Lancaster University

PHYS389 Final Report

Solar System Simulation

Iliya Frolov

(Dated: 10/12/19)

The aims of this investigation is to accurately simulate and analyse an n-body system in space (in this case the Solar System) using the programming language - Python. The accuracy of the model produced will be analysed using a series of techniques such as calculating the conservation of the total energy and total magnitude of the linear momentum of the system. Two methods - the Euler method and the Euler-Cromer method, will also be compared against each other to determine which one produces a more accurate simulation. At the end, for the Euler method, the percentage difference obtained for a time step of 1000, 100 and 10 was: 1.99%, 3.38% and 0.44% respectively. For the Euler-Cromer method, the percentage difference obtained for a time step of 1000, 100 and 10 was: 1.58%, 3.36% and 0.44% respectively. The percentage difference of the momentum of the system was observed to be negligible.

I. INTRODUCTION

A. Sub Intro

In this report, I will be investigating and presenting the results of my code which is designed to simulate n-bodies in orbit (in this case the Solar System). The information of the bodies including initial positions, velocities and mass were taken from JPL Horizons. How accurate the simulation is, will be analysed using techniques such as observing the conservation of the total energy of the system as well as the total magnitude of the linear momentum.

B. Background Theory

Within the code, I used two methods in order to calculate the positions and the velocities of the bodies from their acceleration, the first method is known as the Euler method:

$$\vec{v_{n+1}} \approx \vec{v_n} + \vec{a_n}\triangle t \tag{1}$$

$$\vec{x_{n+1}} \approx \vec{x_n} + \vec{v_n}\triangle t \tag{2}$$

where $\vec{v}$ is the velocity of the body, $\vec{a}$ is the acceleration of the body, $\vec{x}$ is the position of the body, $\triangle t$ is the change in time and n denotes the start of the time-step whilst $n+1$, the end of the time-step. This method gives a numerical solution for the positions and velocities of the bodies at a time $t+\triangle t$, given an earlier time t. It assumes the acceleration is constant for a small interval of $\triangle t$. The second method is known as the Euler-Cromer method:

$$\vec{v_{n+1}} \approx \vec{v_n} + \vec{a_n}\triangle t \tag{3}$$

$$\vec{x_{n+1}} \approx \vec{x_n} + \vec{v_{n+1}}\triangle t \tag{4}$$

This method uses the velocity at the end of the time-step instead of the beginning which should give a more accurate result.

In order to calculate the acceleration of each body, a modified form of Newton's law of gravitation is used along with the principle of superposition:

$$\vec{a_i} = \sum_{i=j}^{N} \frac{Gm_j}{\|r_j - r_i\|} r_j \vec{-} r_i \tag{5}$$

where G is the universal gravitational constant, $\vec{m_j}$ is the mass of the body causing the acceleration and $r_j \vec{-} r_i$ is the difference between the position vector pointing towards the body causing the acceleration and the position vector pointing towards the body being accelerated.

In the design section II part of the report, I will be detailing how my code works in terms of the purpose of each class, method/function used and how they help in simulating or analysing the data produced by the simulation. This section will then be followed up by my results section where I will present the data of the analysis carried out and what it tell us about the simulation.

II. CODE STRUCTURE

The code is run in Python and is structured using OOP (object orientated programming), whereby it utilises two classes and a file to run the simulation. The file used to run the simulation contains the imported bodies from JLP Horizons, as well as the code used to plot the simulation and graphs for the analysis. The two classes are called 'particle' and 'solarsystem' respectively.

The main function of the particle class is to create instances which are the bodies and assign attributes to that instance which are the properties of that body. The following are the attributes assigned: 'position', 'velocity', 'acceleration', 'name' and 'mass'. The particle class also contains two methods - 'update 1' and 'update 2'. 'update 1' contains the code to run the Euler method and 'update 2' contains code to run the Euler-Cromer method.

The 'solarsystem' class contains the main bulk of code that simulates the system. The instances it creates are the systems, the attributes of a system are the following: 'planets', 'kinetic energy', 'potential energy', 'total energy', 'momentum' and 'method'. The class also contains the following methods: 'simulation', 'ke solar', 'pe solar', 'energy conservation', 'momentum solar' and 'method'. Below is a flow chart that shows how the code takes in a list of bodies and simulates a system out of that.

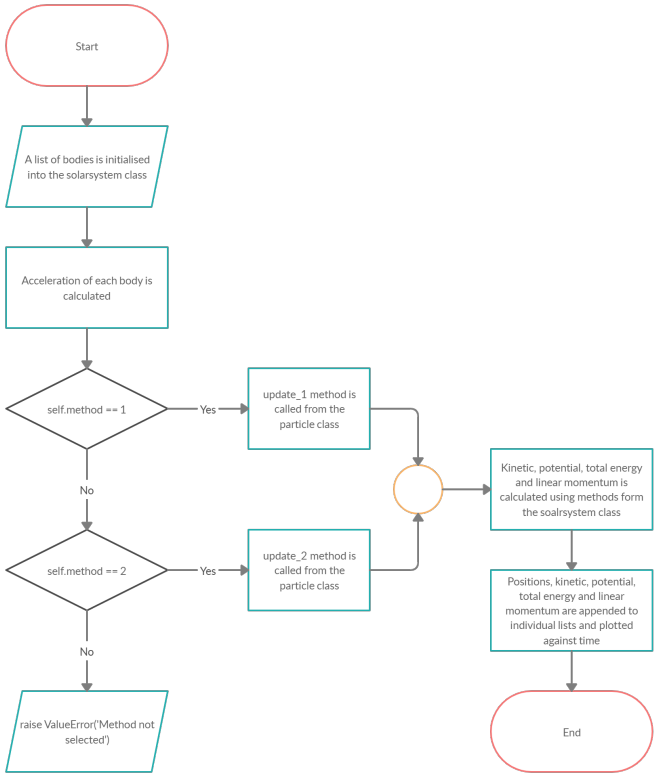

FIG. 1. A flowchart showing the main structure of the simulation code

III. ANALYSIS

A. Does the simulation work?

After running the code, the plot of the simulation can be shown in a graph below:

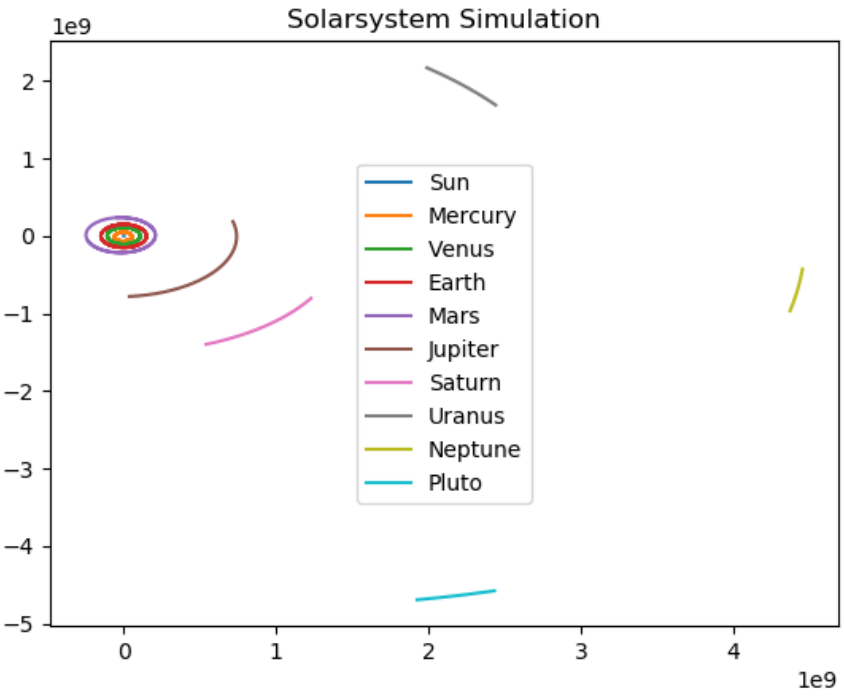

FIG. 2. Plot of the simulation for $\triangle t = 100$

From initial observation, the plot seems to represent an accurate model of the solar system, however the orbits of the planets form Jupiter and above are incomplete due to the range not being long enough. Upon closer examination of the inner orbits (figure below), the thickness of the lines indicate inaccuracy of the simulation as the body orbits are inaccurate. This is due to $\triangle t$ not being small enough and can therefore be corrected by having this value smaller. However, having a smaller $\triangle t$ will require a larger range in order to get sufficient data which would take up a lot of time, so a trade off for accuracy must be made.

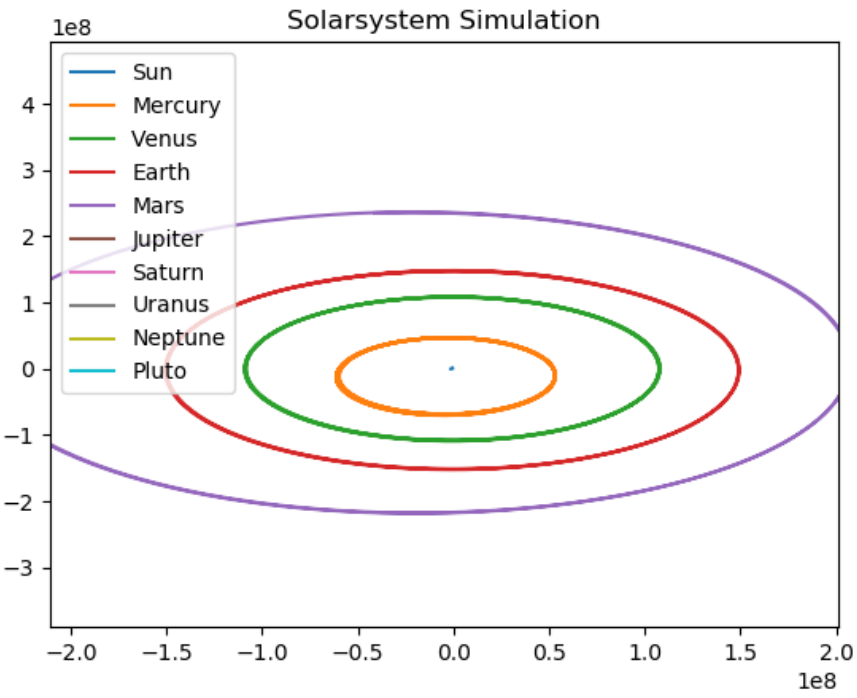

FIG. 3. Zoomed in image of the system

In order for the simulation to be accurate, the total kinetic and total magnitude of the linear momentum taken from the initial conditions of the system must remain constant. The simulation was carried out over a step range of 1,000,000 (1,000,000 repetitions) and was run separately for a time step of 10, 100 and 1000 for each method. The percentage difference of the total energy and momentum was then analysed for each method to determine the reliability and accuracy of the methods. Subsection A will be analysing the results of the simulation using Euler's method, and subsection B will be analysing the results of the simulation using the Euler-Cromer method. The accuracy of the methods will be analysed using percentage difference:

$$percentage \quad difference = \frac{final \quad value - initial \quad value}{initial \quad value} \times 100 \tag{6}$$

A starting value of the total energy was calculated from the initial conditions and then a final value was taken at the end of the simulation, the percentage difference was then calculated. The results of this analysis is shown in the table below as well as the graphs produced by the simulation.

B. Analysis of the Euler method

	Percentage difference	
Time step	Total Energy	Total Linear momentum
1000	1.99%	0
100	3.38%	0
10	0.44%	0

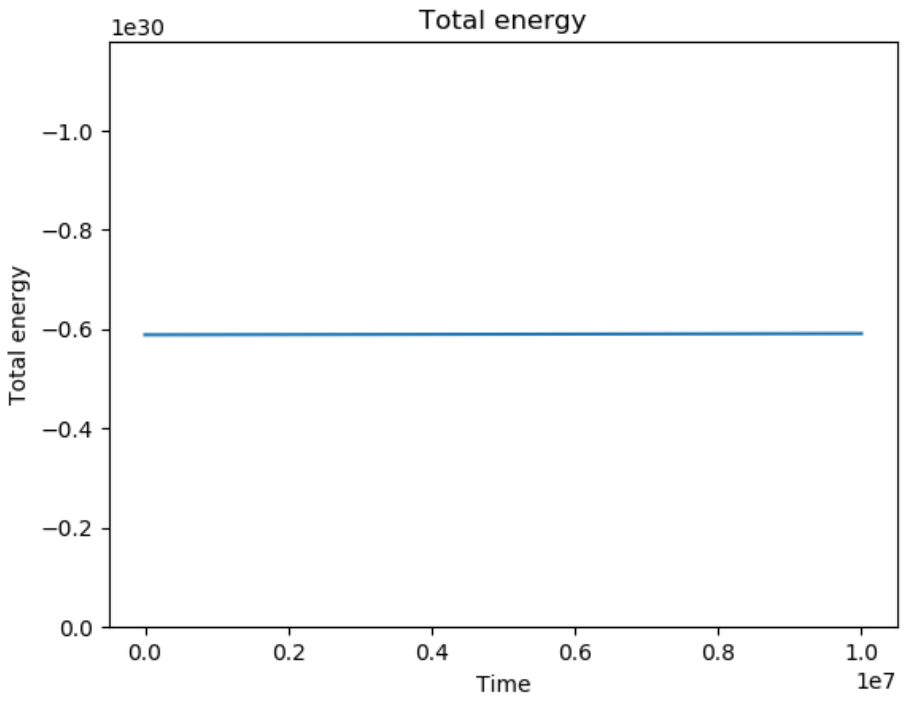

FIG. 4. The total energy of the system with $\triangle t = 10$

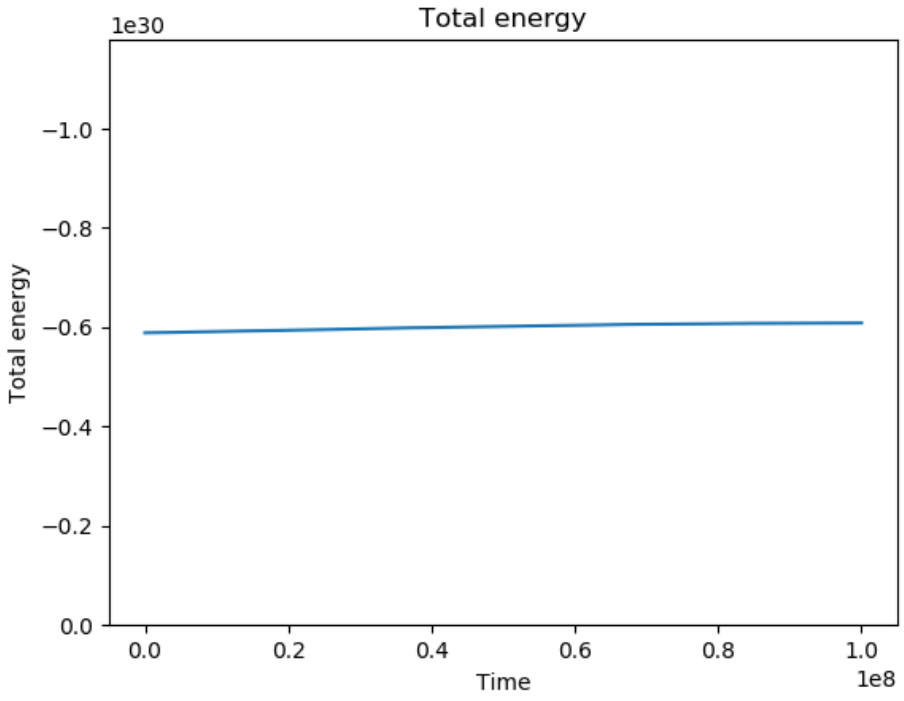

FIG. 5. The total energy of the system with $\triangle t = 100$

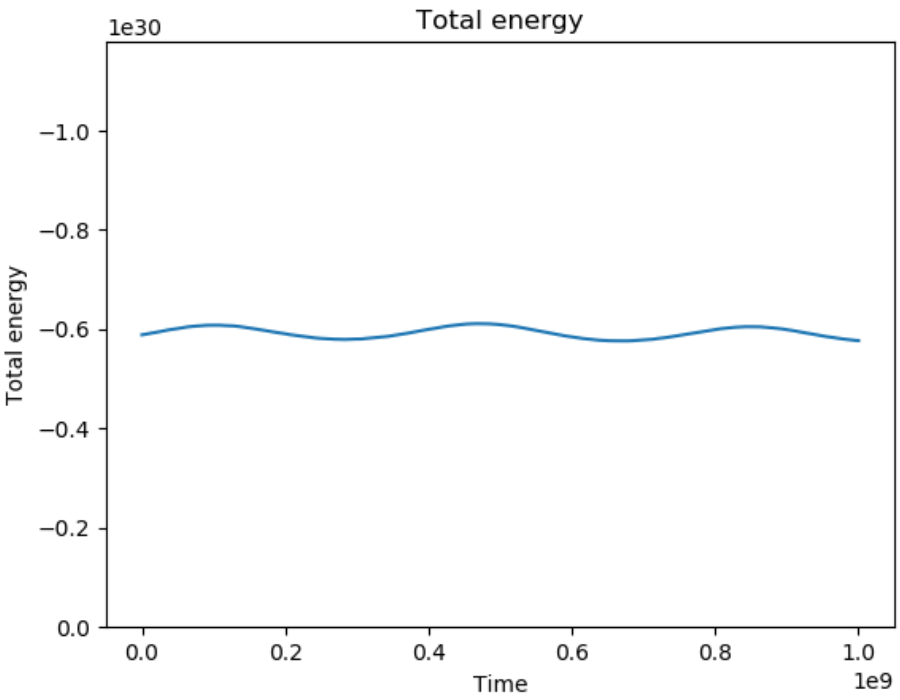

FIG. 6. The total energy of the system with $\triangle t = 1000$

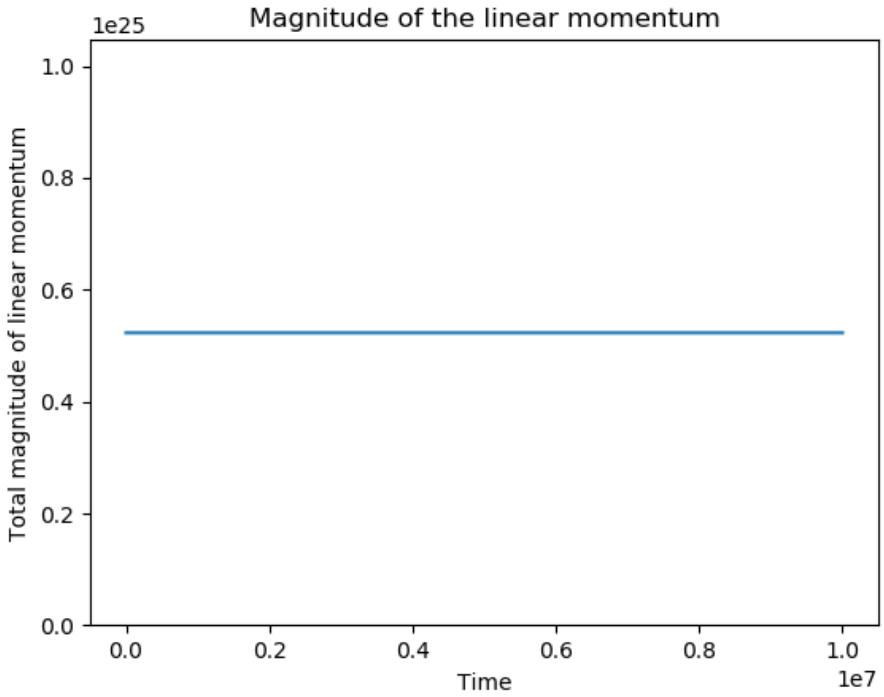

FIG. 7. The total linear momentum of the system with $\triangle t = 10$

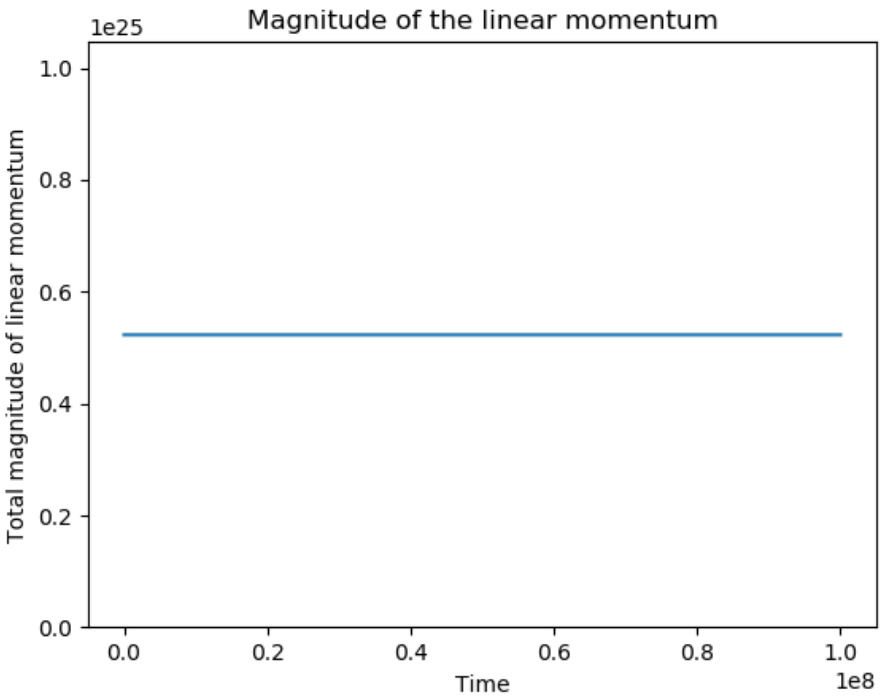

FIG. 8. The total linear momentum of the system with $\triangle t = 100$

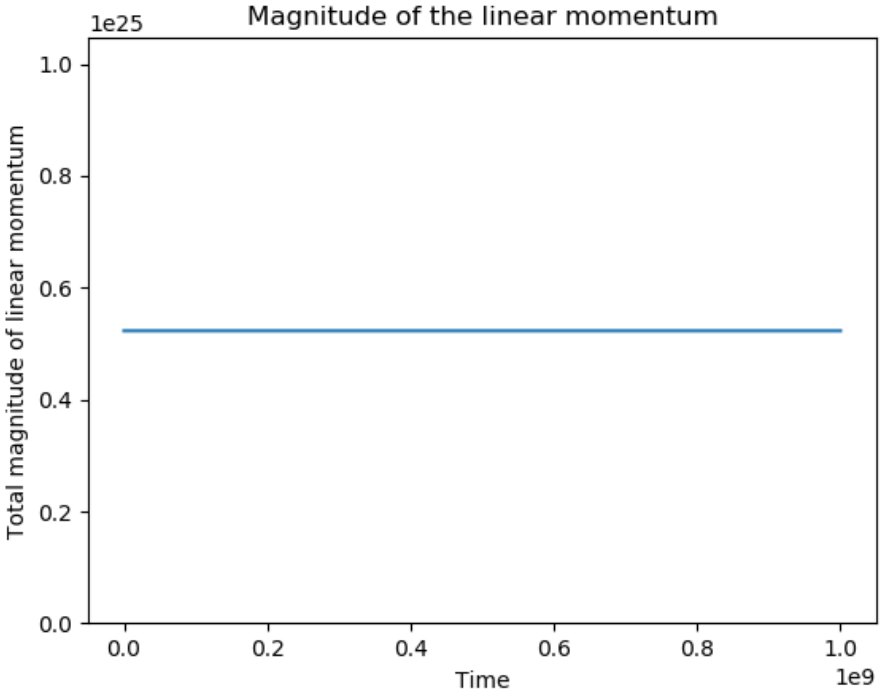

FIG. 9. The total linear momentum of the system with $\triangle t = 1000$

C. Analysis of the Euler-Cromer method

	Percentage difference	
Time step	Total Energy	Total Linear momentum
1000	1.58%	0
100	3.36%	0
10	0.44%	0

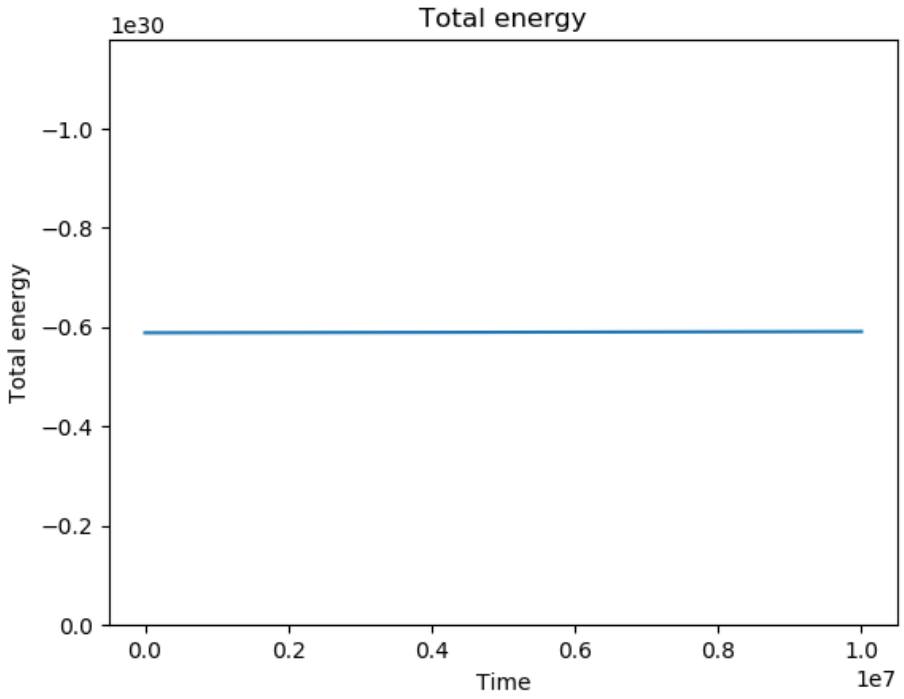

FIG. 10. The total energy of the system with $\triangle t = 10$

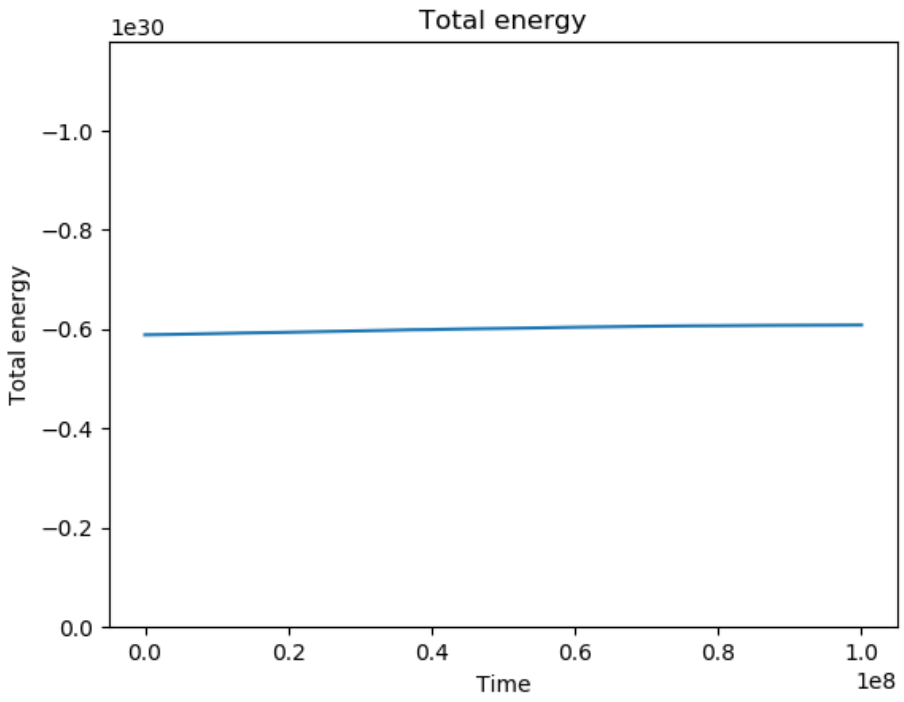

FIG. 11. The total energy of the system with $\triangle t = 100$

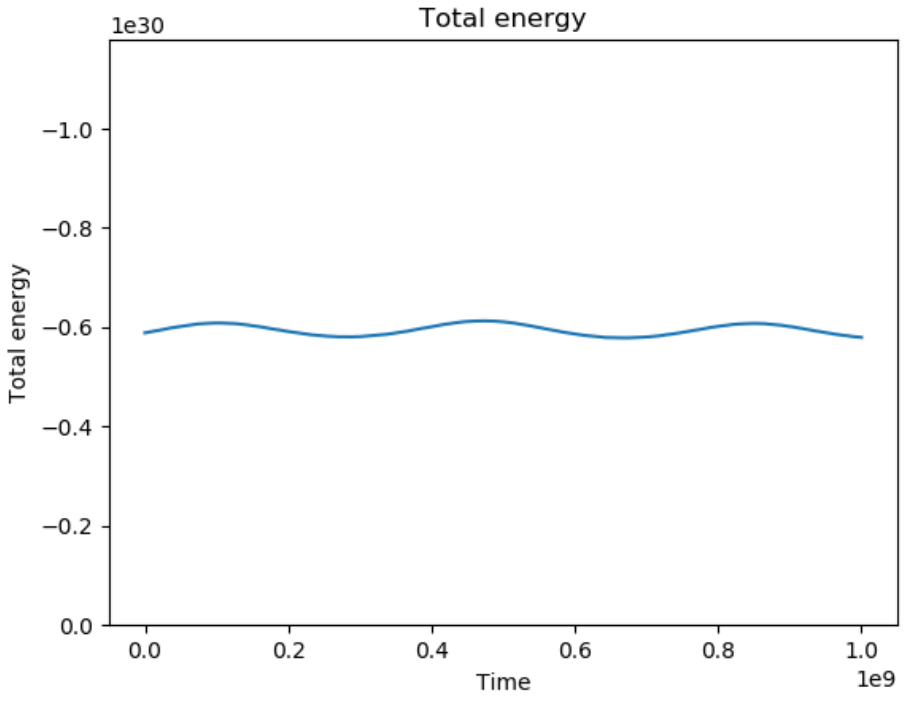

FIG. 12. The total energy of the system with $\triangle t = 1000$

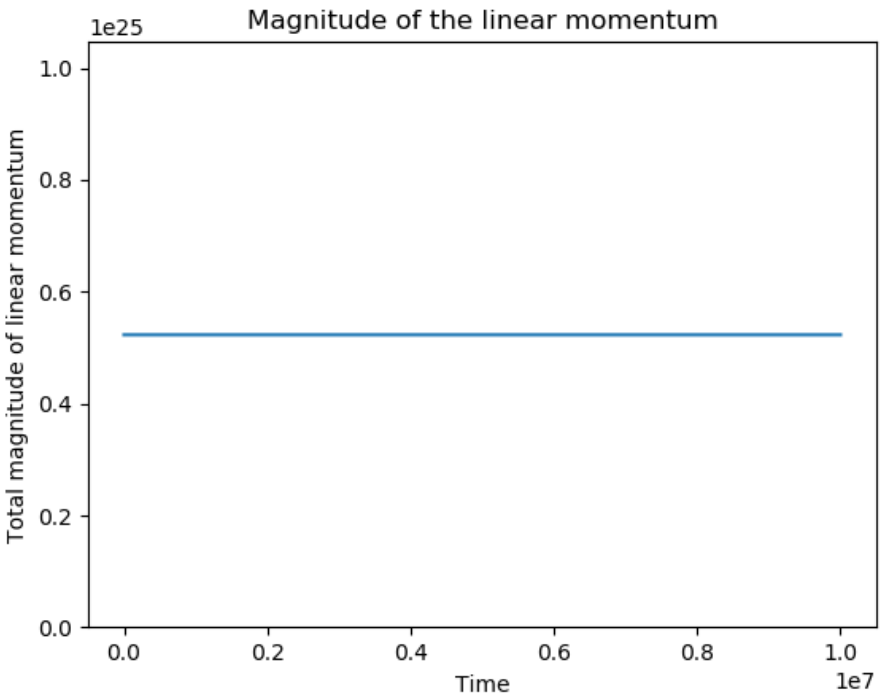

FIG. 13. The total momentum of the system with $\triangle t = 10$

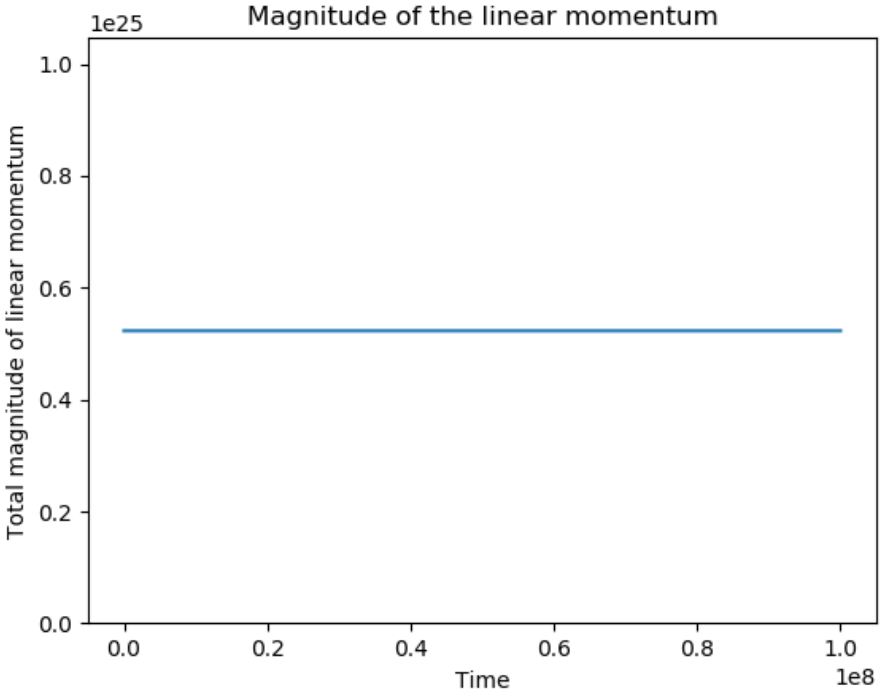

FIG. 14. The total momentum of the system with $\triangle t = 100$

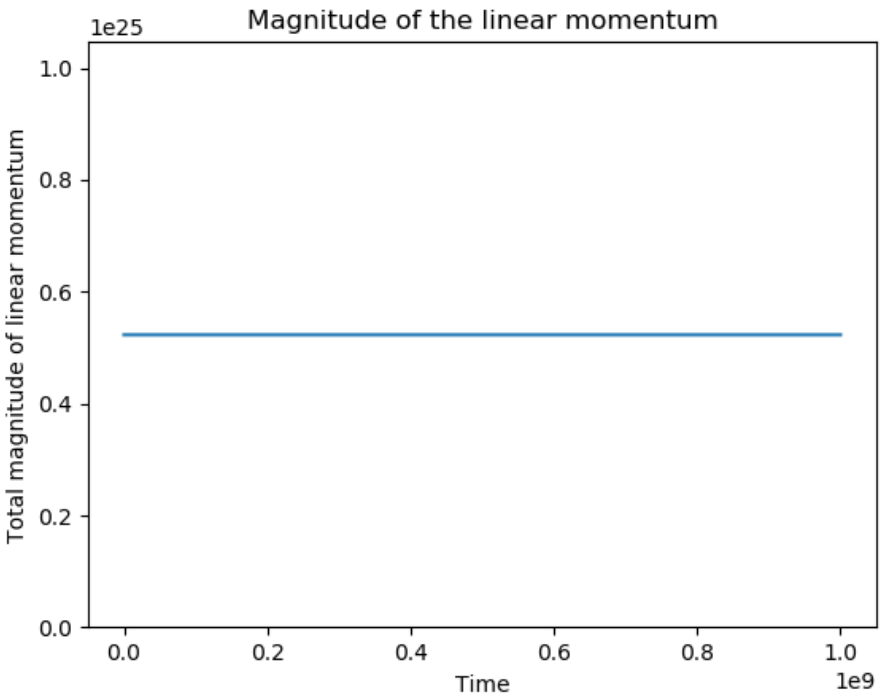

FIG. 15. The total momentum of the system with $\triangle t = 1000$

IV. CONCLUSION

In conclusion, the plot of the system seems to represent a fairly accurate model of the solar system as shown by 2. This is furthermore backed up by the analysis as 5 and 8 show that the total energy and momentum is conserved with a percentage difference of 3.38% and 0% respectively. We can also see from 6 that the accuracy of the simulation starts to deteriorate when using bigger time steps (in this case 1000), which makes sense as the accuracy of the acceleration calculated for each iteration becomes more skewed as it is constant for a longer length of time.

When carrying out the analysis of the Euler method, the results show a small percentage difference with a difference of 0.44% for a time step of 10, 3.38% for a time step and 100 and 1.99% for a time step of 1000 (III B). These results are strange since you would expect the percentage difference to increase with each increase in the times step and not decrease. When calculating the percentage difference of the linear momentum, we observe that there is no visible change recorded for any time step as it is negligible.

For the analysis of the Euler-Cromer method, the results are very similar to that of the Euler method with very small differences in the percentage difference for corresponding time steps. The percentage difference was 0.44% for a time step of 10, 3.36% for a time step of 100 and 1.58% for a time step of 1000 (III C). We can see from the results that the Euler-cromer method yields a slightly smaller percentage difference for the corresponding time steps, with a difference of 0.02% for a time step of 100 and 0.41% for a time step of 1000. This shows that the Euler-cromer method is a better method to use as it provides a more accurate simulation of the system, however this difference quickly becomes negligible for smaller time steps as it is pretty much nonexistent when the time step is 10.

From the analysis it is clear that a major flaw of the investigation is the method by which percentage difference is calculated as the results given are strange. When calculating the percentage difference, the value seemed to decrease when going from a time step of 100 to 1000 which doesn't make sense, this abnormality may come from the flaw of the method by which percentage difference was measured. In the future, a better method to use would be to calculate the percentage difference every set iteration e.g 10 (this would have to be coded), and calculate the average percentage difference across the iterations. In order to further analyse the simulation, we could have also included methods such as the Virial method.

---

[1] https://modules.lancaster.ac.uk/course/view.php?id=11842