Building a model of the Solar System using ordinary differential equations to solve the equations of motion

Fredrik Hoftun & Mikkel Metzsch Jensen

October 26, 2020

Abstract

In this project we have used different numerical ordinary differential equation solvers to simulate a Solar System model. We were able to successfully simulate stable orbits and affirm laws of nature, like conservation of mechanical energy and total angular momentum. We concluded that the velocity Verlet method is the superior ODE-solver for this particular problem, as it conserves energy and have a smaller global error without any significant expense of run time.
We introduced a modified gravitational force, the inverse β force, which scaled the exponent for the distance $r^\beta, \beta \in [2, 3]$ and saw that it affected the stability of the system greatly.
We also looked at how the number of celestial objects simulated in a Newtonian model affected the positional precision of said objects. We found that the precision decreased as the number of objects increased.
Finally, we calculated the perihelion precession of Mercury around the Sun for classical forces to $p_{cy,c} = -8 \pm 13''$ and general relativistic gravitational forces to $p_{cy,r} = 43.5 \pm 0.8''$ from which we found a total precession of $P_{cy} = 51.5 \pm 13$. The big uncertainty originates from possible errors in our classical model, but the results still fits with the predicted value of 43".

Link to our GitHub repository: https://github.com/mikkelme/project3_FYS3150

1 Introduction

The study of the Solar System is a scientific discipline which dates back to the Ancient Era, where mankind observed the nearest planets and stars with the naked eye. As we have gained more insight into this field we have gone through several paradigm shifts regarding our worldview and our small role in the big universe. Some of the biggest shifts came from discoveries like Copernicus' Sun centered system and Newton's theory of gravity, but it is still refined in modern science with the introduction of Einstein's theory of relativity and perhaps also quantum mechanics. In this project we aim to develop a numerical model of the Solar System which makes us able to simulate and predict the movement of several celestial objects. While the analytical solutions are limited to the two-body problem, we can easily compute simulations of a three-body problem or more. Once equipped with a model that describes the force interaction between the celestial objects we can solve the equations of motion as sets of coupled ordinary differential equations (ODE). For this purpose there exists a variety of algorithms. We are going to use the forward Euler method and the velocity Verlet method.

Starting off we are going to implement a hypothetical solar system consisting of only the Earth orbiting the Sun. We will then run a series of test in order to ensure that our method provide stable and energy conserving orbits. We will also use this opportunity to compare the performance of the two ODE-solver algorithms, and settle on the most effective for the further investigations. Then we will calculate the escape velocity analytically and numerically and introduce a variation of the gravitational force as $F_G = \frac{GmM}{r^\beta}$ with $\beta \in [2,3]$ to see how the modified gravitational force affects the system.

Afterwards, we will expand the system to include Jupiter with the purpose of testing how it affects the stability of the Earth-Sun system. Later we will include all the other planets and Pluto to our system, resulting in a ten-body-problem. For this section we will use data from NASA's Jet Propulsion Laboratory (JPL) to initialize the celestial objects with positions and velocities with respect to the Solar System center of mass.

Finally, we will investigate one of the tests of general relativity, the perihelion precession of Mercury. According to observation the perihelion precession of Mercury disagrees from the predicted Newtonian theory by $\approx 43''$ (arcseconds/Julianan century). In general relativity this difference is nonexistent. We will implement a relativistic correctional gravitational force from general relativity to see if we can reproduce this difference and affirm Einstein's theory of general relativity.

2 Method

2.1 Setting up the Earth-Sun system

In the beginning we will limit ourselves to a hypothetical solar system with only the Earth orbiting the Sun. The only force in this problem is the gravitational force F_G acting between the Earth and the Sun. This can be described with Newton's law of gravity as:

$$F_G = \frac{GM_\odot M_{\text{Earth}}}{r^2}$$

where $M_\odot$ is the mass of the Sun and M_{Earth} is the mass of the Earth. G is the gravitational constant and r is the distance between the Earth and the Sun. Since the mass of the Sun is much larger than that of the Earth we can assume that the Sun is fixed at the origin. By using Newtons second law of motion we arrive at the following differential equation that describe the motion of the Earth:

$$\mathbf{a} = \frac{d^2\mathbf{r}}{dt^2} = \frac{\mathbf{F_G}}{M_{\text{Earth}}}, \quad \mathbf{r} = (x, y, z)$$

For the numerical approach we split this into three equations for the Cartesian coordinates x,y and z:

$$\frac{d^2x}{dt^2} = \frac{F_{G,x}}{M_{\text{Earth}}}, \qquad \frac{d^2y}{dt^2} = \frac{F_{G,y}}{M_{\text{Earth}}}, \qquad \frac{d^z}{dt^2} = \frac{F_{G,z}}{M_{\text{Earth}}} \tag{1}$$

where $F_{G,x}$, $F_{G,y}$ and $F_{G,z}$ are the x, y and z components of the gravitational force. By using the fact that the Earth's orbit around the Sun is almost circular we can assume the following relation from the centripetal force:

$$\frac{M_{\text{Earth}}v^2}{r} = \frac{GM_\odot M_{\text{Earth}}}{r^2}$$

where v is the tangential velocity. The latter equation can be solved to show that:

$$v = \sqrt{\frac{GM_\odot}{r}}$$

We introduce "Astronomical units" where we define mass as mass relative to the solar mass $M_\odot$, time as Julianan years 1yr = 365.25 days, 1 day = 86400 s, and distance as one astronomical unit 1AU = 149 597 870 700m (approximately the mean distance between the Sun and Earth). By using the follow unit conversion:

$$[m] = \frac{1}{149\ 597\ 870\ 700}\ \text{AU} \qquad [s] = \frac{1}{365.25 \cdot 86400}\text{yr} \qquad [kg] = 1.98847 \cdot 10^{-30} M_\odot$$

we can recalculate the value of G as:

$$G = 6.67408 \cdot 10^{-11} \frac{m^3}{s^2 kg} = 39.477\ \frac{AU^3}{yr^2 M_\odot} \approx 4\pi^2\ \frac{AU^3}{yr^2 M_\odot}$$

We then see that a circular orbit for the Earth at $r = 1$ AU must have a tangential velocity of $v = 2\pi$ AU/yr.

2.1.1 Algorithms for solving ordinary differential equations

In order to solve the ordinary differential equations (ODE) displayed in the previous section (see equation 1) we will use both the the Forward Euler and the Velocity Verlet method. The Euler method is the simplest possible ODE-solver and can be derived from the Taylor expansion of a function f around a value $t_0 + dt$:

$$f(t_0 + dt) = f(t_0) + dtf'(t_0) + \mathcal{O}(dt^2)$$

where f is a differentiable function of t and dt is a small timestep. This approximates the evolution of f as t is increased by dt. We can substitute in the the equations of motions from (1) for a solution to our problem. For the x-component as an example we get:

$$\begin{aligned} a_x = \frac{d^2x}{dt^2} &= \frac{F_{G,x}(\mathbf{r}(t))}{M_{\text{Earth}}} \\ v_x(t+dt) &= v_x(t) + a_x(t)dt \\ x(t+dt) &= x(t) + v_x(t)dt \end{aligned}$$

This can easily be discretized as:

$$\begin{aligned} a_{x,i} &= \frac{F_{x,i}}{M_{\text{Earth}}} \\ v_{x,i+1} &= v_{x,i} + a_{x,i}dt \\ x_{i+1} &= x_i + v_i dt \end{aligned}$$

The local error done in each step is $\mathcal{O}(dt^2)$ which mean that the global error for a simulation of T years with $N = T/dt$ timesteps is $\mathcal{O}(N)$. The number of FLOPs for the Euler method is $N(5+F(F_G))$ where $F(F_G)$ is the number of FLOPs needed to calculate the gravitational force F_G. For the Earth-Sun system we have $F(F_G) = 4$, two for the numerator, one for the denominator and one for the fraction itself. This however can be reduced to $F(F_G) = 3$ by pre-calculating the denominator (as it is constant). However, in reality the gravitational force is three dimensional, so the time needed to calculate should go like $\approx 3N$. This can be mitigated by vectorizing the gravitational force and algorithms:

$$\mathbf{F_G} = -\frac{GM_{\odot}M_{\text{Earth}}}{r^2}\hat{\mathbf{r}} = -\frac{GM_{\odot}M_{\text{Earth}}}{r^3}\mathbf{r}$$

where $\hat{\mathbf{r}} = \frac{\mathbf{r_S}-\mathbf{r_E}}{|\mathbf{r_S}-\mathbf{r_E}|}$ is the positional unit vector. The time then gets reduced back to $\approx N$ as all three dimensions are calculated simultaneously.

It is a well known fact that the Euler method is quite inefficient for bigger calculations as it needs a very small step length dt to get satisfactory results in most cases. The Euler method is not a symplectic ODE-solver, meaning that it will not conserve energy very well. The velocity Verlet method on the other hand is a symplectic ODE-solver specifically designed for solving the equations of motion[3]. It can be derived by Taylor expanding $x(t+dt)$:

$$\begin{aligned}
x(t+dt) &= x(t) + \frac{v_x(t)}{1!}dt + \frac{a_x(t)}{2!}dt^2 + O(dt^3) \\
v_x(t+dt) &= v_x(t) + \frac{a_x(t)}{1!}dt + \frac{a'_x(t)}{2!}dt^2 + O(dt^3) \\
a_x(t+dt) &= a_x(t) + \frac{a'_x(t)}{1!}dt + O(h^2) \overset{\approx}{\rightarrow} a'_x(t)dt = a_x(t+dt) - a_x(t) \\
v_x(t+dt) &= v_x(t) + \frac{a_x(t+dt) + a_x(t)}{2}dt + O(dt^3)
\end{aligned}$$

which can be easily discretized as

$$\begin{aligned}
a_{x,i} &= \frac{F_{x,i}}{M_{\text{Earth}}} \\
x_{i+1} &= x_i + v_{x,i}dt + a_{x,i}\frac{dt^2}{2} \\
v_{x,i+1} &= v_{x,i} + \frac{a_{x,i} + a_{x,i+1}}{2}dt
\end{aligned}$$

The global error of this method goes like $\mathcal{O}(dt^2)$ while the number of FLOPs required: $N(12 + 2F(F_G))$ is still proportional to N like the Euler method.

2.1.2 Conservation of angular momentum

Kepler's second law of planetary motion states: "A line connecting the Sun and the planet sweeps out equal areas in an equal time interval"[4]. Mathematically we can express this area sweep as:

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \frac{r^2\mathrm{d}\theta}{2\ \mathrm{d}t} = \frac{L}{2m}$$

where L is the angular momentum. The change in the angular momentum with respect to time defines the torque on the system $\tau = \frac{\mathrm{d}L}{\mathrm{d}t}$, which can also be described as $\tau = r \times F$. Since gravity is the only force acting in the Earth-Sun system and it is parallel to the position vector r we get:

$$\tau = r \times F = \frac{\mathrm{d}L}{\mathrm{d}t} = 0$$

From this we can tell that the angular momentum must remain constant in the Earth-Sun system. The angular momentum of a a two-body gravity must therefore always be conserved, since the positional vector and force vector are always parallel. In a simulation with three or more bodies the individual angular momenta are not conserved, as we get cross-products between forces and positional vectors that are not parallel. However the total angular momentum of the system is conserved because of the rotational symmetry of the gravitational force. This means that the total angular momentum is conserved in a system consisting of the Sun, all the planets and Pluto or other similar systems.

2.1.3 Testing forms of the force

Till now we have assumed that gravity is an inverse square force:

$$F_G = \frac{GM_\odot M_{\text{Earth}}}{r^2}$$

but we will investigate the consequences of having an "inverse β" force:

$$F_G = \frac{GM_\odot M_{\text{Earth}}}{r^\beta}$$

with $\beta \in [2, 3]$. We will run simulations on the Earth-Sun system using the velocity Verlet method for different values of β and test whether or not the system is stable in terms of position and energy conservation.

2.1.4 Escape velocity

The escape velocity is defined as the velocity where the kinetic and the potential energy are equal:

$$\frac{1}{2}M_{\text{Earth}}v^2 = \frac{GM_\odot M_{\text{Earth}}}{r}$$

and we find the esacpe velocity v_{esc} as:

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

If $r = 1AU$ we get:

$$v_{esc} = \sqrt{2 \cdot 4\pi^2} \approx 8.857AU/yr = 42.14km/s$$

In order to the test our implementation of the Solar System model, we will approximate this value numerically. This is done by setting up a simplified system where the earth is initialized with $\mathbf{r}$ = (1 AU, 0, 0) and velocity $\mathbf{v}$ = (v_{test}, 0, 0). If the test speed v_{test} is lower than the escape velocity it should eventually change direction and come back to the Sun. We therefore set up a algorithm that checks whether the velocity x-component change sign during the simulation time. We then increase v_{test} until we no longer see the Earth turning around. We use a simulation time of $T = 1000$ yr and $dt = 0.01$ yr.

2.2 Expanding the Earth-Sun system

We now want to expand our Earth-Sun system to include Jupiter. The simplest way to do this is using the vector formulation of Newton's law of gravity:

$$F_G = -G\frac{Mm}{|\mathbf{r}|^3}\mathbf{r}$$

where M and m are the masses of the two objects, $\mathbf{r}$ is the positional vector between them and $|\mathbf{r}|$ is the length of the positional vector (the distance between the objects).

We know that the gravitational force on Earth is the sum of the gravitational forces from the Sun and Jupiter

$$\begin{aligned} F &= F_S + F_J \\ &= -G\frac{M_\odot M_{\text{Earth}}}{|\mathbf{r_S} - \mathbf{r_E}|^3}(\mathbf{r_S} - \mathbf{r_E}) - G\frac{M_{Jupiter} M_{\text{Earth}}}{|\mathbf{r_J} - \mathbf{r_E}|^3}(\mathbf{r_J} - \mathbf{r_E}) \end{aligned}$$

where $\mathbf{r_S} - \mathbf{r_E}$ is the positional vector between the Sun and Earth, $M_{Jupiter}$ is the mass of Jupiter, and $\mathbf{r_J} - \mathbf{r_E}$ is the positional vector between Jupiter and Earth. This can easily be generalized as:

$$F_{ij} = -\sum_{j=1, j\neq i}^{n} G\frac{M_i M_j}{|\mathbf{r_j} - \mathbf{r_i}|^3}(\mathbf{r_j} - \mathbf{r_i}) \tag{2}$$

where M_i is the mass of the planet receiving the forces from planets with mass M_j, n is the number of planets, $\mathbf{r_j} - \mathbf{r_i}$ are the positional vectors from the planets j to planet i and $|\mathbf{r_j} - \mathbf{r_i}|$ are the lengths of the positional vectors, the distances between the planets j and planet i. From this we can easily implement as many celestial objects as we desire.

2.3 Object orientation

In order to generalize the code for a many-body problem using equation 2 we modify it using object orientation. The structure of the code can be briefly outlined by the following summary of the classes we have implemented in `C++` (see our GitHub repository [1] for more details):

- CelestialBody: Contains a constructor for the celestial object with all the associated values: position, velocity, mass, force and energy.
- SolarSystem: Contains all the CelestialBody instances added and functions which manage calculation of force and energy and writing to file.
- Solver: Contains the algorithms for solving the ODE's.
- Force: Contains needed force formulas: Newtons gravitational force, inverse beta and General relativistic correction to Newtons gravitational force. The calculation of energy is placed here as well as the potential energy is related to the force.

The way we deal with the many problem is to run a double for-loop in order to calculate the force between every celestial body. This loop is placed in the SolarSystem class and are called by the solvers for each timestep.

It can look something like this:

Algorithm 1: Calculation of force and energy for a many body system.

```
List of bodies = bodies; Number of bodies = N;
for (i = 0; i < N; i++) do
    body1 = bodies[i];
    for (j = i+1; j < N; j++) do
        body2 = bodies[j];
        //Call force class for calculation of force and energy;
        Force(body1, body2);
        Force(body2, body1);
    end
end
```

The calling of the force class will then link to the chosen force model and calculate the force and energy on body1 with respect to body2 and likewise for body2 with respect to body1. See Appendix for details about the calculation of energy. The values are added to the existing one (not replaced) so we can gather the contribution from every body. Therefore we also have to reset this between every time-step.

2.4 Mercury perihelion precession

As a test of general relativity we want to reproduce the perihelion precession of Mercury around the Sun. The observed precession deviates with 43" per century from the predictions made from the Newtonian gravitational force and all other classical effects. By using a general relativistic correction the force becomes:

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

where M_{Mercury} is the mass of Mercury, r is the distance between Mercury and the Sun, $l = |\mathbf{r} \times \mathbf{v}|$, is the magnitude of Mercury's orbital angular momentum per unit mass, and c is the speed of light in vacuum. We are going to run a simulation of Mercury's orbit starting from the perihelion (position closest to the sun) using the initial values:

$$\mathbf{r} = (0.3075 \text{ AU}, 0, 0), \qquad \mathbf{v} = (0, 12.44 \text{ AU/yr}, 0)$$

with the Sun fixed at the origin. We will then run the simulation for a 100 years. From the Python module SciPy we import a function called scipy.signal.argrelextrema[7] in order to find local minema on the radial distance data. This makes us able to determine the x,y position for every time Mercury passes the perihelion position. We can then calculate the angle between the x-axis and perihelion using:

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

We do this with and without the general relativistic correction and compute the difference. If the time-interval dt is sufficiently small we expect the value from the classical case to be almost zero because we have no disturbing planets in this simulation. We expect the value in the relativistic case to approach the observed difference of 43".

3 Results

3.1 Earth-Sun System

3.1.1 Comparing solvers

In the initial setup we have the Sun fixed at the origin and Earth at position $\mathbf{r} = (1 \text{ AU}, 0, 0)$ and velocity $\mathbf{v} = (0, 2\pi \text{ AU/yr}, 0)$ in order to get a circular orbit in the x,y-plane. We then simulated the motion of the Earth for $T = 10$ yr with $dt = 0.001$ yr for both solvers shown in figure 1. The results are produced with the executables "EarthSun_Euler.exe" and "EarthSun_Verlet.exe" [6].

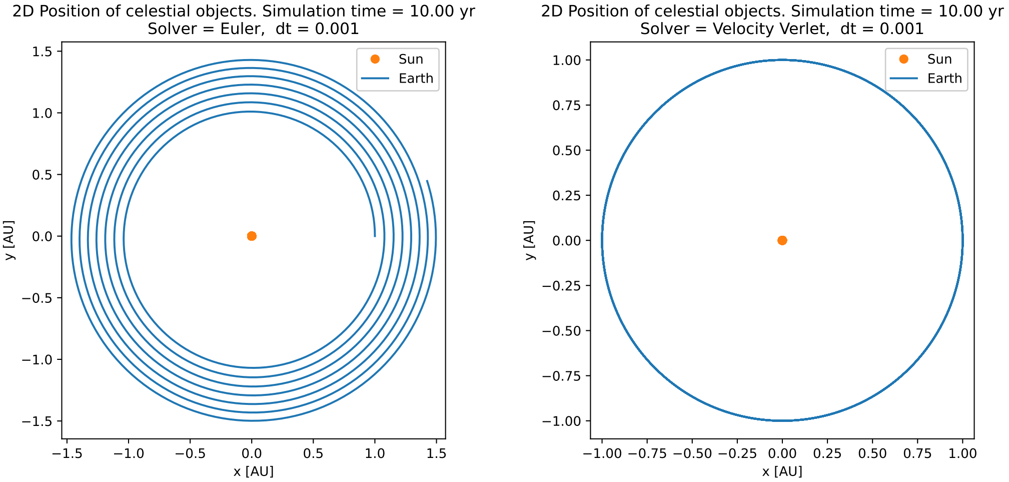


Figure 1: Simulation of Earth-Sun system for $T = 10$ yr, $dt = 0.001$ 1/yr with the Sun fixied at the origin. Shows better stability for the Velocity Verlet salgorithm than the Euler algorithm

From this test we see quite a big difference in the stability of the two solutions. We then ran the same 10 year simulation for both solvers for $dt \in [0.1, 10^{-5}]$. Since we retain the initial values for a circular orbit we expect the Earth to return to its original position after any whole number of years. Therefore we can calculate the positional error as the difference between the first and last positions. In addition we calculate the difference in the total mechanical energy as this should also be conserved. The results are produced with the executables "EarthSun_Euler.exe" and "EarthSun_Verlet.exe" and shown in figure 2.

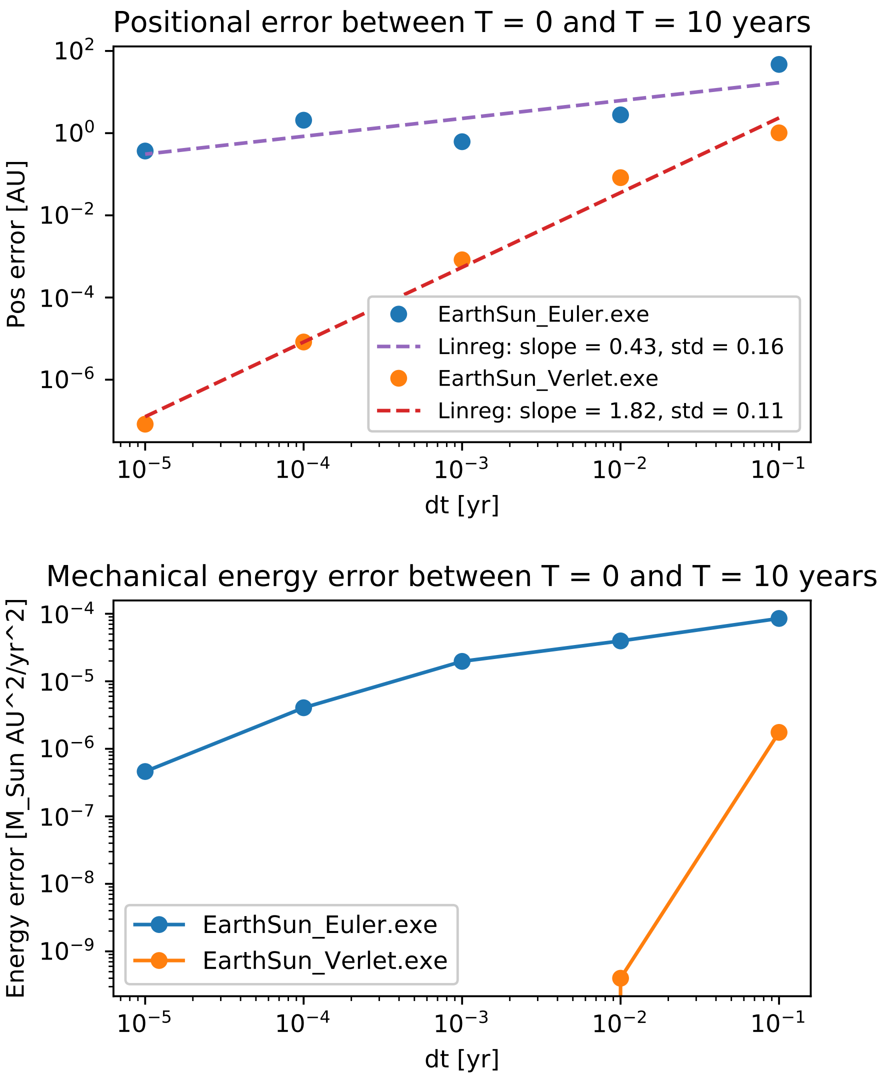


Figure 2: Plot of the relation between positional error and total mechanical error with respect to time.step dt. The simulation is initialized to be circular and the Sun is fixed at the origin. We see clear differences in the performance of the two algorithms.

We also ran a timing of the two algorithms while simulation the circular Earth-Sun system. For this purpose we disabled the writing to file and did not use any compiler flags for speed up. The result are produced with the executables "Euler_timing.exe" and "Verlet_timing.exe" and showed in figure 3.

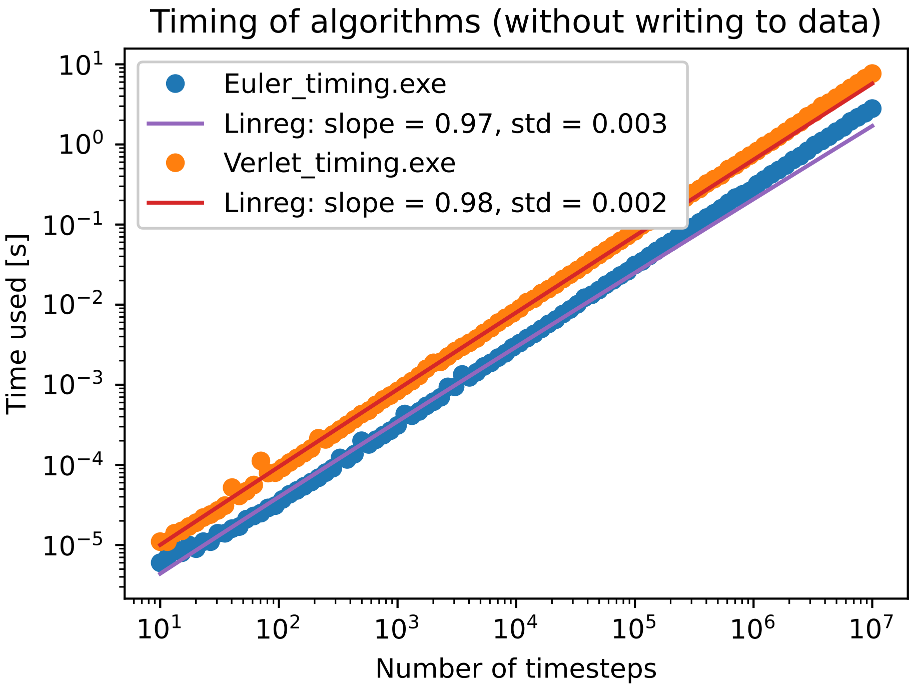


Figure 3: Timing of the two algorithms for a 10 year simulation of the circular Earth-Sun system with dt = 0.001. Writing to file is disabled and no compile flags used.

3.1.2 Testing the inverse β force

We now introduce the inverse beta force to the Earth-Sun system. We keep the sun fixed but now we initialize an elliptical orbit with Earth at position $\mathbf{r} = (1 \text{ AU}, 0, 0)$ and velocity $\mathbf{v} = (0, 5 \text{ AU/yr}, 0)$. First we simulate the position over 10 years for 16 evenly spread out values of $\beta \in [2, 3]$ (see figure 4). The following results in this section are all produced with the executable "EarthSun_InverseBeta_ellipse.exe".

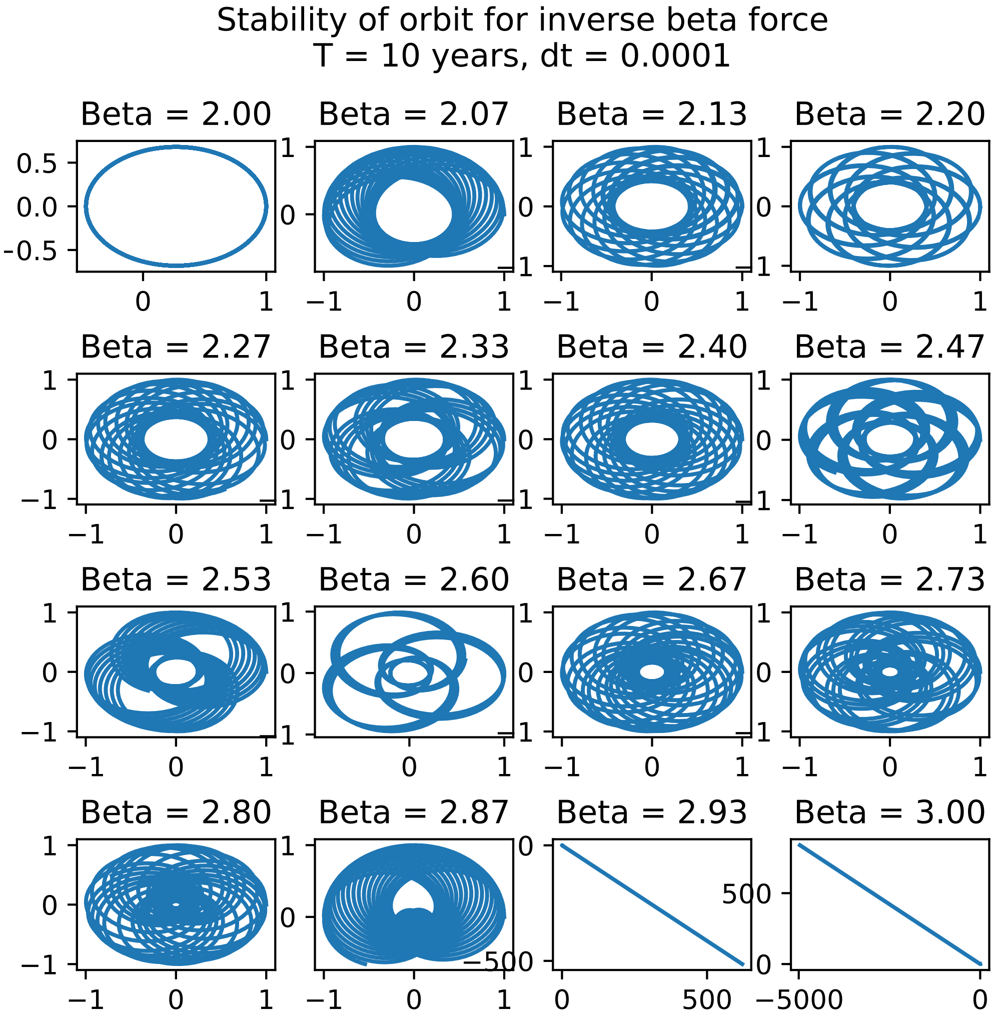


Figure 4: Positional display of the elliptical orbit of the Earth-Sun system for the inverse beta force for different values of β. The axis show x,y position in AU, but this is removed to make is easier to see the results.

We then looked at the energy difference between the first and last time-step of the elliptical orbit with inverse beta force. We reduced the run time to just a half year but increased the number of points for β to 1001. The result are showed in figure 5.

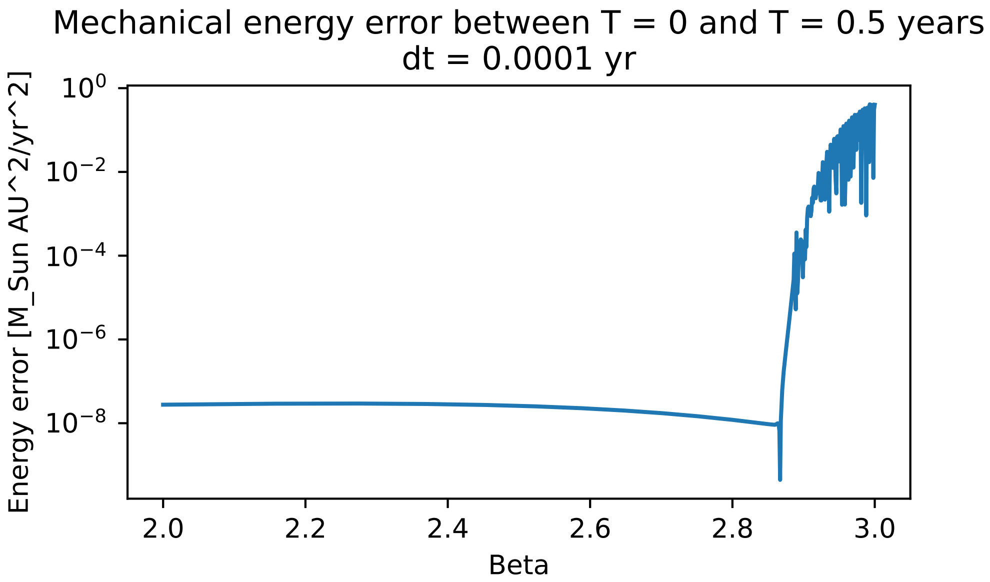


Figure 5: Energy error bewteen T = 0 and T = 0.5 years for the elliptical Earth-Sun system with inverse beta force.

Since the energy error seem to increase around $\beta = [2.87, 2.89]$ we took a closer look at the radial distance between the Earth and the Sun in that interval. This is showed in figure 6.

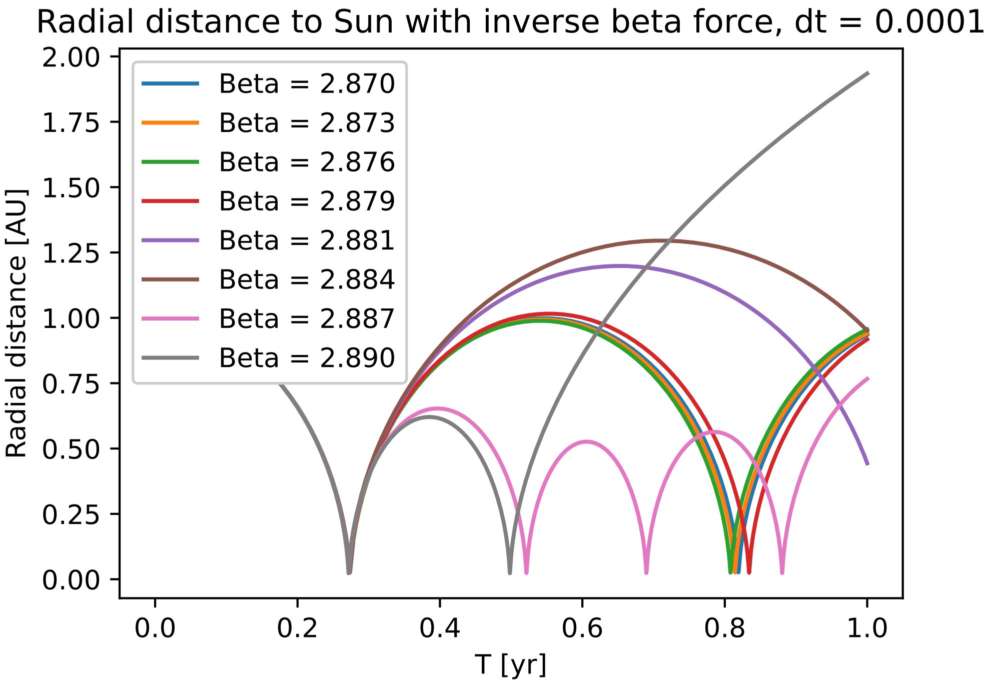


Figure 6: The radial distance between the Earth and the Sun for the elliptical Earth-Sun system with inverse beta force.

Note that the precise tipping point for the stability loss seemed to be dependent on the choice of dt. While the tipping point shown here for $dt = 0.0001$ is in the interval $\beta \in [2.887, 2.890]$ this was in the interval $\beta \in [2.876, 2.879]$ for $dt = 0.001$. On the other hand the angular momentum was constant for any choice of $beta \in [2, 3]$ (se figure 7).

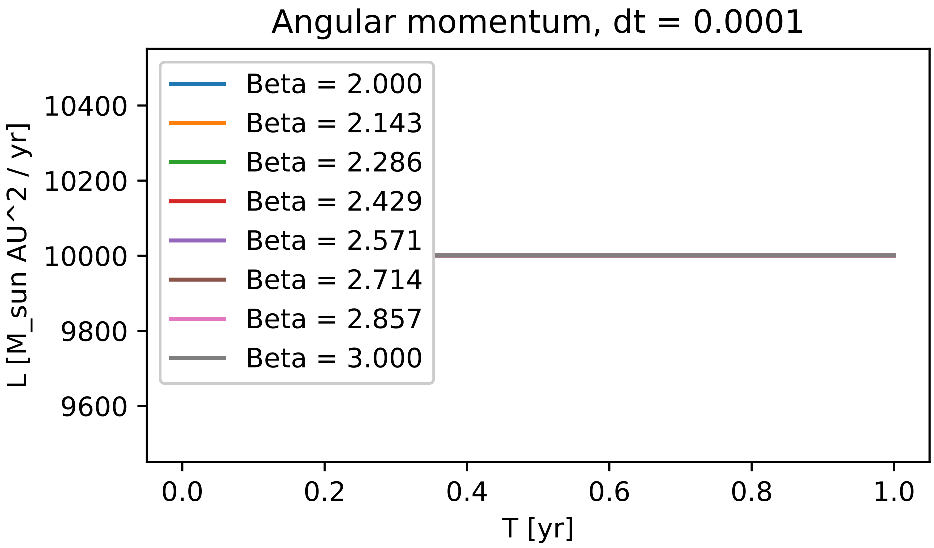


Figure 7: The angular momentum of the Earth with respect to the Sun fixed at the origin for the elliptical Earth-Sun system with the inverse beta force.

3.1.3 Escape velocity

With our knowledge of the analytical escape velocity we ran a test with starting velocity close to the analytical answer and increased the velocity by 0.0001 AU/yr each time the speed was found to be insufficient to satisfy the 1000 years escape criteria. From this we got:

$$v_{escape} = 8.8785 \pm 0.0001 \qquad \text{relative error} = 0.0818\%$$

3.2 Earth-Sun-Jupiter system

Here we have introduced Jupiter to the elliptical Earth-Sun system. We got the initial conditions for Earth and Jupiter from NASA JPL's HORIZON web-interface [5]. The time of initial conditions were: date, time of day = October 23rd 2020, 00:00:00.0000. The Sun was fixed at the origin so we could compare the stability to the Earth-Sun system. For further investigation we decided to create two additional systems, were the mass of Jupiter were 10- or 1000 times its original mass. The results from the Earth-Sun system and Earth-Sun-Jupiter systems are shown side by side on figures 8, 9, 10 and 11 below:

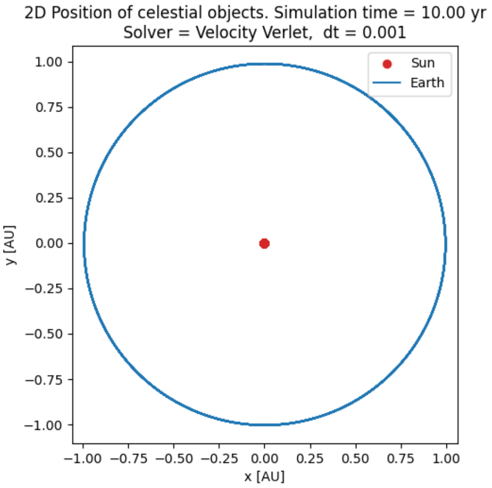


Figure 8: Earth-Sun system.

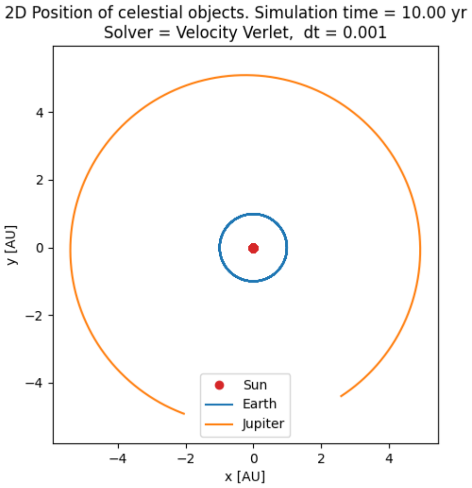


Figure 9: Earth-Sun-Jupiter system.

We then tried increasing Jupiter's mass by a factor 10 and 100 respectively (see figure 10 and 11.

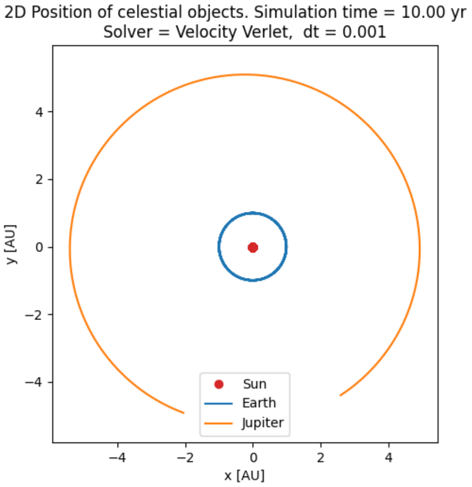


Figure 10: Earth-Sun-Jupiter system, where Jupiter's mass is 10 times it's original mass.

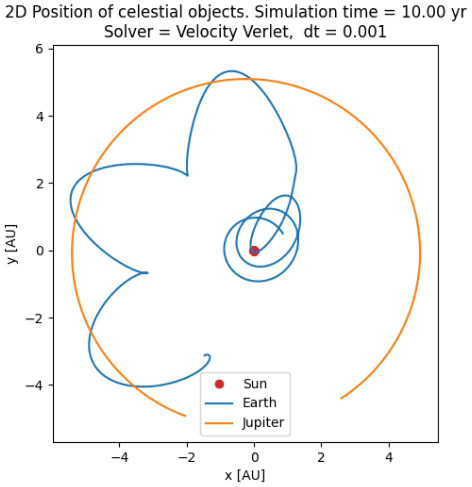


Figure 11: Earth-Sun-Jupiter system, where Jupiter's mass is 1000 times it's original mass.

The positional and mechanical energy stability of the systems:

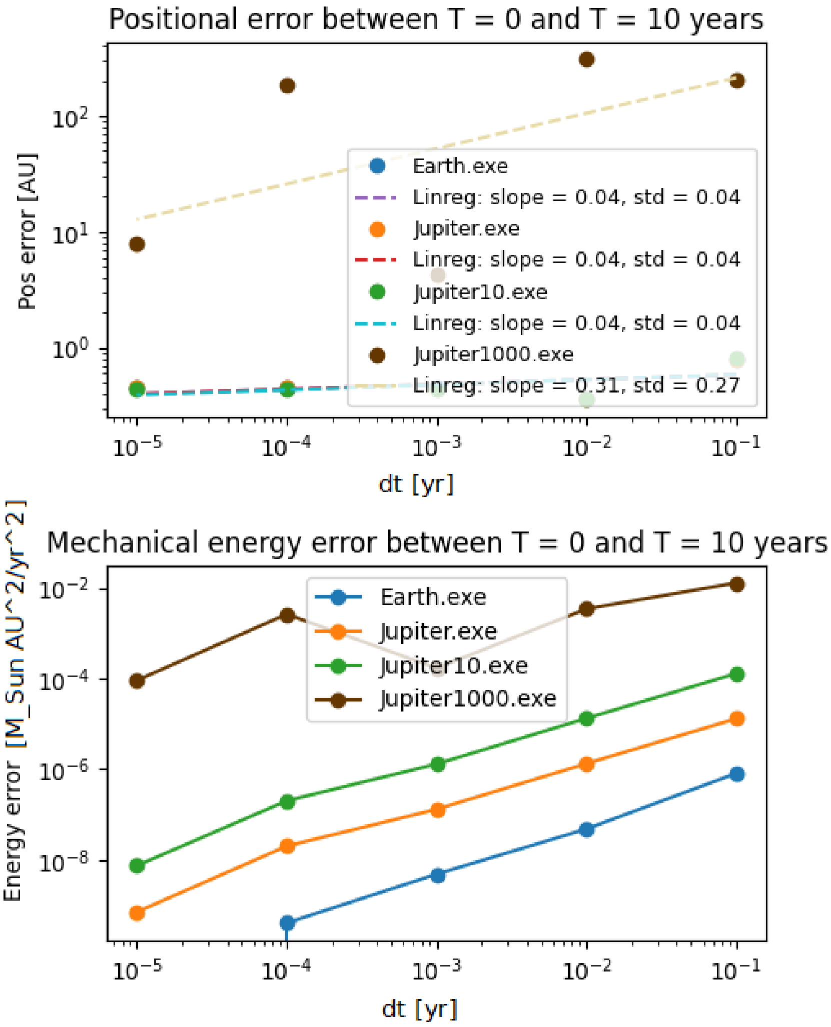


Figure 12: Positional and energy stability of the different systems.

We can see that all the systems, except for the system where Jupiter is 1000 times its original mass, are relatively equal by looking at the slope-values for the linear fit on the positional error or by looking at the data-points which nearly overlap. However, the positional error is quite large in itself at $\approx 10^{-1/2}$, still nothing compared to the 1000 mass Jupiter system with errors $\approx 10^{1,2}$. We also see that the mechanical energy error of the system gets worse as Jupiter's mass increases. These results of the Earth-Sun-Jupiter system were computed with the executables "Earth.exe", "Jupiter.exe", "Jupiter10.exe" and "Jupiter1000.exe". The positional plots had a time-step $dt = 0.001$ yr, number of steps 10000 making the total time $Time = 10$ yr.

3.3 All planets system

3.3.1 Earth-Sun-Jupiter system, Sun not fixed

We made a similar simulation where the Sun was not fixed at the origin, but had initial position from JPLs HORIZON. We set the initial velocity of the Sun equal to the sum of the Earth's and Jupiter's momentum divided by its mass to prevent the origin from drifting. We got the following 2D positional plots showed in figure 13 and 14.

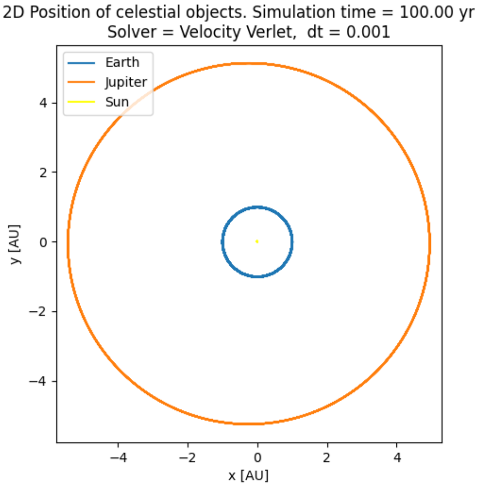


Figure 13: Earth-Sun-Jupiter system, where the Sun's position is not fixed at the origin.

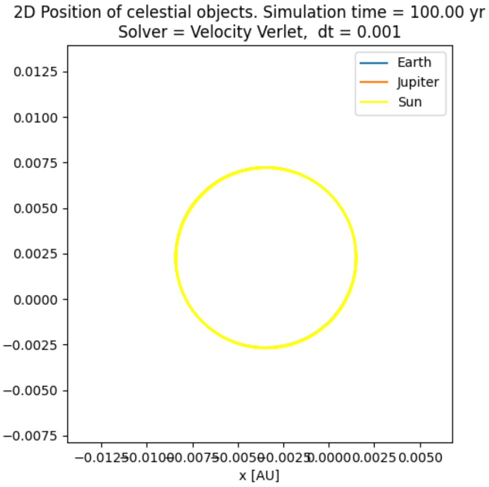


Figure 14: Close-up of the not fixed Earth-Sun-Jupiter system, showing a stable Sun orbit.

3.3.2 All planets system

Finally we simulated the All planets system consisting of the Sun and all the planets, including Pluto. The position of the sun and initial conditions of the planets and Pluto were taken from JPLs HORIZON, while the Sun's initial velocity was set equal to the sum of all the momenta divided by its mass. The 2D positional plots from this simulaion are shown in figure 15 and 16:

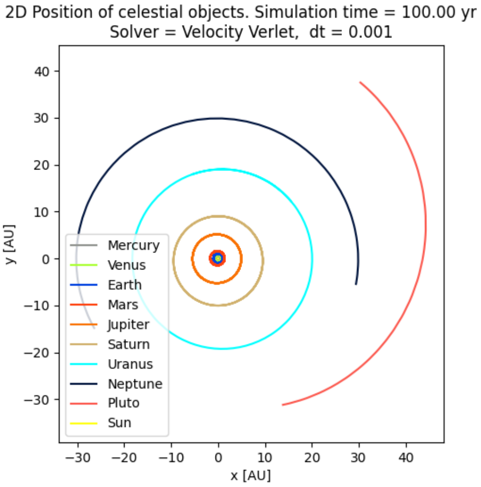


Figure 15: All planets system. The Sun's position is not fixed at the origin.

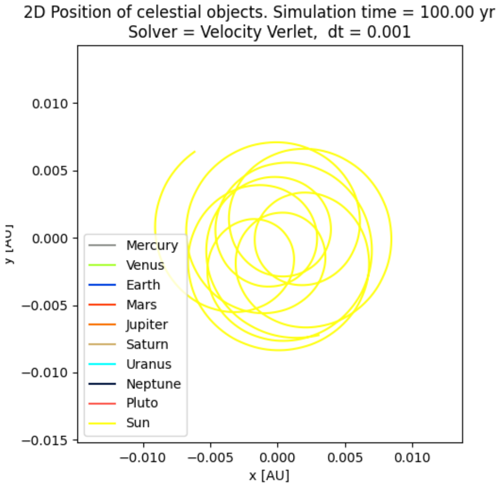
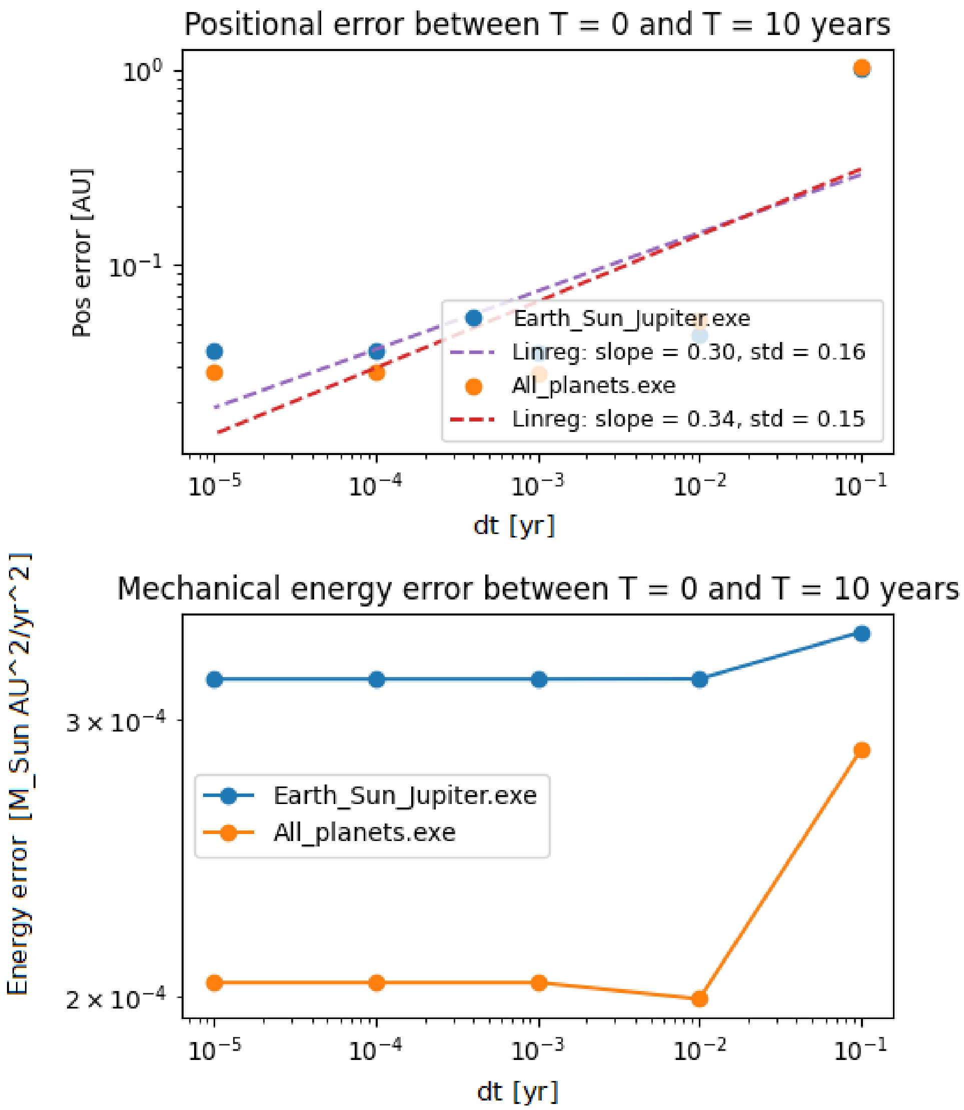


Figure 16: Close-up of the All planets system, showing an unstable Sun orbit.

The positional and mechanical energy stability of the systems are shown in figure 17


Positional error between T = 0 and T = 10 years

Mechanical energy error between T = 0 and T = 10 years

Figure 17: Positional stability of the different systems.

We can see that the positional error does not decrease much as dt decreases, and that the error in itself is quite large at $\approx 10^{-1}$m. The mechanical energy of the system remains almost constant as a function of dt and is more precise compared to the position. These results of the Earth-Sun-Jupiter system and the All planets system were computed with the executables "Earth_Sun_Jupiter.exe" and "All_planets.exe". The positional plots had a time-step $dt = 0.001$ yr, number of steps 100000 making the total time $T = 100$ yr.

3.4 Mercury perihelion precession

We ran the Mercury-Sun simulation for for 100 years with dt = 1e-6 and the Sun fixed at the origin. The result is shown on figure 18 and is produced with the executable "MercurySun_precession_correction.exe".

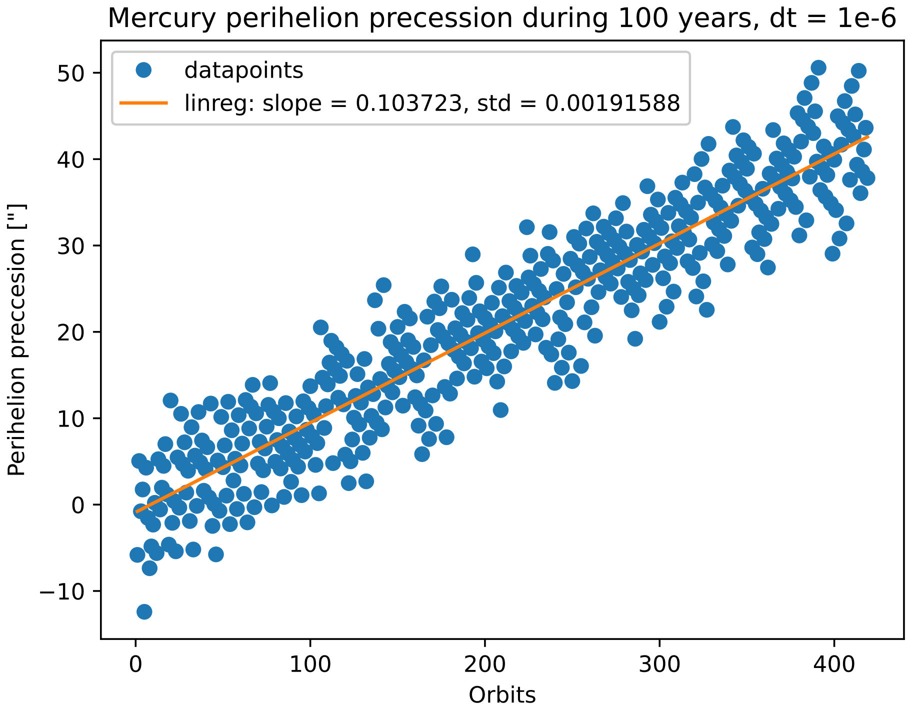


Figure 18: Angle between perihelion and x-axis (starting point) for each orbit of the Mercury-Sun system with the corrected gravitational force during a 100 years period with 419 complete orbits. We see that the angle slowly increases.

From our results we get that the perihelion precession per orbit is $p_{\text{orbits}} = 0.104 \pm 0.002$ "/orbit. This were fitted over 419 complete orbits during the exact time 99.90366 years. We can calculate the estimated precession pr. century (cy) as:

$$p_{cy} = p_{\text{orbit}} \cdot \frac{419 \text{ orbits}}{99.90366 \text{ yr}} 100 \text{ yr} = 43.5 \pm 0.8 \text{ ''}$$

The same simulation with the standard Newtonian gravitational force produced with "MercurySun_precession_newton.exe" yielded the results shown on figure 19.

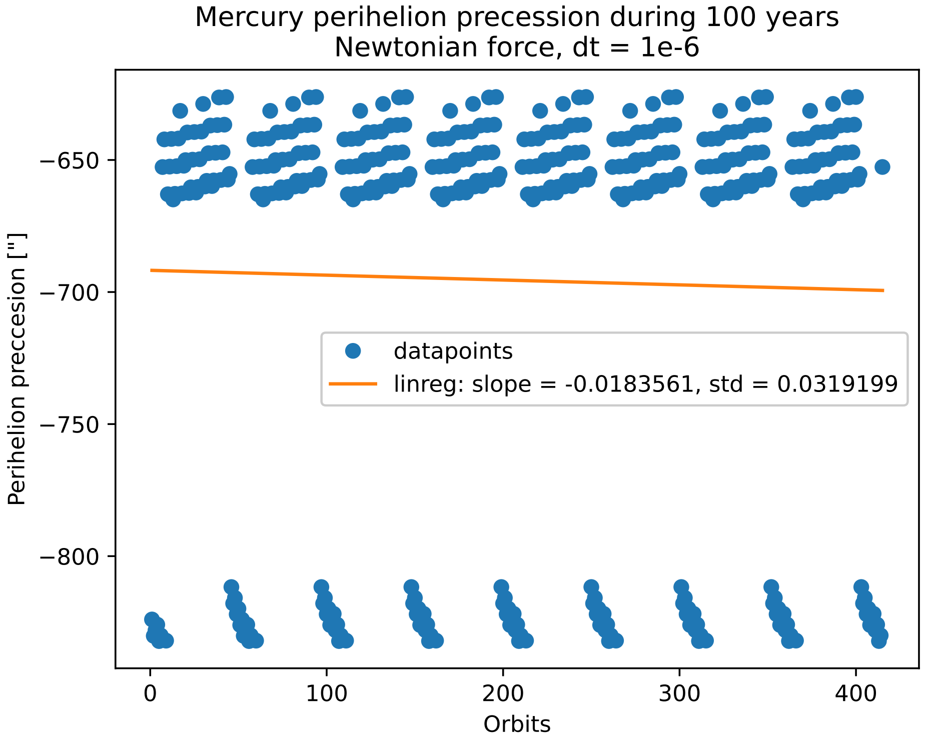


Figure 19: Mercury precession with identical conditions as showed in figure 18 expect that we used the standard Newtonian force in this simulation.

The results from the simulation with the Newtonian force turned out to be worse than expected, but we did not have time to run double check with a new executable. The algorithm for detecting the local minema only found 415 orbits, and thus we get:

$$p_{cy} = -8 \pm 13 \text{ ''}$$

If we use this result the difference in precession becomes:

$$\Delta P_{cy} = 51.5 \pm 13$$

4 Discussion

4.1 Earth-Sun system

We see that we are able to produce stable orbits with conservation of mechanical energy and angular momentum if the step-length dt is sufficiently small. Conservation of energy was expected since gravity is a conservative force, and the angular momentum should be conserved as there is no torque on the system. We have seen this numerically in figures 2 and 7. As expected we also found that the errors in position (with respect to the expected value from a circular orbit) and the mechanical energy decreases as the step length dt decreases. This is of course supported by the idea that a smaller step length will grant more precise calculations of the motion of the celestial objects, which will deviate less and less from the analytical solution.

4.1.1 Comparing solvers

From figure 2 and 3 we generally see that the velocity Verlet is superior in numerical stability while using about the same amount of time. From the results we found that the error and time usage of the two algorithms behaves as shown in table 1.

Table 1: Numerical performance of the Euler and Velocity Verlet method as function of step length dt and total number of time steps N

Sovler	**Global error**	**Time**
Euler	$\mathcal{O}(dt^{0.4})$	$\mathcal{O}(N^{0.97})$
Velocity Verlet	$\mathcal{O}(dt^{1.8})$	$\mathcal{O}(N^{0.98})$

From a theoretical standpoint we expected the global error to be $\mathcal{O}(dt^1)$ for the Euler method and $\mathcal{O}(dt^2)$ for the Verlet method respectively. So in practice the Euler turned out to be even worse while the Velocity Verlet was just slightly worse than the expected value. This can perhaps be explained by the fact that the Velocity Verlet is a symplectic ODE as opposed to the Euler method. This characteristic is clearly showcased in figure 2 where we see that the energy error decreases drastically towards zero in the velocity Verlet for decreasing dt while the Euler method still carried a sizable energy error at dt $= 10^{-5}$.

4.1.2 Inverse β force

From the simulations with the Inverse β force we clearly saw that a change in β affected the stability of the system. This is very well showcased in figure 4 where we see various different orbits with elliptical motion. As we approached $\beta = 3$ the system became unstable and the Earth was thrown out of orbit. We tried to narrow down this limit by looking at the energy conservation and the radial distance. With the use of dt $= 0.0001$ we found the instability limit to be in the interval $\beta \in [2.887, 2.890]$. But this limit was found to be dependent of dt, such that a smaller dt pushed the instability point further towards 3. When looking at the radial distance we see that the orbit became very elliptical, and the small distance to the Sun (high force) and big velocities might lead to numerical errors.

This also explains the sudden loss of mechanical energy and the fact that the Earth was able to escape the gravitational pull from the Sun, as shown in figure 4 for $\beta = 2.93$ and 3.00. Therefore we can't say whether the instability limit is connected to the nature of the force or just pure numerical errors. On the other hand we can tell that the amount nature deviates from a perfect inverse-square law is extremely small, as only values of $\beta \approx 2.00$ were similar to Earth's orbit. Regarding the observation of constant angular momentum, we know that this comes from the fact that the torque on the system is always 0 for a two-body system since the force unit-vector is the same regardless of β.

4.1.3 Escape velocity

Our numerical calculations approached the analytical value for the escape velocity with only a small relative error of 0.0818%. We saw that our numerical value was a bit smaller than the analytical value which can be explained by the nature of our calculation. As we approach the analytical escape velocity the time it takes for the Earth to turn around increases towards infinity. Since we used a finite simulation time of 1000 years it is clear that our result will come short of the exact value in some degree. Therefore we have gained very clear evidence that our system is correctly mimicking the analytical properties of the escape velocity.

4.2 Earth-Sun-Jupiter system

We see obvious differences in the positional error between the circular Earth-Sun system, figure 1, and the slightly elliptical system, figure 8. This most likely stems the fact that the error calculation assumes that the planet returns to its original position every year (circular orbit), but the period of elliptical orbits are not equal to one year as the case of circular orbits (see appendix). The data regarding the positional error are therefore not useful in reflecting the absolute error but can be used for comparison internally within the same simulation conditions. We see that Jupiter in itself has little impact on the Earth-Sun system except for when we multiplied its mass with a factor 1000 making it almost as heavy as the Sun. This is not strange as Jupiter originally have a mass of about 1/1000 the mass of the Sun, resulting in Jupiter contributing 0.1% of the total gravitational force on Earth. When Jupiter's mass is scaled by a factor of 1000 it will be about the same mass as the Sun, resulting in weird orbits as seen in figure 11. The reason for the big positional inaccuracies here is a result of the unstable orbit, as Earth does not even get close its orbit the previous year.

4.3 All planets system

In our first simulation we can see that the Earth-Sun-Jupiter system is stable. It is obvious from figure 16 that the Sun is unstable when all the other planets and Pluto are included. This most likely comes from the fact that the small errors in the forces adds up as the number of celestial objects increases. If that is the case this can be mitigated by lowering the step length (be wary of round off errors) or by implementing a symplectic method with a smaller global error than the Velocity Verlet. Another possibility is that the initialization of the Sun's velocity is wrong, but that is unlikely as it is initialized the same way as the Earth-Sun-Jupiter system.

4.4 Mercury perihelion precession

From the result of the simulation with the corrected gravitational force we get a very promising result. Since we have an isolated system without any disturbances from other planets we expected the simulation with the pure Newtonian gravitational force to yield $p_{cy} \approx 0$. This would have granted the a difference in precession of 43.5 ± 0.8 " which match the real life observed value of 43". When taking the unexpected uncertain result from the simulation with the Newtonian gravitational into account we get $P_{cy} = 51.5 \pm 13$ which still agrees with the observed value with respect to the uncertainty. If we look at the axis on figure 19 from the the Newtonian simulation we see that the angle doesn't change much but that it fluctuates roughly between -600" and -850". This displacement indicates that their might be something wrong with the initialization of the system, or the software for loacting the position of the perihelion each year. As stated, we did not have time to rerun this, but based on the excellent results from the first simulation (with the corrected force) it seems plausible that the Newtonian simulation is somehow flawed.

5 Conclusion

From the simulations of the Earth-Sun system we successfully achieved stable orbits with low to no loss of mechanical energy and a conserved angular momentum. We saw that the velocity Verlet method is the superior ODE-solver for this particular problem, as it is symplectic (energy conserving) and has a smaller global error compared to the forward Euler method, while using almost the same run time. When we swapped the gravitational force with the inverse β force we saw a clear change in the shape of the otherwise elliptical orbit. We found a limit in the interval $\beta \in [2.887, 2.890]$ where any higher choice of β gave unstable results as the Earth was thrown out of orbit. The mechanical energy was also conserved until this point, where the Earth escaped the gravitational pull from the Sun. The reason the Earth escaped might be related to numerical errors as the Earth got too close to the Sun, which resulted in a velocity higher than the escape velocity. The total angular momentum was conserved as the force unit-vector did not change.

When including Jupiter to the Earth-Sun system with Newtonian gravity we still got stable results. We had to increase Jupiter's mass by a factor 1000 (giving it a mass on the same scale as the Sun) in order to lose stability. In the All planets system we saw that the Sun's orbit was stable when we simulated the system consisting of the Sun, Earth and Jupiter, but became unstable as the number of celestial objects increased. Therefore we found the precision of the motion of the planets to be directly linked to the number of planets simulated. Lastly regarding the precession of Mercury around the Sun we got mixed results. While the simulation with corrected gravitational force granted a precession of $p_{cy} = 43.5 \pm 0.8''$, the classical simulation deviated from the expected result and our final result was reduces to the estimate of $P_{cy} = 51.5 \pm 13$. Our result does not disagree with the real life observation 43" but our uncertainty makes it difficult to clearly support it.

References

[1] Hoftun F. and Metzsch-Jensen M. (2020), *Project 3 - GitHub repository*, Available at: https://github.com/mikkelme/project3_FYS3150

[2] Univeristy of Oslo, Department of Physics (2020) *Project 3 description*, Available at: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2020/Project3/pdf/Project3.pdf

[3] Verlet, Loup (1967) *Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.* Physical Review. 159 (1): page 98–103

[4] Hansen F. K. (2008) *AST1100 Lecture notes - Celestial Mechanics*, Availabale at: https://www.uio.no/studier/emner/matnat/astro/AST1100/h08/undervisningsmateriale/lecture1_v2.pdf

[5] NASA, JPL (2020) *HORIZONS web-interface*, Available at: https://ssd.jpl.nasa.gov/horizons.cgi, Accessed October 23rd 2020.

[6] Hoftun F. Metzsch-Jensen M. (2020) *Project 3 - Test-files*, Available at: https://github.com/mikkelme/project3_FYS3150/tree/master/test_files

[7] SciPy (2020) *scipy.signal.argrelextrema documentation*, Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.argrelextrema.html

Appendix

GitHub repository: https://github.com/mikkelme/project3_FYS3150

5.1 Orbital period derivation

For circular orbits we have:

$$v = \sqrt{\frac{GM}{r}}$$

Using $v = r\omega$ and $T = \frac{2\pi}{\omega}$

$$\omega = \sqrt{\frac{GM}{r^3}}, \quad T = 2\pi\sqrt{\frac{r^3}{GM}}$$

For the circular Earth-Sun system this is of course equal to one year. However, for elliptical orbits, the r^3 in the right-most relation gets swapped out with a^3, where a is the semi-major axis so $T \neq 1$ (if $r = 1$), in fact $T_{ellipse} > T_{circle}$ as $a > r$.

5.2 Calculation of energy

$$E_{kin} = \frac{1}{2}mv^2$$

$$E_{pot} = -\int_{\infty}^{r} F(r')dr'$$

For Newtons gravitational force we get:

$$E_{pot} = -\int_{\infty}^{r} -\frac{GmM}{r'^2}dr' = -\frac{GmM}{r}$$

For conservative forces, lik gravity, the mechanical energy is conserved

$$E_{mec} = E_{kin} + E_{pot} = constant$$