Project 5

Anna Stray Rongve
Knut Magnus Aasrud
Amund Midtgard Raniseth

December 18, 2019

Abstract

This report aims to portray a convincing computational model of our solar system, based on basic Newtonian laws of gravitation. Key theory is explained, as well as our approach to making an object oriented simulation of all the solar system's bodies. Following this, we test important aspects of the model: the preferred integration method, the time step dependency, the computational efficiency, energy conservation, which origin to use and how our model handles relativity. The finalized model is convincing and gives satisfactory results on the tests mentioned above.

1 Introduction

The Velocity Verlet method is a widely used method for solving coupled ordinary differential equations. In this report, we will model our solar system's dynamics, utilizing said method. The equations to solve simply come from Newton's laws of motion in gravitational fields, although we will make a small modification down the road to account for relativistic effects and achieve greater accuracy.
Due to the sheer number of variables and methods required to calculate the motion of this many bodies, we will object orient our code - simplifying the process of adding bodies and making them interact with each other.
On our way to the final model of the solar system, we are going to explore the accuracy-differences between the Velocity Verlet and the Euler Forward method, as well as which timestep is needed for sufficient accuracy. The last check before we get going is to make sure that energy is conserved in the system. Thereafter we investigate different aspects of the solar system with our model. This includes what impact the massive planet Jupiter might have on the orbit of Earth, and at what speed the perihelion of Mercury precesses around the sun, when relativistic effects are accounted for.
The report has a theory part explaining the physical theory and the thought behind our computational implementation of it. Following this are our results and finally a discussion of them.

2 Theory

2.1 The Earth-Sun system

We begin by looking at the problem simply using the Euler Forward and the Velocity Verlet method. In two dimensions we have the following for the Earth-Sun system:

The gravitational force F_G:

$$F_G = \frac{GM_\odot M_E}{r^2}$$

where $M_E = 6 \times 10^{24}$ kg, $M_\odot = 2 \times 10^{30}$ kg, and $r = 1.5 \times 10^{11}$ m
The force acting on Earth is given by Newtons 2. law, here given in x- and y-direction

$$\frac{d^2x}{dt^2} = \frac{F_x}{M_E}, \quad \frac{d^2y}{dt^2} = \frac{F_y}{M_E}$$

By using the equalities $x = r\cos(\theta)$, $y = r\sin(\theta)$ and $r = \sqrt{x^2 + y^2}$ we obtain

$$F_x = -\frac{GM_\odot M_E}{r^2}\cos(\theta) = -\frac{GM_\odot M_E}{r^3}x$$

$$F_y = -\frac{GM_\odot M_E}{r^2}\sin(\theta) = -\frac{GM_\odot M_E}{r^3}y$$

This gives the following first order coupled differential equations:

$$\frac{dv_x}{dt} = -\frac{GM_\odot}{r^3}x, \qquad \frac{dv_y}{dt} = -\frac{GM_\odot}{r^3}y$$

$$\frac{dx}{dt} = v_x, \qquad \frac{dy}{dt} = v_y$$

In order to simplify we will use astronomical units (AU) defined by r, which is the avarage distance between Earth and the Sun. 1 AU $= r = 1.5 \times 10^{11}$ m

$$\frac{M_e v^2}{r} = F = \frac{GM_\odot M_E}{r^2}$$

Since $GM_\odot = v^2 r$, we get the velocity of Earth, assuming it moves in circular motion:

$$v = 2\pi \mathrm{r/years} = 2\pi \mathrm{AU/years}$$

Then we have the following relationship:

$$GM_\odot = v^2 r = 4\pi^2 \frac{(\mathrm{AU})^2}{\mathrm{years}^2}$$

With the mathematical relations established, we can discretize the equations for use in the code.

$$v_{x,i+1} = v_{x,i} - h\frac{4\pi^2}{r_i^3}x_i, \qquad x_{i+1} = x_i + hv_{x,i}$$

$$v_{y,i+1} = v_{xy,i} - h\frac{4\pi^2}{r_i^3}y_i, \qquad y_{i+1} = y_i + hv_{y,i}$$

2.2 The Verlet method

Another numerical method to be used to evalute the motion of planets in our Solar system is the Verlet method. This is a method that is pretty easy to implement, as well as giving stable results.

If we look at Newton's second law in the form of a second order differential equation in one dimension.

$$m\frac{d^2x}{dt^2} = F(x,t)$$

In coupled differential equations, we obtain

$$\frac{dx}{dt} = v(x,t), \qquad \frac{dv}{dt} = F(x,t)/m = a(x,t)$$

Using a Taylor expansion:

$$x(t,h) = x(t) + hc^{(1)}(t) + \frac{h^2}{2}x^{(2)}(t) + O(h^3).$$

From Newtons second law we already have obtained the second derivative, $x^{(2)}(t) = a(x,t)$.
Using Taylor for $x(t-h)$ and the discretized expressions $x(t_i, \pm h) = x_{i\pm1}$ and $x_i = x(t_i)$ we obtain

$$x_{i+1} = 2x_i - x_{i-1} + h^2x_i^{(2)} + O(h^4)$$

The orresponding velocity Taylor expansion is

$$v_{i+1} = v_i + hv_{(1)} + \frac{h^2}{2}v_i^{(2)} + O(h^3)$$

With Newtons second law:

$$v_i^{(1)} = \frac{d^2x}{dt^2} = \frac{F(x_i,t_i)}{m},$$

Adding the expansion of the derivative of the velocity

$$v_{i+1} = v_i + \frac{h}{2}\left(v_{i+1}^{(1)} + v_i^{(1)}\right) + O(h^3)$$

Since our error goes as $O(h^3)$ we only use the terms up to the second derivative of the velocity.

$$hv_i^{(2)} \approx v_{i+1}^{(1)} - v_i^{(1)}$$

Rewriting the Taylor expansions for the velocity:

$$x_{i+1} = x_i + hv_i + \frac{h^2}{2}v_i^{(1)} + O(h^3)$$

and

$$v_{i+1} = v_i + \frac{h}{2}\left(v_{i+1}^{(1)} + v_i^{(1)}\right) + O(h^3)$$

The implementation of the Verlet algorithm can be seen on line 54 in the non object oriented code: `/Earth-Sun_Verlet/main.cpp` or in the object oriented code: `/complete_solar-system/src/verlet.cpp`.

2.3 Code development and object orientation

In order to have full control over our implementation of these methods, we first implemented them in a non-object oriented code. This is found in these folders found on our github repository[1]: `/code/Earth-Sun_Verlet/` and `/code/Earth-Sun_Euler-FWD/`. When this was done, we were ready to move on to an object oriented approach which would be of great benefit when adding more planets to our system. The object oriented code is found in `/code/complete_solar-system/` and has most different functions split into its respective files located in `/complete_solar-system/src/`.
The gist of our approach is having a `Body` object containing the mass, position, name and velocity of a specific body, as well as methods to calculate the distance to other bodies and the gravitational acceleration due to the presence of other bodies.
In addition, we also have a `System` object that contains all of the system's bodies. This also has a `solve`-method that applies the specified algorithm on all the bodies based on its current acceleration (that is calculated firsthand).

2.4 Testing of the algorithm

Before inserting all the planets in the solar system, we would like to thoroughly test the simple Sun-Earth case. This is done by finding initial values for a perfectly circular orbit, and then test stability with different stepsizes and check for conservation of energy. We will also compare the performance of Eulers forward method to the Verlet method.

2.4.1 Initial values

When using astronomical units the radius between the Sun and Earth is quite easily set to $1AU$. The mass of the Sun is also set to 1, and the Earth mass is relative to this mass. For the orbit to be circular we set the centripetal

force equal and opposite to the gravitational force. For finding the velocity, the equations are formulated as the following:

$$\begin{aligned} F_g &= \frac{GM_\odot M_E}{r^2} \\ F_c &= \frac{M_E v^2}{r} \\ \frac{GM_\odot M_E}{r^2} &= \frac{M_E v^2}{r} \\ v &= \sqrt{\frac{GM_\odot}{r}} \end{aligned}$$

Where G is the gravitational constant commonly set to $4\pi^2$ for solar system computations. Since $M_\odot$ and r is 1, we get:

$$v = 2\pi \frac{AU}{yr} \tag{1}$$

Thus our initial value for the velocity of Earth should be 2π. This is achieved setting x-position to 0, x-velocity to 2π, y-position to 1 and y-velocity to 0.

2.4.2 Stability with varying timestep

Changing the timestep Δt is crucial for finding a good balance between calculation speed and accuracy. We simulated over a period of a millennium. The timesteps simulated was $\Delta t = \{0.01, 0.02, 0.05, 0.1\}$ years. The results are shown in section 3.2.1.

2.4.3 Energy and angular momentum conservation

As we have a circular orbit, we would expect the potential and kinetic energy to be conserved since the velocity is the same, and the distance from the sun should also be the same. These energies should be conserved since the only forces acting on the system is conservative forces, namely the gravitational force.

$$E_{tot} = E_k + E_p = \frac{1}{2} M_E v^2 + M_E \gamma r$$

Conservation of angular momentum is true if the system is not acted upon by a torque. Since the only force acting on the system is the gravitational pull of the sun, the angular momentum must be conserved. This can be shown by the following:

$$\begin{aligned} L &= I\omega \\ I &= r^2 m, \qquad \omega = \frac{v}{r} \\ L &= r^2 m \frac{v}{r} \\ L &= rmv \end{aligned}$$

Thus the angular momentum is conserved as long as the orbit velocity and radius is constant, which it is if the kinetic and potential energy is conserved.

2.4.4 Velocity Verlet versus Euler's forward algorithm

The last test is to check whether we want to use the velocity Verlet algorithm, or Eulers forward algorithm. As we've seen, in resuts section 3.1, the accuracy of our Verlet solver is superior. However it also takes more time. We tested our algorithms with 10^6 integration points, with and without saving the data to files. The results are shown in section 3.2.3.

2.5 Escape velocity

The escape velocity is the minimum veocity needed by an object to be projected to overcome the pull from the gravitational force in order to escape the gravitanonal field and the orbit. We will try to find this velocity by changig the initial conditions of Earth until it looks like it has escaped the gravitational field of the Sun. Then we will compare this to the analytical result found from this equation:

$$v_{esc} = \sqrt{\frac{2GM_{\odot}}{r}} \tag{2}$$

where G is the unversal gravitatonal constant, $4\pi^2$, $M_{\odot}$ is the mass of the sun and r is the distance between the Sun and Earth.

We will also look at what will happen if we let the gravitational force deviate from the original by changing the exponential of the distance:

$$F_G = \frac{GM_{\odot}M_E}{r^{\beta}}$$

with $\beta \in [2, 3]$, e.i. changing the exponetial from 2 towards 3 and study the difference. This is visualized in plot 5.

2.6 The three-body problem

In order to find out how much the planet with the greatest mass, Jupiter, alters the motion of the Earth, we will include the planet in our solar system.

This is done by simply adding the magnitude of the force between Earth and Jupiter,

$$F_{Earth-Jupiter} = \frac{GM_{Jupiter}}{r^2_{Earth-Jupiter}} \tag{3}$$

Where $M_{Jupiter}$ is the mass of Jupiter, and M_{Earth} is the mass of Earth, r is the distance between the two planets, and G is the gravitational constant.

The problem is solved by modifying the first order differential equations to accomodate the motion of Earth and Jupiter. The way we do this is to also consider the force from equation (3) when calculating the acceleration of Earth.

We will also study the effect of altering the mass of Jupiter by a factor of 10 and 1000, which gives Jupiter roughly the same mass as the Sun.

Finally we would like to check if the fixed-sun approximation was a good approximation. To check this, will see what changes when we let the sun also be a free body. The initial velocity of the sun is set such that the total momentum of the system is zero, and thus the center of mass is fixed.

2.7 All planets

After testing what timestep is needed for the Velocity Verlet algorithm, as well as testing our solver thoroughly for borth conservation of energy, and angular momentum, and figuring out if it is really necessary to let the sun move as a free body, we will run our computations with all planets of the solar system present. The initial velocities and positons are taken from NASA's webpage [2] which means that we are using the solar systems Barycenter as our origin. The result is shown in section 3.5.

2.8 The perihelion precession of Mercury

The observed perihelion precession of Mercury is 43 arc seconds per century, which translates to roughly 0.01194° per century. The perihelion of a planet means, in its simplicty, where it is at its closest position to the sun. Assuming a planar orbit of Mercury, we want to see how this perihelion position changes in regards to the Sun over time. This is what is called the perihelion precession, and is often measured in arc seconds per century, of which 3600 equals one degree per century.

As closed elliptical orbits is a special feature of the Newtonian $\frac{1}{r^2}$ force, we would assume that the perihelion precession of the Sun-Mercury scenario is zero when we only use this force. However, by adding a general relativistic correction to the Newtonian gravitational force, we would expect our computed perihelion precession of Mercury to be very close to the observed 43 arc seconds per century. This is a good test of the general theory of relativity. In this case we will look at the system with no other bodies than the Sun and Mercury itself.

The new gravitational force including the relativistic correction is as follows:

$$F_G = \frac{GM_{Sun}M_{Mercury}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right]$$

Where $l = |\vec{r} \times \vec{v}|$ is the absolute angular momentum of Mercury and c is the speed of light. The perihelion angle θ_p is given by

$$\tan\theta_p = \frac{y_p}{x_p}$$

Where y_p and x_p is the position of Mercury at perihelion. Our simulation will run over a century, where we choose our y_p and x_p values as the last time in the simulation where Mercury reaches perihelion. The speed of Mercury at perihelion is set to $12.44AU/yr$ and the distance to the Sun is set to $0.3075AU$ for our calculations.

3 Results

3.1 Euler and Verlet without object orientation

In order to make sure that our algorithm is running correctly, we started by solving the differential equation using both Euler's and Verlet's method without using object oriented(oo) code. The results are seen in Figure 1, calculated by the algorithms located in the following folders: `/code/Earth-Sun_Euler-FWD` and `/code/Earth-Sun_Verlet`.

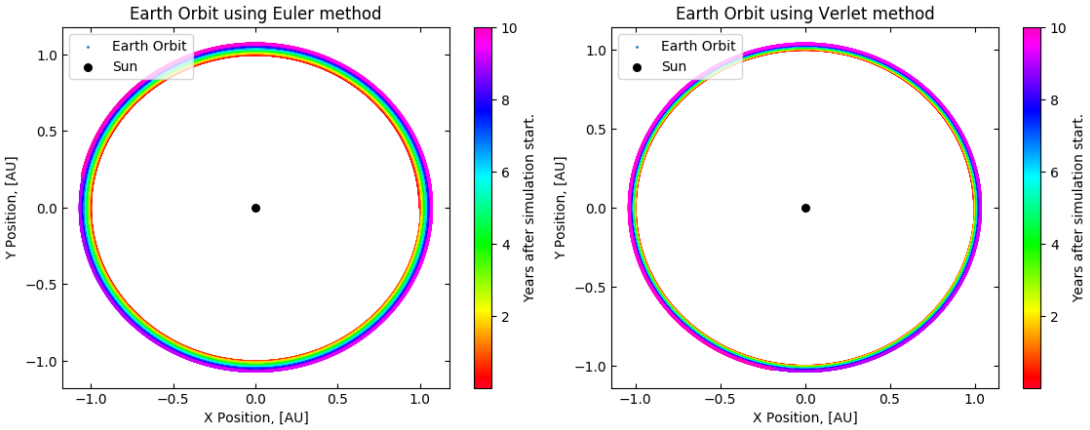

Figure 1: Earth orbit around the Sun using Eulers method and Verlets method respectively

3.2 Testing

3.2.1 Stability with varying timestep

The figures 2 and 3 show Earth's orbit over a thousand years, calculated with different timesteps.

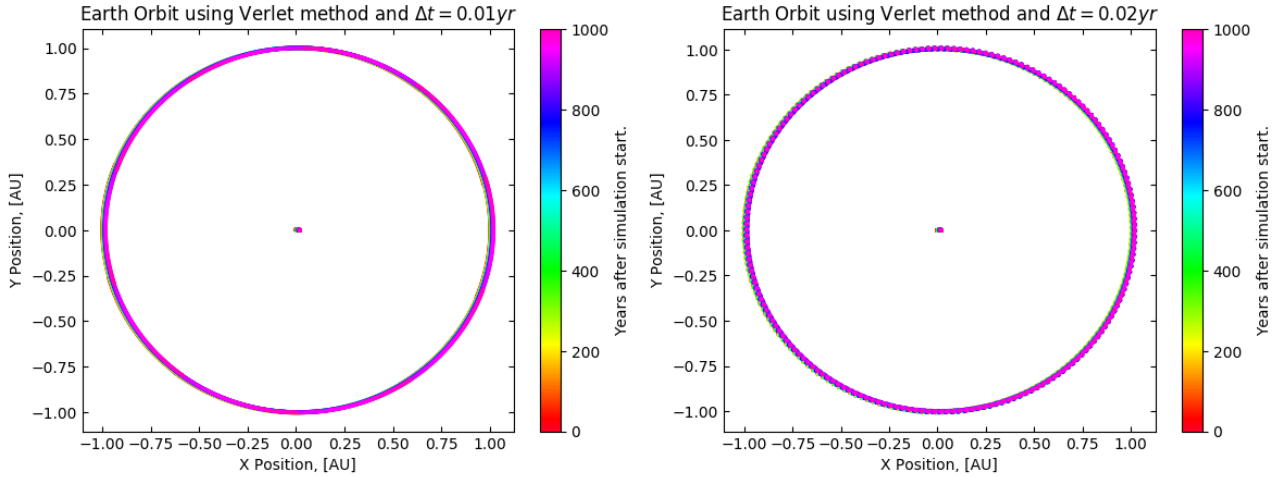

Figure 2: Earth orbit with time steps $\Delta t = 0.01$years and 0.02years respectively

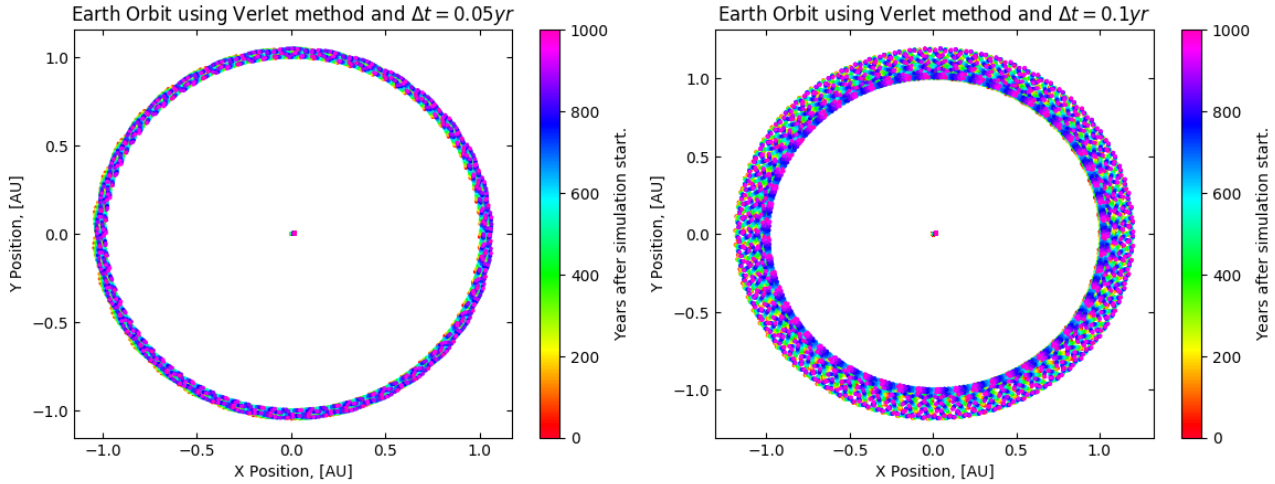

Figure 3: Earth orbit with time steps $\Delta t = 0.05$year and 0.1year respectively

3.2.2 Energy and angular momentum conservation

In figure 4 below, kinetic energy and potential energy is plotted as a function of time in the system. We chose to simulate over a millennium with a timestep of $\Delta t = 0.01$, as this timestep is sufficient.

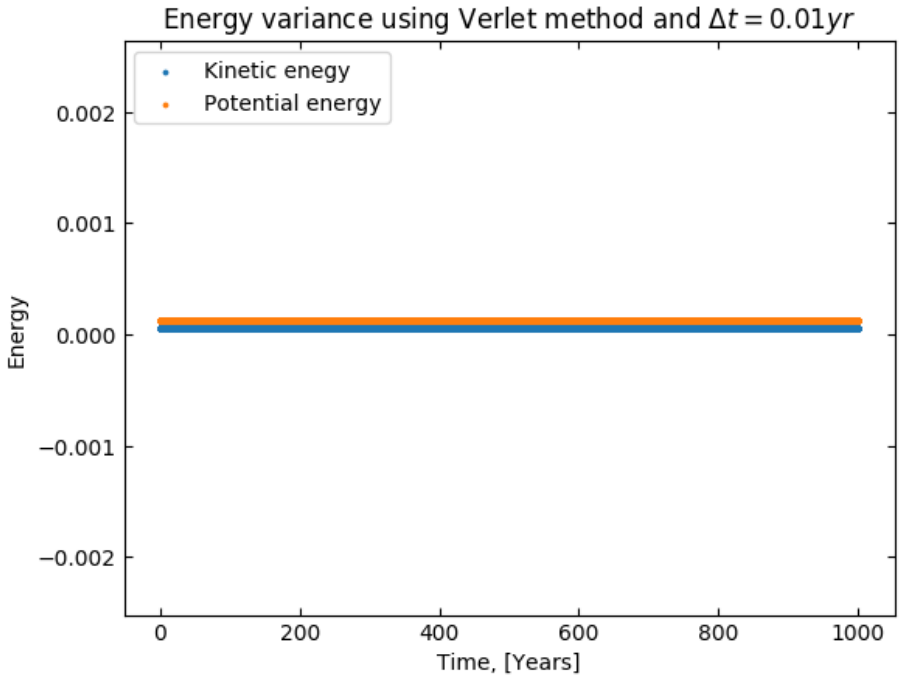

Figure 4: Kinetic and Potential energy with timestep $\Delta t = 0.01$year.

3.2.3 Verlet vs. Euler

Table 1 shows the efficiency of the different methods compared.

	Flops	**Time spent**
With filesave		
Euler's method:	10N	126 s
Verlet's method:	6N	119 s
Without filesave		
Euler's method:	10N	67 s
Verlet's method:	6N	71 s

Table 1: Comparison of flops and time for the Verlet and Euler method for 10^6 iterations over a period of 100 years simulating all planets

3.3 Escape velocity

By trial and error we found the escape velocity of planet Earth. We found the same value from section 2.5 using equation (2), which is the calculated value shown in table 2.

Table 2: Escape velocities of Earth, found by trail and error, and calculations respectively.

Experimental value:	2.828π
Calculated value:	2.8284π

We also looked at what happens when changing the exponent of the denominator the force of gravity from 2 towards 3 with initial velocity $v_{initial,x} = 2.2\pi$. This is shown in figure 5.

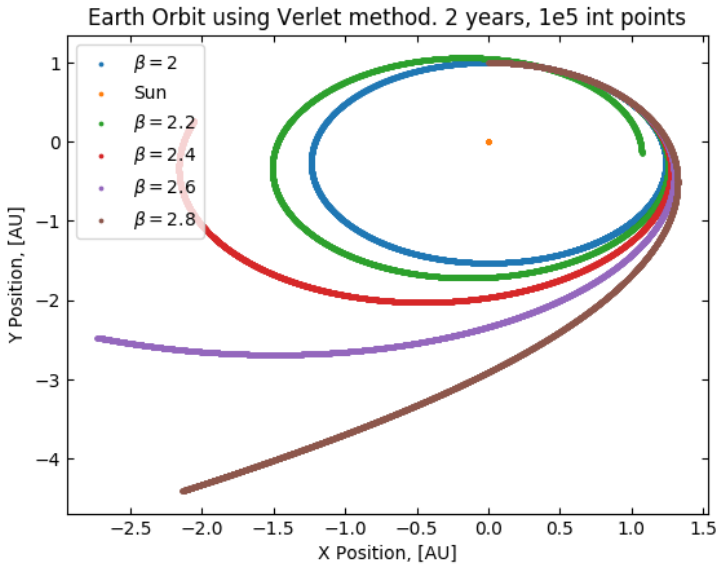

Figure 5: Escape velocity with increasing exponent of denominator of the gravitational force.

All this was found using the initializer found in `/src/initializer_test.cpp` inside our `complete_solar-system` code folder by changing the initial velocity of the Earth and the exponent of the distance r on line 40 in our body-code found in `/src/body.cpp`. Then the plotter found in `/Plotter/2d-plot_beta.py` was used to plot the figure above.

3.4 Three-body problem: Sun, Earth and Jupiter.

The first four plots below (figures 6 and 7) show the three-body simulation where the sun is fixed in the origin, with varying masses of Jupiter.

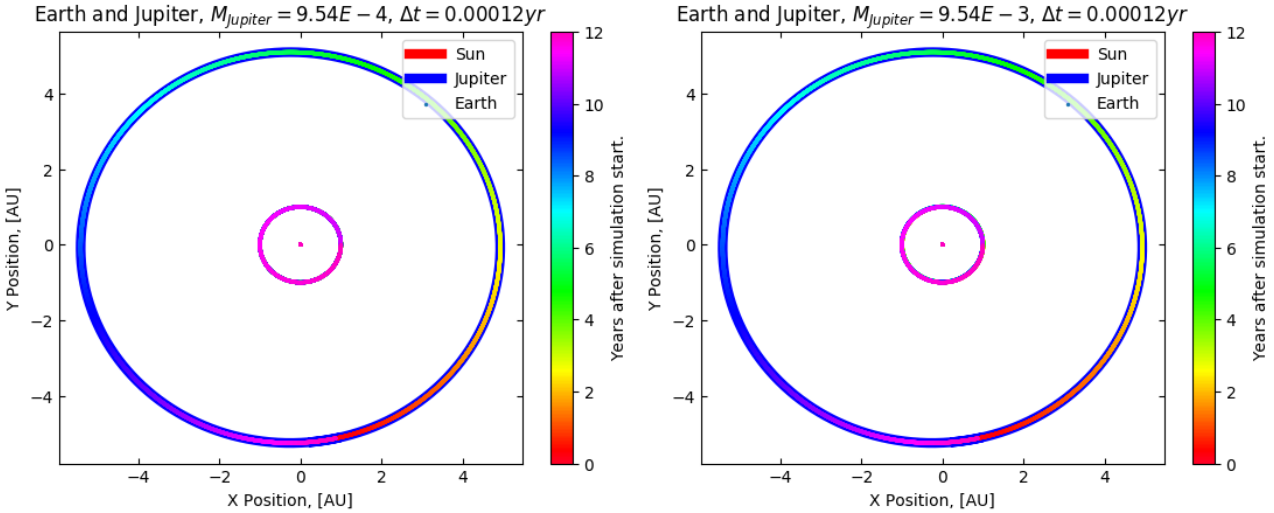

Figure 6: Positions of Sun in the middle, Earth second and outermost Jupiter, calculated using the velocity Verlet method with with original mass of Jupiter and an increase of mass of a factor of 10 respectively. The sun is in a fixed position.

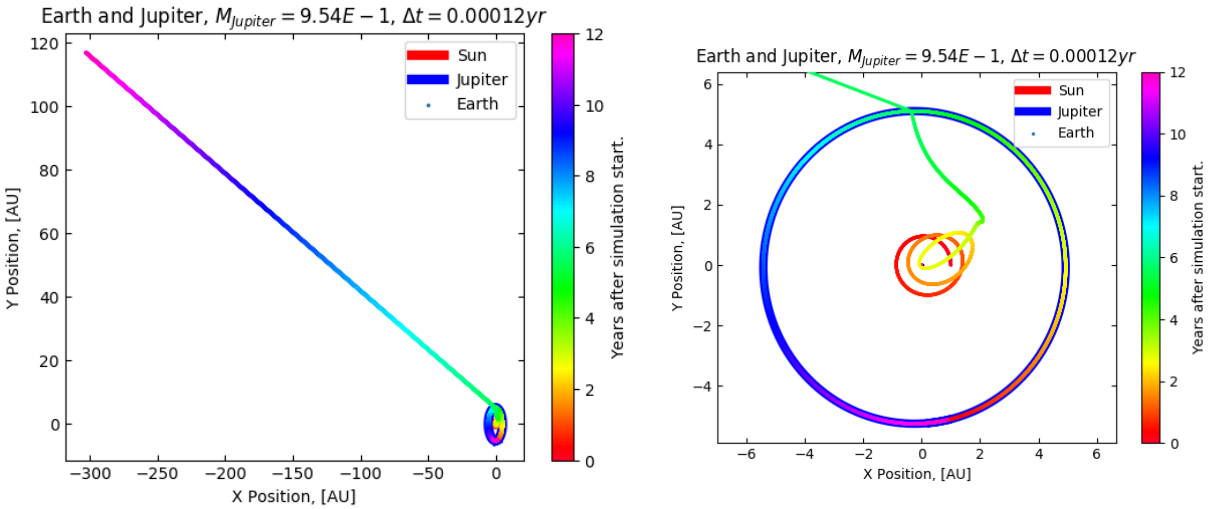

Figure 7: Positions of Earth and Jupiter using the velocity Verlet method with an increase of mass with a factor of 1000. After approximately 5 years, Earth escapes the system. The illustration to the right is a magnified version of the first plot and shows the three body interaction when the sun is fixed.

In order to test the fixed sun approximation, we also calculated this scenario for a dynamic Sun, shown below in figures 8 and 9.

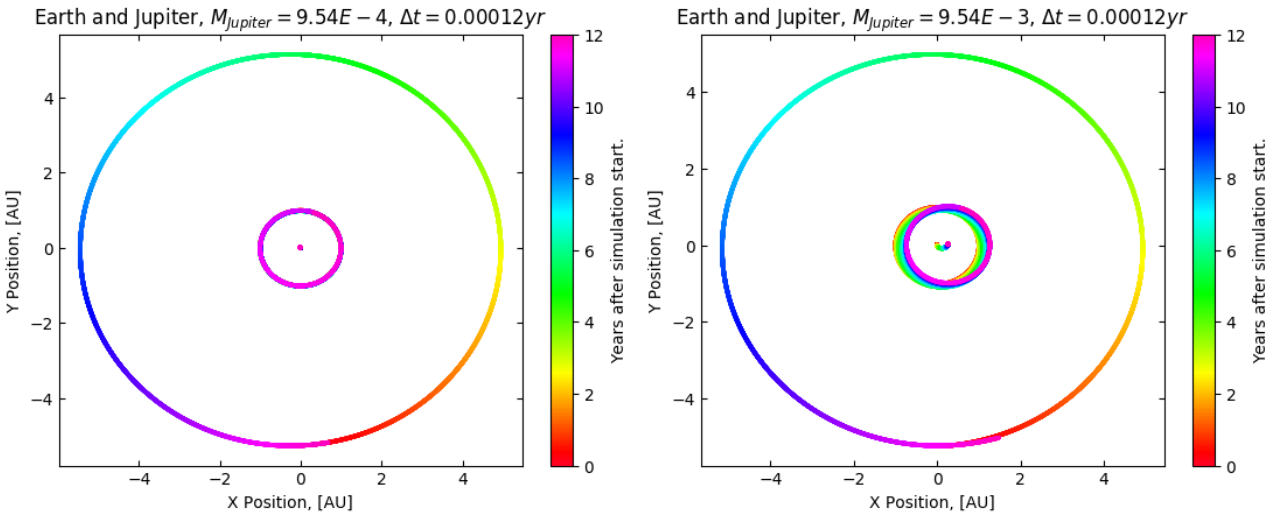

Figure 8: Positions of Sun in the middle, Earth second and outermost Jupiter, calculated using the velocity Verlet method with with original mass of Jupiter and an increase of mass of a factor of 10 respectively

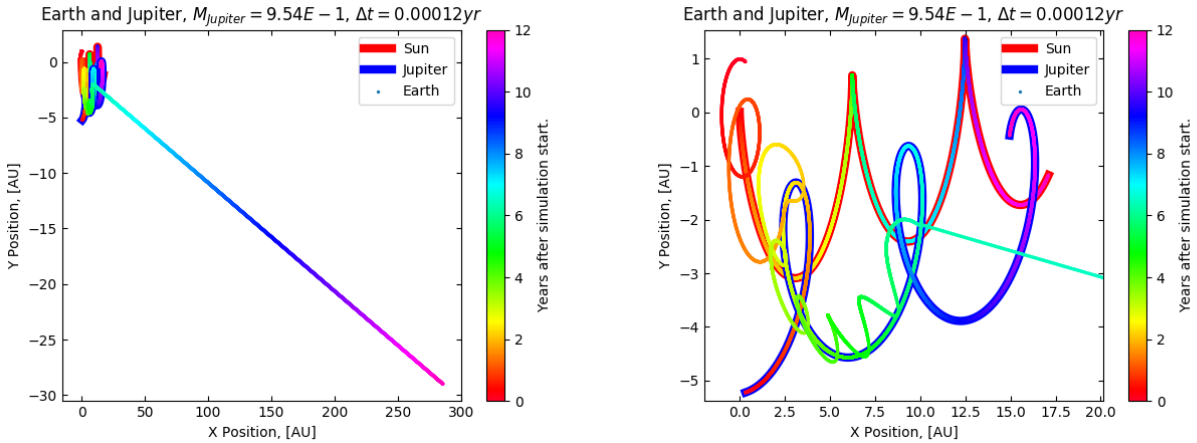

Figure 9: Positions of Earth and Jupiter using the velocity Verlet method with an increase of mass with a factor of 1000. After approximately 7 years, Earth escapes the system. The illustration to the right is a magnified version of the first plot and shows the beautiful interaction between two equally heavy objects.

3.5 The Solar system

In figure 10 we see the entire solar system plotted. The origin is set at the barycenter of the solar system, and all the initial conditions were taken from NASA's HORIZONS Web-interface (dated December 4th). [2]

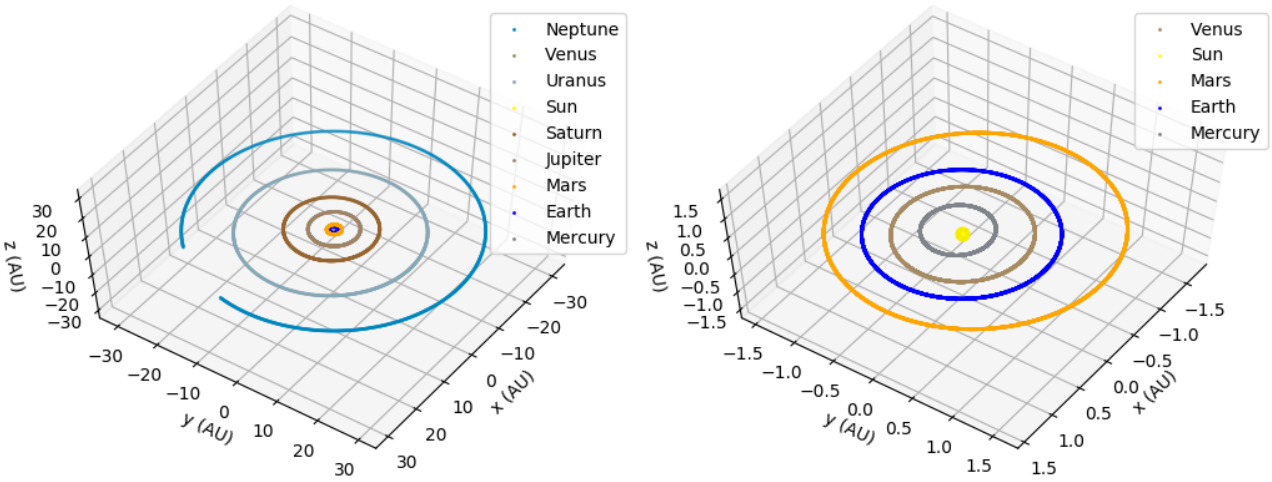

Figure 10: All planets in The Solar System.

3.6 The perihelion precession of Mercury

By using initial conditions of $2 \cdot 10^8$ integration points over a century, we are using a timestep $\Delta t = 5 \cdot 10^{-7} yr$. We also changed the code to only save the position of Mercury to file in the last year of simulation to save space. The initializer used is found in `/src/initializer_perihelion.cpp` inside our `complete_solar-system` code folder, and the results is found in table 3.

Table 3: Table of Mercury's calculated perihelion precession with and without relativistic correction.

	θ_p(Degrees)
Newtonian	$-0.000426°$
Relativistic	$0.0117318°$

4 Discussion

4.1 Euler and Verlet without object orientation

Comparing the Verlet and Euler method, as seen in figure 1, both methods work just fine. However, we can see that we have a larger error in the Euler method

than the Verlet method, since the orbit of Earth in the Euler plot is expanding faster.

4.2 Testing

As we can se from section 3.2.1, the Verlet method is quite stable. Even with a step length of 0.02 years, we see that the orbit for the first thousand years is still quite good. However, with larger steplengths, the calculations soon become unreliable.
From results 3.2.2 we see that both the kinetic and potential energy is constant over time, thus it is conserved. Since these values are constant, the angular momentum must also be constant, since it only depends on distance, speed and mass.
When comparing the performance between the Euler and Verlet algorithm in results section 3.2.3 it is not clear which is the best. This is because of all the other things going on in our program. For example we illustrated that saving the file to disk is what is both the biggest uncertainty, and the biggest time taker. This is because the write speed is dependent on how much the rest of the system wants to access the hard drive. Due to this uncertainty, the Euler method - running with writing to disk - actually took longer than the same Verlet run. We also see without filesave, that the time saved by the Euler algorithm is not substantial, and not worth the loss of accuracy. For this reason, the velocity Verlet algorithm is used in the rest of this project.

4.3 Escape velocity

As shown in section 3.3, the escape velocity found by trial and error on the initial conditions resulted in an incredibly good value compared to our analytical expression. This is yet another proof that our code is solid.
When modifying the exponent of the distance in the force formula, we get the expected result in figure 5. When β increases we would expect all objects with orbit radius greater than $1AU$ to feel a weaker force, which the trajectories in the mentioned figure clearly shows.

4.4 Three-body problem.

We see in figure 6 that after adding Jupiter, Earth's orbit is not altered much. Even with Jupiter ten times its mass, the orbit is virtually unchanged. However, increasing Jupiter's mass close to the Suns mass, we see that Earths orbit gets unstable, and slingshots out of the system after around 5 years. We also found that the fixed Sun approximation is quite good. This can be seen from figure 8, where we see that for its original mass, Jupiter does not distort the orbits in any impactful way. However we see that when Jupiter becomes 10 times its mass, the sun starts to change position, which in turns affect the Earths orbit also. We also plotted the scenario when Jupiter and the Sun has approximately the same mass, where all three bodies are free to move. This resulted in the

beautiful plot in figure 9, where we can see the Sun and Jupiter beautifully dancing around each other while the Earth violently moves between the two strong gravitational fields.

4.5 The Solar system

In figure 10 we see the solar system with all its planets orbiting beautifully over a period of 150 years. As we can see, Neptune still has 15 years to complete a full orbit, while most other planets has already completed several orbits. As a matter of fact, Mercury has completed a staggering 622 full orbits.
It is important to note that in the plot, the planet's sizes are obviously not to scale. However, the orbit sizes are to scale, and that lets us visually see why Neptune takes such a long time to complete an orbit, while Mercury is so quick.

4.6 The perihelion precession of Mercury

In order to test the general theory of relativity, we found the perihelion precession of Mercury with and without the relative correction. We found or non-corrected force to result in zero perihelion precession, which is expected, and we found the corrected force to result in a value very close to the observed perihelion precession. This means that our calculations, and the general theory of relativity holds true for this scenario, which means that the perihelion precession of Mercury can indeed be explained by the general theory of relativity.

5 Conclusion

Using Newtonian mechanics coupled with the Velocity Verlet integration method to simulate the solar system gives convincing results. Velocity Verlet being the preferred method is further backed by testing its efficiency against Euler Forward and it has the advantage of conserving the energy of the simulated solar system, as well as provably being stable. We've also found the model can handle addition of relativistic effects and consider this report successful.

6 Appendix

GitHub repository

References

[1] Amund Midtgard Raniseth, Anna Stray Rongve, Knut Magnus Aasrud. *Github repository*. 2019. URL: `https://github.com/kmaasrud/Project-5/`.

[2] NASA JPL Solar System Dynamics Group. *Horizons System*. 2017. URL: `https://ssd.jpl.nasa.gov/?horizons`.