Max Sours

PHY407 Final Project: N-body Problems

Computational Background

Euler Methods

For a system of 1st order ODEs:

$$\frac{d\mathbf{v}}{dt} = \mathbf{F}(\mathbf{x}), \frac{d\mathbf{x}}{dt} = \mathbf{v}$$

Forward Euler uses the following formula to integrate the system.

$$\mathbf{v_{i+1}} = \mathbf{v_i} + \mathbf{F}(\mathbf{x_i})\Delta t$$

$$\mathbf{x_{i+1}} = \mathbf{x_i} + \mathbf{v_i}\Delta t$$

for some timestep Δt. Forward Euler is often unstable, but there is a slightly different version that is much more stable.

Semi-Implicit Euler integrates the system like so:

$$\mathbf{v_{i+1}} = \mathbf{v_i} + \mathbf{F}(\mathbf{x_i})\Delta t$$

$$\mathbf{x_{i+1}} = \mathbf{x_i} + \mathbf{v_{i+1}}\Delta t$$

The difference being that in the last equation, $\mathbf{v_{i+1}}$ is used instead of $\mathbf{v_i}$.

Verlet Method

Another method of solving the system

$$\frac{d\mathbf{v}}{dt} = \mathbf{F}(\mathbf{x}), \frac{d\mathbf{x}}{dt} = \mathbf{v}$$

Is the following:

$$\mathbf{x_{i+1}} = \mathbf{x_i} + \mathbf{v_i}\Delta t + \frac{1}{2}\mathbf{F}(\mathbf{x_i})\Delta t^2$$

$$\mathbf{v_{i+1}} = \mathbf{v_i} + \frac{1}{2}(\mathbf{F}(\mathbf{x_i}) + \mathbf{F}(\mathbf{x_{i+1}}))\Delta t$$

Adaptive Runge-Kutta-Fehlberg

The adaptive Runge-Kutta-Fehlberg algorithm uses a 4th order Runge-Kutta method along with a 5th order Runge-Kutta method to estimate the error. If the error exceeds the tolerance, decrease the timestep. Otherwise, increase the timestep. There is a way to calculate the optimal timestep for any time, but it leads to irrational timesteps, so you should just stick with decreasing or increasing by powers of 2. Runge-Kutta takes a system of 1st order ODEs, so you should define a vector $\mathbf{q} = (\mathbf{v}, \mathbf{x})^T$ and function $\mathbf{f}(\mathbf{q}) = (\mathbf{a}(\mathbf{x}), \mathbf{v})^T$. Additionally, you will need two separate tolerances xtol and vtol.

The formula for RKF is as follows [1]:

$$k_1 = f(t_i, q_i)$$

$$k_2 = f(t_i + \frac{1}{4}h, q_i + \frac{1}{4}k_1)$$

$$k_3 = f(t_i + \frac{3}{8}h, q_i + \frac{3}{32}k_1 + \frac{9}{32}k_2)$$
$$k_4 = f(t_i + \frac{12}{13}h, q_i + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3)$$
$$k_5 = f(t_i + h, q_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4)$$
$$k_6 = f(t_i + \frac{1}{2}h, q_i - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5)$$

$$q_{i+1} = q_i + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5$$
$$\tilde{q}_{i+1} = q_i + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6$$

Note that k_2 is not used when calculating q_{i+1}, $\tilde{q}_{i+1}$.
Error can be calculated with

$$E = \|\tilde{q}_{i+1} - q_{i+1}\|_2$$

When using two different tolerances, you'll want

$$E_v = \|\tilde{v}_{i+1} - v_{i+1}\|_2 < \text{vtol}$$
$$E_x = \|\tilde{x}_{i+1} - x_{i+1}\|_2 < \text{xtol}$$

And when you're calculating more than one body, you'll want to make sure no error term breaches the tolerance.

Physics Background

n-Body Gravitational Potential

The force exerted on the ith point mass in an n-body system is given by

$$F_i = \sum_{j=1, j\neq i}^{n} \frac{Gm_i m_j}{|x_j - x_i|^3}(x_j - x_i)$$

And the potential energy of the system is given by

$$U = -\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{Gm_i m_j}{|x_j - x_i|}$$

The force equation will be used in the ODE function, while the potential energy equation can be used along with kinetic energy as a way to check to see if the code is correct, as well as a way to measure the effectiveness of an algorithm by how well it obeys the conservation of energy.

Lagrange Points

In the 3 body problem where the 3rd object has negligible mass, and in the limiting case where $M_1 >> M_2$, there exist 5 points of equilibrium called Lagrange Points. These points exist in the second object's frame of reference, which is non-inertial.

These points are given by [2]

$$\text{L1: } (R(1-\sqrt[3]{\frac{M_2}{3M_1}}),0)$$

$$\text{L2: } (R(1+\sqrt[3]{\frac{M_2}{3M_1}}),0)$$

$$\text{L3: } (-R(1+\frac{5}{12}\frac{M_2}{M_1}),0)$$

$$\text{L4: } (\frac{R}{2}(\frac{M_1-M_2}{M_1+M_2}),\frac{\sqrt{3}}{2}R)$$

$$\text{L5: } (\frac{R}{2}(\frac{M_1-M_2}{M_1+M_2}),-\frac{\sqrt{3}}{2}R)$$

Initial Conditions

Setting up initial conditions for the simulation can be tricky. With a planet orbiting a star, it is often best to assume a circular orbit around that star.

$$\frac{GMm}{r^2}=\frac{mv^2}{r}\rightarrow v=\sqrt{\frac{GM}{r}}$$

Set the star stationary at the origin and the planet at $(r,0)$. Then the planet's velocity is $(0,v)$.

However, for the Lagrange Points, it is better to try and match the angular velocity of the Earth.

$$\omega=\frac{v}{r}\rightarrow\frac{v_E}{r_E}=\frac{v_S}{r_S}\rightarrow v_S=v_E\frac{r_S}{r_E}$$

For L4 and L5, you'll have to rotate the velocity vector to stay perpendicular to the position vector.

With the current setup, the center of mass is off the origin, and the system has a nonzero net momentum. It is useful to have the center of mass be at the origin and for the net momentum to be 0. This can be accomplished by subtracting the center of mass's position from all other positions, and subtracting the center of mass's velocity from all other velocities. That is,

$$v_i'=v_i-v_{CoM},x_i'=x_i-x_{CoM}$$

where

$$x_{CoM}=\frac{\sum_i x_im_i}{\sum_i m_i},v_{CoM}=\frac{\sum_i v_im_i}{\sum_i m_i}$$

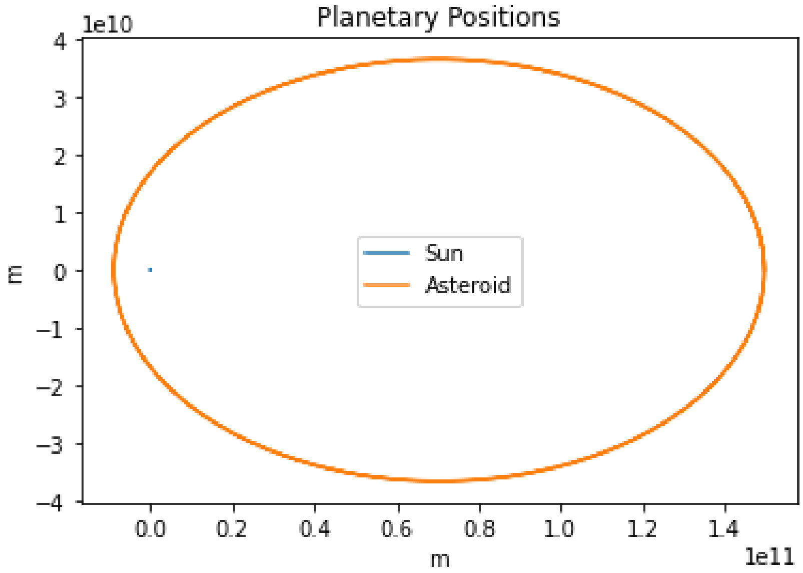
Questions

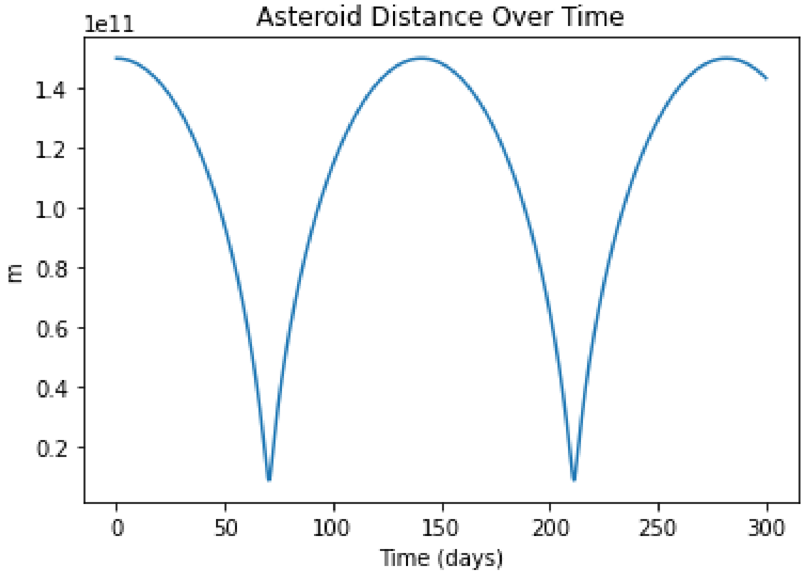
1. **Errors in n-Body Integration**

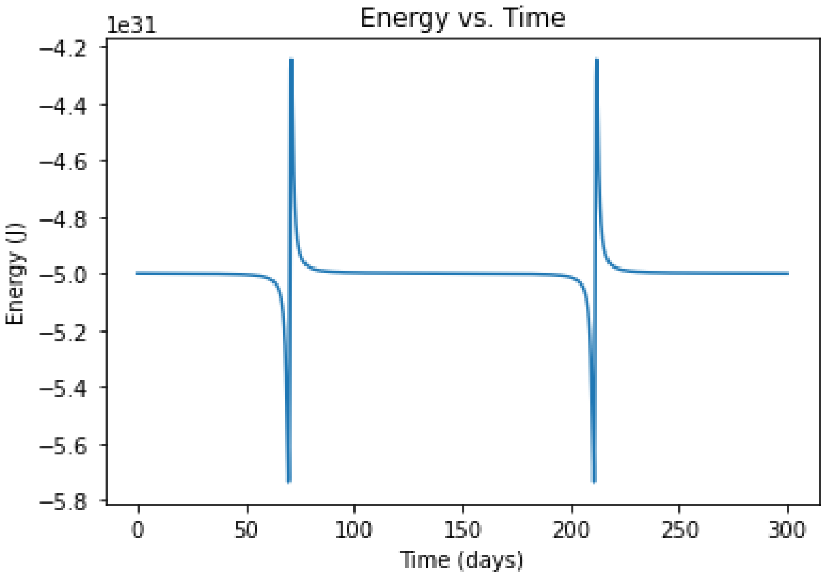
   (a) We will test the accuracy of several integration methods using a two-body system of the Sun and a large asteroid. Set the initial position of the asteroid to be 1 AU from the Sun and the initial velocity at 10 km/s. Remember that the center of mass and net momentum should both be 0.

   Nothing to submit.

(b) In Lab 1 you learned that the Semi-Implicit Euler (also called Euler-Cromer) method gave satisfactory results for the Sun-Mercury system (with the Sun being a fixed point). We will use this method to simulate the Sun-Asteroid system. Use the initial conditions from above and integrate the system over 300 days with a timestep of 1 hour (3600 s). Output plots of the positions of the Sun and the asteroid, distance of the asteroid over time, and the total energy of the system over time. What does the energy plot tell you about the error in the system? Is Semi-Implicit Euler a good integration method?

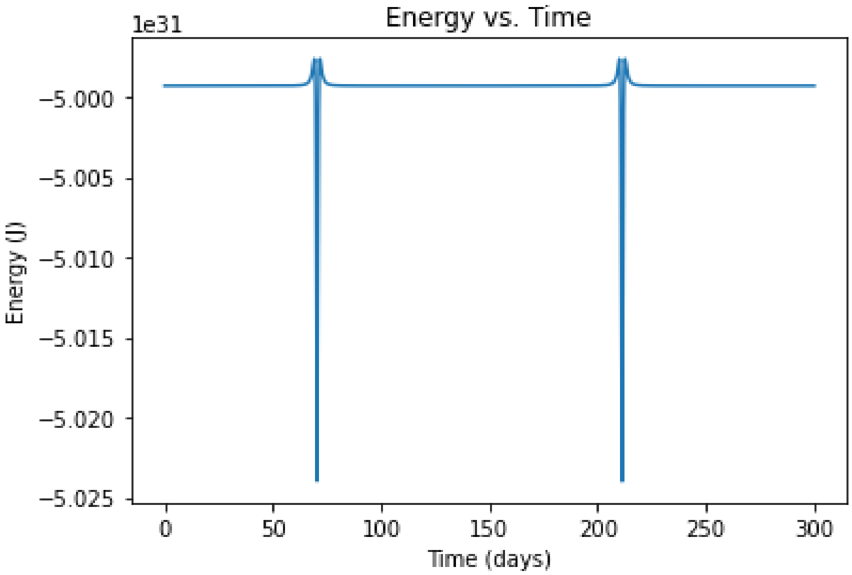






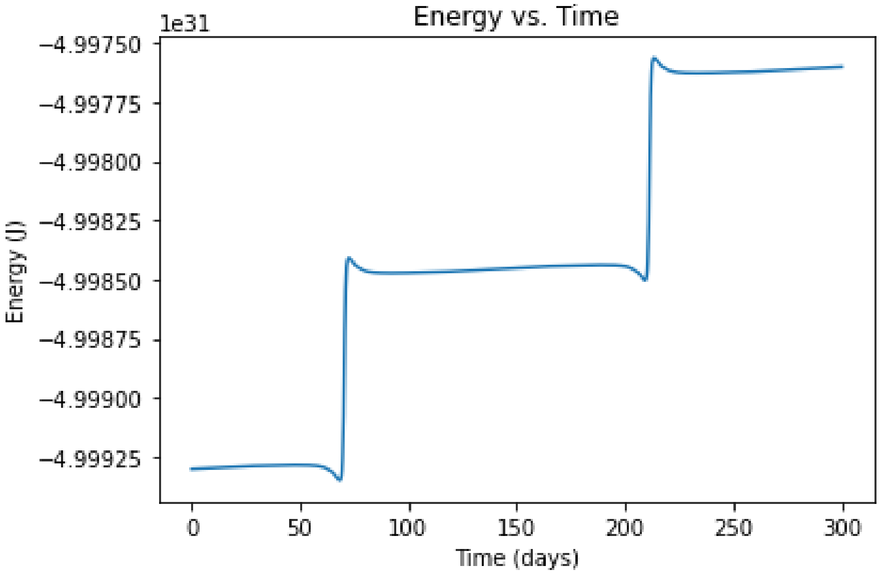
At first glance, the paths of the Sun and the asteroid make sense: the Sun barely moves and the asteroid makes a highly elliptical orbit around the sun. However, as you can see from the energy plot, energy is not conserved at specific points when the asteroid is close the sun. The relative error bound on the energy is $\frac{5.8-4.2}{4.2} = 38\%$, which is very high. While Symplectic Euler does seem to cancel out its errors, that is, on one side of the ellipse the errors have on sign, and on the other side they have the opposite sign, the error in energy is still alarmingly high. While the method appears to be stable since when you complete a revolution it has roughly the same energy as when it started, it is highly inaccurate, especially close to the sun. This fact makes Semi-Implicit Euler a poor integration method.

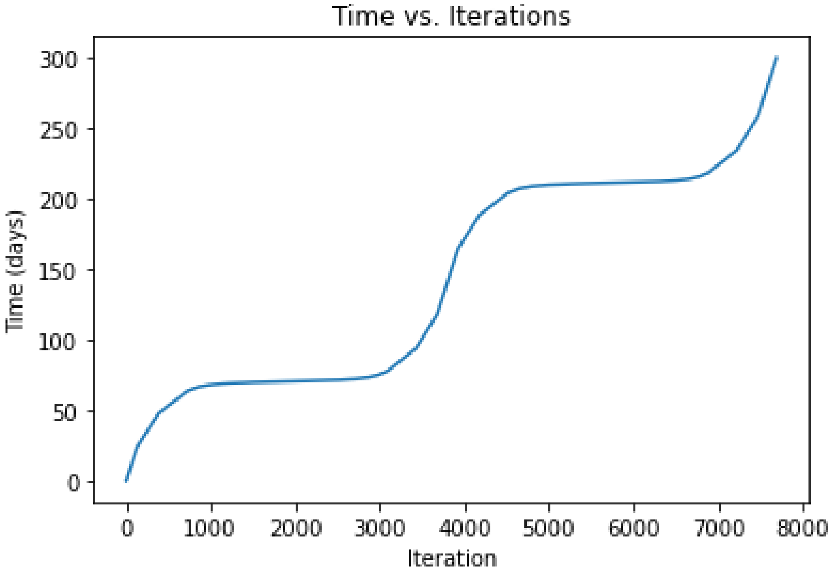
(c) Repeat part (b) for the Verlet method. Report energy over time. You should generate the other 2 plots to verify the algorithm is generating good results, but you don't need to report them. Is energy conserved at all times? When is energy not conserved?



Verlet shows much improvement over Symplectic Euler, but it still does not conserve energy well when the asteroid is near the Sun. The relative error bound with the Verlet method is $\frac{5.025-4.995}{5} = 0.6\%$, which is a lot better than the previous method. Verlet is a good choice for a molecular dynamics simulation such as this one, and the energy mostly conserved over the simulation. However, Verlet still performs poorly near the Sun compared to everywhere else. When there is one area that is significantly harder to integrate than everywhere else, it is often a good idea to use an adaptive method, which is the next step.

(d) Implement the adaptive Runge-Kutta-Fehlberg method. Run the same simulation with a starting timestep of 2048 s (or any power of 2), an xtol of 100 km, and a vtol of 0.1 m/s. Output the energy plot and comment on the results. Additionally, plot time as a function of iterations. When is the algorithm taking smaller timesteps?

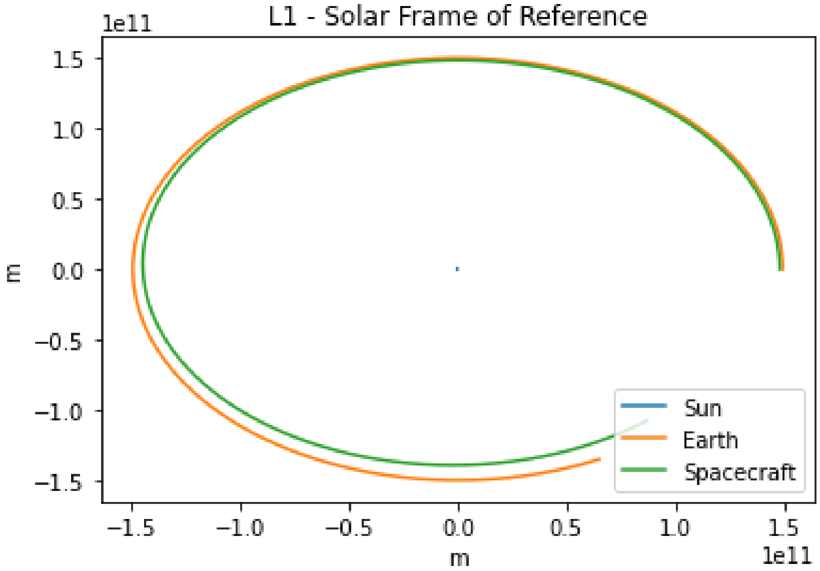


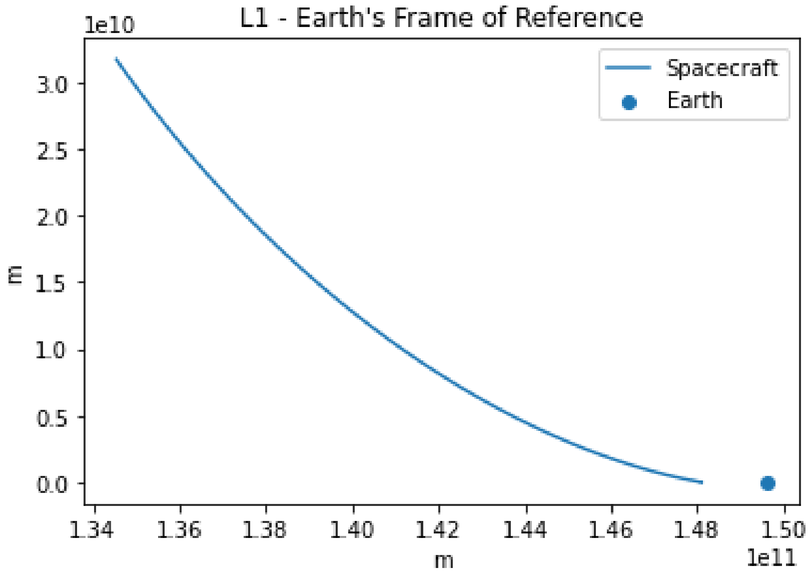


First, one can see from the scale on the energy plot that we are conserving energy much better than either of the previous two methods, especially when the asteroid is close to the Sun. The relative error bound here is $\frac{4.99935-4.99750}{4.99750} = 0.035\%$, which is 20 times better than the Verlet method. From the Time vs. Iterations plot, we can see that, indeed, the algorithm is taken many more timesteps at the times when the asteroid is closest to the Sun, which serves to reduce the error found at that point. There is one issue, however. The energy appears to be increasing slightly for each iteration, which means this algorithm is unstable. That being said, it is also far more accurate than the previous models while taking roughly the same amount of time to complete, so this is the method I will use, especially since my simulations will not go on for many revolutions. It is also possible that the previous algorithms are actually unstable, but the errors in energy at points of closeness are large enough that the instability can't be seen. Additionally, thanks to the position and velocity tolerances, it is possible to put an upper bound on the error of the energy.

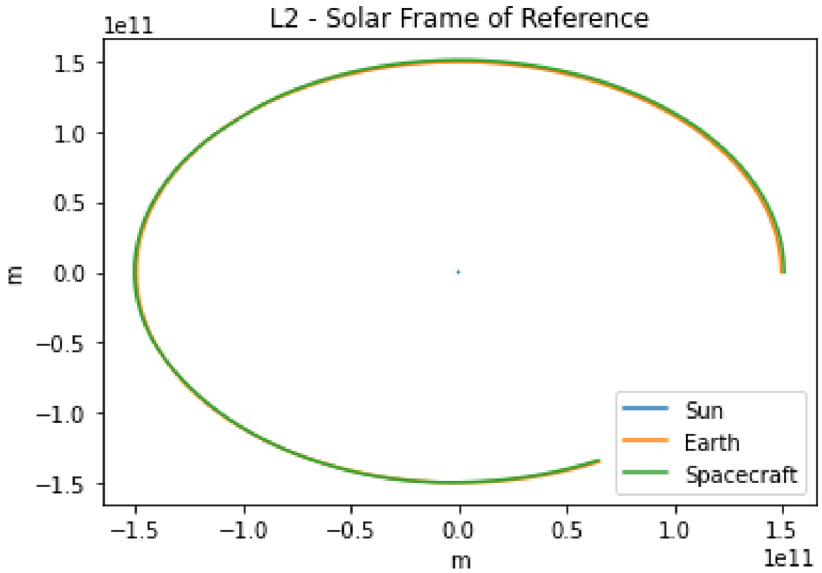
2. **Lagrange Points**

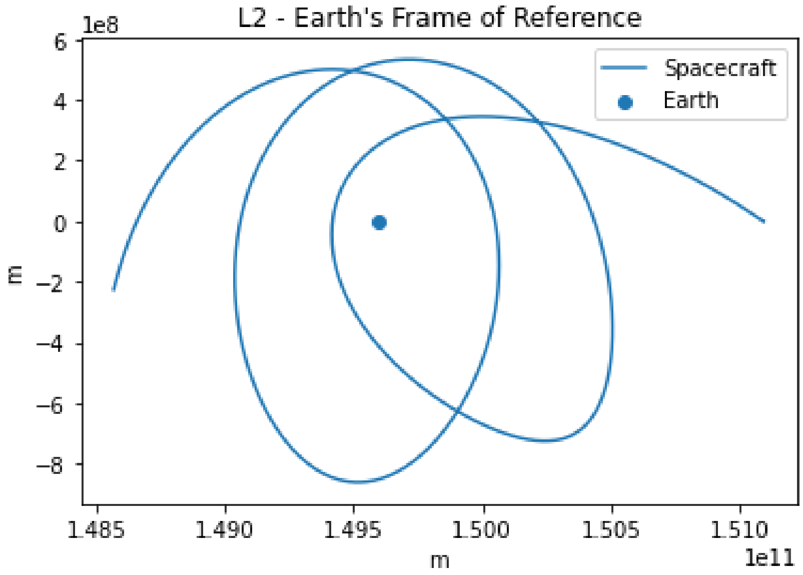
   (a) Simulate the Sun-Earth system with a small (1 kg) mass at L1 for 300 days, h0 = 2048 s, xtol = 100 km, vtol = 0.1 m/s. The initial velocity should be perpendicular to the position vector, and the angular velocity of the small mass should be the same as Earth's. Output position plots in the Solar Reference Frame and Earth's Reference Frame (you can leave out the Sun and make Earth just the point (1 AU, 0) for the second graph). Repeat this process for L2, L3 and comment on your findings. Are these points stable?



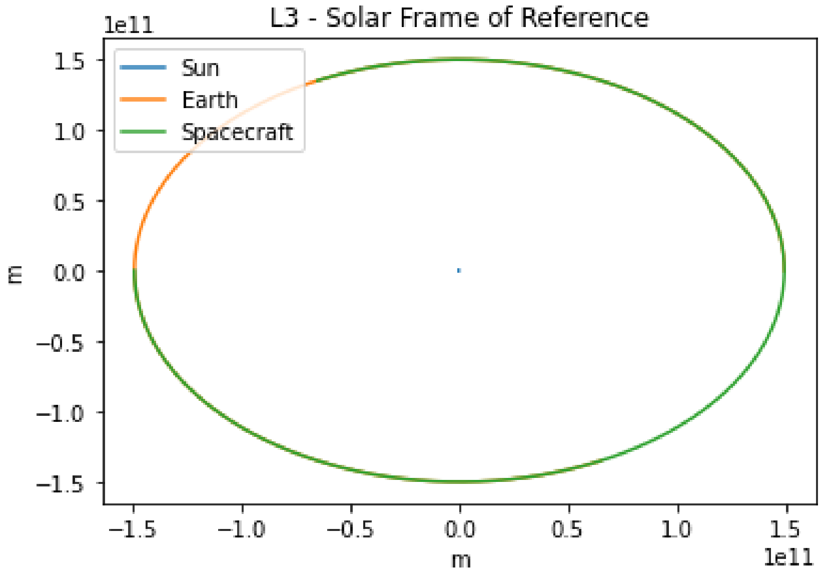


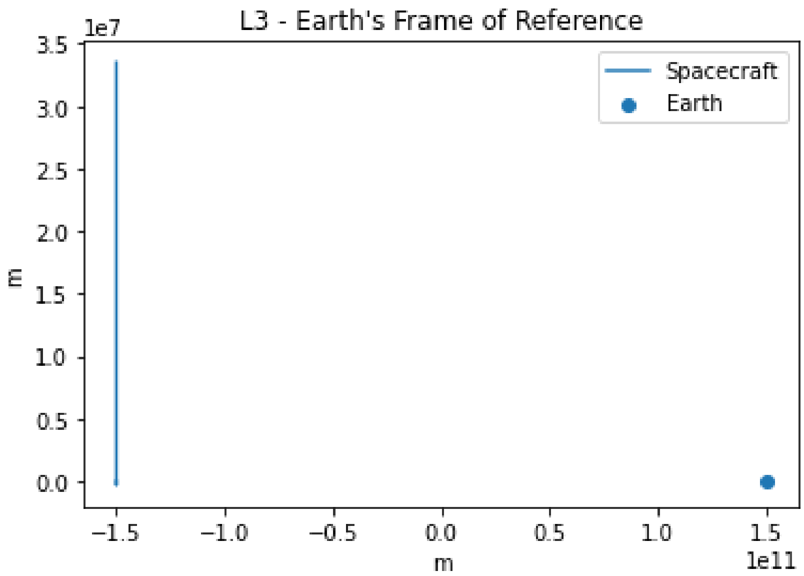
The spacecraft distinctly moves off the L1 point, as can be seen from both frames of reference, showing that this point is definitely not stable.





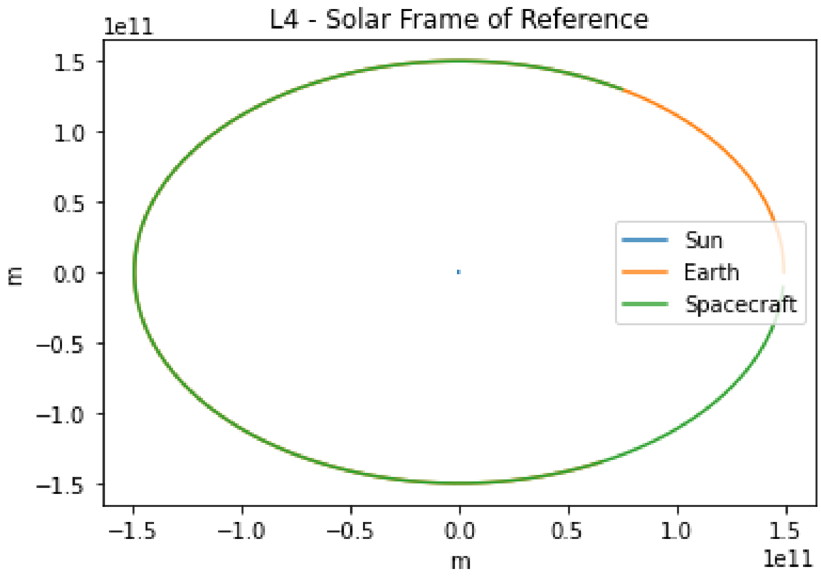
The spacecraft also moves away from the L2 point quickly, falling into Earth's gravity well, orbiting a few times, and it looks like it is making its way out of Earth's gravitational influence. Interestingly, the spacecraft has fallen sunwards both times, so it is possible that I need to increase the spacecraft's angular velocity. However, I suspect that it would merely cause the spacecraft to fall off L2 in a different direction.

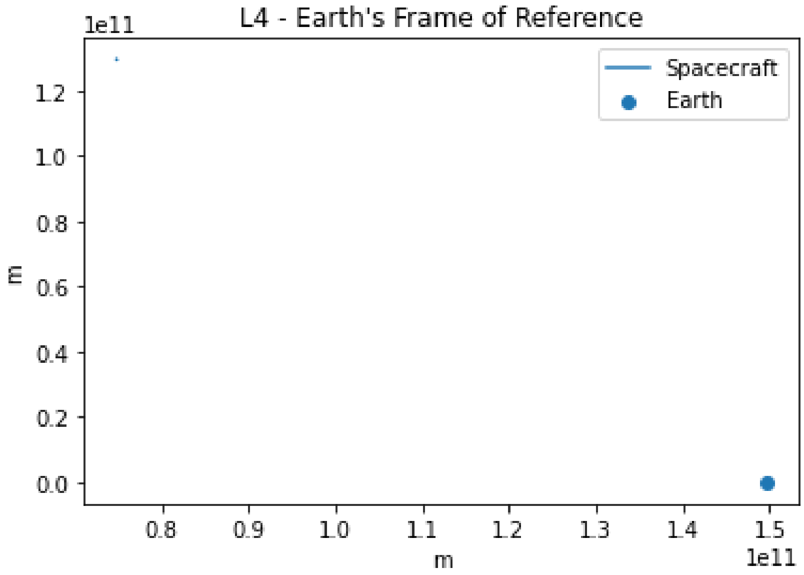




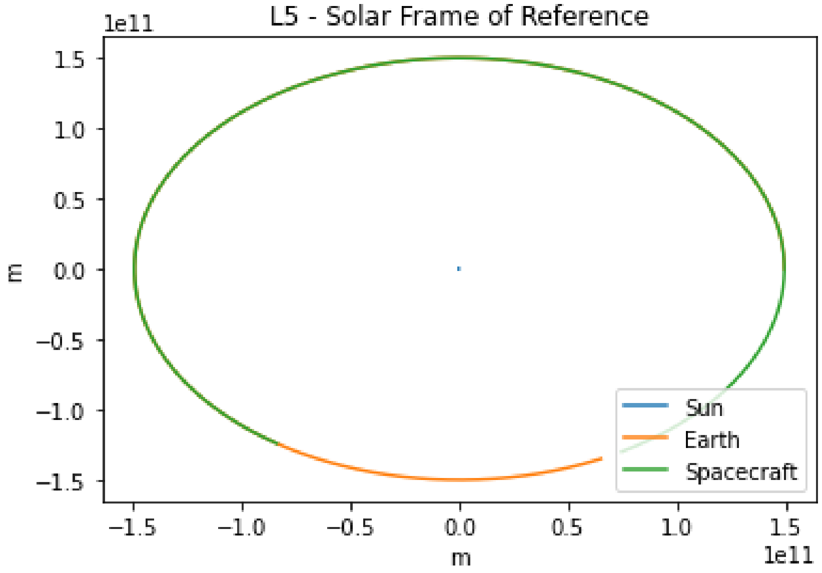
Interestingly, the spacecraft doesn't move that much compared to the movements off L1 and L2. 10^7 is just an order of magnitude above the position tolerance. But literature shows that L3, too, is unstable, but this graph implies that it is less unstable than L1 and L2.

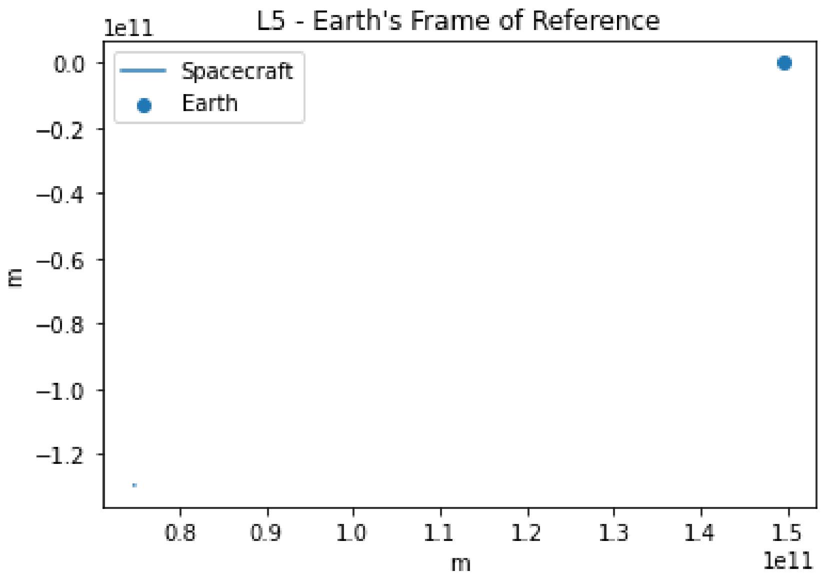
(b) Do the same thing for L4, L5 and comment on your results. Are these points stable?





Note: the spacecraft is in the top left.





Note: the spacecraft is in the bottom left

It's pretty clear from these plots that L4 and L5 are stable points. Interestingly, literature says that these two points are actually unstable potential maxima, but the Coriolis force (since this is a rotating reference frame) serves to keep the spacecraft stable. Natural objects tend to cluster around the L4 and L5 points of massive planets, most famously are Jupiter's Trojan asteroids, located at its stable Lagrange points.

References

[1] C. Christara. CSC 436. Class Lecture, Topic: ”Numerical methods for solving IVPs for ODEs.” Faculty of Arts and Science, University of Toronto, Toronto, ON, Canada, Fall, 2020

[2] N. Cornish, ”The Lagrange Points,” *map.gsfc.nasa.gov*, 1998. [Online]. Available: https://map.gsfc.nasa.gov/ContentMedia/lagrange.pdf. [Accessed Dec 17, 2020].