Final Write up: Star Clusters and Power optimization

Daniel R Herrera

May 11, 2019

1 Introduction

In today's write up, we will be discussing 2 different subjects. The first of which will be star clusters and the analysis of such systems. We will be taking a look at how their positional vectors are effected depending on the relative distance each star has from one another. We will also be taking a look at the approximate speeds each of these stars will have within their given system.

For he second project, we will be discussing how to maximize a given batteries power given its internal resistance values. We will be primarily looking at the variations of Ohm's Law and will see how the optimization functions in python will help us determine that both resistance values must equal each other in order for power to be maximized.

2 Coding

As usual, the actual code will be posted alongside the LaTex file on my github account. However, I will be walking through the logic and process I went through in order to get the information I got.

2.1 Star Clusters

In order for us to understand the positional vectors created from these star systems, we will be focusing our code for this section on the pandas, mplot3d, and mpl-tookits for mplot3d. Once we read our information into an array for the code to read, we then have to set up a 3d plot in order to show the position of these star systems at a given time. At t=0, we plotted the following figure:

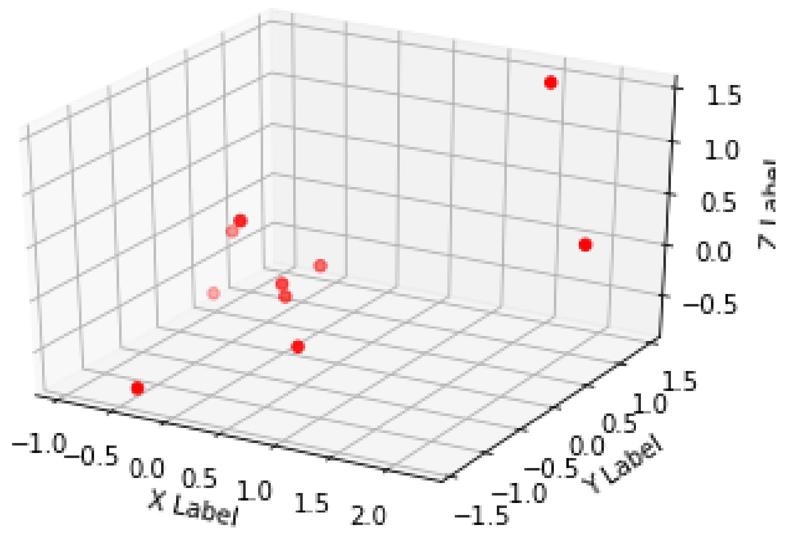

Figure 1: Position of first 10 stars at t= 0.

Note that this is only the first 10 stars in the given spreadsheet as there were over 10,000 stars in the csv spreadsheet and plotting them all among themselves would be difficult to interpret.

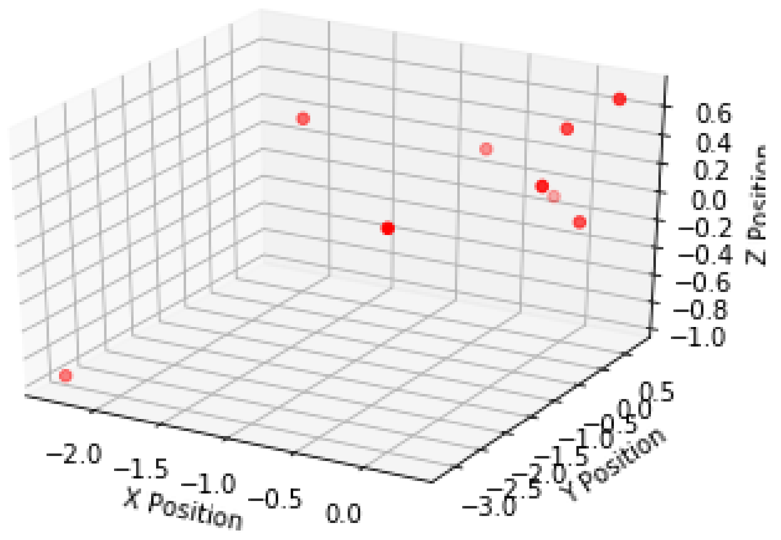

Figure 2: Position of first 10 stars at t= 1.

After inputting the quiver function, we were able to implement it into the code to determine the distance in which the stars traveled from t = 0 to t = 1 as shown below:

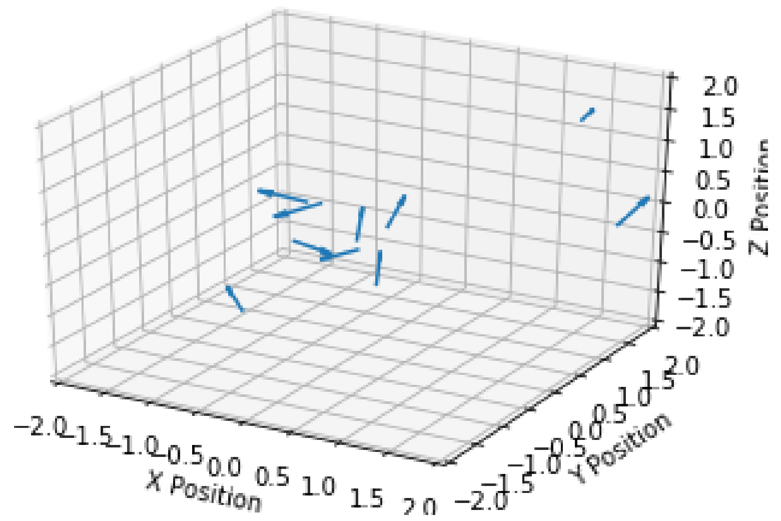

Figure 3: Positional vectors of first 10 stars from t=0 to t= 1.

We did this a few more times to give us an idea of how the stars were moving in accordance to the others around them. Implementing the methods from above, we get the following:

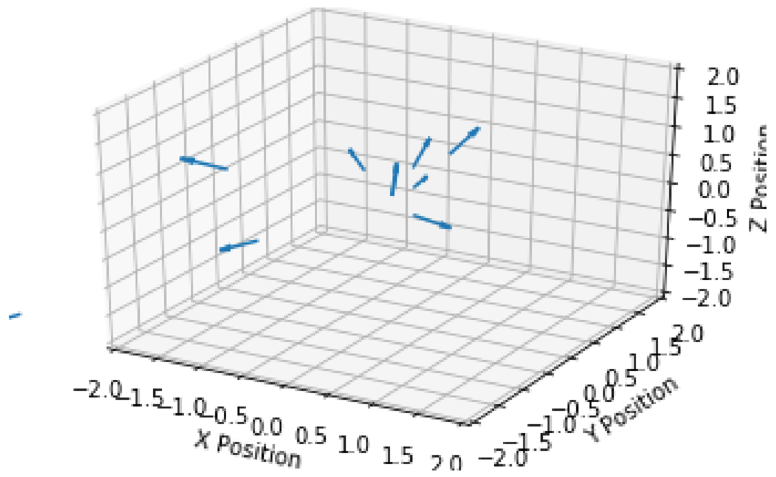

Figure 4: Positional vectors of first 10 stars from t=1 to t= 2.

Finally, we were able to see the distribution of the overall velocity magnitude for each individual star using the following equation:

$$|V| = \sqrt{(v_x)^2 + (v_y)^2 + (v_z)^2)} \tag{1}$$

Once the velocities magnitude was determined for each star, we were able to create a histogram distribution for the first group of stars.

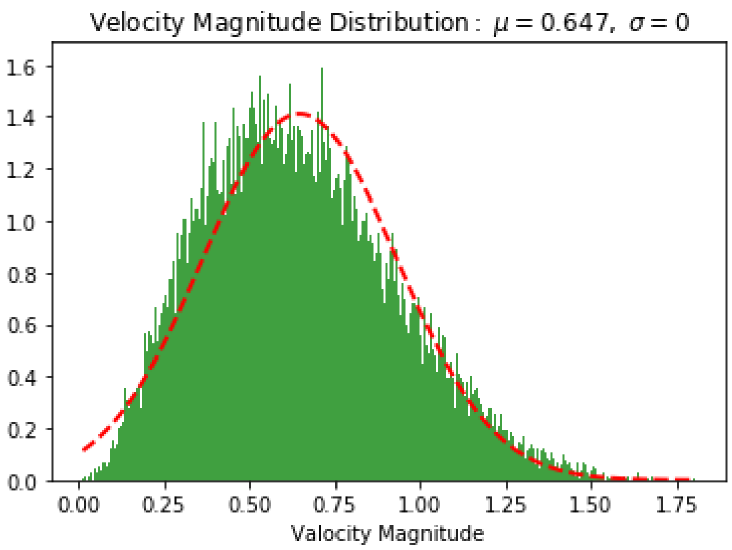

Figure 5: Distribution of velocity magnitudes for star cluster.[4]

2.2 Maximum Power

For this section, it was my intention to type out some code that will potentially optimize the maximum power of a battery by varying the resistance values. This, in turn, will help establish the Maximum Power transfer theorem.

First I had to import the relevant libraries: this includes math, pylab, scipy.stats for norm as well as scipy.optimize for minimize. This may have been clear to many before the fact but the optimization for scipy.optimize does not include a maximize function and instead only comes with a minimize feature. This means I had to manipulate the code and the data in order to get the data that I had wanted to get.

I then had to input several self defined functions as well as my objective function. The objective function is[2]

$$P = \frac{V^2 R}{(R + r)^2} \tag{2}$$

In order for us to determine the maximum value for power, however, we need to take the derivative of the Power function with respect to (R). [1]

$$P_{max} = \frac{\partial P}{\partial R} = V^2 \frac{r - R}{(R + r)^3} \tag{3}$$

This shows that the derivative will be 0 if r = R. Keeping this in mind, we will manipulate the last function which then gets reduced to:

$$P_{max} = \frac{V^2}{4r} \tag{4}$$

Knowing this, we now have the constraints we need in order for us to use the minimize function given to us by the scipy.optimize library. However, my earlier concern then pops up: We are only able to minimize and not maximize the function outputs. This leads to my first failed attempt. We next have to incorporate the derivative and new constraints in order to see if we can fully maximize the results. When viewing my notebook, you can see that this also fails. My third attempt finally tries to maximize the solutions by putting a sign flipping argument for the objective function as well as the constraints. This in turn returns two values in my array that results in r and R being extremely close values having a % error of only about 1.16%.

3 Analysis

3.1 Star Cluster

From figure 1 and figure 2, we can see that the star clusters tend to move overtime (obviously). but its not until we take a look at the positional vectors when we realize how these star clusters move. If you take a look at figure 3, the positional variance from t=0 to t=1, we can see that how the position of each star moves relative to other stars around it. My original assumption was that the stars would uniformly move towards other stars, however, the positional vectors tell a different story. They are moving away form one another in different directions in 3D space. It was not until this moment that i realized that these distances are in the magnitude of 5 light years. For every 1 positional variance in the position and variance graphs, there are 5 light years between them. What this means is that the distance is so far that their gravity essentially is too weak to radically shift other star positional vectors and speeds.[3]

Next, we have the histogram of the magnitude of velocities. We can see that we have a typical distribution of speeds resulting in a bell curve with a slight lean off going to the right of the spectrum. We can seer that the velocities peak around 0.6 on the velocity graphs.

3.2 Maximum Power

When we already know how to maximize the Power output of a battery, the development of the code becomes a lot more simple. We were aiming for the resistor values of R and r to be the exact same and, as how most optimization tools are, we were able to get a close approximation of these values. When we take a look at our last two variables in the x array, we can see a value of 3.567 and 3.609. It was to be expected that the optimization tool would not get it down to a 100% accuracy but it did come to about 1.1637% of the expected values.

4 Conclusion

From my findings of the star cluster, I was surprised to find out that the positions of the stars had little to no significant impact on the position or velocity shifts of other stars from the long distance apart they had form one another. In fact, according to further positional vectors of the same 10 stars, the majority of the stars stay on the same trajectory for the majority of the time period given to us on the data sheet. The distribution of the velocities was also not much of a surprise. the vast majority of 'fast moving stars' move at approximately 65 km/s to 100 km/s and we can see that a vast majority of our stars within this cluster are within that sweet spot range with the average actually being around 0.65 or 65 km/s.

As for the maximum power theorem of the battery, since we already knew how the interaction between the two r and R values would react with one another, it's not surprising to see that the maximization code we made in python was able to come rather close to the predicted value. Overall, I was pleasantly shocked to see that my code was able to give such accurate optimization readings for the maximum power theorem.

References

[1] Optimized maximum power, 2014.

[2] Maximum power theorem, 2017.

[3] Boojum. Milky way galaxy and other stars, 2003.

[4] Mario Pasquato. Star clusters, 2016.