A description of the Solar System simulation and numerical methods used

Anton Belov

18.12.2020

Abstract

This report describes a physics simulation written in Python in which Solar system evolves with time. It discusses two numerical methods used to approximate calculations: Euler and Euler-Cromer. Both methods reproduce stable orbits with a small enough change in time, however as it gets larger it becomes clear that Euler-Cromer is more accurate. For example, after 10 years the difference between final and initial momentum using Euler-Cromer is 0.161%, using Euler - considerably greater (0.219%). The superiority of Euler-Cromer method could also be seen when looking at an x-y position graph of all the bodies of the system, where it produces much stabler orbits compared to Euler. Euler-Cromer has been proven to be more versatile as the errors do not scale as Δt increases, meaning it is more efficient in large time period simulations. With slight adjustments to the code, other systems, such as a system of electrically charged particles could be simulated, meaning there could be applications of this code beyond the scope of this project.

1 Introduction

Simulating a system where each body interacts with every other body requires finding acceleration, velocity and position of each particle at different time intervals after some initial values have been provided. In this section there is a description of how it is done in my code and what precautions need to be taken when calculating individual values.

To calculate gravitational force between two bodies Newton's law of universal gravitation has been used, which states that force is directly proportional to the product of the masses and inversely proportional to the distance between them squared ($F = Gm_1m_2/r^2$). Having more than two bodies in the system though means the superposition principle has to be used. It states that the total force on a body is a vector sum of individual forces. Implementing this into Newton's formula by using the summation sign to include all the bodies and making sure acceleration is not updated with respect to the body it is calculated of (to obtain body's acceleration, force is divided by the mass of that body), the final equation is:

$$\overrightarrow{a_i} = \sum_{j \neq i}^{N} \frac{-Gm_j}{|r_{ij}|^2} \hat{r}_{ij} \tag{1}$$

Here r_{ij} is vector distance between particles of index i and j, G is gravitational constant and $\overrightarrow{a_i}$ and m_j are vector acceleration and mass of particles of index i and j respectively. N is the number of time steps and can be written as total time span over the duration of one time step ($N = T/\Delta t$).

Forces of one particle on others and vice versa had to all have been calculated at the same time, without any motion of the bodies prior, as it would otherwise violate Newton's third law. The reason for that could be seen in Equation 1: acceleration of a body is inversely proportional to the magnitude of distance between the two body, meaning if one's acceleration is calculated after some time period of another, the force exerted on the first body by the second would be different to force exerted by the first on the second.

In order to calculate values of position and velocity of every body in the system with non-uniform acceleration, numerical methods have to be used. It can be assumed that for a small interval of time Δt, the change in acceleration is small, hence Taylor expansion of standard equations of motion can be used:

$$\overrightarrow{x}_{n+1} = \overrightarrow{x_n} + \overrightarrow{v_n}\Delta t + O((\Delta t)^2) \tag{2}$$

$$\overrightarrow{v}_{n+1} = \overrightarrow{v_n} + \overrightarrow{a_n}\Delta t + O((\Delta t)^2) \tag{3}$$

Here $O((\Delta t)^2)$ signifies contribution of $(\Delta t)^2$ and higher orders. If Δt is small enough, those could be omitted, which yields approximate equations for x_{n+1} and v_{n+1} (position and velocity after a time step).

Applying Equation 2 and then Equation 3 to some initial conditions leads to Euler's method. However, due to not being stable for oscillatory systems (proof in Section 2) another numerical method could also be used: Euler-Cromer. It instead calculates the position at the end of the step rather than the beginning (i.e. Equation 3, then Equation 2).

The error in each time step is given by the truncation of the expansion, hence will be of order $(\Delta t)^2$. By definition $N = T/(\Delta t)$ (from Equation 1) and therefore the total error will be proportional to $N(\Delta t)^2 = T(\Delta t)^2/(\Delta t)$, hence $\propto \Delta t$. As already mentioned, if the time step is small, ignoring contributions of higher orders will not make a significant difference. However, if simulation is run with multiple bodies over a longer time period, Δt has to be increased, as otherwise it would take significantly longer to compute values of all particles. If that is done, the error will be large enough to impact orbits of planets greatly even over a relatively short period of time. (See Section 2)

2 Results

At the start of the project simpler models of the simulation were constructed in order to test and debug functions and methods before using them in the final code.

First attempt was to simulate Earth's gravitational field acting on a particle close to the surface of Earth. This way the acceleration would be constant allowing me to check if functions with numerical methods for updating position and velocity are working correctly. Because acceleration is assumed to be uniform, equations of motions can be solved analytically using basic equations of motion - "suvat" equations (Equations 4 and 5). Results obtained using these formulas are closest to how the particle would actually behave without the effects of any external forces. This means approximated values obtained using numerical methods could be checked against the "true" values to see which ones are more accurate at the same Δt.

$$x = ut + \frac{1}{2}at^2 \tag{4}$$

$$v = u + at \tag{5}$$

A test particle "Ball" of mass 1 kg has been given initial velocity of 1 ms^{-1} and 50 ms^{-1} in x and y directions respectively, starting from origin. It was subject to acceleration of -10 ms^{-2} in y-direction, which is approximately that of Earth's near the surface. A graph of x-y position was then plotted to compare the trajectory of the particle between introduced methods:

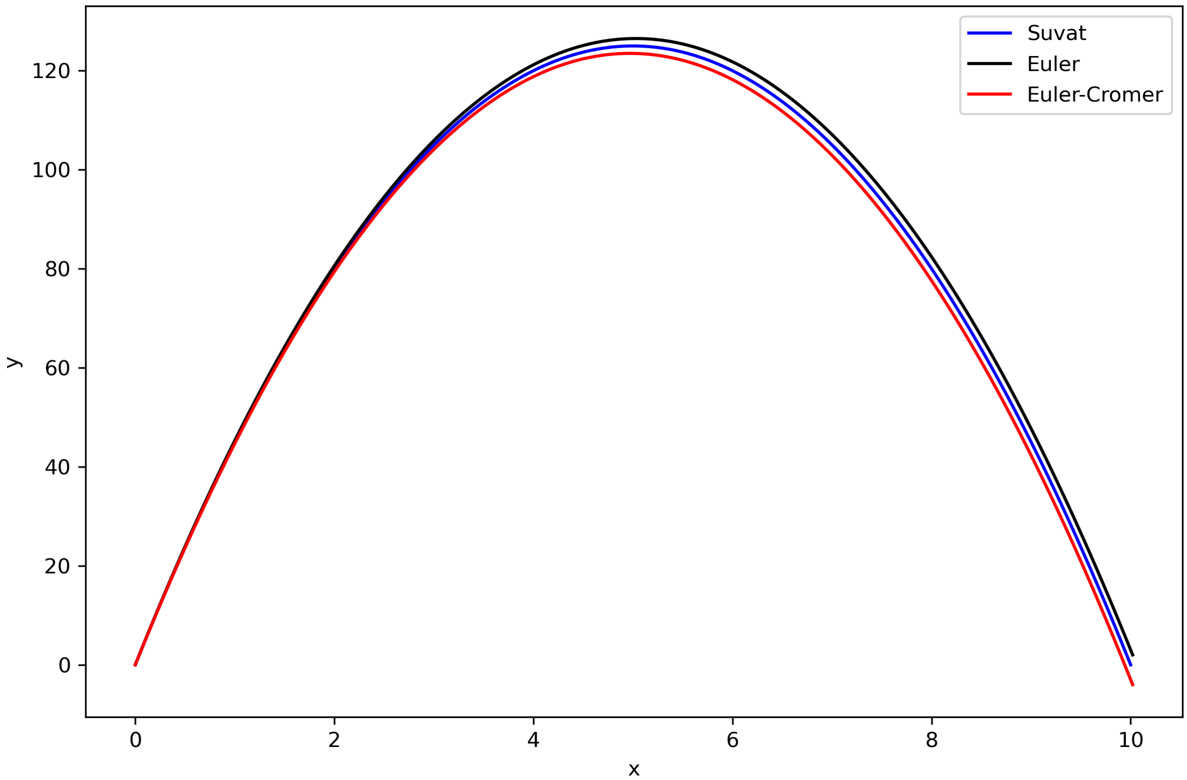

Figure 1: Graph of x-y positions of particle "Ball" using different methods of calculating and updating its position and velocity

Figure 1 shows trajectories of "Ball" where $\Delta t = 0.06\ s$. The total time period is $10s$, meaning there were 166 iterations in total. It was noted that increasing Δt past roughly $0.1\ s$ drastically decreases the accuracy of calculations and does not produce a smooth curve on a graph due to lack of data points, as they are updated and plotted every Δt seconds.

Because Euler method updates particle's position before its velocity, the particle as expected "overshoots" the actual path as its velocity magnitude will be slightly greater when moving in the positive y-direction and slightly smaller when moving in the negative y-direction due to truncation of the t^2 factor. Opposite effect is observed for Euler-Cromer method - particle's velocity magnitude is smaller on the "rise" and greater on the "fall" compared to actual values, because it is updated before particle's position is.

Next progression was to simulate a system in which there are two bodies, one of which is a lot more massive than the other (such as Earth and the Sun). An assumption that Sun's acceleration due to Earth is negligible could then be made to simplify the system, allowing to test the function, which updates acceleration of some body. Note, that Sun is therefore stationary throughout the duration of the simulation; only Earth's values are updated. A graph of x-y position was then plotted with values obtained using Euler and Euler-Cromer methods:

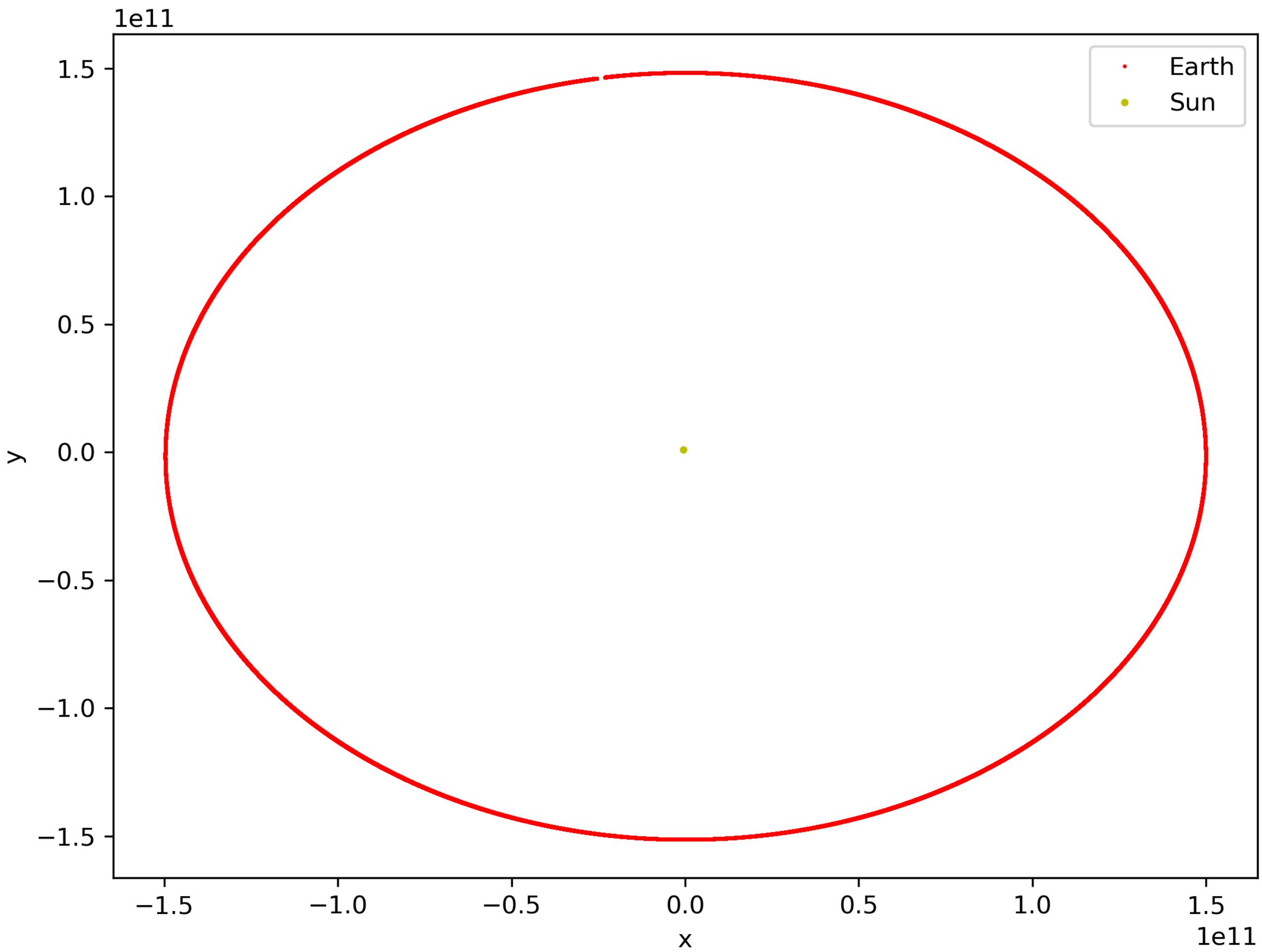

Figure 2: Graph of x-y positions of Earth and Sun in a two body system with $\Delta t = 180\ s$

Figure 2 is a plot of values of both methods, however from this perspective differences cannot be seen. If the point of initial position coordinates on the graph is enlarged, it is possible to see that when using Euler-Cromer Earth's last position coordinates are slightly closer to initial ones than those calculated using Euler's method, meaning those are slightly more accurate. In general the graph shows that both methods are capable of producing stable orbits with relatively short time step. The initial and final angular momentum difference for Euler is equal to 0.0178% to 4 d.p.t, whereas for Euler-Cromer it is 0.0037%, which is almost five times more accurate. It was noted that with specified Δt both methods are precise enough for Earth not to do a full orbit around the Sun (note a little separation between the orbit trajectory at the top center of the graph). This is as expected, since the total time period was set to 365 days, whereas a whole year contains roughly 365.25 days.

The final step was to add other Solar system planets and the Moon's initial conditions and make them an instance of the main class 'Particle'. Code was adjusted to update every body's data (velocity, position and acceleration including Sun) after each time step and save it to the file called "MultipleBodies". Saving data of every iteration would allow the user to access it without having to re-run the code as it may take a long time to complete if Δt is relatively small or total time period is relatively large. Different files were then written to perform various checks on the final simulation such as "XYPositionGraph.py", "AngularMometumDifference.py" and "PrintData.py" (a description of what each one does precisely can be found in the "README" file).

The code was made more user-friendly by allowing user to input the total number of days, set a method that would be used and add any other bodies to the system (up to 3 bodies) trough the terminal. The way to improve would be to either add similar features for Δt or come up with an equation for it so that the time it takes for the simulation to complete is always the same (e.g. 30 seconds). This would ensure the simulation does not run for too long if total time period and/or number of bodies are relatively large. Some features, such as restricting the total time period to 100 years were added to aid achieving low enough simulation run time.

Initial conditions of the system were set to values given in Ephemeris on Jet Propulsion Laboratory website with starting date of 1st of January 2020. To start with, data obtained after running the simulation for a year using both numerical methods were tested against the one given in Ephemeris. Note that 2020 is a leap year, hence the total time period was set to 366 days when comparing values. A table (Table 1) displaying components of position and velocity of Earth relative to the Solar System Barycenter after a full year at $\Delta t = 720$ seconds was constructed, where all values are given in km and km^{-1} respectively to 3 d.p.. Note that in the code all values are given in SI units (i.e. m and ms^{-1}).

	Ephemeris	**Euler**	**Euler-Cromer**
x-position	$-2.521 \cdot 10^7$	$-2.452 \cdot 10^7$	$-2.575 \cdot 10^7$
y-position	$1.460 \cdot 10^8$	$1.464 \cdot 10^8$	$1.459 \cdot 10^8$
z-position	$8.816 \cdot 10^3$	$9.481 \cdot 10^3$	$8.828 \cdot 10^3$
x-velocity	$-2.987 \cdot 10^1$	$-2.987 \cdot 10^1$	$-2.985 \cdot 10^1$
y-velocity	$-5.023 \cdot 10^0$	$-4.882 \cdot 10^0$	$-5.130 \cdot 10^0$
z-velocity	$-2.631 \cdot 10^{-4}$	$1.512 \cdot 10^{-4}$	$-2.599 \cdot 10^{-4}$

Table 1: Comparison of position coordinates and velocity of Earth between Ephemeris and numerical methods after 366 days

Although, both methods output values in the same order of magnitude, from Table 1 it is clear that Euler-Cromer method generally produces more accurate results than Euler's. To further examine this statement, total angular momentum of the system was calculated at the beginning and the end of the time period. The percentage difference between final and initial angular momentum is displayed in Table 2.

Time period (years) / Δt (s)	**Euler**	**Euler-Cromer**
1 / 180	0.0201%	0.0202%
1 / 7200	0.0145%	0.0202%
10 / 7200	0.2186%	0.1613%

Table 2: Table displaying the percentage difference between final and initial angular momentum for various time periods and Δt

From Table 2 it could be seen that angular momentum difference of both methods is similar at small time steps. However, as Δt and total time period increase, a simulation using Euler-Cromer method conserves angular momentum better, meaning final values of position and velocity of individual planets are likely closer to the true values (given in Ephemeris). It was also noted that changing Δt at the same time period did not affect the accuracy of values produced using Euler-Cromer method (as momentum difference is almost identical after a year with $\Delta t = 180$ and 7200 seconds) as much as it did using Euler's.

A graph of x-y positions was then plotted for every planet for further testing. The plot only shows the six closest planets to the Sun, as it was impossible to fit outer planets of the Solar System on the same graph, as they are relatively far away from inner ones and their displacement is negligible after one year, since their orbit time period is very large compared to inner planets:

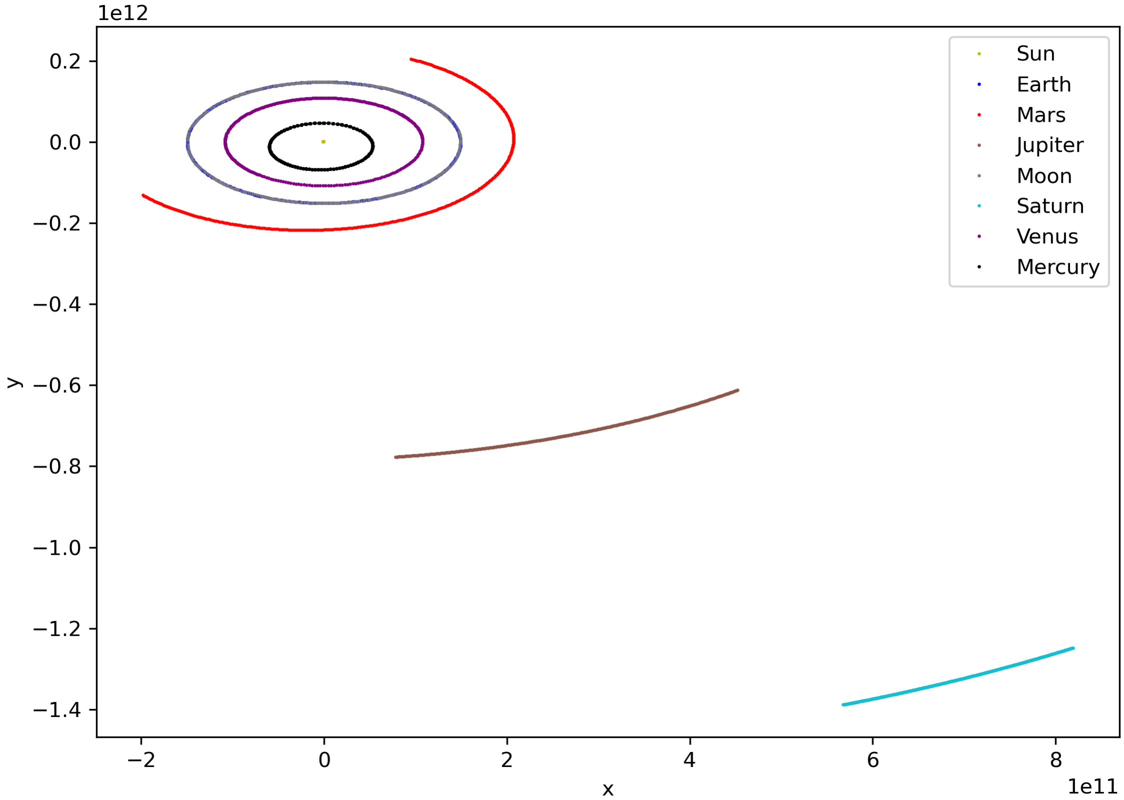

Figure 3: Graph of x-y positions of planets of Solar System over one year with $\Delta t = 180\ s$

Both numerical methods visually produce a very similar graph with specified Δt, so Figure 3 represents values obtained using Euler and Euler-Cromer. This plot demonstrates that both methods are able to reproduce stable orbits at a relatively small Δt. If Earth's trajectory is enlarged, it could be seen that Moon orbits Earth as expected.

However, as Δt increases values become less accurate and the shape of the graph changes. To observe the effects better the total time period was increased to 10 years and Δt to 7200 seconds. Graph 4 uses Euler's method to calculate the values and only shows Mercury, Venus, Earth, Moon and Mars.

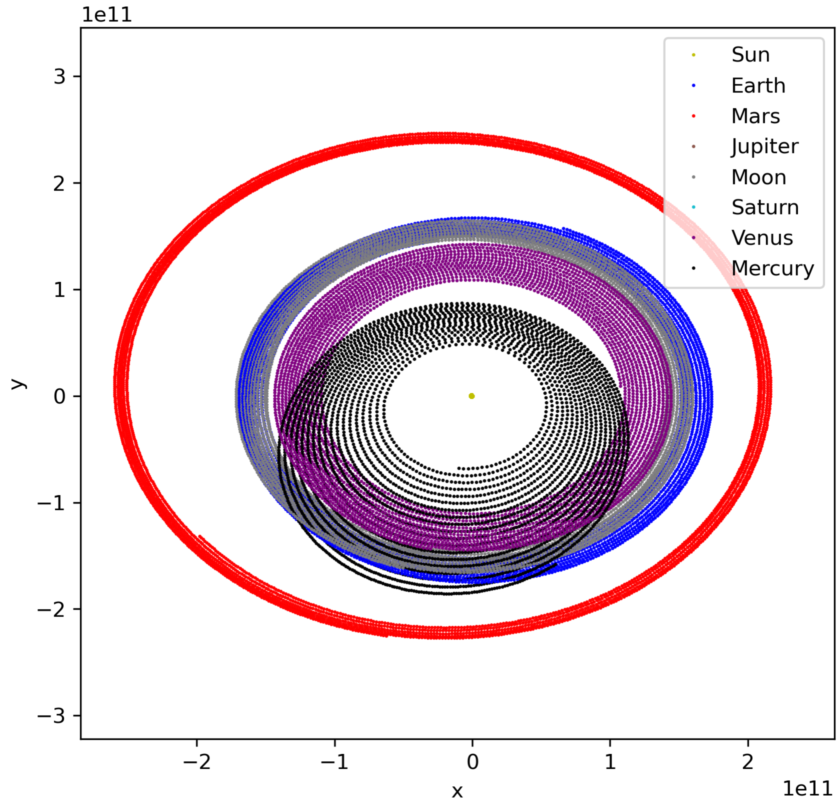

Figure 4: Graph of x-y positions of planets of Solar System over ten years with $\Delta t = 7200\ s$ using Euler's method

Using the same initial conditions and Δt another graph was plotted with values obtained through Euler-Cromer method:

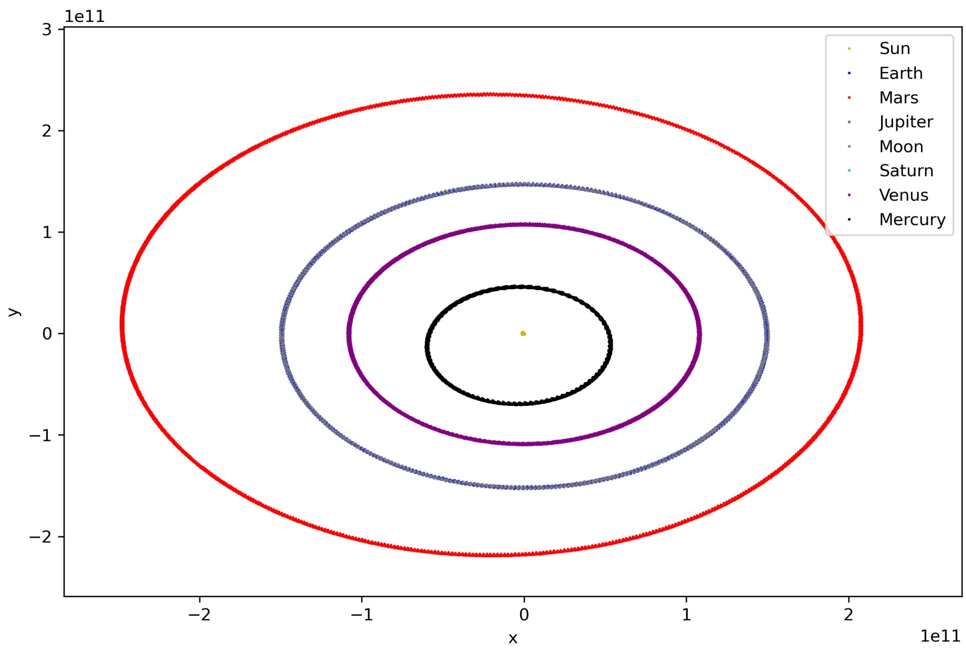

Figure 5: Graph of x-y positions of planets of Solar System over ten years with $\Delta t = 7200\ s$ using Euler-Cromer method

It could be seen that although according to momentum calculations, Euler's method is very slightly more accurate at smaller Δt than Euler-Cromer, it does not reproduce stable orbits at greater time steps. Euler-Cromer on the other hand provides values of position and velocity of reasonable accuracy (withing 0.1% for $\Delta t < 6000$ seconds), conserves angular momentum relatively well and reproduces stable orbits for all bodies of the system at a wide range of Δt.

In conclusion, calculations of momentum and comparisons of individual components of position coordinates and velocity provides limited insight into how effective and accurate a particular numerical method is. It was found that visual presentation of trajectories provides information about particle's final position as well as all the intermediate iterations. In terms of a practical example, the difference between initial and final angular momentum after 10 years with $\Delta t = 7200$ using Euler's method was just above 0.2%. This is only greater than that of Euler-Cromer by roughly a third, which is relatively not a lot. However, plotting a graph reveals a completely different picture, which could not have been implied from just calculating momentum difference. My suggestion of explanation of this limitation of the angular momentum test revolves around the following: even though the orbits of individual planets are clearly not stable, because some are moving towards and some away from the barycenter of the system (could be seen on Figure 4), the increasing angular momentum of one body would partially cancel out the decrease in momentum of the other. An improvement could be to calculate angular momentum difference for each planet separately and comparing those independently to avoid cancellations described.

Another area of improvement is introduction of more numerical methods. Numerical methods discussed in this report are simple and inaccurate relative to others, such as Euler-Richardson and Verlet. Implementing them into the code would likely improve the accuracy of values, since e.g. Verlet takes into calculation the second order of Δt, meaning the truncation results in an overall error $\propto (\Delta t)^2$ instead. Note that Verlet method was implemented into the code and is available for use, however has not been tested to an extend that other methods have been and is not discussed in this report.

Based on quantitative results obtained using both numerical methods and their comparisons with the true values (Emphemeris data), it could be concluded that for a relatively small Δt (< 360 seconds) both methods are capable of reproducing stable orbits with Euler-Cromer method yielding slightly better values, affecting the final result negligibly. However, as Δt increases, Euler-Cromer proves to be much more stable compared to Euler and outputs much more accurate values, hence is more effective at performing calculations for simulations of shorter and longer time period.

References

[1] California Institute of Technology 2008, NASA, accessed 2 December 2020, `https://ssd.jpl.nasa.gov/horizons.cgi`