William Steltz

Professor Clark and Teuben

Physics 265

March 9th, 2025

Lab 1 Report

## Section I

I was able to create a code that can be used to calculate several aspects of gravitation and how it will affect a rocket or launcher. I was able to create functions to calculate the gravitational potential of the Earth, of the Earth-Moon system, the gravitational force field of the Earth-Moon system, and calculate the run-time and altitude of the Saturn V stage 1. I was able to complete this using JupyterLab in python using imported libraries like numpy, matplotlib and scipy.integrate.

## Section II

In order to calculate the gravitational force of the Earth-Moon system, I had to obtain the mass and size of these bodies, along with their distance from each other. I used 5.9e24 kilograms and 6.372e6 meters for the mass and radius of the Earth. I used 3.8e8 meters for the Moon's distance, and 7.3e22 kilograms for the mass of the Moon. I oriented my plot on Earth at the origin (0,0), and created a function that calculated the gravitational potential of the Moon and Earth separately using (-G * M)/R where G is the gravitational constant, M is the mass, and R is the distance away from the center of mass. Then I added these potentials together so I could see them related to each other. I then made a plot using matplotlib.pyplot so that I can see the potential related to location. I made the plot three dimensional using the “projection = ‘3d’” command. I then added a colormap called “binary” as it made it easy to see both potentials. This code gave me an easy to read plot that shows the gravitational potential energy on the z-axis related to position on the x and y axes.

Next, I made a contour plot of the potentials. I used the same values and function as I did in the previous plot, but instead of making it a subplot, I used the contour function. I used the same axes with x and y being positional and z showing the potential. Since gravity is much stronger when you are close to the center of mass, the contour lines were all bunched up close to the Earth and Moon making it hard to read. To fix this, I used the "levels =" which affects the number of lines, and set it to a logarithmic scale. This made the plot easy to understand and shows the potential.

## Section III

For the gravitational force, I created a similar function to the one I made earlier but this one calculated the x and y components of the gravitational force from the Earth and the Moon and added them together. I then created x and y values that spanned from -1.5 times the distance of the Moon to +1.5 times the distance. I then used the "vectorize" command to vectorize my force function with my x and y values. I then created a color norm using a logarithmic scale so that I can create a streamplot with readable colors. I then created a subplot and made it a stream plot showing the force in space with vectors according to x and y coordinates. I used the colormap "hot" and added a color bar that was logarithmic. This code resulted in a streamplot with vectors all pointing towards the middle with a little disturbance where the moon is, which is accurate to the force of gravity.

## Section IV

I later wrote code predicting the performance of the Saturn V Stage 1 rocket. I found the burn time by taking the wet mass, dry mass, and the fuel burn rate. I created an equation that was (wet mass - dry mass) / burn rate to get an estimated burn time of about 157 seconds.

Next, I found the change in velocity for a given time by creating a function that takes an input for time. The user inputs a number of seconds that is used in the function. The function requires five variables, the time given by the user, wet mass, dry mass, fuel burn rate, and the acceleration due to gravity. The function also returns a change in velocity of 0 m/s when the fuel is exhausted. I used Equation 3 in the packet and put that into my function.

Last, I found the altitude of the rocket by using Equation 4. To do this, I had to use the line “from scipy.integrate import quad.” This allows me to compute an integral in my code by using the command “quad.” With this, I computed the integral of the change in velocity from 0 to T, where T is the total burn time, which I calculated earlier. I defined two variables when using the integral, since running this command returns the answer and the error, and the error was not necessary.

**Section V**

With the real rocket burning for 160 seconds and reaching an altitude of 70 kilometers, my predictions of 157 seconds and 65.9 kilometers seems slightly off. This may be due to rounding errors in the different masses. I only used two significant figures in numbers that had four to seven figures. I also never factored drag into my functions, which may have also affected the estimate.