FYS3150 Computational Physics – Project 3

Harald Moholt

October, 2016

Abstract

The objective of this project was to build a model for simulating the solar system using ordinary differential equations. Positions and velocities of celestial bodies were calculated by using Newton's gravitational law. Both Euler and Verlet was implemented and compared with each other regarding computation speed, FLOPS and relative error. It was found that the Verlet solver was superior to the Euler solver due to its high accuracy. This lead to a precise model of our solar system. The general theory of relativity was illustrated by showing the perihelion precession of Mercury.

GitHub repository with code can be found at: github.com/harmoh/FYS3150_Project_3

1 Introduction

The aim of this project is building a model for the solar system using ordinary differential equations. This involves finding the planets orbital positions over time. Positions and velocities will be found solving coupled ordinary differential equations, using both Euler and Verlet integration algorithms. Both algorithms will be written in code in order to simulate the system. In the beginning, a two-body system, only containing the Sun and the Earth, will be tested to see if the total energy and angular momentum of the system is conserved. The Sun will be set as the center of the solar system, and this system is co-planar, using the xy-plane. Then, Jupiter will be added to see its effect on the Earth's orbit, and see how this changes when the mass of Jupiter is increased.

Finally, the Sun and all planets (including Pluto) will be a part of the final model of the system in three dimensions. Data from NASA[1] will be used for initial positions and velocities of all included celestial bodies. When all planets are included, Solar System Barycenter will be defined as the center of mass in order to give all objects their correct positions and velocities. The perihelion precession of Mercury will also be calculated and simulated. Different scripts will be created to run and plot the different parts of the program.

This report is set up in the following way: Section 2 presents the theory relevant in this project, section 3 goes through the method and the algorithms used. Section 4 explains how the algorithms are implemented in the code and how the program is build up. Section 5 presents the results achieved from the program and shows plots of the different systems implemented. A discussion of the results are done in section 6 before a final conclusion is made in section 7.

[1] NASA website for generating data: http://ssd.jpl.nasa.gov/horizons.cgi

2 Theory

A central part of this project is calculating the forces between the celestial bodies. We use Newton's gravitational law, which is given by a force F_G

$$F_G = \frac{GM_{\text{Sun}}M_{\text{Earth}}}{r^2} \tag{1}$$

where G is the gravitational constant ($G = 6.67408 \cdot 10^{-11}$ m^3 kg^{-1} s^{-2}), M_{Sun} is the mass of the Sun, M_{Earth} is the mass of the Earth and r is the average distance between the Sun and the earth. Using Newton's second law of motion, we get the coupled equations

$$\frac{d^2x}{dt^2} = \frac{F_{G,x}}{M_{\text{Earth}}} \tag{2}$$

and

$$\frac{d^2y}{dt^2} = \frac{F_{G,y}}{M_{\text{Earth}}} \tag{3}$$

where $F_{G,x}$ and $F_{G,y}$ are the gravitational forces in direction x and y.

Table 1 shows the mass of celestial bodies and their distance from the Sun. In this project, we use astronomical units (AU) for convenience. 1 AU is the average distance between the Sun and the Earth, given as 1 AU $= 1.5 \cdot 10^{11}$ m. For time unit, years is used so that one orbit of the Earth around the Sun takes one year. This way, unit for positions are AU and velocities are given in AU/year.

Table 1: Mass of planets and distance from the Sun

Planet	Notation	Mass (kg)	Distance from the Sun
Sun	M_{Sun}	$2 \cdot 10^{30}$	–
Mercury	M_{Mercury}	$3.3 \cdot 10^{23}$	0.39 AU
Venus	M_{Venus}	$4.9 \cdot 10^{24}$	0.72 AU
Earth	M_{Earth}	$6 \cdot 10^{24}$	1.00 AU
Mars	M_{Mars}	$6.6 \cdot 10^{23}$	1.52 AU
Jupiter	M_{Jupiter}	$1.9 \cdot 10^{27}$	5.20 AU
Saturn	M_{Saturn}	$5.5 \cdot 10^{26}$	9.54 AU
Uranus	M_{Uranus}	$8.8 \cdot 10^{25}$	19.19 AU
Neptune	M_{Neptune}	$1.03 \cdot 10^{26}$	30.06 AU

This project also studies the perihelion precession of Mercury, where the Newtonian gravitational force is changed by adding a relativistic correction such that the corrected force becomes

$$F_G = \frac{GM_{\text{Sun}}M_{\text{Mercury}}}{r^2}\left(1 + \frac{3l^2}{r^2c^2}\right) \tag{4}$$

where c is the speed of light and l is Mercury's orbital angular momentum given by:

$$l = |\vec{r} \times \vec{v}| \tag{5}$$

Over time, the perihelion precession of Mercury is visible. The perihelion angle, given by

$$\tan\theta_p = \frac{y_p}{x_p} \tag{6}$$

changes by 43 arc seconds per century.

3 Method

3.1 Euler algorithm

For a circular motion, we know that the force can be written as

$$F_G = \frac{M_{\text{Earth}} v^2}{r} = \frac{G M_{\text{Sun}} M_{\text{Earth}}}{r^2} \tag{7}$$

where v is the velocity of the earth. We can then write

$$v^2 r = G M_{\text{Sun}} = \frac{4\pi^2 (\text{AU})^3}{(\text{year})^2} \tag{8}$$

We then use this and Euler's forward method in two dimension to get:

$$x_{i+1} = x_i + h v_{x,i} \tag{9}$$

$$v_{x,i+1} = v_{x,i} + h a_{x,i} \tag{10}$$

$$y_{i+1} = y_i + h v_{y,i} \tag{11}$$

$$v_{y,i+1} = v_{y,i} + h a_{y,i} \tag{12}$$

The acceleration is given as

$$a = \frac{F_G}{M_{\text{Earth}}} \tag{13}$$

where

$$F_G = \frac{G M_{\text{Sun}} M_{\text{Earth}}}{r^2} \frac{\hat{r}}{r} \tag{14}$$

where G is a constant that we set to

$$G = 4\pi^2 \tag{15}$$

and the mass of the Sun and the Earth can be changed to the two opposing bodies when calculating the force. The step length Δt is defined as:

$$\Delta t = \frac{t_{final} - t_{initial}}{n} \tag{16}$$

When including the third dimension, another set of coupled equations have to be added. The example above only focuses on the two body system between the Sun and the Earth. When adding several bodies in the system, the force between the different bodies are still the same, but r is then the distance between the two celestial bodies.

3.2 Verlet algorithm

The Verlet method uses the same acceleration and can be written (in two dimensions):

$$x_{i+1} = x_i + h v_{x,i} + \frac{h^2}{2} a_{x,i} \tag{17}$$

$$v_{x,i+1} = v_{x,i} + \frac{h}{2}(a_{x,i+1} + a_{x,i}) \tag{18}$$

$$y_{i+1} = y_i + h v_{y,i} + \frac{h^2}{2} a_{y,i} \tag{19}$$

$$v_{y,i+1} = v_{y,i} + \frac{h}{2}(a_{y,i+1} + a_{y,i}) \tag{20}$$

The implemtation of the Verlet method i shown below. The system forces are calculated before iterating through the bodies and calculating positions and velocities. Forces are updated before finding the velocities in order to get the force for the next step.

```
1     system.calculateForcesAndEnergy();
2     for(CelestialBody &body : system.bodies())
3     {
4         body.acceleration = body.force / body.mass;
5         body.position += body.velocity * m_dt + body.acceleration * m_dt*m_dt / 2;
6         system.calculateForcesAndEnergy();
7         body.velocity += (body.force / body.mass + body.acceleration) * m_dt / 2;
8     }
```

3.3 Escape velocity

When calculating the velocity an object need in order to escape from the gravitational pull of another object, energy conservation has to be studied. Kinetic energy is defined as:

$$E_k = \frac{1}{2}mv^2 \tag{21}$$

where m is mass of the object and v is the velocity. The potential energy needed can be written as:

$$E_p = -\frac{GMm}{r} \tag{22}$$

where G is Newton's gravitational constant, M is the mass of the larger object with gravitational pull, m is tha mass of the smaller object in question and r is the distance between the two masses' center. In order for the object to escape the gravitational pull, the energy can not be conserved, leading to $E_k + E_p = 0$:

$$\frac{1}{2}mv^2 - \frac{GMm}{r} = 0 \tag{23}$$

This can be rewritten as:

$$v = \sqrt{\frac{2GM}{r}} \tag{24}$$

where v is the needed velocity to escape out of orbit. In our case with the Sun and the Earth, we get:

$$v = \sqrt{\frac{2 \cdot 6.67 \cdot 10^{-11} \cdot 2 \cdot 10^{30}}{1.5 \cdot 10^{11}}} \text{ m/s} = 42.174 \ km/s \approx 8.9 \ AU/year \tag{25}$$

4 Implementation

This project is written in C++ and Python. C++ includes the algorithms and the code necessary for calculating the orbits of all celestial bodies and functionality for writing to file. The part of this project written in Python reads the produced text files and plots them in two dimensions. The program includes four different parts:

1. Sun and Earth system
2. Sun, Earth and Jupiter
3. The whole solar system (including Pluto, excluding moons)
4. Sun and Mercury

Each part includes its own Python file which runs the program and produce the corresponding plots. The C++ program is the same for the different parts, but is modified by input arguments "1", "2", "3", or "4", respectively. The Python scripts use the correct input arguments. The C++ program can be modified by changing the variables `t_final` for final time (in years) or `totalSteps` for the number of integration points. Only the third part uses data from NASA for initial positions and velocities.

The program starts with creating an instance of the class `SolarSystem`, which contain energy, forces and all celestial bodies, like the Sun and the Earth. Each body is a new instance of the class `CelestialBody`, which includes the bodies name, position, velocity, mass, etc. Name, initial position, velocity and mass are set when the bodies are created. Positions and velocities are created using the class `vec3`, which is a vector class that contains three elements for x, y and z coordinates. The program uses either the Euler method or the Verlet method. They are each their own class with the algorithm implemented. A loop in `main.cpp` runs the integrators a number of times equal to `totalSteps`. The loop also writes to file using two methods, one creating `.txt` files for Python and one creating `.xyz` files compatible with Ovito for animations of the solar system. The C++ program produces output comments regarding different variables and properties.

5 Result

5.1 Sun and Earth

To start with, a system only containing the Sun and the Earth was created. The Earth's orbit around the Sun was calculated using two different methods, Euler and Verlet, as discussed in section 3. Table 2 shows computation time and error (for the Earth given in distance from coordinates (1,0) in AU) for the Euler and Verlet integration methods. The computation times are calculated from the average of three tests (except where Δt = 1e-07, for obvious reasons). A pattern is visible here; computation time increases proportionally with the number of integrations. These calculations are done for the system only containing the Sun and the earth. Calculations for computation time included writing to file for plotting. Computation time without writing to file was approximately 1/9 of the time (e.g. $\sim$ 10 seconds for n = 1e-07).

Table 2: Error and time comparison between Euler and Verlet

n	Δt	Error (Euler)	Error (Verlet)	Time (Euler)	Time (Verlet)
1e01	1e-01	4.18692e+0	4.72901e-03	0.00050 s	0.00037 s
1e02	1e-02	7.16590e-01	4.39566e-07	0.00131 s	0.00107 s
1e03	1e-03	7.68826e-02	5.85470e-08	0.00821 s	0.00924 s
1e04	1e-04	7.86856e-03	6.79437e-09	0.08568 s	0.09441 s
1e05	1e-05	7.89233e-04	1.48026e-09	0.85293 s	0.93869 s
1e06	1e-06	7.89485e-05	9.47389e-10	8.55050 s	9.46985 s
1e07	1e-07	7.89590e-06	8.93766e-10	90.13670 s	93.28570 s

The Euler method is highly inaccurate at high Δt, accuracy increases with decreasing step length Δt. The Verlet method is very accurate for step lengths of $\Delta t \leq 0.01$. The difference in computation time between the methods is not significant, the Verlet method uses slightly more computation time. This is due to an extra calculation of the forces in the integration. Figures 1 and 2 shows

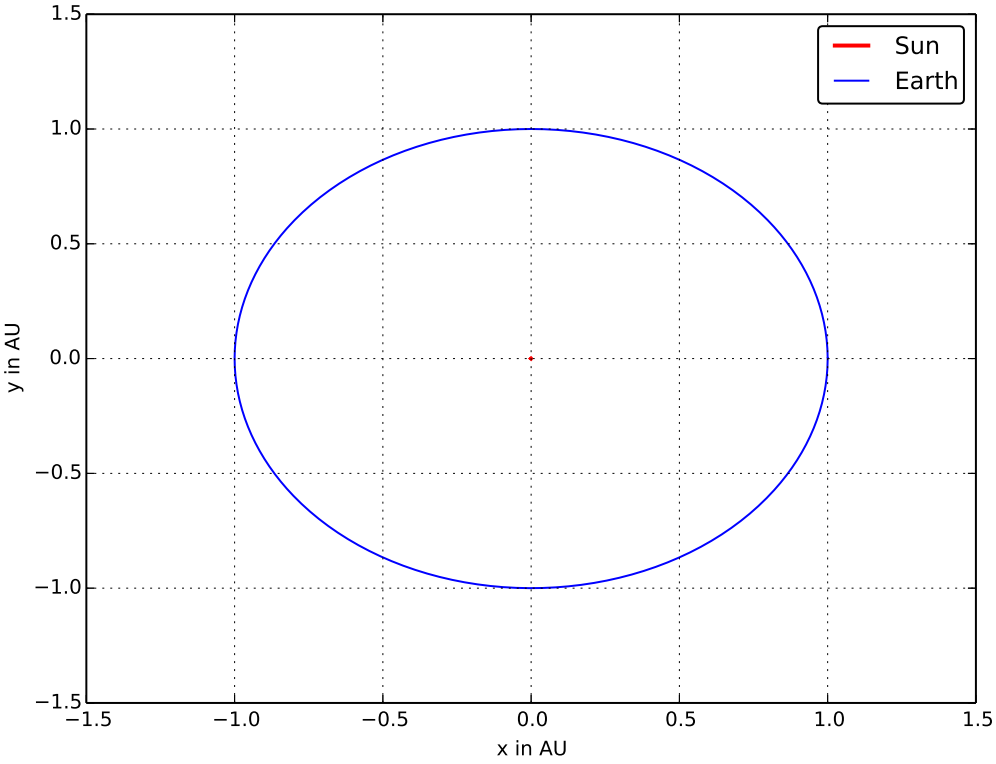

Figure 1: Sun-Earth system using Verlet when $\Delta t = 10^{-3}$ and final time is 1 year

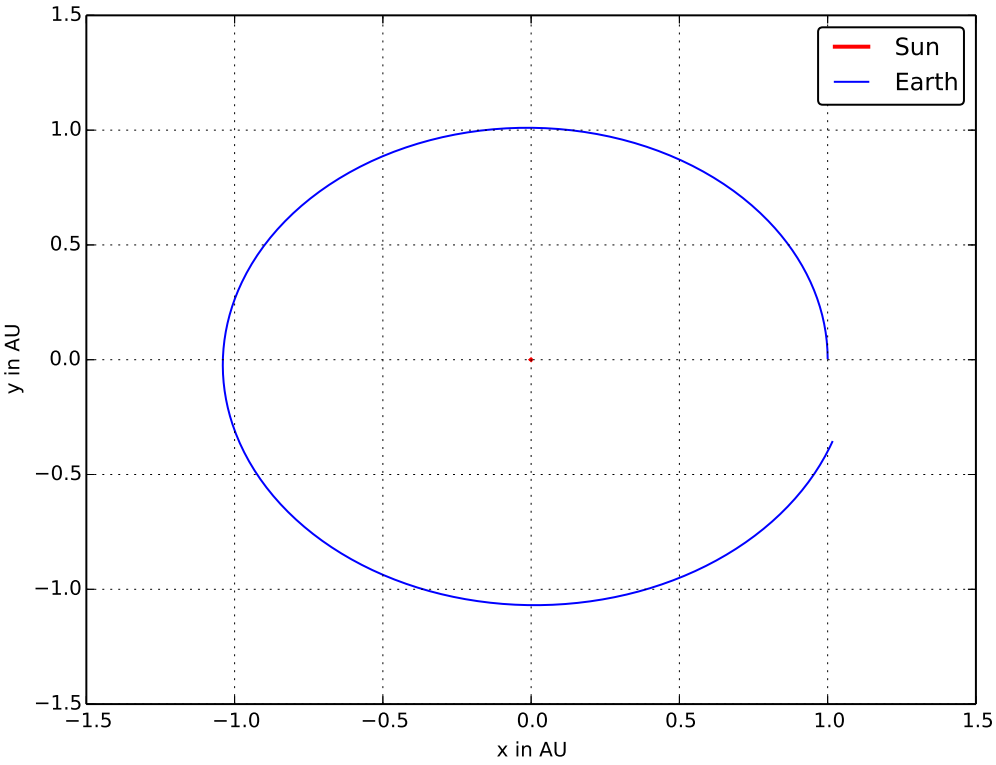

Figure 2: Sun-Earth system using Euler when $\Delta t = 10^{-3}$ and final time is 1 year

the difference in accuracy between the methods when $\Delta t = 10^{-3}$ and final time is set to one year.

If the of energy of a system is conserved, it means that the system is indeed closed, meaning that there is no external forces acting on the system or that the system does not affect anything outside of that system. If the energy is conserved, then the forces between the Sun and Earth are contained and the Earth will continue to orbit the Sun into infinity. When calculating the total energy for the system including the Sun and the Earth during 1 year, the result is $-5.9\text{e-}05$, which is not zero, but not far from it either. The angular momentum is $[0,\ 0,\ -1.56\text{e-}14]$, which is approximately zero.

The initial velocity of the Earth in the system with Sun and the Earth was increased in order to find an escape velocity by experimentation. This would need integration over a very long period of time (tested with 1000 years) with many integration points (tested with $n = 10\text{e}07$). After some trial and error, the escape velocity was found to be somewhere between 8.86 AU/year and 8.92 AU/year. As seen in section 3, the calculated escape velocity for the Earth out of the Sun's gravitational pull is 8.9 AU/year. In other words, an approximation of the escape velocity can be found, but higher accuracy needs a very high final time ($t_{final} \geq 10000$) in order to observe the effect and a small step length ($\Delta t \leq 1\text{e-}05$) for an accurate calculation. This leads to a large number of integrations ($n \geq 1\text{e+}09$), which can easily take half an hour or more.

5.2 Three body system with Jupiter

A three body system was simulated by adding Jupiter to the existing system containing the Sun and the Earth. Figure 3 shows the three body simulation with the Sun, the Earth and Jupiter over a duration of 11.9 years (duration for Jupiter to orbit the Sun once).

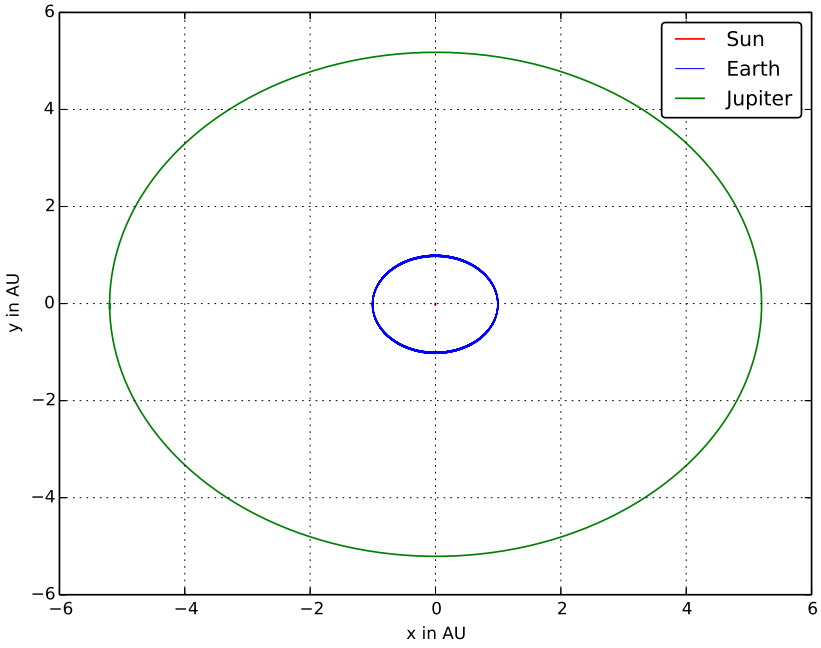

Figure 3: Solar system including only the Sun, the Earth and Jupiter with final time of 11.9 years and a step length $\Delta t = 1.19\text{e-}06$

This shows how the Sun and the Earth react when Jupiter has its actual mass. As seen, the Sun and the Earth's orbit stays the same (as far as visible from the figure). All plots of the three body system is produced using a final time of 11.9 years with a step length of Δt = 1.19e-05. However, increasing the mass of Jupiter will increase the forces acting on both the Sun and the Earth. Figure 4 shows the system when the mass of Jupiter is increased by a factor of 10, which is approximately 100 times smaller than the Sun. It is now visible that the Sun is following Jupiter's orbit by moving in a circular motion. The Earth is still orbiting the Sun and is now following the Sun.

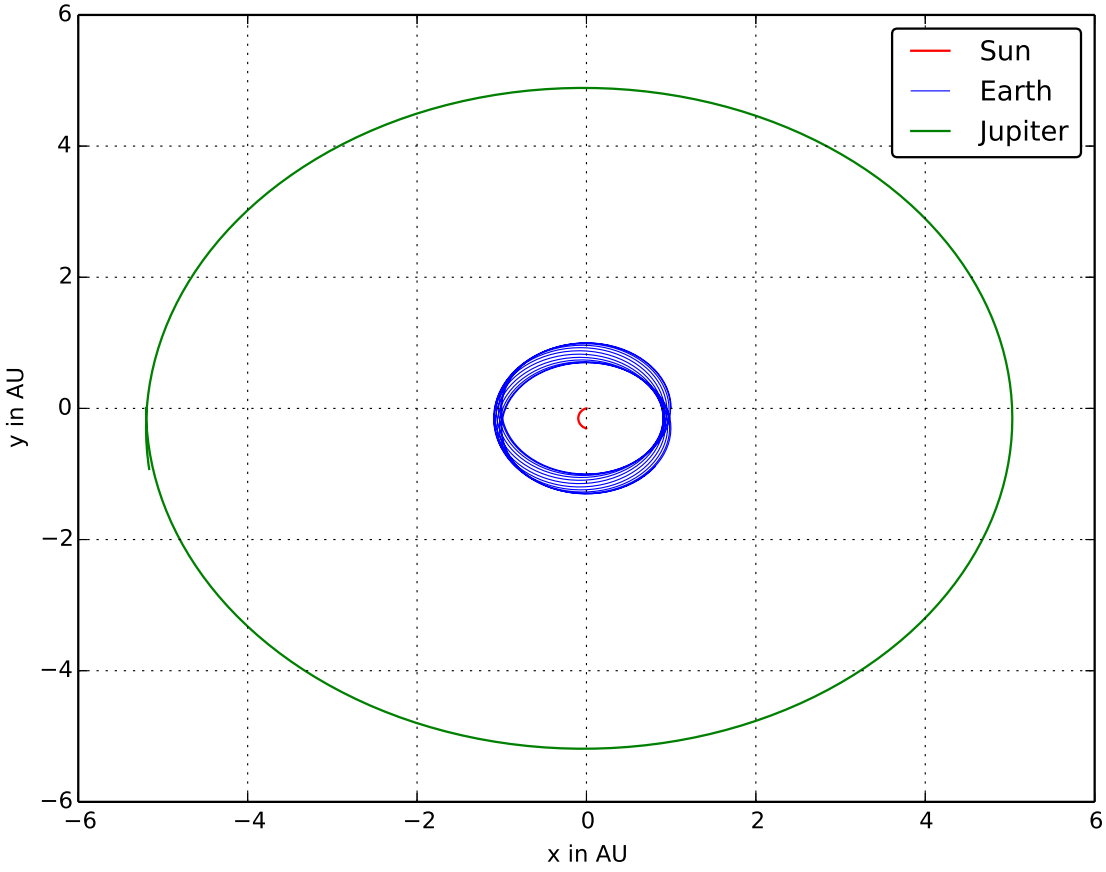

Figure 4: System including the Sun, the Earth and Jupiter where the mass of Jupiter increased by a factor of 10 with a final time of 11.9 years and a step length Δt = 1.19e-06

Figure 5 shows the system when the mass of Jupiter is increased by a factor of 1000, which is approximately the same as the mass of the Sun. This results in the Sun and Jupiter orbiting around each other while the Earth is unstably orbiting around the Sun before getting shoot out of orbit. The stability of the Verlet solver may be challenged here, as different step lengths lead to widely different results regarding the orbit of the Earth. Like the other simulations, low enough step length (Δt = 1e-06) leads to higher accuracy, which in this case may be essential in order to get the correct orbit of the celestial bodies.

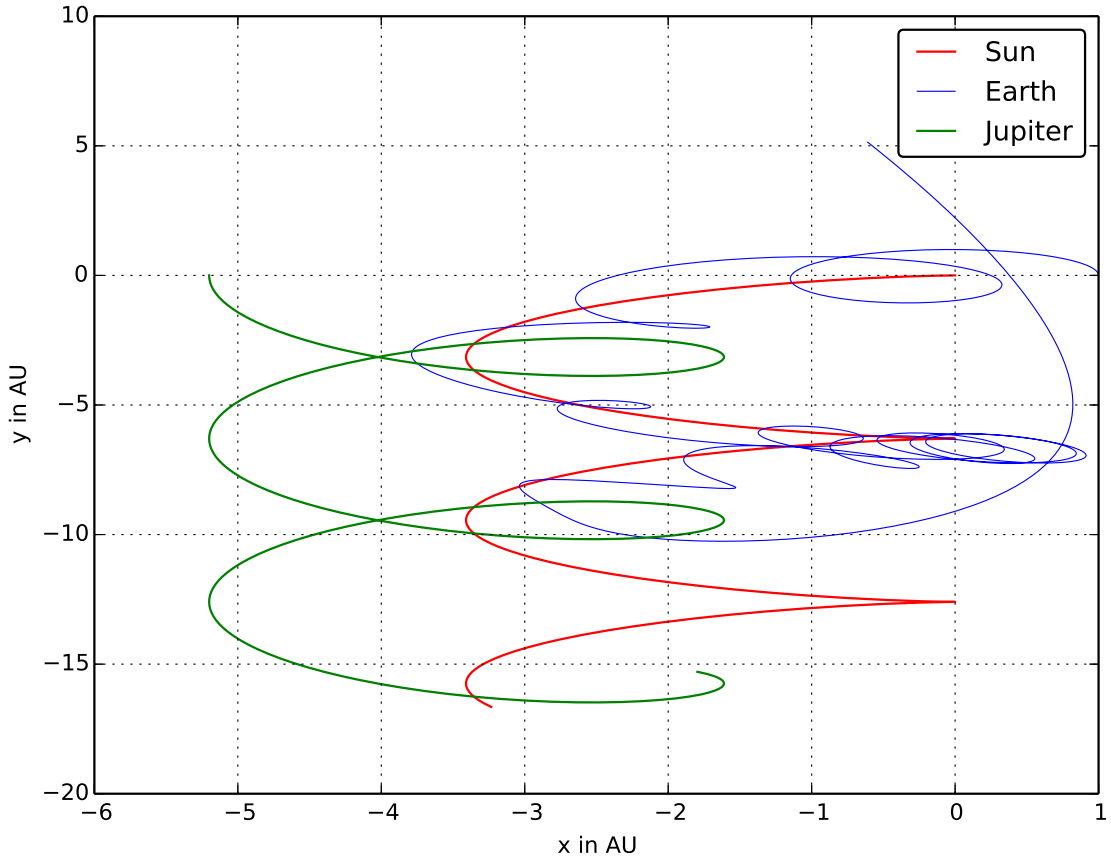

Figure 5: System including the Sun, the Earth and Jupiter where the mass of Jupiter increased by a factor of 1000 with a final time of 11.9 years and a step length Δt = 1.19e-06

5.3 Entire solar system

Now, all the planets (including Pluto), is simulated in the system. Initial positions and velocity from NASA's website is used for an actual simulation of our solar system. Figure 6 shows the entire system over at period of 250 years with a step length of Δt = 2.5e-04. Final time of 250 years was chosen to get an entire orbit of Pluto, which lasts equivalent to almost 250 years on Earth. In order to achieve this, the step length ended up being low. This affected the accuracy of the whole system and is especially visible for the inner planets that have a low orbital time. Although only two dimensions are shown in the figures, three dimensions were used when calculating the positions and velocities. Small changes can be made to the plotting part of the program in order to show the system in three dimensions. Alternatively can the `.xyz` be imported into Ovito for an animation in three dimensions over time.

When using data from NASA, the Solar System Barycenter was chosen as the center of mass. This allowed also the Sun to get its correct initial position (which is not the mass-center of the solar system). The center of mass is also now in origin of the coordinate system. Figure 8 shows the Sun's position over 250 years with a step length of Δt = 2.5e-04 when simulating the entire solar system. Notice the scale. Even though the Sun's position changes a lot, it is always within 0.01 AU of the center of mass of the solar system.

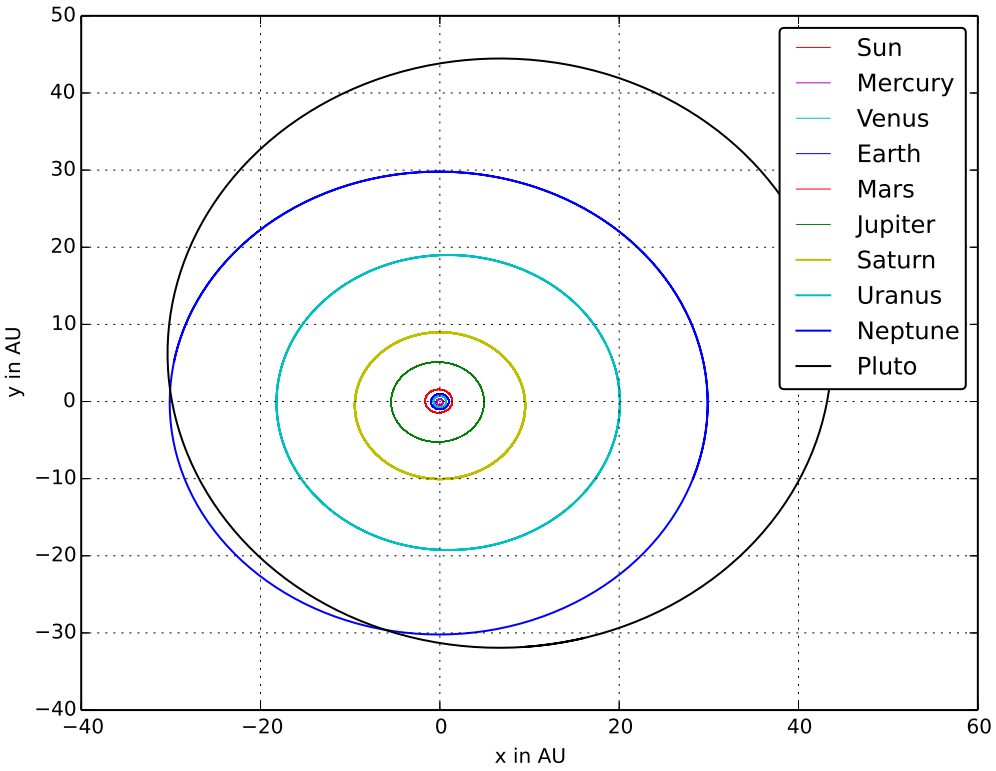

Figure 6: Solar system with final time of 250 years and Δt = 2.5e-05

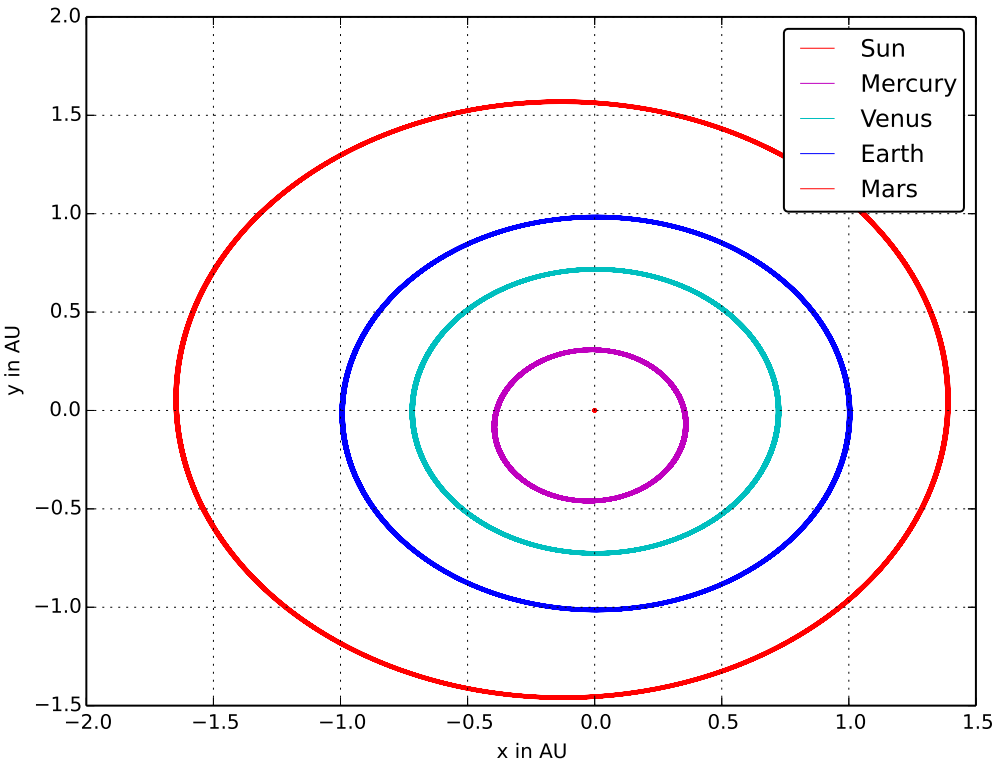

Figure 7: Inner part of the solar system with final time of 250 years and Δt = 2.5e-05

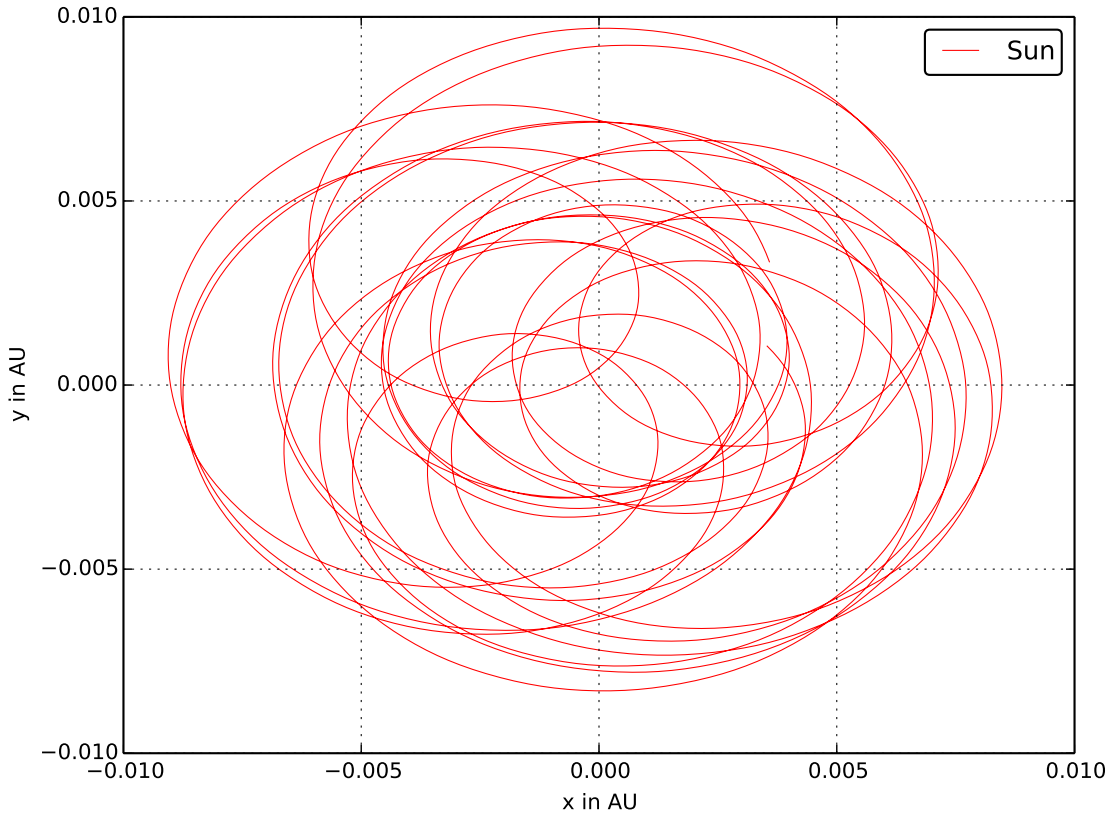

Figure 8: Solar system, only displaying the Sun, final time of 250 years and Δt = 2.5e-05

5.4 Perihelion precession of Mercury

The final part of this project involved studying the perihelion precession of Mercury. Here, we use the Newtonian gravitational force with relativistic correction presented in section 2. In order to show the effect of this correction, the perihelion angle over one century was calculated both with and without relativistic gravity.

Figure 9 shows the perihelion angle of Mercury without relativistic gravity over 100 years with a step length of Δt = 1e-07. The perihelion angle changes between -0.4 and 0.4 arc seconds, but does not exceed those limits. This indicates that the perihelion angle does not change over time.

Figure 10 shows the perihelion angle of Mercury with relativistic gravity over 100 years with a step length of Δt = 1e-07. This shows a clear effect of the relativistic correction in Newtonian's gravitational force. The perihelion angle increases linearly from 0 to approximately 43 arc seconds during one century. The exact numbers are 42.905 arc seconds after 99.929 years (taken from the printed text file), which is very accurate compared to the 43 arc seconds per century presented in section 2.

In order to achieve this accuracy, the step length has to be set low enough ($\Delta t \leq$ 1e-07). To be able to see the effect of the perihelion precession at all, the step length needs to be $\Delta t \leq$ 1e-06.

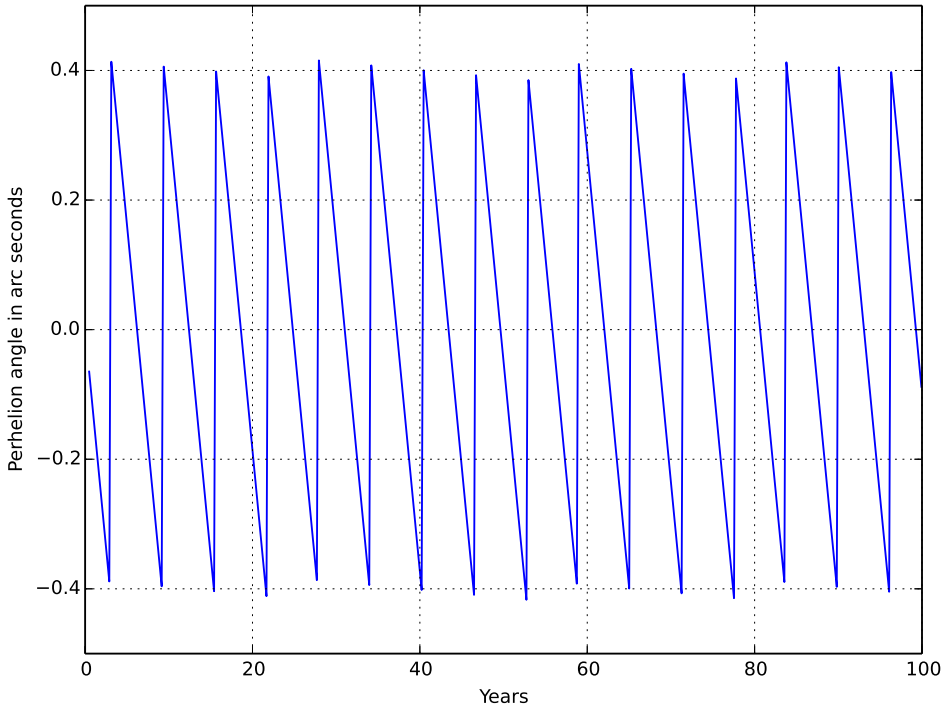

Figure 9: Perihelion angle of Mercury without relativistic gravity over a period of 100 years with a step length of Δt = 1e-07

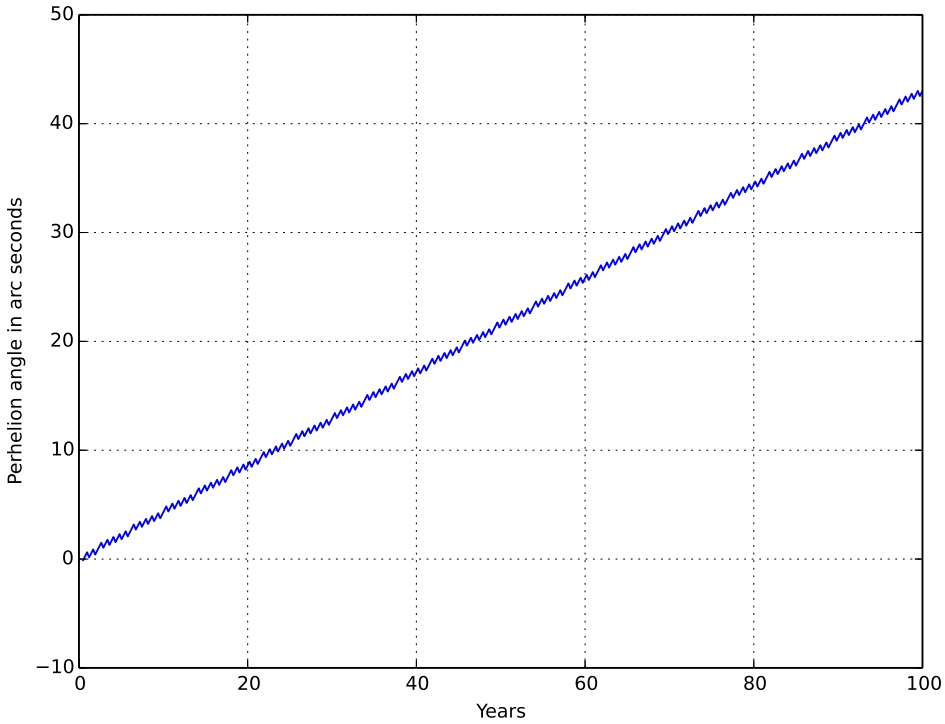

Figure 10: Perihelion angle of Mercury with relativistic gravity over a period of 100 years with a step length of Δt = 1e-07

6 Discussion

The system containing only the Sun and the Earth was used to test the benefits of the different algorithms. The forward Euler solver showed slightly shorter computations time than the Verlet solver. This is due to the number of FLOPS performed by the algorithms. Since the Verlet uses acceleration for both the position and the velocity, this leads to a few extra calculations for every time step, which was visible by up to 10 % of extra computation time (seen in table 2).

The Verlet solver showed significantly higher accuracy than the Euler solver, especially with few integration points (seen in figs. 1 and 2). The reason for this is that is does not only include the acceleration when finding the next position, but also takes the acceleration for both the next and the current step when finding the velocity. This makes it very accurate for step lengths up to $\Delta t =$ 1e-02. The high accuracy for this usage is the reason why the Verlet was chosen for the rest of this project.

7 Conclusion

This project has studied the solar system and created a model for simulating it using ordinary differential equations. The aim was to develop a code for implementing both the Euler and the Verlet algorithms. The simulation started with a two body system containing the Sun and the Earth and was expanded with Jupiter and then the rest of the planets (including Pluto) to simulate our entire solar system (excluding moons, satellites, meteors, asteroids and other minor celestial bodies). Finally, the perihelion precession of Mercury was simulated.

The Verlet solver was proved superior to the Euler solver due to its high accuracy in early tests. The escape velocity of the Earth from the Sun's gravitational pull can be approximated by trial and error, which were close to the numerical result. The mass of Jupiter has a large effect on both the Sun and the Earth in the three body system. Setting the mass of Jupiter equal to the mass of the Sun results in them orbiting each other, which confirms that Newton's gravitational forces also apply in this model of our solar system. The entire solar system was simulated as expected using actual positions and velocities gathered from NASA. The general theory of relativity could be illustrated by showing the perihelion precession of Mercury by adding a relativistic correction to the Newtonian force.