Computational Project 5: Solar System

Aksel Graneng, Michael Bitney and Silvia Morales

December 19, 2019

Abstract

In this paper we have looked at solving the n-body problem consisting of celestial bodies interacting through Newtonian and relativistic gravity. This was done making a model by implementing the Euler Forward and Velocity Verlet integration methods. The model was used for different initial conditions, such as planetary data from NASA for November 29th 2019 of our solar system. We used this model for 2, 3 and n-body simulations. We also used it in order to see the general relativistic effects on the perihelion precession of Mercury, comparing this to the observed 43" precession per century.
We found that Velocity Verlet conserves energy and angular momentum, while Euler Forward gained angular momentum and lost energy. Following this, we concluded that Velocity Verlet gives better and more stable orbits than Euler Forward, therefore preferable when simulating orbits.
Through experimentation, we adjusted the gravitional force to be propertional to $1/r^3$, instead of $1/r^2$, and found that it lead to a more chaotic solar system, where the only stable orbit was a circular orbit.
Using the n-body simulator we simulated a full solar system using initial values for our solar system and found to be accurate.
We did not succeed to correctly model the perihelion precession of Mercury. This was likely due to precission and coding errors.

The material for this project can be found in our GitHub repository at: `https://github.com/michaesb/Solar_system_simulation`

Introduction

Astronomy is one of the oldest branches of science. We know of people trying to describe the movement of celestial bodies as far back as Eudoksos in ancient Greece, who described a system with the Earth as its center [2].

As time has passed, models of the solar system have become more complex. From Galileo Galilei's heliocentric model and Kepler's description of elliptic orbits to Sir Isaac Newton discovering the law of gravity and Albert Einsteins general theory of relativity, great leaps have been made in astronomy.

With today's numerical tools, we can not only look at the past movements of the planets, but also predict their future movements through implementing the laws that govern their movement in numerical models.

In this project, we will be looking to make models that solve first the 2-body using 2 different numerical integration methods, and then the n-body problem using Velocity Verlet and using it to calculate planetary orbits in the solar system. We will also be looking at the effect general relativity has on celestial bodies that orbit close to the sun, such as

Mercury.

This paper includes a theory part explaining the basic theory on Newtonian mechanics, the relativistic effect on gravity, basics behind numerical integration and descriptions of the Euler Forward and Velocity Verlet integration methods. It also includes a method part detailing what was done, results on the form of plots, a discussion and a conclusion.

Theory

Newtonian mechanics

Forces and motion

When looking to understand the motion of celestial bodies, one must first understand the relations of force, mass and acceleration. The root of it all lies in Sir Isaac Newtons famous second law of motion, which explains the relationship between force, mass and acceleration. Mathematically, it can be written as equation 1:

$$\vec{F} = m \cdot \vec{a} \tag{1}$$

Where $\vec{F}$ is force, m is mass and $\vec{a}$ is acceleration, the time derivative of velocity:

$$\vec{a} = \frac{d\vec{v}}{dt}$$

Which in turn is the time derivative of position:

$$\vec{v} = \frac{d\vec{x}}{dt}$$

Equation 1 can be rewritten as:

$$\vec{a} = \frac{\vec{F}}{m} \tag{2}$$

From this, we can see that understanding the forces applied to a body, along with its mass, can let us understand its position and velocity, which is what encompasses its motion.

Gravity

By combining an understanding of the laws of motion with observations of the planets, Sir Isaac Newton managed to come up with a force which attracts 2 bodies to each other, explaining the planetary orbits in great detail, which is equation 3.

$$\vec{F}_G = G\frac{mM}{R^2}\hat{R} \tag{3}$$

Here, m and M are the masses of the 2 attracting bodies, R is the distance between the 2 bodies, $\hat{R}$ is a unit vector pointing from the body the force is working on to the other body, and G is the gravitational constant: 6.67408 $\mathrm{m^3kg^{-1}s^{-2}}$.

When looking at our solar system, one can simplify gravity by measuring distance in astronomical units (AU) and time in years. One AU then becomes the mean distance between earth and the sun: $1.5 \cdot 10^{11}$ m. Using these units, we can see by assuming the earths orbit to be constant, that the gravitational constant multiplied with the mass of the sun simply becomes $GM_\odot = 4\pi^2$ with $M_\odot$ being the mass of the sun.

This lets us rewrite gravitational force:

$$\vec{F}_G = 4\pi^2\frac{m}{R^2}\hat{R}$$

Subsequently, gravitational acceleration becomes:

$$\vec{a} = \frac{4\pi^2}{R^2}\hat{R}$$

Conservation

An important consequence of Newtons second law is the concept of conservation [1]. Two of the most important conservation laws state that energy and angular momentum is conserved in a closed system.

For energy, this means that the sum of potential and kinetic energy must be constant. When the potential energy comes from only gravity, the formula for total mechanical energy of a body with mass M_1 becomes:

$$\frac{1}{2}M_1V^2 + G\frac{M_1M_2}{r} = E_0 \tag{4}$$

Where M_2 is the mass of the body attracting it.

For angular momentum, the equation is:

$$\vec{R}_1 \times \vec{p}_1 = \vec{R}_2 \times \vec{p}_2 = \vec{L}_0 \tag{5}$$

Where $\vec{R}$ and $\vec{p}$ is the position and momentum of the body.

General Relativity

In 1915, Albert Einstein released his paper on general relativity. The general theory of relativity boils down to time and space being connected, so that time progresses faster, or space is bent more, closer to large masses such as the sun, instead of the previous notion that gravity being a force acting on a object. A result of this is that Newtonian gravity is slightly incorrect. This effect is especially noticeable in orbits closer to the sun.

The effect of general relativity causes orbits to not be perfect ellipsoids, as one would expect from Newtonian gravity. After one period (one year in the case of the Earth), one would normally expect the position of the celestial body to be exactly what it was one period earlier. If one adds a general relativistic correction to Newtonian gravity, this is not the case. Instead, the position will differ slightly from the initial one. In the long term, such a correction would lead to the perihelion of the orbital ellipse precessing around the Sun.

In our Solar System, this is most notable for Mercury as it has the orbit closest to the sun. Therefore, looking at the perihelion precession of Mercury is a good way to test general relativity. After all classical effects have been subtracted, the perihelion precession of Mercury has been observed to 43 arcseconds per century [3]. The gravitational force on Mercury, corrected for general relativistic effects, becomes equation 6.

$$\vec{F}_G = \frac{GM_{\odot}M_{\text{mercury}}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right] \tag{6}$$

Numerical Integration

Differential equations are a powerful tool when it comes to describe the behaviour of nature. In the case of the force between two masses, the resulting differential equation can tell us much about the system, for example, the velocity and position of an object at different time steps. In order to solve these equations we must resort to integration. Several algorithms exist in order to approximate integral results, but all of them work similarly:

- Have a function $y(t)$ with an initial value, $y(t_0)$
- Create a step dependant on the interval you want to integrate, $h(n)$ where n is the number of integration points and a, b is the area we want to integrate over. In our case, since we need a fixed step, we will use:

$$h(n) = \frac{b-a}{n} \tag{7}$$

- Construct a definition for the next value of $y(t)$ using the step function.

The difference between the algorithms is the definition of the next value of the function $y(t)$. This change can lead to loss or gain of precision, which in sensible systems like ours, is incredibly important.

Eulers method

The Euler's method will define the next value of $y(t)$ as:

$$y(t_i + h) \equiv y_{i+1} = y_i + hf(t_i, y_i) + \mathcal{O}(h^2), \tag{8}$$

where $f(t,y)$ is the first derivative of the function $y(t)$ which is found by a truncation of the first term of a Taylor expansion. The proof taken from The MIT site, see [5]. Because of the usage of the first term of the Taylor expansion only, we expect an error which is noted as $\mathcal{O}(h^2)$ In this case, the next element of the function $y(t)$ is given explicitly by the previous element. Because of this, this case of the Euler method is know as the Forward Euler algorithm.

Velocity Verlet

The Verlet algorithm is a easy to program and numerically stable integration algorithm which will define the next value of $y(t)$ as a set of coupled differential equations. For example, in the case of Newton's second Law:

$$\begin{aligned}
m\frac{d^2x}{dx^2} &= F(x,t), \\
\frac{dx}{dt} &= v, \\
\frac{dv}{dt} &= \frac{F(x,t)}{m} \equiv a,
\end{aligned}$$

from which we use the Taylor expansion and truncation on both. Full proof can be found at References nr [6]. The Verlet algorithm is usually used to find the position of the object using the acceleration and the previous position as parameters. If you include the contribution of the velocity in the calculation of the position, the algorithm is known as the Velocity Verlet approximation. The expression found is:

$$\begin{aligned}
y_{i+1} &= y_i + hv_i + \frac{1}{2}a_i(h^2) + \mathcal{O}(h^4) \\
v_{i+1} &= v_i + h\frac{1}{2}(a_{i+1} + a_i)
\end{aligned}$$

where the velocity has been defined as a separate expression which $y(t)$ is dependent from. The full proof can be found in the reference section. The error in the Velocity Verlet algorithm is smaller than the one given by the Euler Forward method. This allows us to simulate with higher precision without increasing the computational power needed.

Method

We want to simulate the orbit of celestial bodies. To do this, we used the Euler Forward and Velocity Verlet integration methods to integrate a discretized gravity (equation 3):

$$\vec{F}_i(\vec{R}_i) = G\frac{mM}{R_i^2}\hat{R}_i$$

We started off by solving a 2-body problem with a stationary sun and a moving earth, by using both integration methods. In this case, by using astronomical units (AU) and years, the gravitational acceleration on the earth became:

$$\vec{a}_i(\vec{R}_i) = \frac{4\pi^2}{R^2}\hat{R}$$

By placing the Earth at an initial position of 1 AU, this simply became $4\pi^2$.
We tested physical properties of our 2-body simulator, giving earth different initial velocities. We also tested the simulator with different gravity functions by changing a β value in ranges of $\beta \in [2, 3]$ in

$$a_G = 4\pi^2\frac{1}{R^\beta}$$

Once this was done, we made a n-body simulator using Velocity Verlet. This simulator could allow one celestial body to remain stationary, letting us simulate celestial body orbits with a stationary sun.
Using this simulator, we modeled a 3 body problem with Earth, the Sun and Jupiter with initial conditions retrieved from NASA [4] for November 29th 2019. We did this again, multiplying Jupiter's mass with 10, 100 and 1000.

We used the n-body simulator to model all the major celestial bodies in the solar system, including the Sun, all the planets and Pluto. This was simulated over 250 years using 10^6 integration points.

We wanted to model the perihelion precession of Mercury, and compare our results to the observed precession of 43" per century. To do this, we added a relativistic factor to gravity (see equation 6) and simulated the movement of Mercury in a system consisting of Mercury and a stationary sun for 100 years. We did this with a timestep of $\Delta t = 10^{-6}$. To find the precession, we then calculated its orbital angle at its final orbital perihelion by $\theta = \arctan(\frac{y_p}{x_p})$.
Due to the discretized nature of the orbital data, the position of perihelion was ambiguous. We therefore approximated the perihelion precession as the mean of precessions for the final 5 years of simulation.

Results

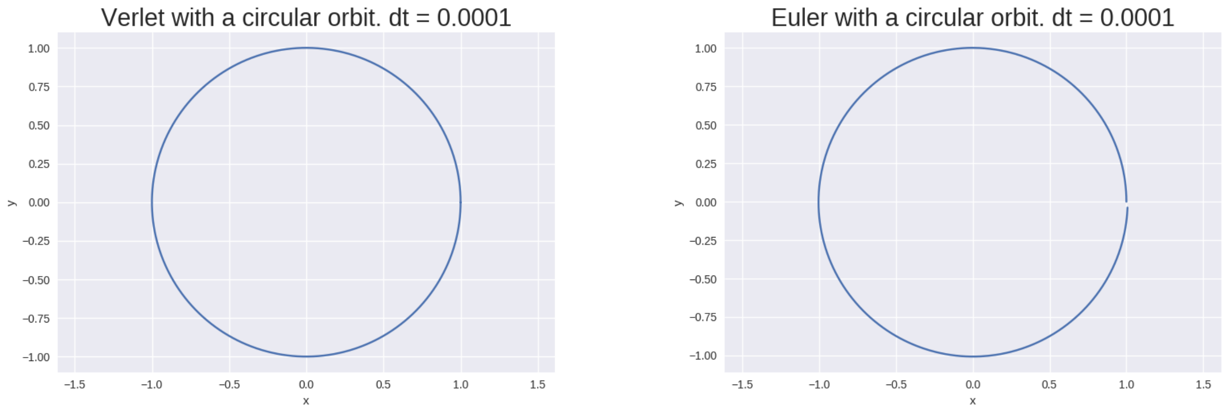


Figure 1: Comparisons of orbital simulations for the Verlet (left) and Euler (right) methods, for a circular orbit around a stationary sun with a $dt = 0.0001$.

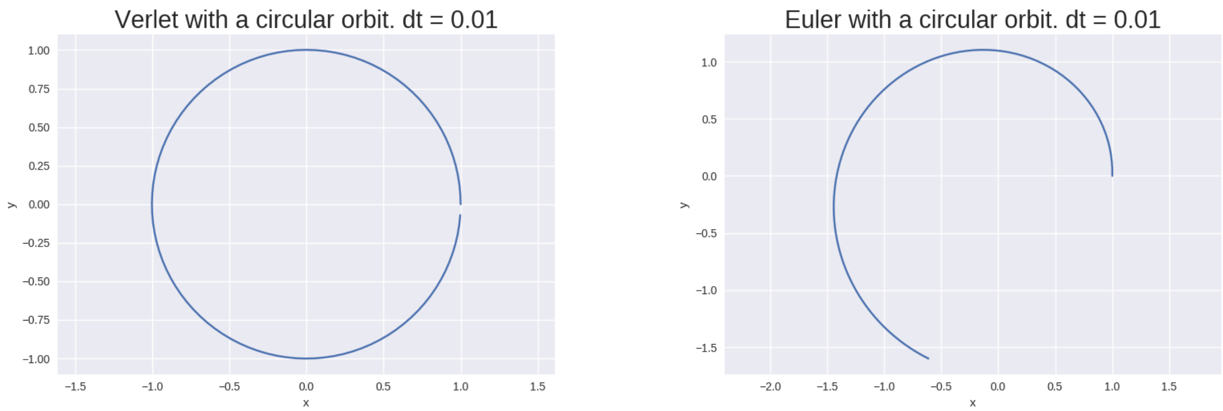


Figure 2: Comparisons of orbital simulations for the Verlet (left) and Euler (right) methods, for a circular orbit around a stationary sun with a $dt = 0.01$.

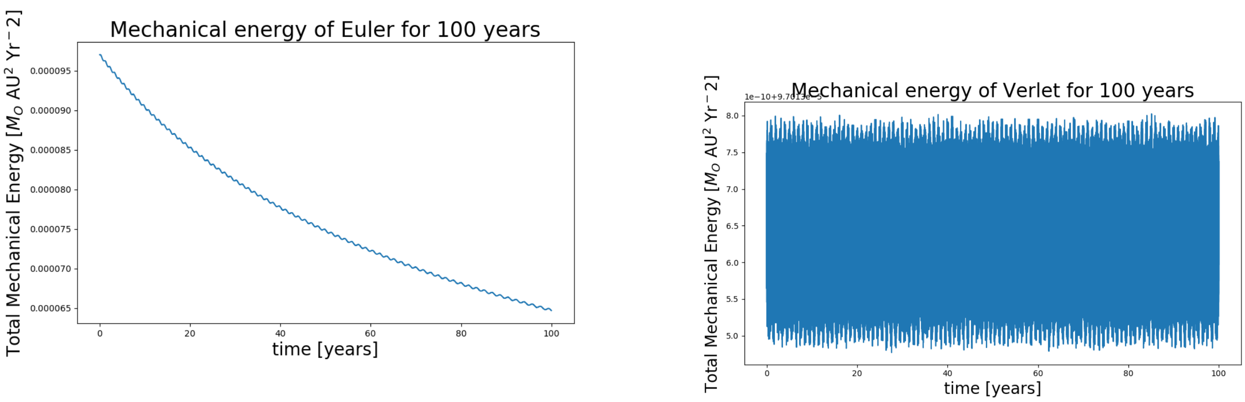


Figure 3: To the left we plotted the mechanical energy for a system computed with Euler Forward algorithm. To the right, the same plot, but using Velocity Verlet method

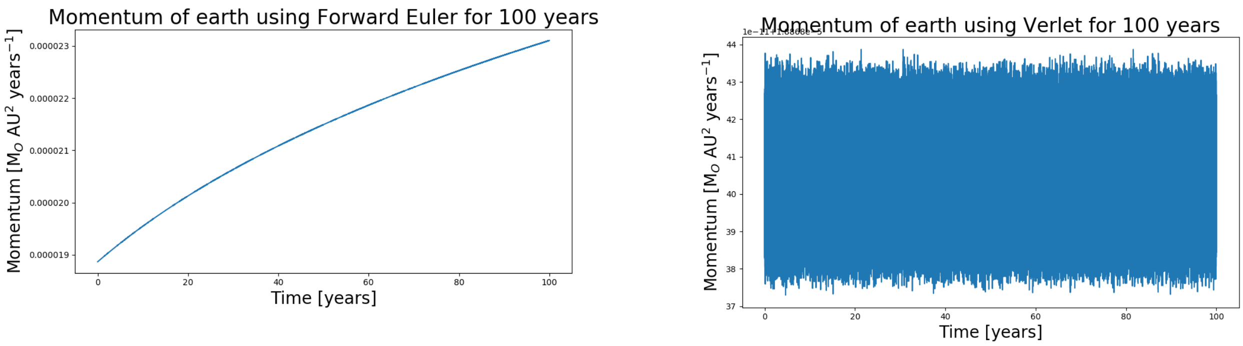


Figure 4: To the left we plotted the momentum of Earth using Euler Forward to compute the data. On the right, Velocity Verlet is used.

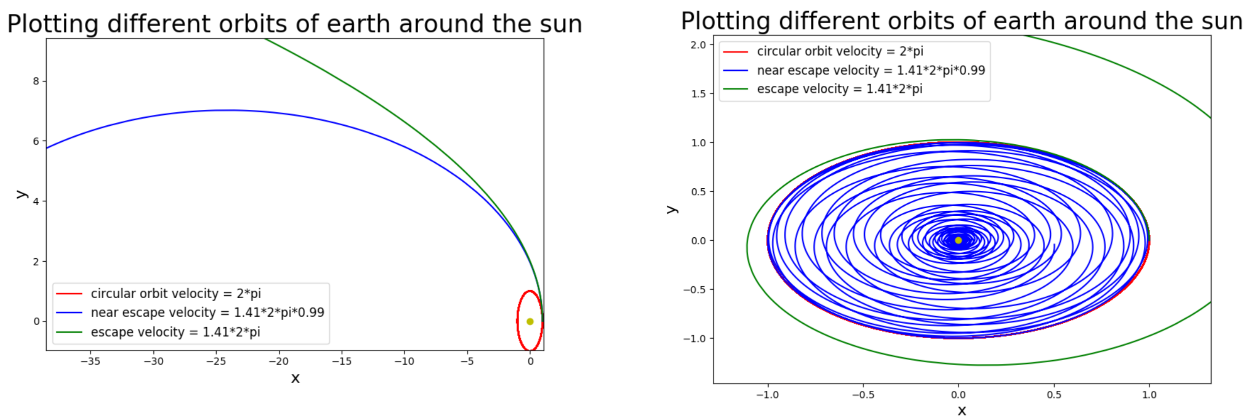
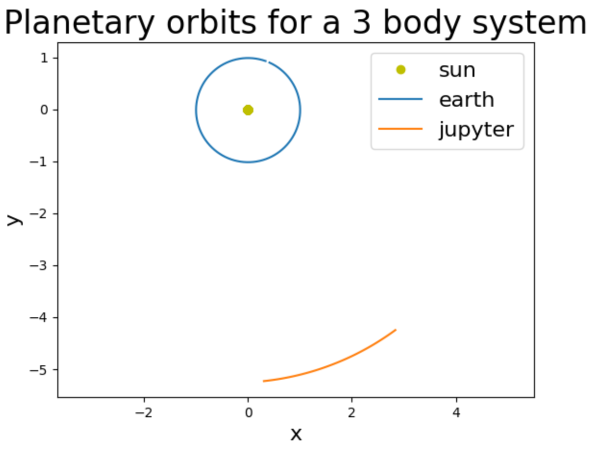
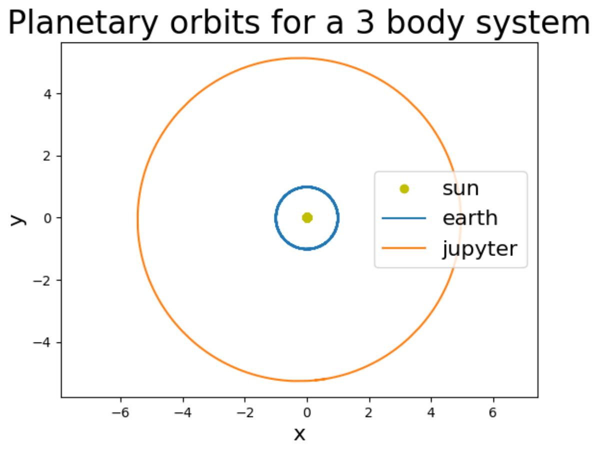


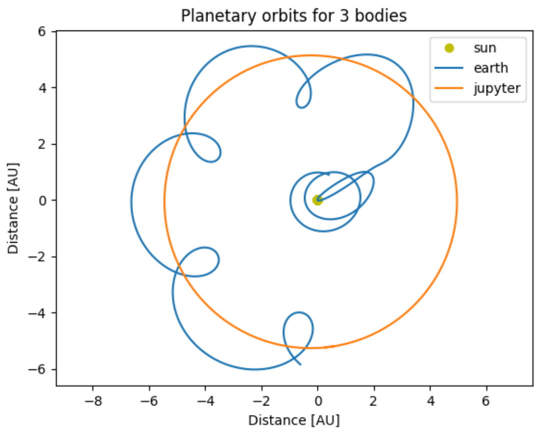
Figure 5: To the left we plotted the orbit of Earth around the sun for 100 years, for different start velocities of earth. To the right we adjusted the gravitational force with to be proportional to $1/r^3$ instead of $1/r^2$ (see equation 3).



(a)



(b)



(c)

Figure 6: Figure 6a shows the orbit of the Sun, Earth and Jupiter system (SEJs) for 1 year, with Jupiter having its usual mass (dt = 1e-2). Figure 6b shows the SEJs for Jupiter being 10 times heavier than usual (dt = 1e-6). The last figure 6c shows the SEJs for Jupiter being 1000 times heavier than its real mass (dt = 1e-6).

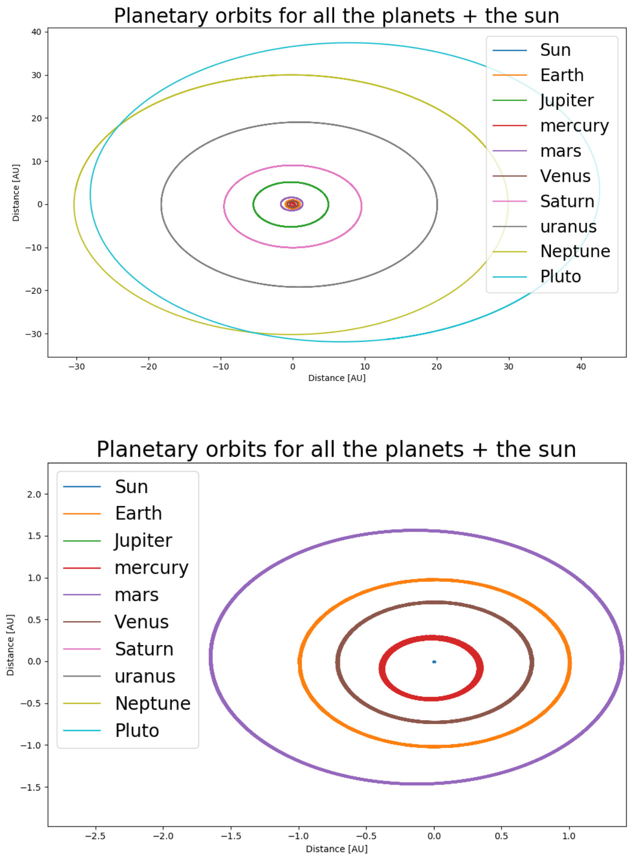


Figure 7: At the top we plotted the orbits of all planets in the Solar System with the initial conditions provided by NASA. To the right we zoomed into the orbits of Mercury, Venus, Earth and Mars, with the Sun as the origin

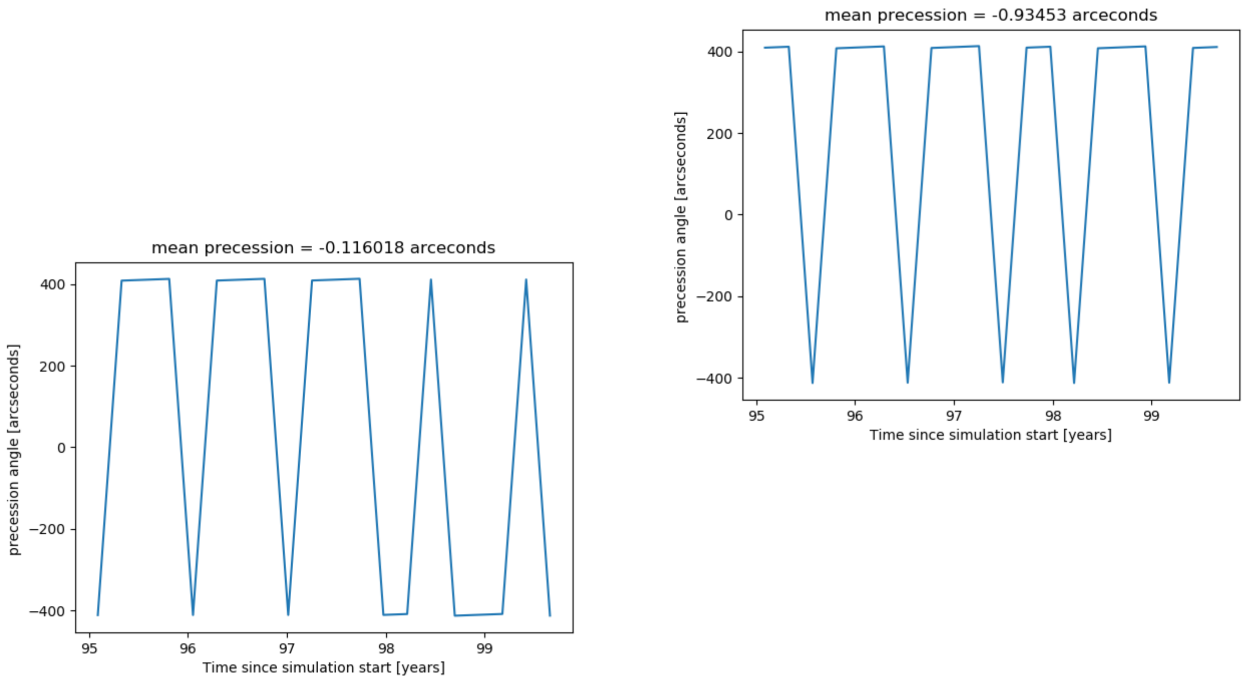


Figure 8: Here we have the angle of Mercury at perihelion for the last 5 years of simulation. The lefternmost figure is without a relativistic correction, while the rightnmost is with relativistic gravity. The mean values are in the figure titles, with -0.12″ without relativity and -0.94″ with relativity.

Discussion

Euler Forward and Velocity Verlet Comparison

The purpose of figures 1 and 2 is to compare the accuracy between the Euler Forward and Velocity Verlet algorithms. Note that in figure 1 the orbit in Euler is not perfectly circular and starts to lose it's accuracy. Velocity Verlet is performs better in this aspect, being able to complete an entire orbit without any (visible) loss of accuracy.

Note that in figure 2, the orbit plotted with Verlet method is not perfectly circular and start to lose accuracy with a small time step. Still, we can see that system is stable. Euler here is no longer a circular orbit and can't function at all with this low dt.

The Euler and the Verlet algorithm was tested with a single orbit around the sun and found that the Euler method needed a $dt \leq 0.0001$ in order to be stable. Verlet did significantly better needing only $dt \leq 0.01$. These two plots make us aware of the great loss of precision Euler Forward has, and Therefore we see that Velocity Verlet is preferable.

We also plotted the mechanical energy of an object around the sun in figure 3. We can clearly observe a loss in mechanical energy using the Euler algorithm, which explains why the planets slowly spiral into the Sun. In the case of the Verlet algorithm, we observe the mechanical energy oscillating, but being stable over a long timespan. This also shows that Velocity Verlet is preferable in simulating orbits, where conservation of energy and momentum is essential.

Escape Velocity and adjustment of gravity

For figure 5 we observe the orbit of the Earth with different starting velocities with two different laws of gravitation. What we call the "near escape velocity" (NEV) is a starting velocity which is 99 % of the normal escape velocity for the Earth, which was calculated earlier. Note that, in the left figure using NEV as a start velocity will make the Earth eventually go back to orbit the Sun. The right plot shows the same, but for a new law of gravitation which is proportional not to $1/r^2$, but to $1/r^3$. We observe that a slight change in the starting velocity will affect the orbits much more, as a change in 1% can either slingshot the Earth to space or spiral it into the Sun.

3 Body System

We now move to solve the three body system. This is shown in figure 6. As you increase Jupiter's mass, we can observe how the orbit of the Earth becomes slowly affected by it. It comes a point, when the mass of Jupiter is 1000 times heavier than usual, where the Earth feels more gravitational pull from Jupiter and starts orbiting the gas planet instead of the Sun. This provides with a really interesting plot.It's worth noting that if Jupiter increased in size by a factor of 1000, it will become star, which make this a two star system instead.

a Full Solar System

Now, we observe the Solar System as a whole. Figure 7 shows the 8 planets and the dwarf planet Pluto plus the Sun as the origin. The orbit of Pluto is unusual compared to the other planet's orbits, which was one of the reasons why Pluto got its planet status taken away in 2006. As we zoom into the closest orbits, we can observe Mercury's orbit with a thicker line than other orbits. This is because this plot shows 250 years worth of orbits, and with Mercury's orbit being 88 days, we see a lot of overlapping orbits. This might be due to an changing elleptic orbit or the orbit rate might be to high compared to the time resolution.

b Perihelion precession of Mercury

Finally we wanted to study the Perihelion precession of Mercury, and compare our results to the observed precession of 43" per century. From figure 8 we see perihelion angles for the last 5 years of simulation. We see that the resulting angles are going back and forth between approximately 400 and -400 arcseconds. This might be because the algorithm used to find the perihelion cannot find the exact physical point of perihelion, as the discretized nature of our data set only lets you get so far.

We therefore see that the resolution, and therefore also error, in our calculated angles to be about 800 arcseconds. As this error is around 20 times larger than the value we are looking for, our data becomes statistically insignificant. Alternatively the actual perihelion could be oscillating between orbits. A way to distinguish this would be to increase the number of integration points to something at least 20 times higher than what is used. This would significantly increase computation time however, making additional debugging difficult.

Conclusion

Euler Forward and Velocity Verlet

From figure 1 to 4 we can see quite clear differences between the Velocity Verlet and Euler Forward integration methods. Velocity Verlet seems to conserve energy and angular momentum, while Euler Forward loses energy and gains angular momentum. This leads to Velocity Verlet giving us much more stable, and therefore realistic, orbits, even at quite low time resolutions.

Stability of the solar system

From figure 5 to 6 we see how changing physical properties even slightly can lead to a much less stable system.

First we see how, if gravity's dependence on distance had been tweaked slightly, the amount of possible stable orbits would decrease drastically. In fact, if gravity was proportional to $1/r^3$, the only stable orbit would be a perfect circle.

Looking at figure 6 we can see how an extra celestial body with a mass similar to the sun would affect Earths orbit. While this model is unrealistic as the sun and "Jupiter" would orbit around a common center of mass, it gives us a good idea on how chaotic earths orbit would be. If Earths orbit was as chaotic as seen in figure 6c, chances of life on would have been minimal as conditions such as temperature would be changing quite rapidly among many other effects.

Solar system simulation

In figure 7 we see a realistic model of the solar system, simulated over 250 years. The accuracy of this data could be increased by increasing the number of integration points, adding all known celestial bodies and adding a relativistic correction.

Perihelion precession of Mercury

From figure 8 there are 3 conclusions we can draw.

Firstly: The angle of Mercury's perihelion is in fact oscillating with $\pm$ 400 arc seconds. In this case we would have to find the perihelion precession by making a linear approximation of the data over a long period of time (preferably 100 years or more).

Secondly: There is some fault in our model or the way we read the data. This would hinder us in finding the minute values we are looking for.

Thirdly: Our current data is unusable in this aspect, making us need a higher resolution and more data points in order to find the actual precession.

Future suggestions

For future work, one can look into optimizing the computation time of the model.
More work should also be done on finding the perihelion precession of Mercury. This can be done by thoroughly testing every part of the model to remove any potential bugs, finding better ways to process the data, getting data with a higher resolution, and doing a linear approximation on perihelion angle data for 100 years. Our plots were programmed to hold only 1000 points, in order to minimize the processing time of the plots. It would be an improvement to create plots with more points graphed in order to observe the precession more clearly.

References

[1] Malthe-Sørensen, A (2015) *Elementary mechanics using python*. Cham Heidelberg New York Dordrecht London: Springer.

[2] Elgarøy, Ø (2017) *Astronomi - en kosmisk reise*. Oslo: Universitetsforlaget.

[3] Lecture Notes made by Morten Hjorth-Jensen in class Fys3150/4150 at UiO `https://github.com/CompPhysics/ComputationalPhysics`

[4] HORIZONS Web-interface, retrieved from `https://ssd.jpl.nasa.gov/horizons.cgi#top` at 15. des 2019.

[5] MIT Course Notes for Differential Equations retrieved from: `http://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node3.html`

[6] University of Delaware notes on Computational Methods for Physics retrieved from: `http://www.physics.udel.edu/~bnikolic/teaching/phys660/numerical_ode/node5.html`

Acknowledgements

Thanks to Morten Hjorth-Jensen and student teachers in FYS-STK-4150/3150 for help during this project.
We thank Bernhard Lotsberg for help during debugging of code.