Phys 430

Project I: Simulating real-world spacecraft

Joseph Chang

Mar. 15th, 2020

Introduction:

A famous quote came out a few years ago that says the computers we use daily are more powerful than the supercomputer NASA used in the 1960s for the Apollo program. Therefore, I have always wanted to simulate launching a rocket on my personal computer. In this project, I use the Euler-Cromer method to simulate space crafts on two dimensions (both cartesian and polar coordinates) with Newtonian mechanics

$\left(\ddot{\vec{x}} + \ddot{\vec{y}}\right) =- \frac{GM}{x^2+y^2}\hat{R} - \frac{\rho(h)}{2\,m(t)}v^2 C_d A\hat{v} - \frac{\vec{v}_{exhaust}\dot{m}}{m(t)}$. In the last simulation (section 4 in the project), I simulate stage I and stage II of Saturn V rocket. In that simulation, I include the effect of gravity from the earth, air drag caused by the ground atmosphere and low orbit atmosphere, and the propulsion from the trust engine on the rocket. I found the drop position of stage I and the orbit of stage II.

Method:

1. Dimensions

Using three dimensions, X, Y, and Z, would be the most realistic way to define the motion of a classical object. However, due to the conservation of angular momentum, two dimensions X and Y, are good enough to simulate an orbit. Also, a three-dimension system is more expensive than a two-dimension system in terms of computation. As a result, using a two-dimension system is a reasonable trade-off while simulating an orbit. To make the code of simulation robust, I define a class object called Motion2D. Motion2D can perform the conversion between cartesian coordinates and polar coordinates. Also, Motion2D is a 2-D vector that can perform vector addition, subtraction, and scale.

2. Newtonian mechanics versus Lagrangian mechanics

The Lagrangian mechanics, $\ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r}$, I learned in Phys 419, is a great tool while calculating orbits because it can maintain the conservation of energy. Emily Saaen shows that using Lagrangian mechanics to compute orbit can be easily done [1]. Nevertheless, my final goal is to simulate a rocket launch, and drag is a critical factor when flying a rocket. Unfortunately, Lagrangian mechanics suffer from friction forces since the system is losing energy. Therefore, I will use Newtonian mechanics to mimic gravity $\vec{F}_g = -\frac{GMm}{R^2}\hat{R}$, drag $\vec{F}_D = -\frac{1}{2}\rho v^2 C_d A\hat{v}$, and thrust $d\vec{v}_{Rocket} = -\frac{\vec{v}_{exhaust}dm}{m(t)}$.

3. The numerical method

I have learned many great ways to calculate differential equations in this course. Nonetheless, this project is complicated already. So, I will be using the Euler-Cromer method to solve the differential equation of the system. Although the Euler-Cromer method does not keep the conservation of energy, it constrains the error scale of the energy in an acceptable range. The Euler-Cromer method is the following:

$$dv = a(t)$$
$$v_{t+1} = v_t + dv$$
$$x_{t+1} = x_t + v_{t+1}$$

4. Air density

As I mention above, drag force is a critical factor when simulating rockets. Because the rocket is launched vertically and the height varies air density, I need to find the function of air density at height Rho(h). To obtain the function Rho(h), I collect standard air density data [2] at a height range from -1,000m to 80,000m and Jacchia air density data [3] at a height range from 90,000m to 2,500,000m. First, I shift the height data 2000m upward, so the data is in the domain of logarithm. Then, I take a logarithm on the data. Third, I fit the logarithm data with a quadratic equation. Finally, after getting the coefficients of the quadratic equation, I reverse the process to obtain the air density at the height function Rho(h). The program for finding Rho(h) is in the "AirDensity_Height_regression.m" file. The equation I obtained was:

$$\rho(h) = e^{-35.49} \times (h + 2000)^{10.63} \times (h + 2000)^{-0.7334 log(h+2000)}$$

5. Defining the properties of a spacecraft

I defined another class object called SpaceCraft that can record the classical motion, mass, fuel, and exhaust velocity of a spacecraft. This object was not utilized until section 4 of the project. SpaceCraft class also includes the rocket equation to calculate the acceleration of the spacecraft.

6. Minor testing code

I wrote two .m files of code to test before I started the code on the project. "test_Motion2D.m" is to test if the Motion2D class can make the range of theta on polar coordinates to (-pi, pi] after converting from cartesian coordinates to polar coordinates. "Rocket_equation.m" is just to test if the Euler-Cromer method works rocket equation.

Verification of program:

Other than the minor testing codes, I use real-world data to be the initial condition of the system. I plotted the result of my calculation and compared them to the real-world result. My result is physically corrected.

Calculating and data:

1. Section 1: Simulate International Space Station with real-world data

In this part, I calculate the orbit of the ISS (International Space Station) in an ideal condition in which the ISS is only affected by the gravitational force. I let the center of the earth be (0, 0). I plugged in the data of gravitational constant, earth mass, and earth radius, and I obtained the data of ISS from the ISS Wikipedia page [4] as the initial conditions. Then, I let ISS fly for a period (about 90 minutes). The result is shown in the following graphs.

Figure 1 is the angle of ISS versus time in seconds. Because the orbit of the ISS is nearly circular, the plot shows a linear line. The reason why there is a big angle drop at half period is due to the domain of angle I defined in the program.

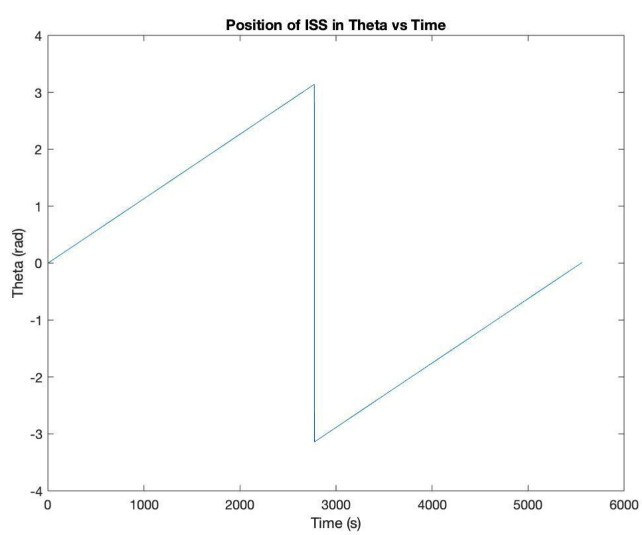
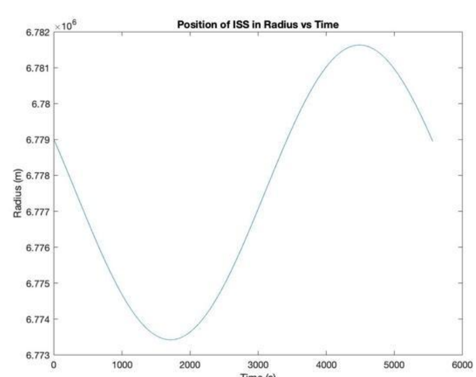


Figure 2 is the radius of ISS versus time. This graph shows that the orbit of the ISS is not exactly circular.



The difference between the major axis and the minor axis is on a scale of 1/100.

Figure SEQ Figure * ARABIC 2

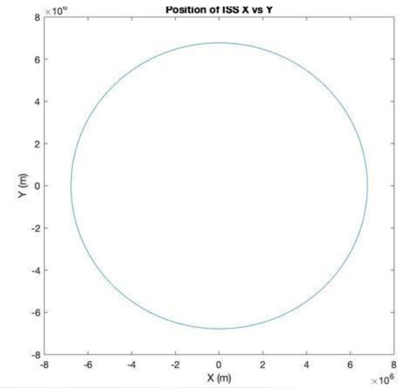
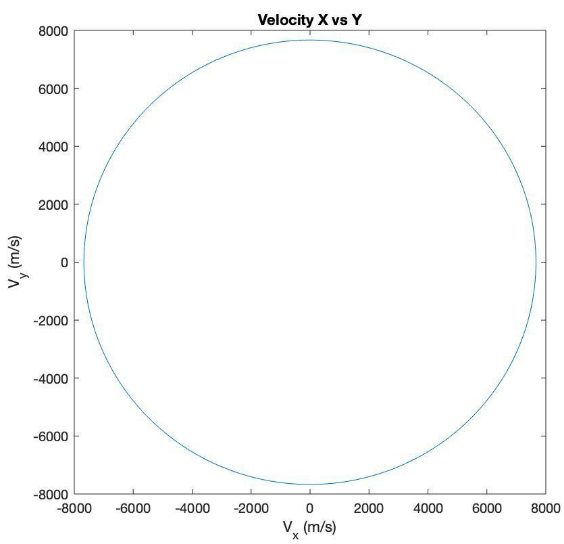


Figure 3 shows the orbit of ISS in X and Y Cartesian coordinates. It is a circular orbit.

Figures 4 and 5 are the velocity and acceleration of the orbit in X and Y Cartesian coordinates. The acceleration plot shows that the acceleration that ISS experiences are about 8.7 m/s^2.



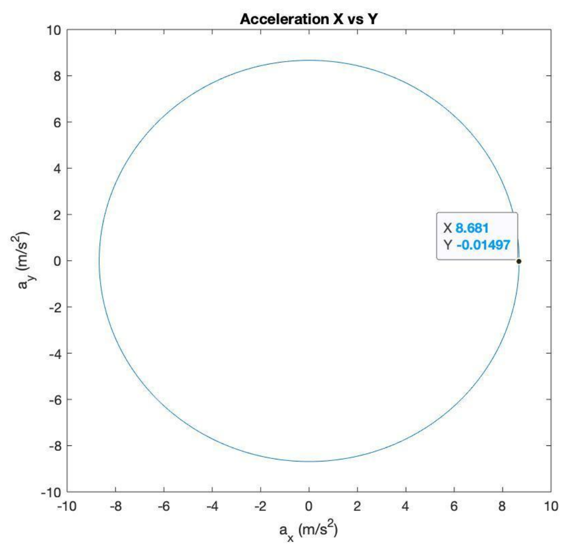
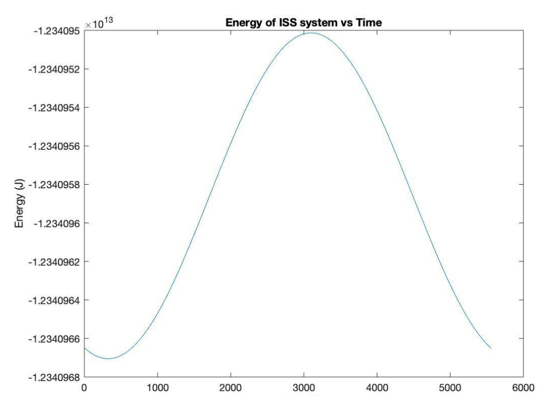


Figure 6 shows the energy of the orbit system varies with time which is the issue of the Euler-Cromer method because the energy was supposed to be conserved. However, since the



energy oscillates back, the result is still considered a good approximation.

2. Section 2: Simulating ISS orbit for more periods

I let ISS travels for 100,000 seconds (18 periods) to see if it stably can stay in the orbit. The result is shown in the following graphs.

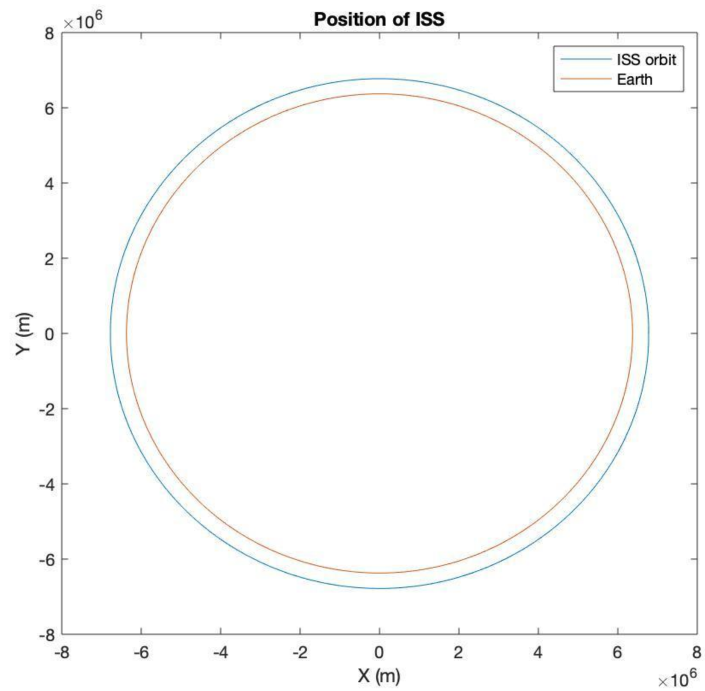


Figure 7 shows the position of ISS for 18 periods in XY. I also plot the earth to give an intuition of the earth's scale. As the figure shows, the orbit is stable.

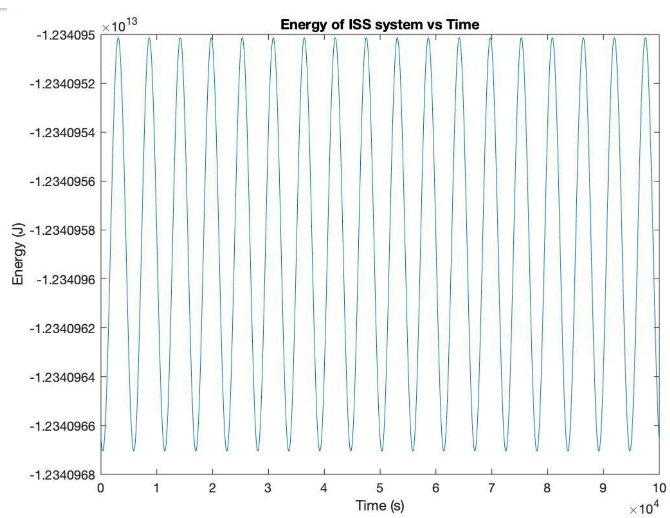


Figure 8 shows the energy of the orbit simulated by the Euler-Cromer method. Although the energy does not conserve, the change of energy is small compares to the total energy of the orbit. Also, the energy always oscillates back after every period. As a result, the change of energy is ignorable.

From the results of section 1 and section 2, I have proven that the Euler-Cromer method can simulate the physical orbit in the real world.

3. Section 3: Simulate ISS in realistic

In reality, ISS need to re-boost every once in a while to maintain the orbit height because the drag does exist in low orbit. As I mention in the method, I collect the data of air density and obtain an air density at height Rho(h) function with non-linear regression. Then, I plug the information into the quadratic drag equation $F_D = \frac{1}{2}\rho v^2 C_d A$ from Wikipedia [5] to calculate drag, and I combine the drag into the Euler-Cromer algorithm. I let ISS flies for four periods, and the result of the calculation is shown in the following graphs.

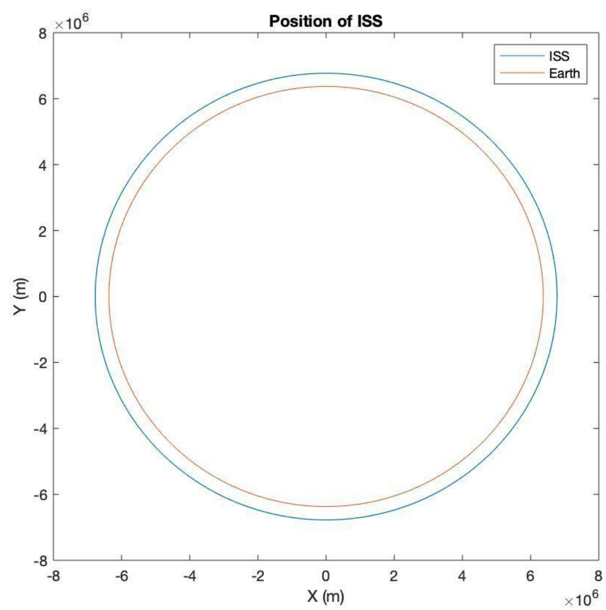
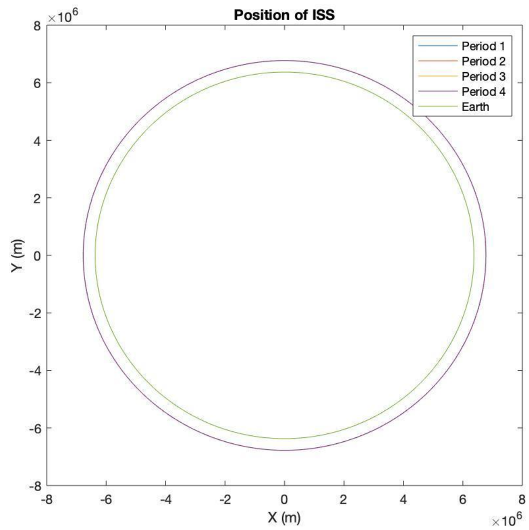


Figure 9 is the position of ISS for 4 periods. This plot is similar to figure 7 because drag is small on the earth scale. In reality, it takes months for ISS to fall at a significant altitude, and 90 minutes is one period. As a result, thousands of periods need to be calculated to see ISS falls. It is too expensive to do such a simulation, so I won't do it here.

Figure 10 is the same as figure 9, except that I label different periods in different colors. Figure 11 is the zoom-in view of figure 10. It shows that the orbit height falls about 20,000m every 90 minutes.



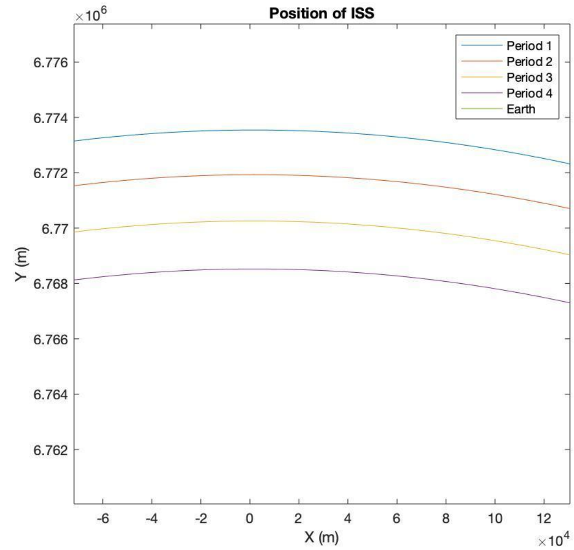
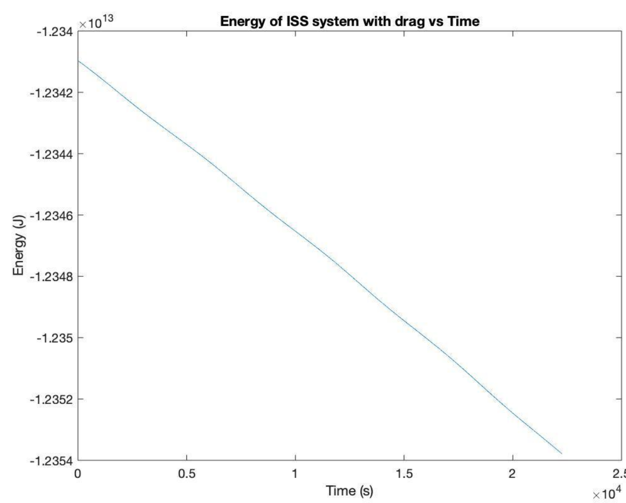


Figure 12 is the energy versus time plot. It shows that the energy of the orbit decreases linearly due to drag. Also, it shows that the oscillation of energy caused by the Euler-Cromer method is ignorable.



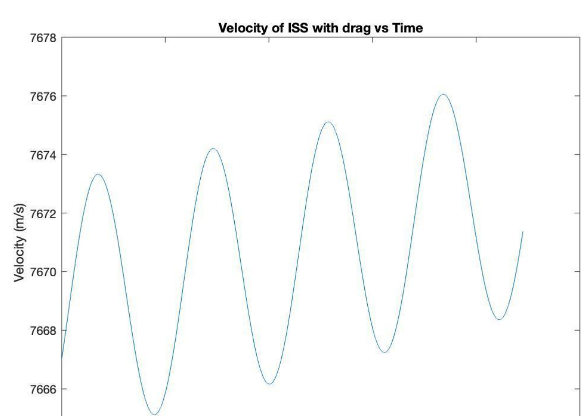
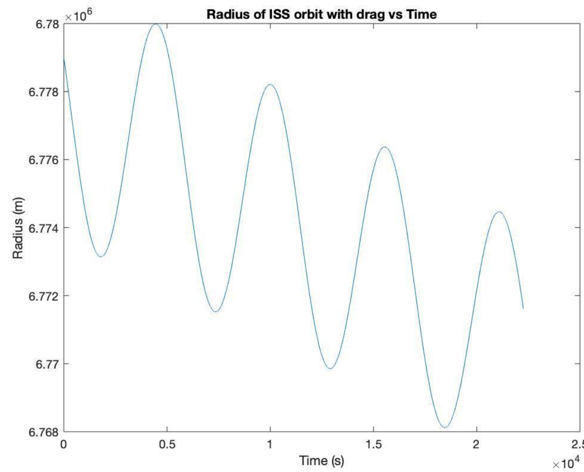


Figure 13 is the speed of ISS versus time. The local maximum speed increases because the height decreases.

Figure 14 is the radius of the orbit verses. It shows that the height of the ISS falls due to drag force.



4. Section 4: Simulating Saturn V's affected by gravitational force, drag, and propulsion.

This is the final goal of the project which simulates launching a rocket realistically. I have shown that the Euler-Cromer method works on 2-D gravitational and drag forces in sections 1 to section 3, and the new factor I add to this section is the propulsion system (rocket equation). As I mentioned above in the method, I have proven the rocket equation, given by the MIT website [6], works for the Euler-Cromer method in the code file “Rocket_equation.m”. And, I include all calculations needed for the rocket equation in SpaceCraft class object. I used real Saturn V data, from Eastern Oklahoma State University [7] and Wikipedia [8], for the calculation. I simulate stage 1 and stage two of Saturn V. The result is shown in the following graph.

Figure 15 tracks the launching projectile of Saturn V stage 1 and stage 2 on the earth scale.

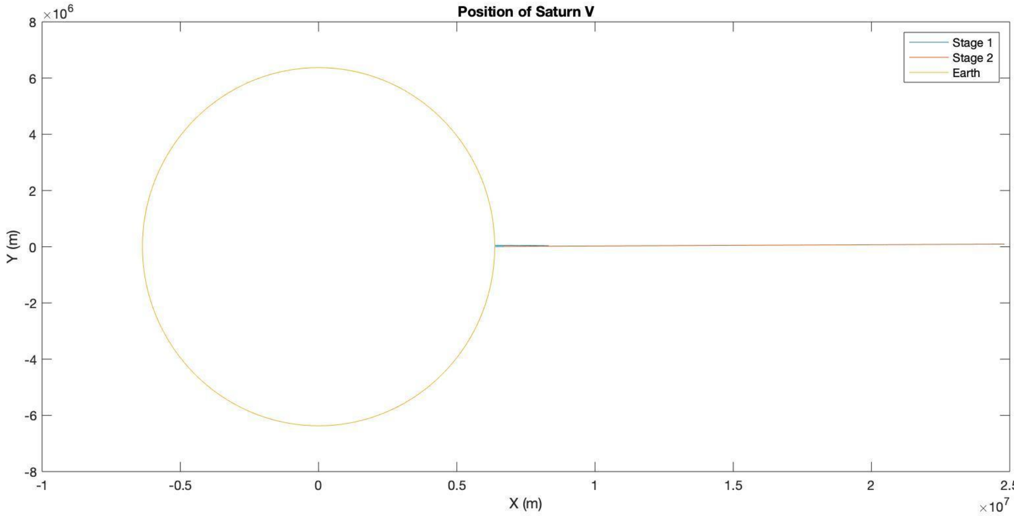


Figure 15

Figure 16 is the zoom-in view of figure 15. The point where stage 2 occurs is the point when stage 2 separates from stage 1.

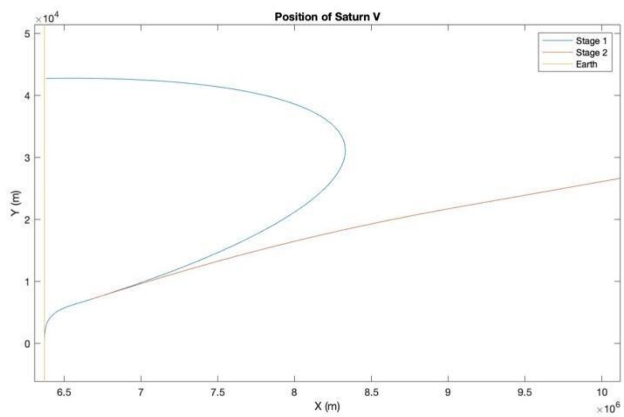


Figure 17 is the height time plot of Saturn V stage 1. It is close to a quadratic equation except for the part where time is under 100 seconds which is caused by strong drag in low altitude.

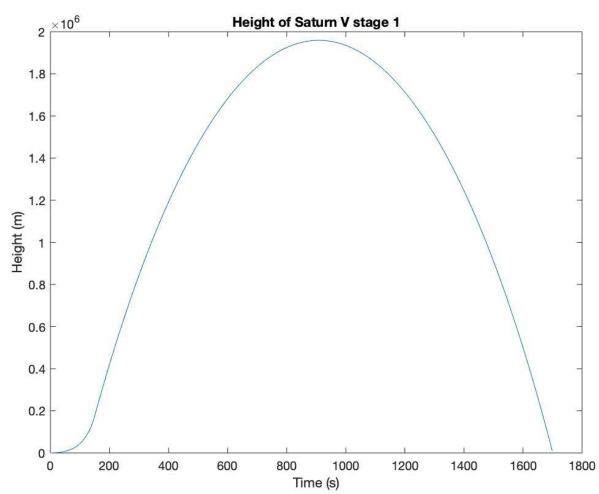


Figure 18 is the same as figure 17, but it only shows the first 200 seconds after launching the rocket where the rocket experience strong drag.

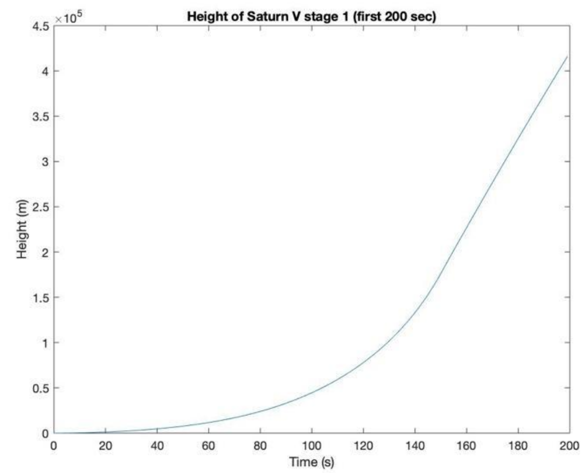


Figure 19 is the height time plot of Saturn V stage 2. The plot starts at 180 seconds, where stage 2 is separated from stage 1.

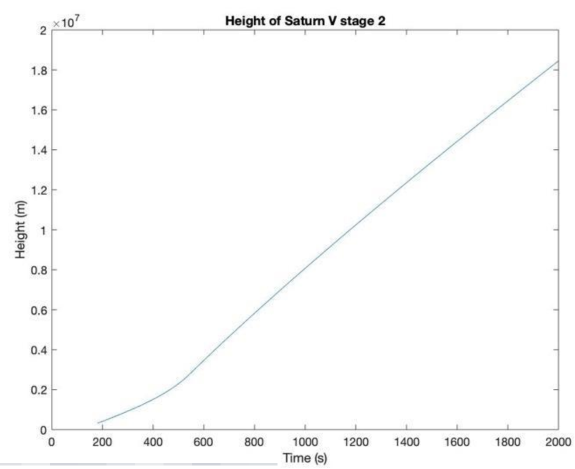


Figure 20 combines figure 17 and figure 19 to give a sense of the height scale between stage 1 and stage 2.

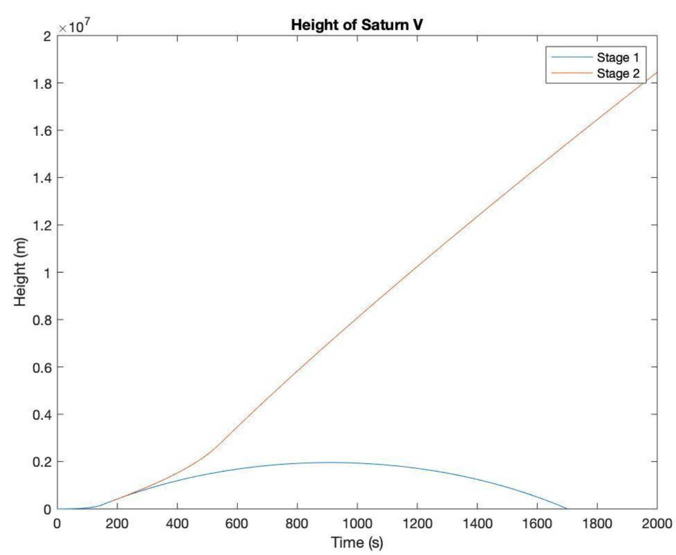


Figure 21 is the speed time plot of both stage 1 and stage 2.

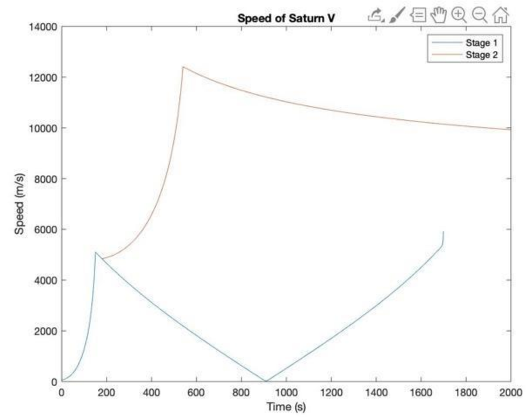


Figure 22 is the angle time plot of both stage 1 and earth. I have obtained the flight duration of stage 1 in the height time plot. At the end of the flight duration is the moment that stage 1 touches the ground. And, the angle difference between stage 1 and the earth when stage 1 touches the ground can be used to predict the ground-touching coordinates of stage 1.

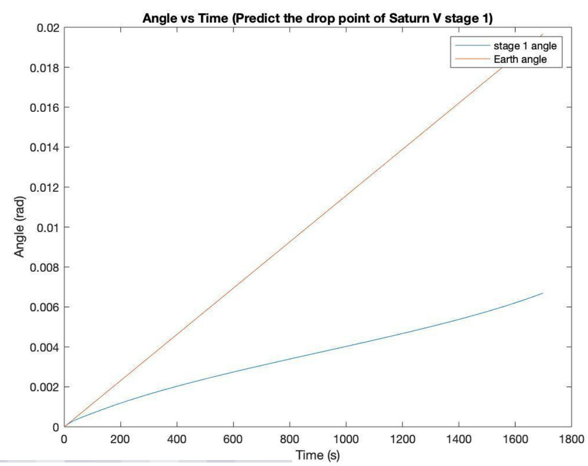
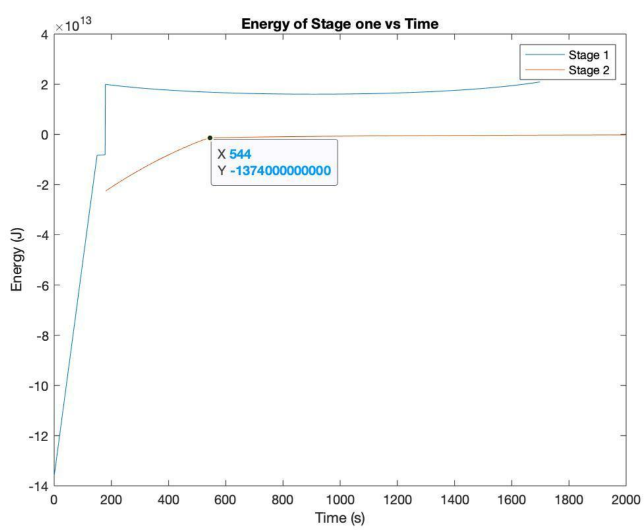


Figure 23 is the energy of stage 1 and stage 2. The reason why the energy of stage 1 suddenly jumps at 180s is that stage 1 is separated from stage 1. The graph shows that the energy of stage 2 after burning all fuel is still negative. It tells that stage



2 does not escape earth's gravity after burning all fuel. Stage 2 is still in an orbit of the earth.

Conclusion:

In this project, I apply the Euler-Cromer method on two-dimensional space to simulate gravitational force, drag force, and propulsion with a rocket equation $\left(\ddot{\vec{x}} + \ddot{\vec{y}}\right) = -\frac{GM}{x^2+y^2}\hat{R} - \frac{\rho(h)}{2\,m(t)}v^2 C_d A\hat{v} - \frac{\vec{v}_{exhaust}\,\dot{m}}{m(t)}$. I verify that the simulation is done by Euler-Cromer method is physical even though it does not conserve energy. I simulate International Space Station and Saturn V Rocket with real data to predict their orbits. Also, I define classes and functions to make the code robust. Finally, I show the calculation result in the graph. From the result, I determine that ISS is constantly falling and Saturn V stage 2 is still in earth orbit after burning all the fuel.

Critique:

From this project, I learned how to apply the Euler-Cromer method in a two-dimensional system of both Cartesian coordinates and polar coordinates. I learned how to write and use class objects in MATLAB.

Log:

Since the completion of the project was done over multiple days during spring break, I failed to track how much time this project cost me.

Reference

[1] E. Saaen, *Programming a simulation of the Earth orbiting the Sun*, Evgenii, Aug. 2016. Accessed on: Mar. 15, 2020. [Online]. Available: https://evgenii.com/blog/earth-orbit-simulation/.

[2] "U.S. Standard Atmosphere," *The Engineering ToolBox*, Accessed on: Mar. 15, 2020. [Online]. Available: https://www.engineeringtoolbox.com/standard-atmosphere-d_604.html.

[3] L. Johnson, Jacchia – Lineberry Upper Atmosphere Density Model, National Aeronautics and Space Administration, Mar. 1983. Accessed on: Mar. 15, 2020. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19830012203.pdf.

[4] "International Space Station," *Wikipedia*, Mar. 2020. Accessed on: Mar. 15, 2020. [Online]. Available: https://en.wikipedia.org/wiki/International_Space_Station.

[5] "Drag (physics)," *Wikipedia*, Mar. 2020. Accessed on: Mar. 15, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Drag_(physics).

[6] "14.2 The Rocket Equation," MIT, Accessed on: Mar. 15, 2020. [Online]. Available: http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node103.html.

[7] M. Nealon, Rocket Science, EOS, 2014. Accessed on: Mar. 15, 2020. [Online]. Available: http://www.mnealon.eosc.edu/RocketSciencePage5.htm.

[8] "Saturn V," Wikipedia, Mar. 2020. Accessed on: Mar. 15, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Saturn_V.