Simulating Our Solar System

Annabelle Carver

14/12/20

Abstract

This paper accounts for the simulation of the solar system using three different numerical approximations, Euler Method, Euler-Cromer Method and Verlet's algorithm. It explains how the code functions to support an n-body simulation producing qualitative plots of their orbital path and quantitative figures for the energies of the system. Each aspect of the project and results are analysed and tested to deduce which algorithm is most apt in modelling our solar system and providing accurate information comparable to NASA's. Limitations on accuracy along with their cause are widely discussed.

Contents

1 Introduction

Leonhard Euler (1707-1783) was a Swiss-born mathematician known as one of the top 5 mathematicians of all time writing over five hundred books and articles. Euler is arguably the most influential creator of everyday pure mathematics establishing critical identities such as $e^{i\theta} = \cos\theta + i\sin\theta$[1]. Before dying in St.Petersburg, Russia on September 18^{th} 1783 Euler created an approximation based off numerical integration of ordinary differential equations which

is comparable to the Taylor series expansion[2]. This method is widely known as the 'Forward Euler Method'. Given initial values the forward method can be used to determine position and velocity at time $t + \delta t$ assuming that acceleration stays constant during δt.

$$x_{n+1} = x_n + v_n \Delta t[4] \tag{1}$$

$$v_{n+1} = v_n + a_n \Delta t[4] \tag{2}$$

By way of comparison this paper also considers two further approximations, the Euler-Cromer Method (also known as the semi-implicit method) and the Verlet algorithm (commonly used in molecular dynamics). Although similar, all three of these approaches contain slightly different variations. Distinctly, Euler-Cromer focuses on determining the next velocity in order to compute the associated position and the Verlet method requires both current acceleration at t in addition to the acceleration of the subsequent time step, $t + \delta t$.

$$v_{n+1} = v_n + a_n \Delta t[4] \tag{3}$$

$$x_{n+1} = x_n + v_{n+1} \Delta t[4] \tag{4}$$

$$x_{n+1} = x_n + v_n \Delta t + \frac{1}{2} a_n (\delta t)^2[4] \tag{5}$$

$$v_{n+1} = v_n + \frac{1}{2}(a_{n+1} + a_n)\Delta t[4] \tag{6}$$

a denotes acceleration, v is velocity, x is position, δt is a time step and n symbolises a current position while $n+1$ symbolises the next position. For this report acceleration was not a provided value, hence the following equation was utilised alongside imported data from NASA:

$$a = \frac{GM}{r^2} \hat{r}[4] \tag{7}$$

Here, G is the gravitational constant 6.67x10$^{-11} m^3 kg^{-1} s^{-2}$, M is the mass of the attracting body and r is the distance between the two bodies. The r outside of the fraction is styled with a hat to show that the acceleration is of vector form. Although the code associated with this paper does not incorporate this form of the gravitational acceleration it is displayed here for ease of integrating where gravitational potential energy (used for the energy graphs) is the result (8).

$$U = \frac{-GMm}{r} \tag{8}$$

For analysing the n-body results produced from the varying approximations, qualitative data in the form of 3D orbital colour plots and quantitative data in the form of 3 layer graphs are created. It is clear this project resembles the N-body problem, 'a large number of bodies that interact exclusively by gravity', making Virial's Theorem appropriate for analysis[3].

$$W + 2K = \frac{d^2 I}{dt^2}[3] \tag{9}$$

As our system is time independent the theorem simplifies to $W + 2K = 0$ where W is gravitational potential energy, K is kinetic energy and I was inertia. The following sections provide an account of the simulation, results, testing and analysis. Where there are extensions beyond the scope of the exercise extra details shall be provided.

2 Simulation layout

2.1 Particle class

This file contains a class for a generic body in the solar system simulation. In addition to containing all base variables required for manipulating the system's planets, this class also calculates kinetic energy and momentum of a body at a given position.

2.2 Solar system class

Utilising the methods from the particle file, the solar system class coalesces all the given n-bodies to simulate the solar system. The file's fundamental purpose is running two methods which update a particle's acceleration to approximate the direct position and velocity of said particle via any of the three mentioned algorithms summoned by a key word. Employing these base methods, the file then calculates the following positions and velocities as well as the total kinetic energy, total gravitational potential energy, total energy, total momentum and total angular momentum of the system as a whole.

2.3 Solar system data

Npy files storing all critical data for any programmed simulation is made in this file. Here, data from NASA's HORIZONS interface is ran through the solar system's methods to predict each particle's journey through space and its associated energy[6]. By changing one key word, the time step and range of iteration, six different data files can be made for a certain algorithm over the chosen time frame.

2.4 Simulation analysis

The simulation analysis file assembles the code and data together by way of graphics to allow for quantitative analysis. This file creates a 3D labelled orbit plot displaying all n-bodies for one chosen numerical approximation. Furthermore, for the energy data collected, all three numerical approximations are shown on a 2D labelled graph. Not only does this file supply figures but also numerical answers for maxima and minima, something hard to acquire from the code beforehand.

2.5 Testing

Throughout the simulation many qualitative tests, such as the print() statement, are incorporated to allow debugging to be easier. However, the main source of analytical testing came from this file to confirm whether the code complies with the laws of physics. The classes are put though small simulations to examine the gravitational acceleration and angular momentum.

3 Results

3.1 Solar system simulation

The three simulations below are primarily showing how the terrestrial planets (Mercury through to Mars) and Jupiter orbit around the largest mass in the system; this is the sun which accounts for $1.9891\text{x}10^{30}kg$ of our system. As a result, the planets closest to the sun move faster

to counteract the gravitational force pulling them inwards. This is clearly demonstrated in the figures (1)(2)(3) where the planets proximate to the sun perform more complete orbits displaying thicker coloured lines. For a simulation of the entire solar system up to Pluto please reference the appendices (18)

3.2 Solar system simulation

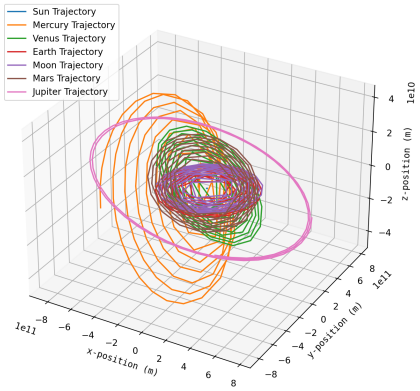

Figure 1: Euler method

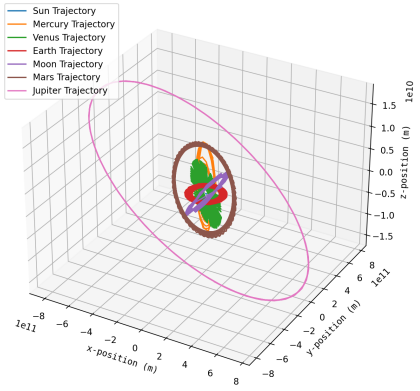

Figure 2: Verlet's algorithm

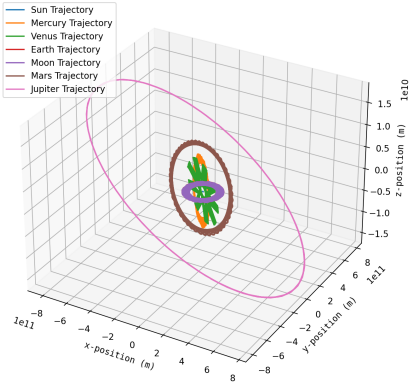

Figure 3: Euler-Cromer method

Figure 4: 3D orbit plots of the planets up to Jupiter at a large time step of one day (86400 seconds) over 10000 iterations showing the growing instability moving from right to left. For a larger view of these figures reference the appendices.

However, it is evident that the paths produced by the planets under Euler Method show greater motion outwards than the other methods; remarkably so, each rotation can be distinctly pointed out. This inaccuracy can be demonstrated by comparison to the Taylor's expansion[2]. Euler method is derived simply from taking derivative at a point on a curve more commonly defined by $f(x_n, y_n) = (y_{n+1} - y_n)/(x_{n+1} - x_n)$. When compared to the Taylor series below (an approximation of higher degree) we see the truncation error is caused by the absence of the of next term. Therefore, when looking at Euler method results, the local error of $(\delta t)^2$ per step should be remembered. Although the Euler-Cromer method is also a first-order integrator, giving a negative error of $(\delta t)^2$ per step in the velocity calculation, this is counteracted by the positive error produced when the next velocity value v_{n+1} is used to calculate the next position. It can be visualised as finding the derivative at the end of a time step rather than in the middle.

$$x_{n+1} = x_n + f(x_n, t)\delta t + \frac{1}{2}(\delta t)^2 \frac{df(x,t)}{dt} [5] \tag{10}$$

Verlet's algorithm can be seen as an improved Euler method. Its position calculation is of a higher order than Euler's giving a smaller local truncation error of $(\delta t)^4$ per step. Additionally, the velocity calculation bears greater precision by using the average acceleration (6).

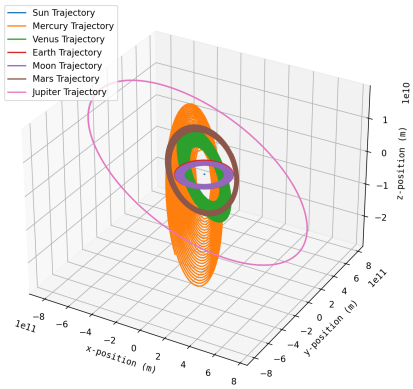

Figure 5: Euler Method

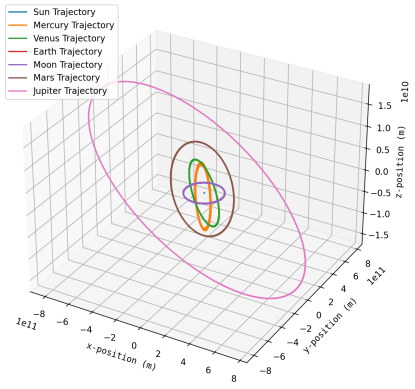

Figure 6: Verlet's algorithm

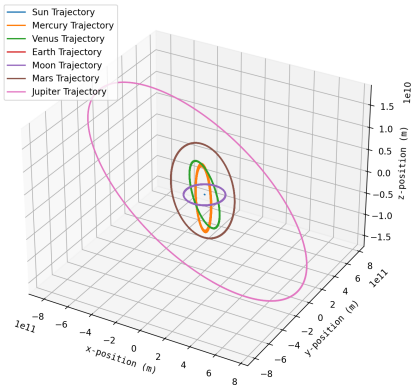

Figure 7: Euler-Cromer method

Figure 8: 3D orbit plots of the planets up to Jupiter at a medium time step of 8640(s) over 100000 iterations. There is growing similarity of the Verlet and Euler-Cromer methods. For a larger view of the figures reference the appendices.

Visible accuracy of the orbits dramatically changes as the time step interval decreases. As the time step decreased by a magnitude of 10 the iterations were increased accordingly. Even though the time frame is the same the number of orbits appears smaller given the greater accuracy. It is clear that the planets nearer to the sun are seemingly less stable. Their period of rotation is smaller, their velocity is greater and therefore the distance they travel each time step is larger. For example Mercury finishes its orbit in 88 days, 2.55 faster than the next planet. The error these larger distances face are noticeably significant when put together over a smaller volume contrasting to say Jupiter. This affect is considerably easier to spot with the Euler simulation, its lack in accuracy pushes trajectories of planets like Mercury outwards. From qualitative data alone the Euler method is facing many limitations for the use of approximating an n-body solar system.

The Euler-Cromer and Verlet plots in figures (4) and (8) are very similar only showing discrepancies on large time steps. However, Both Mercury and Earth's moon are evidence that the Euler-Cromer method is more accurate. The precession of Mercury and the moon under Verlet's algorithm move outwards dramatically and even show the moon to go off orbit from earth, a phenomena that would not occur. An explanation for this could be Verlet's dependence on gravitational acceleration for calculating velocity (6). As each planets' path deviates outwards (particularly the closet ones to the sun) from its orbit due to local error the gravitational force between them may increase if their distancing reduces (7). For the Verlet calculations this would increase the affected planet's velocity causing its orbit to precess outwards; demonstrated by the moon's proximity to the Earth.

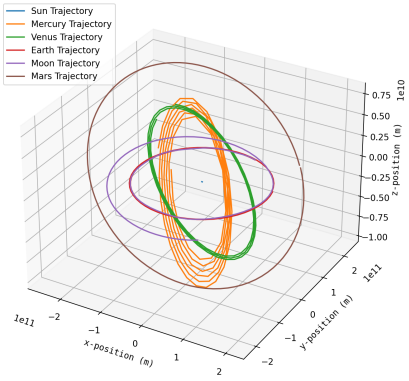

Figure 9: Euler Method

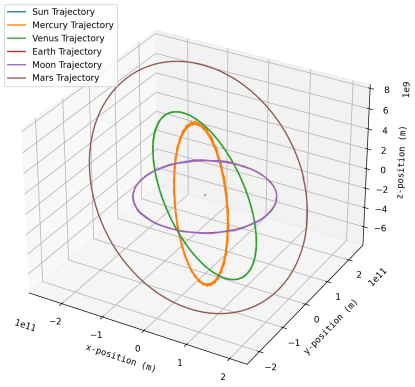

Figure 10: Verlet's algorithm

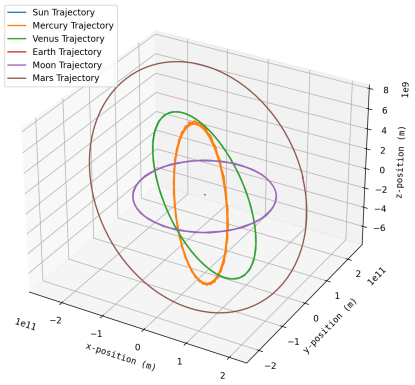

Figure 11: Euler-Cromer method

Figure 12: 3D orbit plots of the terrestrial planets at a medium time step medium of 6000(s) over 10,000 iterations. For a larger view of the figures reference the appendices.

Interestingly, when Jupiter (the largest mass of the planets) is taken out of the n-body system the Euler-Cromer method and Verlet's algorithm become almost identical. The two are remarkably accurate that even at a time step of 6000, such as in figure 12, the motion of the moon about the Earth can be distinguished. Jupiter's mass of $1.898\text{x}10^{27}$ (about 150 times larger than all the terrestrial planets put together) clearly affected the accuracy of Verlet's algorithm by increasing gravitational acceleration (6); The addition of Jupiter's mass did not counteract its distance from the other planets which could be expected as radius is also a variable in gravitational acceleration(6). Consequently, Verlet's algorithm is hindered by large variation of mass.

3.3 Energy of the System

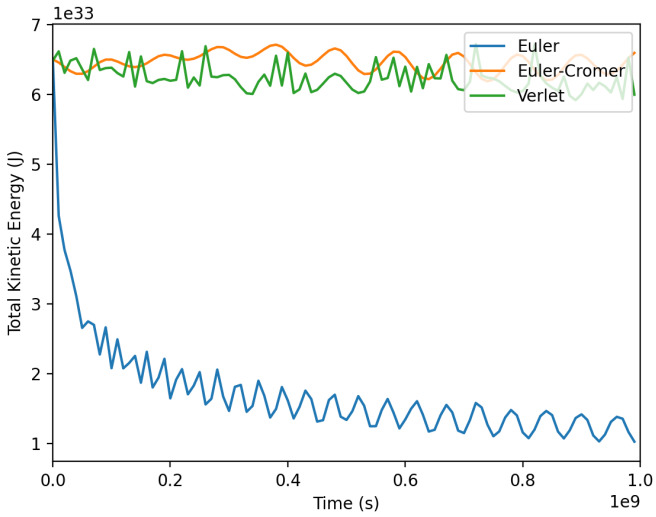

Figure 13: This graph shows the total kinetic energy of the solar system. Here the time step is large at 100000(s) over 10000 iterations.

The total kinetic energy of the system shows clear oscillations, demonstrating simple harmonic motion. Within a closed system an object moves about a centre point without dissipating energy but rather transferring it between kinetic energy and potential energy; here the potential energy is gravitational. Varying peaks and troughs of Figure 13 displays that our n-body system is not typical harmonic motion. The variation of orbits and periods of the planets in our simulation suggests they are all at differing phase with one another. By deduction this means they construct a certain set of gravitational frequencies. In fact the whole solar system is observed to oscillate with a base period of 178.38 years and is known as "the music of the spheres"[7].
In the above graph Euler's method shows a substantial loss of kinetic energy whereas Euler-Cromer and Verlet's algorithm do not. All three simulations start at 3.2x10^{33}J. It seems that Euler's method is not only limiting for position plotting but also for the whole dynamics of the system. The loss of velocity at each time step causes the planet to exponentially lose kinetic energy (tending to $1J$); this could be predicted as $K = \frac{mv^2}{/}2$.

Interestingly Euler-Cromer exhibits a very smooth graph oscillating between 3.3x10^{33}J and 3.1x10^{33}J about a steady position. On the other hand, Verlet's algorithm is very spiky and gradually starts to stray away from Euler-Cromer dipping below 3x10^{33}J. This suggests that the Verlet algorithm potentially over estimates the change in kinetic energy and the Euler-Cromer method under estimates the kinetic energy. Given Euler-Cromer calculates the velocity first with $(\delta t)^2$ error per time step but Verlet using a more accurate acceleration this theory does not seem amiss.

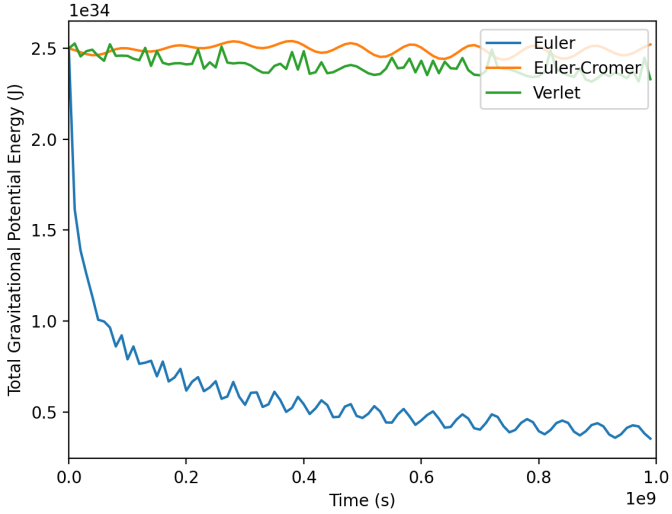

Figure 14: This graph shows the total potential energy of the solar system. Here the time step is large at 100000(s) over 10000 iterations.

The potential energy of the planets is also oscillating, displaying the same simple harmonic motion. Here, however, the average energy is much higher with all three simulations starting at 2.5x10^{34}J. A potential reason for this could be that gravitational potential energy requires the use of the gravitational constant (8), 6.67x10$^{-11}m^3kg^{-1}s^{-2}$, whereas kinetic energy does not.

When looking at figure (14 it seems that the oscillations are minor when really they're one magnitude larger than those in figure (13). Similarly to kinetic energy, Euler's Method is inaccurate and Verlet's simulation seems to drop energy ever so slightly. This can be explained by the domination of the increasing radii within the gravitational potential equation, something

more prevalent with Euler's simulation than Verlet's. Euler's method dissipates its potential energy exponentially again but this time towards $0J$. Why does Euler's method lose gravitational potential energy quicker than kinetic energy? The only answer we can suggest is that the inverse of the decreasing radii in gravitational potential is of a larger proportion to its starting value than the decrease of velocity squared in kinetic energy to its initial value. Conversely Euler-Cromer moves consistently from $2.55\text{x}10^{34}J$ to $2.45\text{x}10^{34}J$ while Verlet gradually loses potential energy bearing $2.4\text{x}10^{34}J$ after $1\text{x}10^{9}s$.

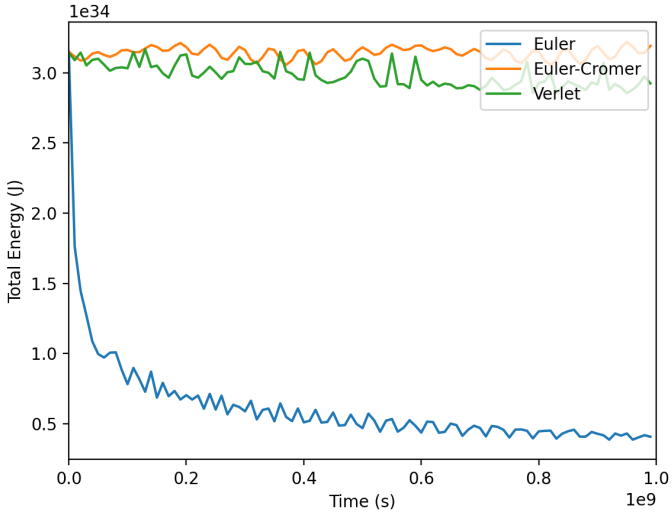

Figure 15: This graph shows the total energy of the solar system by summing the kinetic energy with the potential energy. Here the time step is large at 100000(s) over 10000 iterations.

The initial value of total energy for all three methods sits at $3.2\text{x}10^{34}J$. As expected from the previous figures as well as Virial theorem (9), the start value provides evidence that the potential energy dominants kinetic energy. This observation is crucial to testing whether our code functions the way its meant to. To see the results of the Virial theorem test please refer to section (3.5).
This graph of total kinetic energy is imperative over the potential energy or kinetic energy graphs as it shows a larger deviation between the algorithms. In comparison to before, Euler's method decreases in total energy more dramatically at the start raising the question as to whether the Euler Method is incapable of approximating the energies of an n-body system over a large time frame? Verlet's algorithm drops to $0.4\text{x}10^{34}J$ lower than the Euler-Cromer method. Suggesting that when it comes to analysing the energy of the n-body system Euler-Cromer method is superior to the accuracy of Verlet's Algorithm.

3.4 Time steps

With a time step decrease in magnitude 10, Verlet's simulation and Euler-Cromer's simulation start to map onto one another and demonstrate how the system of planets oscillate together with a set of gravitational frequencies or phases. Periodically these phases match up to create a node like point in space where the energies cancel to their mid position; In the graph below this period is about $0.22\text{x}10^{8}s$. This example perfectly leads onto the point that the time step of these figures and graphs are crucial for analysing the difference between numerical methods.

As demonstrated in the simulation section (3.2) and here in figure 16 as time steps get smaller the approximations map onto one another. The only exception we have seen to this is in figure 16 itself where the Euler method is so inaccurate it drops out of path with the others in about $0.2\text{x}10^8 s$. Apart from the decline in the Euler simulation, all approximations show similar oscillations from smaller to larger and back again repeatedly.

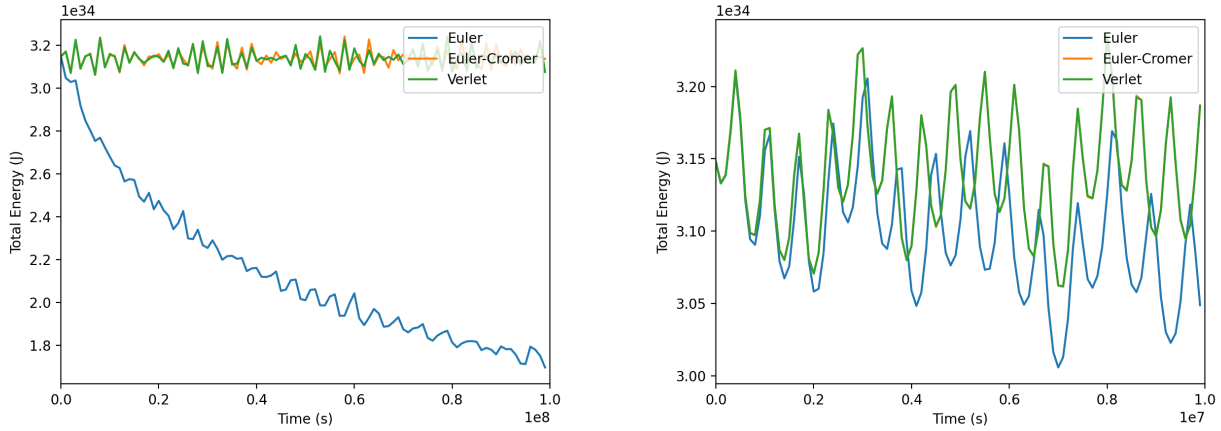

Figure 16: This graph shows the total energy of the solar system at a medium time step of 10000(s) (left) and small time step 1000(s) (right) over 10000 iterations.

Consistently it appears that Euler-Cromer is more accurate than Verlet's algorithm, nevertheless, here there is evidence for Verlet's use. As suggested in figure (13) the Euler method estimates the changes in kinetic energy to be lesser than expected whereas the Verlet method over estimates the changes in kinetic energy. This is evident in (16) where the time step is smaller and Euler-Cromer's method becomes more spiky and Verlet's becomes smoother.
In the left graph of figure (16) shows the similarity of all three approximations in the time before the Euler method loses energy dramatically. Interestingly, looking at the time frame we should note that 10000000 seconds have past; the equivalent of almost a year. In spite of Euler's method being the least trustworthy it does manage to keep the energy of the planets fairly consistent for almost a year. At such a small time step we can hardly even see Euler-Cromer as it is exactly the same as the Verlet simulation.

3.5 Testing the simulation

By virtue of Euler-Cromer simulations proven accuracy, the maxima and minima of its potential and kinetic energy for the medium time step of 10000(s) are used below to find the simulations error when performing the Virial theorem on this data (9). Computed from the simulation and rounded to 3 significant figures the maximum and minimum of the total potential energy were $2.55\text{x}10^{34} J$ and $2.45\text{x}10^{34} J$ giving $2.5\text{x}10^{34} J$ as the average. For total kinetic energy the maximum and minimum were $6.75\text{x}10^{33} J$ and $6.25\text{x}10^{33} J$ giving $6.5\text{x}10^{33} J$ as the average. It must be noted that Virial's theorem considers gravitational potential energy to be 0 and consequently adds on $2K$ to the potential. Given the nature of my code and how the simulation graphed the energies, the computed potential energy was positive. For this reason, kinetic energy shall be subtracted from potential.

Potential energy (J)	Kinetic energy (J)	W-2K (J)
$2.5x10^{34}$	$6.5x10^{33}$	$1.2x10^{34}$

Table 1: Table for testing the validity of the simulation using Virial's theorem.

Given that the expected output of Virial's Theorem should be 0 shows that the energy of the simulation is very inaccurate. The possible explanation for this occurrence is due to the limited number of planets the computer could run through the simulation. As a result the total mass of the system and distancing between all planets is significantly lower, not only would this affect the potential energy directly but indirectly slow down the planets movements by decreasing gravitational acceleration. Potentially all of the energy graphs could have been affected by this but seemed to create act as expected.
Having questioned the potential energy calculation it seemed fit to determine whether the code could return accurate values for the gravitational acceleration on 2 planets compared to NASA'S data[6]. Below are the results which are all within 0.5% error. Thus demonstrating that the gravitational potential energy calculations are correct and that maybe the discrepancy in Virial theorem comparison is from the approximations, kinetic energy calculation or creation of the .npy files.

Planet	Expected value m/s^2	actual value m/s^2	% error
Earth	9.800	9.793	0.07
Mercury	3.711	3.728	-0.46

Table 2: Table to show the gravitational acceleration of the surface of Earth and Mars.

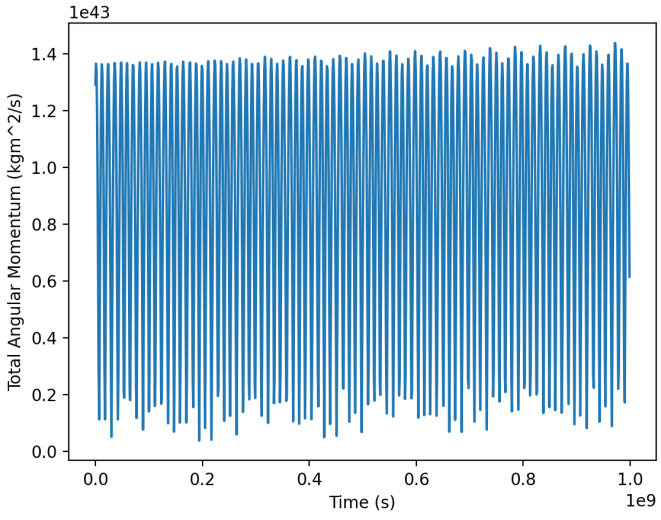

Figure 17: Graph to show the angular momentum of the planets up to Jupiter with a time step of 10000(s) over 100000 (s).

Momentum was the final test used for the simulation. Euler-Cromer's method was used to simulate this as it has proven thus far to be the most trustworthy approximation. The

graph oscillates dramatically between $0.05\text{x}10^{43} kgm^2/s$ and $1.4\text{x}10^{43} kgm^2/s$ suggesting that the planets are rotating around the sun with varying speeds as suggested by Kepler's second law. Most significantly this figure demonstrates the conservation of angular momentum by the code. The time frame is over $1\text{x}10^9$ seconds, which is almost three times as long as Jupiter's orbit, and even after this range the angular momentum oscillates about the average value of $0.725\text{x}10^{43} kgm^2/s$. Hypothetically, if the code is creating a reliable output for angular momentum then one could assume that the kinetic energy plots are also correct (given their distinct similarity). Potentially the reason Virial's theorem is an insufficient test is because the simulation lacks all of the planets and therefore cannot be considered a closed system.

4 Conclusion

In theory, Verlet and Euler-Cromer approximations have much smaller local error and better accuracy than Euler's method. The limitations of these 2 approximations sits within their equations (2) and (6).
The Euler-Cromer method has proven itself to be inferior to the Euler method and Verlet's algorithm when simulating our n-body solar system. In both qualitative orbit plots and quantitative energy graphs the Euler-Cromer method follows a rigid nature deviating only slightly, if not at all, from the expected theoretical path. In the solar system simulation the Euler method and Verlet algorithm produce thick lines suggesting outward trajectories of the planets and exposing instability in their calculations. The planets certainly do not stick rigorously to their orbital paths but over this small time range planets would show negligible aberration; Euler's plots are false for most cases.

The significance of time steps is the pinnacle deciding factor between approximations. It does appear that the Verlet's algorithm is only inaccurate over large time frames such as in figures 2 and 15; otherwise this method is reliable and almost indistinguishable from the Euler method. For total energy at a time step of 10000 both Euler-Cromer and Verlet methods oscillate continuously between $3.25\text{x}10^{34} J$ and $3.1\text{x}10^{34} J$. On the other hand, at a time step of 100000 Verlet drops to $0.15\text{x}10^{34} J$, a significant amount when compared to the oscillation amplitude. The simulation's range of time affects the accuracy that can be inferred. Figures like 16 over $1\text{x}10^7 s$ display Euler's method losing only $0.1\text{x}10^{34} J$. This can suggest that Euler is arguably quite accurate over small ranges like 116 days (the period here) but misrepresents the differences between the methods; Euler's method is not apt for simulating our n-body system. On the other hand, although the total accuracy of the methods are hindered by time range it is useful for investigating how the code and each method performs at each individual calculation. Difference between figures 15 and 16 displays that Euler-Cromer's oscillations become more pointed. This implies that over a longer time step and range, values are under estimated. Could Verlet's algorithm be more suited for an n-body system over smaller time frames and time ranges?

For our n-body solar system it is clear that Euler-Cromer was most suitable for modelling the simulation. Nevertheless, limitations obstructed further investigation into the nature of the approximations. Smaller time steps could not be run for longer ranges of time and in order to get any smaller time steps the number of planets had to be limited. Furthermore, the solar system does not just contain our planets and Earth's moon. There are many more moons and satellites that could be analysed for all three approximations and could affect the interactions and energies of the figures above. For a longer simulation of the whole solar system up to Pluto

see the appendices below(18).
Improvements for this simulation lie in the angular momentum investigation and time step range. Angular momentum could not be plotted for the planets under Jupiter but when it was included time steps could not go below 10000 (s). Additionally, due to the complex code of Verlet's algorithm angular momentum could not be simulated at all for this approximation. Revising this area of the investigation could present more differences between the methods.

References

[1] Nahin, Paul J. (2006) "Dr. Euler's Fabulous Formula : Cures Many Mathematical Ills", *Princeton University Press*

[2] Ydri, Badis (16 March 2016) "Computational Physics: An Introduction to Monte Carlo Simulations of Matrix Field Theory", *arXiv:1506.02567v2*

[3] Astro320, "Lecture 10: N-body problem: virial theorem", `www.astro.umd.edu`, *University of Maryland*

[4] `http://ma.ttpitk.in/teaching/PHYS281/gravity/`, Date accessed 13/11/20

[5] Chai, Patrick (August 2015) "Numerical Integration Techniques in Orbital Mechanics Applications", *ResearchGate*

[6] `https://ssd.jpl.nasa.gov/horizons.cgi`, Date accessed 1/12/20

[7] Nicola Scafetta (16 January 2014) "The complex planetary synchronization structure of the solar system", Cornell University, *arXiv:1405.0193*

[8] Carl D. Murray, Stanley F. Dermott (1999), "Solar System Dynamics", *Cambridge University Press*

5 Appendices

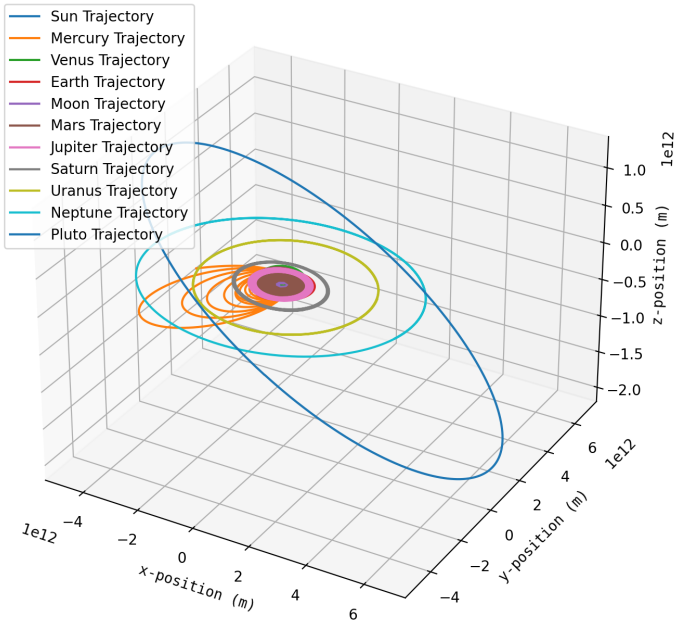

Figure 18: Simulation of the whole universe up to Pluto.

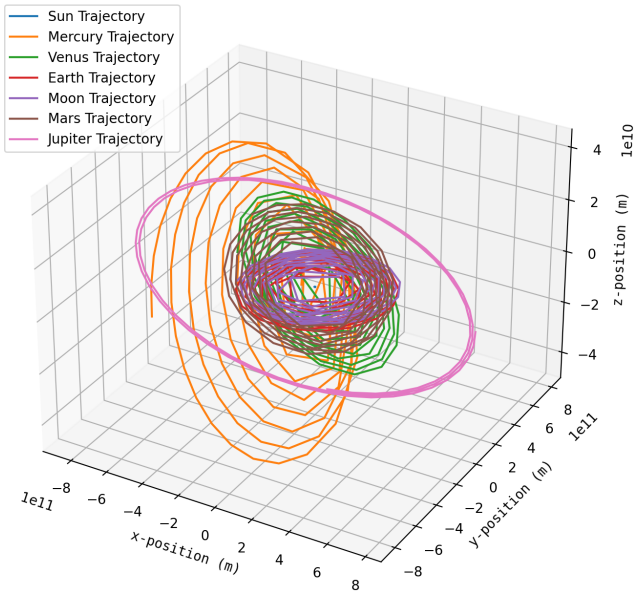

Figure 19: Larger version of figure 1.

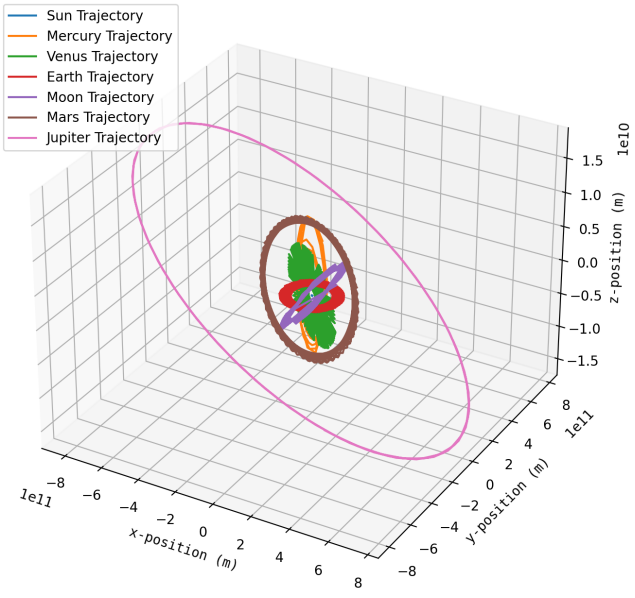

Figure 20: Larger version of figure 2.

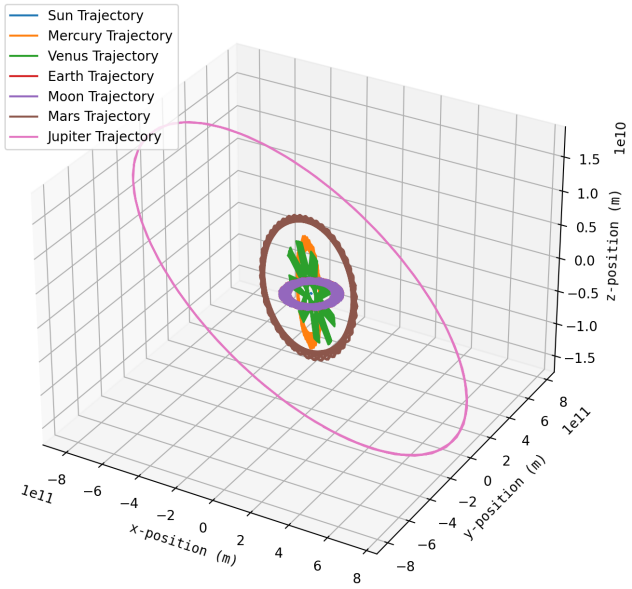

Figure 21: Larger version of figure 3.

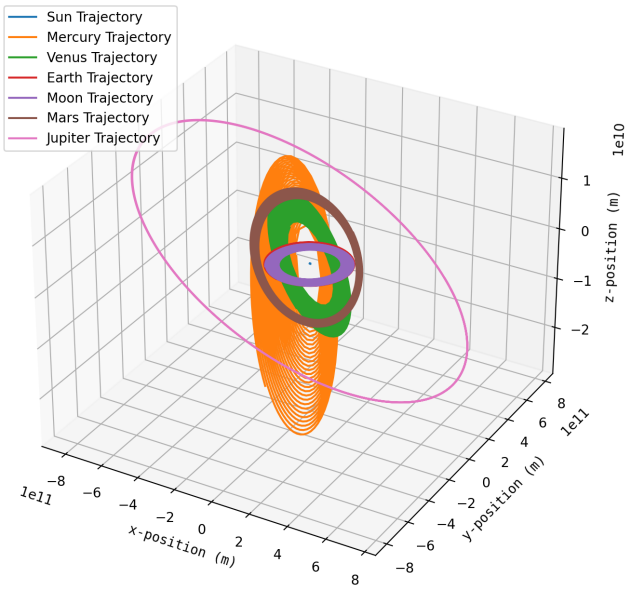

Figure 22: Larger version of figure 5.

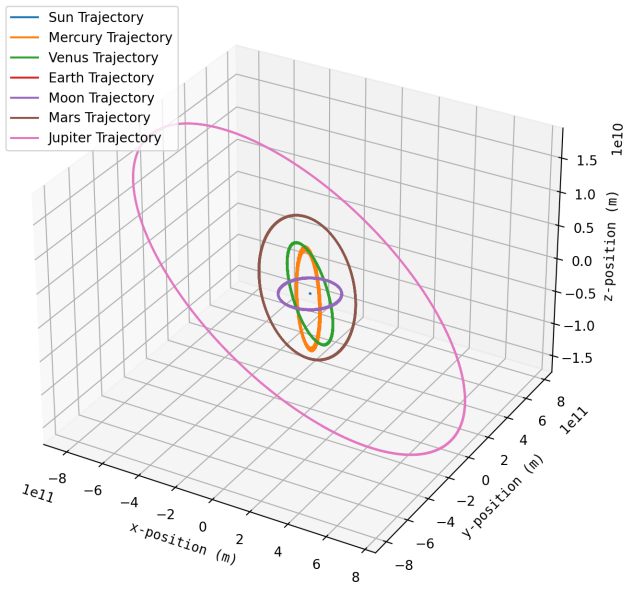

Figure 23: Larger version of figure 6.

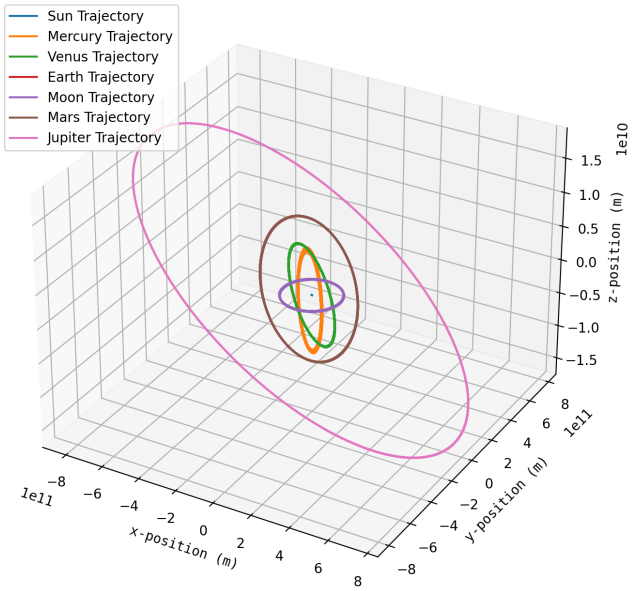

Figure 24: Larger version of figure 7.

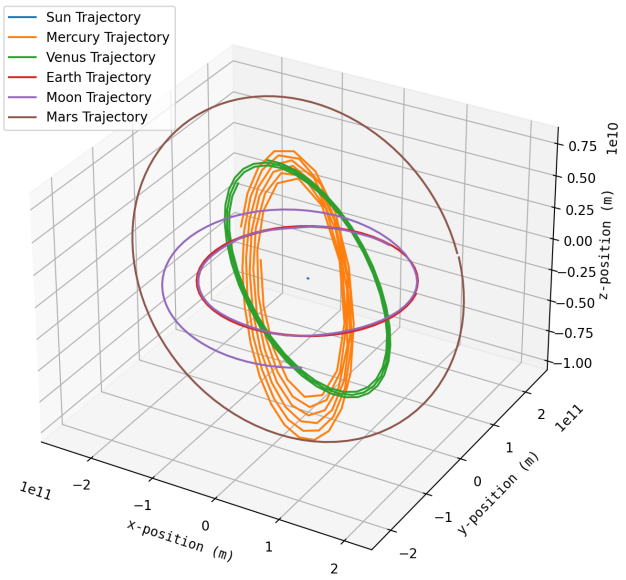

Figure 25: Larger version of figure 9.

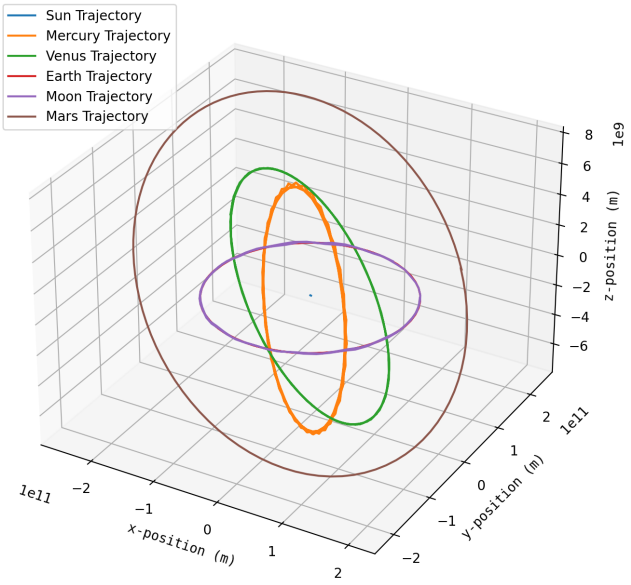

Figure 26: Larger version of figure 10.

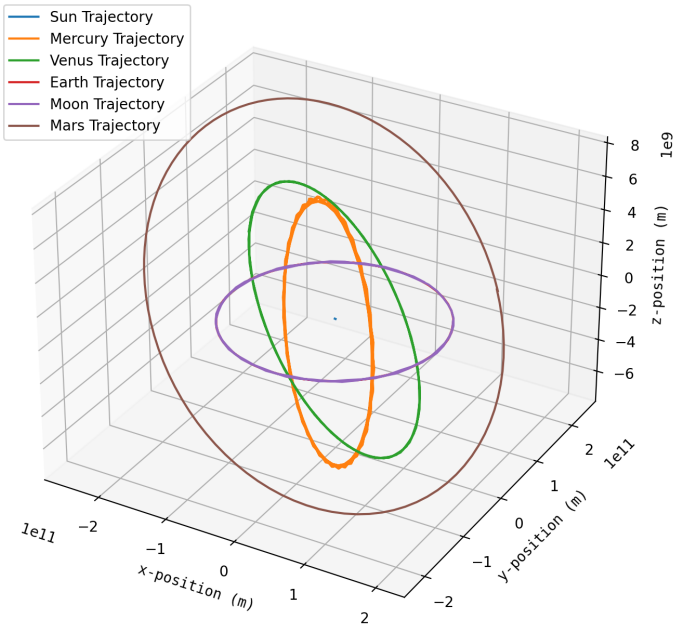

Figure 27: Larger version of figure 11.