold derived data, that is values that are to be calculated from the provided data.

For example, the GalaxyData class has fields for Galaxy Name (‘Name’), Recession Velocity (‘Recession’) and Extinction (‘A\_V\_MW’) which are read from galaxy\_data.dat. I have added a further column for the galaxy’s distance (‘dpc\_v’) which is initially set to a null value.

In order to calculate H0, it is necessary to populate this dpc\_v column, and this is achieved by the steps described below.

Note – I shall describe the procedure for the Visual band – the data for the IR band is processed in exactly the same way, thus there is also a dpc\_i column.

**Outline Of Procedure**

We can obtain a reasonable value for the dpc of a galaxy by calculating an average of the dpc’s of the cepheids within it. These can be calculated from the Absolute and Apparent Magnitudes of the cepheids along with the Extinction value of their galaxy. We have already been given the Apparent Magnitude and Extinction so we need to calculate the Absolute Magnitudes, M\_V in the case of the Visual band.

For cepheids within the Milky Way, we can calculate M\_V from its Apparent Magnitude, m\_V, and parallax. However, for stars in distance galaxies we do not have the luxury of using the parallax method because of the large distances involved.

Since Absolute Magnitude can be obtained from the formula:

M\_V = alpha \* log(P) + beta

where P is the period of the cepheid. We can calculate alpha and beta (in both the Visual and IR bands) from the Milky Way galaxies and then use these values to calculate M\_V in the galaxy cepheids.

**Milky Way Cepheids**

I added additional columns to the MW Cepheid data to hold log(P), dpc, mu, M\_V (and M\_I). log(P) was simply populated from the log of the Period. This enabled me to calculate dpc from the formula:

M\_V = m\_V – 5log(dpc) + 5 – A\_V

Using the python graph plotting library matplotlib.pyplot, I entered my values of M\_V and log(P) and obtained values for the gradient and y-intercept of the graph, giving me values for alpha and beta.

Unfortunately, I did not have time to calculate the errors associated with my input data.

**Galaxy Cepheids**

To allow access to data across the three classes I had constructed, I created a new class Galaxy, which contains galaxy data for a specific galaxy along with its list of cepheids. I also added MWCepheids to Galaxy in order to provide the alpha and beta values that this class had calculated.

A dict of Galaxy objects is created in main() keyed on Galaxy Name (as obtained from galaxy\_data.dat).

I was now able to calculate M\_V for GalCepheids from log(P), alpha and beta. This then enabled me to calculate dpc for each Cepheid within the galaxy. The average of these was then taken to populate dpc in the Galaxy class.

At this point the Visual and IR values of dpc are combined to give an average dpc for each galaxy. This is then used to calculate H0 from the formula:

H0 = v\_rec / dpc

Taking an average of the values of H0 for each of the galaxies gives a value for the Hubble Constant.

This is then used to calculate the age of the universe from the formula:

tau = 1 / H0