Project 3
FYS3150

Ann-Silje Kirkvik

github.com/annsilje/fys3150

October 23, 2016

Abstract

This project models the orbits of the planets and the Sun in the solar system by numerically solving coupled ordinary differential equations describing the motion of the celestial bodies based on the gravitational force. For the Earth-Sun system the numerical algorithms Forward Euler and Velocity Verlet has been tested. The Velocity Verlet algorithm turned out to perform significantly better than the Forward Euler algorithm. The tree-body problem of the Earth-Jupiter-Sun system has been simulated with different masses for the planet Jupiter, and the 9-body problem of the solar system has been solved. The Sun is the by far most important contributor to the planets orbits, but the planets also have an effect on the motion of the Sun, which again affects the motion of the planets.. Finally, a modified Newtonian gravitational force with a correction to account for the effects of general relativity is tested to investigate the impact on Mercury's orbit. The discrepancy between the expected precession of Mercury's orbit from the classical newtonian gravitational force of the other celestial bodies in the solar system and the observed value can be reproduced with the modified Newtonian gravitational force.

1 Introduction

This project models the orbits of the planets and the Sun in the solar system by numerically solving coupled ordinary differential equations describing the motion of the celestial bodies based on the gravitational force. To solve these equations numerically the Forward Euler and Velocity Verlet methods are tested, with emphasis on the latter. Several scenarios are tested, ranging from the two-body problem of the Earth-Sun system to the 9-body problem of all planets and the star of the solar system. In addition, a modified gravitational force model is tested for the two-body problem of the Mercury-Sun system to look at the effects of general relativity on the planet's orbit. Section 2 describes the theoretical background of the project and implementation. Section 3 shows the results of the numerical experiments and section 4 has some final remarks.

2 Description

Newton's law of gravitation describes the gravitational force exerted on a body by another body and is given by

$$\boldsymbol{F}_{12} = \frac{Gm_1m_2}{r^3}\boldsymbol{r} \tag{1}$$

where G is the gravitational constant, m_1 and m_2 are the masses of the two bodies involved, $\boldsymbol{r} = \boldsymbol{x_2} - \boldsymbol{x_1}$, where $\boldsymbol{x}_k$ is the position vector of body k and r is the relative distance between the bodies. By applying Newton's second and third laws of motion, $\Sigma \boldsymbol{F} = m\ddot{\boldsymbol{x}}$ and $\boldsymbol{F}_{12} = -\boldsymbol{F}_{21}$, the following second order differential equations are obtained

$$\frac{d^2\boldsymbol{x}_1}{dt^2} = \frac{\boldsymbol{F}_{12}}{m_1} \quad \text{and} \quad \frac{d^2\boldsymbol{x}_2}{dt^2} = \frac{\boldsymbol{F}_{21}}{m_2} \tag{2}$$

where each equation can be rewritten into a set of first order differential equations:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v} \quad \text{and} \quad \frac{d\boldsymbol{v}}{dt} = \frac{\boldsymbol{F}}{m} \tag{3}$$

For a two-body problem in three dimensions this leads to 12 coupled first order differential equations. For the three-body problem in three dimensions the number of equations increases to 18. In general, there are 6 coupled first order differential equations for each body. For the 9-body problem of the solar system this becomes 54 equations. Each equation also has an initial condition for the time $t_0 = 0$, yielding the initial position and velocity of each body. Since the Sun is large compared to the other celestial bodies, it can be reasonable in some cases to ignore the forces the planets exert on the Sun and keep the Sun's position fixed. This will reduce the number of equations by 6 for each case.

Choosing a step size, h, and discretizing the problem with $t_{i+1} = t_i + ih$, for $i = 0, 1, ..., n$, these differential equations can be solved numerically using numerous different methods. Here, only two of these methods will be tested, namely the Forward Euler and Velocity Verlet algorithms, which is also described in [1].

2.1 Forward Euler

Making a Taylor expansion of the position and velocity vectors $\boldsymbol{x}$ and $\boldsymbol{v}$ gives

$$\boldsymbol{x}(t_{i+1}) = \boldsymbol{x}_{i+1} = \boldsymbol{x}_i + h\frac{d\boldsymbol{x}_i}{dt} + \mathcal{O}(h^2) \tag{4}$$

$$\boldsymbol{v}(t_{i+1}) = \boldsymbol{v}_{i+1} = \boldsymbol{v}_i + h\frac{d\boldsymbol{v}_i}{dt} + \mathcal{O}(h^2) \tag{5}$$

This gives the Forward Euler algorithm:

1. $\boldsymbol{a}_i = \Sigma \boldsymbol{F}_i / m$
2. $\boldsymbol{v}_{i+1} = \boldsymbol{v}_i + h\boldsymbol{a}_i$
3. $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + h\boldsymbol{v}_i$

where the acceleration $\boldsymbol{a}_i$ is determined by Newton's second law. The local error of the Forward Euler algorithm goes as $\mathcal{O}(h^2)$ and the updated positions and velocities are linear extrapolations from the previous step. The advantage of this algorithm though, is its simplicity. Updating the positions require 2 floating point operations for each dimension, and the same is required for updating the velocities.

2.2 Velocity Verlet

Again, starting with a Taylor expansion of the position and velocity vectors $\boldsymbol{x}$ and $\boldsymbol{v}$, but also the acceleration vector $\boldsymbol{a}$:

$$\boldsymbol{x}(t_{i+1}) = \boldsymbol{x}_{i+1} = \boldsymbol{x}_i + h\frac{d\boldsymbol{x}_i}{dt} + \frac{h^2}{2}\frac{d^2\boldsymbol{x}_i}{dt^2} + \mathcal{O}(h^3) \tag{6}$$

$$\boldsymbol{v}(t_{i+1}) = \boldsymbol{v}_{i+1} = \boldsymbol{v}_i + h\frac{d\boldsymbol{v}_i}{dt} + \frac{h^2}{2}\frac{d^2\boldsymbol{v}_i}{dt^2} + \mathcal{O}(h^3) \tag{7}$$

$$\boldsymbol{a}(t_{i+1}) = \boldsymbol{a}_{i+1} = \boldsymbol{a}_i + h\frac{d\boldsymbol{a}_i}{dt} + \mathcal{O}(h^2) \tag{8}$$

Solving equation 8 for $h\frac{d\boldsymbol{a}_i}{dt} = h\frac{d^2\boldsymbol{v}_i}{dt^2}$ and inserting into equation 7 gives

$$\begin{aligned}\boldsymbol{v}_{i+1} &= \boldsymbol{v}_i + h\frac{d\boldsymbol{v}_i}{dt} + \frac{h}{2}(\boldsymbol{a}_{i+1} - \boldsymbol{a}_i) + \mathcal{O}(h^3)\\ &= \boldsymbol{v}_i + \frac{h}{2}(\boldsymbol{a}_i + \boldsymbol{a}_{i+1}) + \mathcal{O}(h^3)\end{aligned} \tag{9}$$

The Velocity Verlet algorithm then becomes:

1. $\boldsymbol{a}_i = \Sigma\boldsymbol{F}_i/m$
2. $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + h\boldsymbol{v}_i + h^2\boldsymbol{a}_i/2$
3. $\boldsymbol{a}_{i+1} = \Sigma\boldsymbol{F}_{i+1}/m$
4. $\boldsymbol{v}_{i+1} = \boldsymbol{v}_i + h(\boldsymbol{a}_i + \boldsymbol{a}_{i+1})/2$

where the accelerations $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i+1}$ are determined by Newton's second law, and $\boldsymbol{F}$ depends on the position of the bodies. The Velocity Verlet algorithm has a local error that goes as $\mathcal{O}(h^3)$, which is an improvement over the Forward Euler algorithm. In addition, the updated position now also takes into account the acceleration at the previous position and the velocity is updated using the average acceleration over the time step $[t_i, t_{i+1}]$.

Updating the positions require 5 floating point operations for each dimension, and updating the velocities require 4 floating point operations for each dimension. Since the step size is constant these numbers can be reduced to 4 and 3 respectively by computing the constants $h^2/2$ and $h/2$ only once. Still, this is slightly more operations than in Forward Euler, but both algorithms scale linearly with the number of steps. The Velocity Verlet algorithm is also slightly more complicated because the accelerations and velocities from the previous step need to saved.

2.3 Scaling and assumptions

Distances in space are very large compared to here on Earth and selecting a different unit to measure distances seem appropriate. One such unit is the Astronomical Unit (AU) where 1 AU is the average distance between the Earth and the Sun, which is close to 1.5×10^{11}meter. To measure time year is selected as the unit, with 365.25 days per year and 86400 seconds per day, which is also the time the Earth takes to do one revolution around the Sun. Under the assumption of uniform circular orbits the speed and acceleration of the Earth can be expressed as

$$v = \frac{2\pi r}{T} \tag{10}$$

$$a = \frac{v^2}{r} = \frac{F}{M_{Earth}} = \frac{GM_\odot}{r^2} \tag{11}$$

Combining equation 10 and 11 yields

$$\begin{aligned} \frac{GM_\odot}{r} &= \frac{(2\pi r)^2}{T^2} \\ GM_\odot &= \frac{4\pi^2 r^3}{T^2} = 4\pi^2 \frac{AU^3}{yr^2} \end{aligned} \tag{12}$$

By selecting the unit of mass as 1 solar mass the physical constant G simply becomes $4\pi^2$ (skipping units). Applying the same conversion to the speed of light it becomes roughly $c = 63241$AU/yr. The masses of the bodies in the solar system is listed in 1. Initial positions and velocities for the these bodies can be obtained from NASA's websites [4] or [5], but the latter requires a parser library to read the files. One such library is [6].

Table 1: The mass of bodies in the solar system

Body	Mass (kg)	Mass ($M_\odot$)
Sun	2×10^{30}	1
Earth	6×10^{24}	1.20×10^{-7}
Jupiter	1.9×10^{27}	2.45×10^{-6}
Mars	6.6×10^{23}	3.00×10^{-6}
Venus	4.9×10^{24}	3.30×10^{-7}
Saturn	5.5×10^{26}	9.50×10^{-4}
Mercury	3.3×10^{23}	1.65×10^{-7}
Uranus	8.8×10^{25}	4.40×10^{-5}
Neptune	1.03×10^{26}	5.15×10^{-5}

2.4 Conserved quantities

To assess the quality of the different solvers and different time steps some basic physical laws can be used as tests. These laws are described in various basic textbooks on physics, such as [2].

As long as the system is closed and does not interact with bodies outside the system the total energy of the system should always be conserved. The only force involved is the gravitational force and the total energy, E, of the N-body system then becomes:

$$E = \sum_{i=0}^{N} \frac{1}{2} m_i v_i^2 - \sum_{i=0}^{N} \sum_{j>i}^{N} \frac{G m_i m_j}{r} \tag{13}$$

The total linear momentum $\boldsymbol{P}$ is also a conserved quantity, as long as the net force on the system is zero, and is given by:

$$\boldsymbol{P} = \sum_{i=0}^{N} m_i \boldsymbol{v}_i \tag{14}$$

Similarly, the total angular momentum $\boldsymbol{L}$ is conserved, as long as the net torque on the system is zero, and is given by:

$$\boldsymbol{L} = \sum_{i=0}^{N} m_i (\boldsymbol{r} \times \boldsymbol{v}) \tag{15}$$

under the assumption of circular motion and the celestial bodies being modeled as point masses.

2.5 Escape Velocity

A system where the total energy is less than zero is considered a bound system and the bodies will orbit each other. When the energy reaches zero, the system is no longer bound and the bodies will escape each others' gravity fields. With the Earth-Sun system, assuming the Sun is fixed at the origin and $r = 1$AU, solving equation 13 for $E = 0$ will yield the escape velocity.

$$\begin{aligned} E &= 0 \\ \frac{1}{2} M_{Earth} v_{Earth}^2 &= \frac{G M_{\odot} M_{Earth}}{r} \\ v_{Earth}^2 &= \frac{2 G M_{\odot}}{r} \\ v_{Earth} &= \sqrt{\frac{2 G M_{\odot}}{r}} \\ v_{Earth} &= \sqrt{8\pi^2} AU/yr \end{aligned} \tag{16}$$

Note that Earth's escape velocity does not depends on the mass of the Earth, only the mass of the Sun and the distance to the Sun.

2.6 General relativity

According to general relativity, space-time is distorted by large celestial bodies and the Newtonian gravity model is no longer sufficient to describe the motion of celestial bodies close to large celestial bodies. In our solar system, the Sun is large enough to have an observable effect on Mercury's orbit that is not accounted for by the classical Newtonian model. The point in the orbit where the planet has its closest approach to the Sun is called the perihelion. Removing the gravitational effect from the other planets in the solar system, the position of Mercury's perihelion is observed to have an angular velocity of $43''$ per century. By applying a correction term to the Newtonian model to account for general relativity the force model becomes:

$$\boldsymbol{F} = \frac{GM_{\odot}M_{Mercury}}{r^3}\boldsymbol{r}\left(1 + \frac{3l^2}{r^2c^2}\right) \tag{17}$$

where $\boldsymbol{r}$ is the vector between the Mercury and the Sun, r is the length of this vector, $l = |\boldsymbol{r} \times \boldsymbol{v}|$, where $\boldsymbol{v}$ is the relative speed of the two bodies and c is the speed of light. $M_{\odot}$ and $M_{Mercury}$ is the mass of the Sun and the mass of Mercury, respectively.

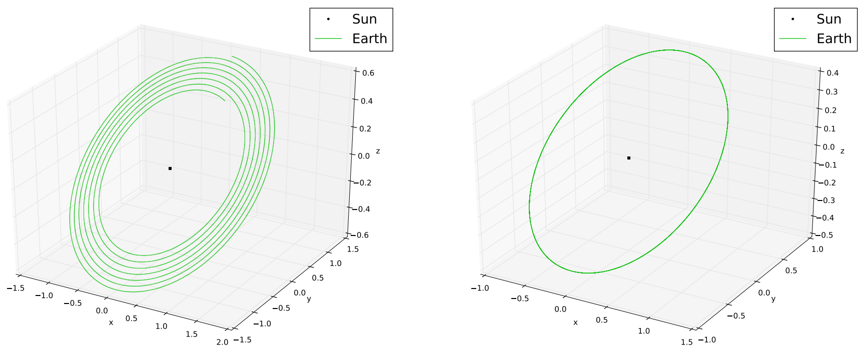
3 Results

The source code for this project is located at http://github.com/annsilje/fys3150. The implementation is based on the framework provided at [3]. Several scenarios have been tested. For the Earth-Sun system both the Forward Euler and the Velocity Verlet algorithm has been tested. The tree-body problem of the Earth-Jupiter-Sun system has been simulated with different masses for the planet Jupiter, and the 9-body problem of the solar system has been solved. Finally, the modified Newtonian gravitational force is tested to investigate the impact on Mercury's orbit.

3.1 Two-body problem

Using three-dimensional coordinates and velocities from [5] as initial conditions the Earth-Sun system has energy $E = -5.60443 \times 10^{-5} M_{\odot}AU^2/yr^2$, linear momentum $P = 2.5 \times 10^{-2} M_{\odot}AU/yr$ and angular momentum $L = 2.75 \times 10^{-5} M_{\odot}AU^2/yr$.

Running a simulation over 10 years the solution quickly start to decay using the Forward Euler algorithm. With step size $h = 10^{-2}yr$ the system's energy has increased by a factor of 1.7 after 10 years and as the Earth spirals further and further away from the Sun the angular momentum is nearly doubled. The linear momentum, however, seem to be stable, which indicated that the speed of the Earth is constant as long as the Sun is fixed. Using the same step size the Velocity Verlet algorithm performs much better. Both energy and angular momentum is conserved, but will probably require a slightly smaller step size



(a) Forward Euler (b) Velocity Verlet

Figure 1: Earth-Sun system solved with the Forward Euler algorithm in (a) and the Velocity Verlet algorithm in (b). The time step is set to 0.001 years and the simulation is run for 10 years for both algorithms.

for longer simulations. After 10 years the energy was conserved (with 5 digits) and the orbit is stable.

Figure 1 shows the Earth's orbit around the Sun, fixed to the origin, computed by the two algorithms with step size $h = 10^{-3}yr$. Despite smaller step size, the Forward Euler algorithm still computes an outward spiraling orbit, and even with step size $h = 10^{-7}yr$ it conserves energy with only 3 digits, which is fewer digits than the Velocity Verlet algorithm with step size $h = 10^{-2}yr$.

As shown in 2.1 and 2.2, the local error of the Velocity Verlet algorithm is one order of magnitude smaller than that of the Forward Euler algorithm, and is therefore expected to perform better in terms of precision. In addition, the equations better represents the physical situation by including the acceleration when updating the positions. The Forward Euler algorithm conserved linear momentum fairly well, and is therefore not too bad at updating the velocities. But the algorithm performs poorly when it comes to positions.

The Forward Euler algorithm is slightly faster than the Velocity Verlet algorithm, as expected since updating the positions and velocities requires less floating point operations. There is, however, additional calculations that are needed for both algorithms for each step, such as computing the forces and accelerations. Therefore, computing the new positions and velocities is only a part of the total number of operations needed and the difference between the two algorithms is not as large as might be expected when only comparing the difference in floating point operations of the two algorithms. Table 2 shows typical execution times for the 2-body problem using the two algorithms for different step sizes. The table indicates that the execution time scale linearly with the number of steps for the two-body problem. In addition to the computations needed to solve the problem, the positions are written to a file for each step, which also adds a constant overhead in both algorithms in execution times.

Table 2: Execution times for the Forward Euler algorithm and the Velocity Verlet algorithm for the two-body problem

n	Forward Euler	Velocity Verlet
10^2	0.003407s	0.004039s
10^3	0.012861s	0.013250s
10^4	0.078251s	0.080261s
10^5	0.550461s	0.613225s
10^6	4.283050s	4.408770s
10^7	40.80890s	42.68170s
10^8	399.5480s	426.0720s

With a fixed Sun at the origin the theoretical escape velocity, or more correctly escape speed, of the Earth was in equation 16 shown to be (skipping units) $v = \sqrt{8}\pi \approx 8.8857658763$. Figure 2 shows the simulations of the 2-body problem with the Sun fixed at the origin with different speeds closer and closer to the theoretical escape speed. For smaller speeds the Earth is bound in an elliptical orbit, but as the speed increases the planet goes into a larger and larger elliptical orbit, before it finally turns into a parabola and the Earth escapes. As expected the energy $E = -1.53653 \times 10^{-7}$ for $v = 8$ is negative, and goes towards zero as the speed goes toward the escape speed. Calculating the Earth's speed for a random sample from [5] gives a value of $v_{Earth} \approx 6.37$, which means the Earth will remain safely in orbit around the Sun.

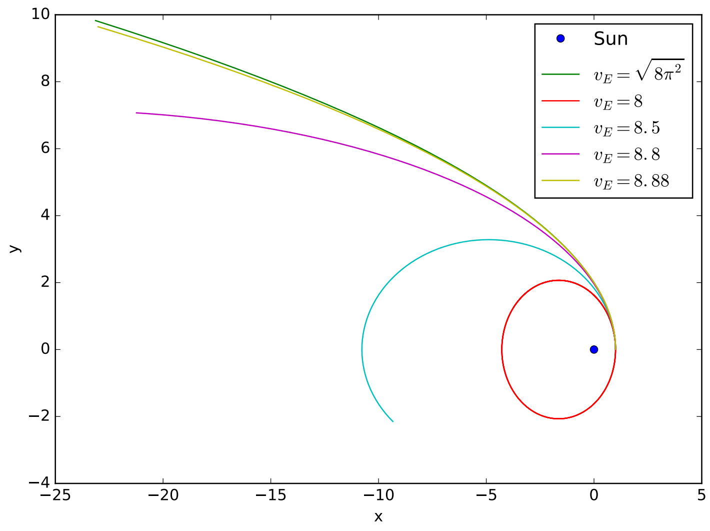


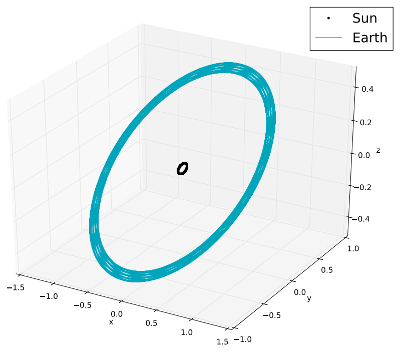
Figure 2: Earth orbits as Earth's speed approach the theoretical escape speed at $\sqrt{8}\pi$

3.2 Three-body problem

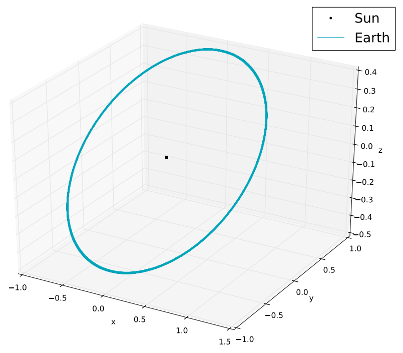
Adding Jupiter to the Earth-Sun system, the problem turns into a three-body problem. Jupiter is the largest planet in the solar system and may have an effect on the orbits of other celestial bodies. Using the Velocity Verlet algorithm the Earth-Jupiter-Sun system has been solved both with and without considering the gravitational forces acting on the Sun. The mass of Jupiter has been increased to see how it affects the orbits. Figure 3 shows the calculated orbits as the mass of Jupiter increases and figure 4a shows the three-body problem with normal mass.

As seen in figure 3b the Earth's orbit around the Sun is still stable even with the mass of Jupiter being increased by a factor of 10. The perturbations in the orbit due to Jupiter is difficult to see at this scale. However, if the forces on the Sun is also considered, the motion of the Sun due to the gravitational force from Jupiter can be observed. The Sun is approximately 1AU from the Earth while Jupiter is approximately 5.20AU from the Sun and therefore approximately 4.20AU from the Earth when the planets are aligned. The gravitational force falls off as $1/r^2$. Therefore, the main contribution to the change in the orbit of the Earth is not from Jupiter itself, but from the change in the Sun's position due to the gravitational force from Jupiter, as can be seen in figure 3a.

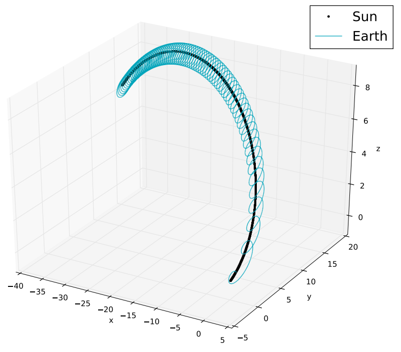
Increasing the mass of Jupiter further, with a factor of 1000, the system becomes unstable for the fixed Sun and the Earth escapes the solar system. Assuming the Sun is not affected by the gravitational force from the planets is clearly a poor approximation now that Jupiter's mass is of the same order



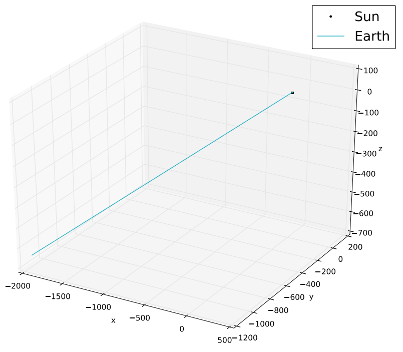
(a) Big Jupiter



(b) Big Jupiter, Sun fixed



(c) Huge Jupiter

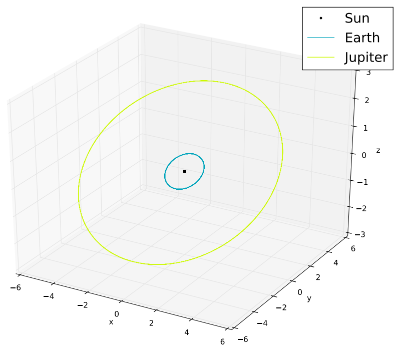


(d) Huge Jupiter, Sun fixed

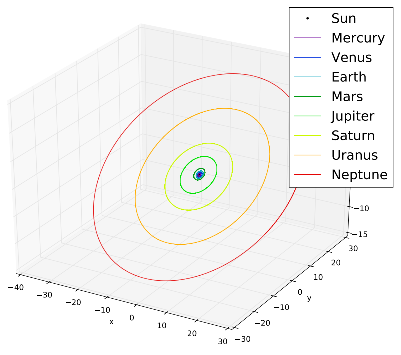
Figure 3: Earth-Jupiter-Sun system solved with the Velocity Verlet algorithm. The time step is set to 0.001 years and the simulation is run for 100 years for all cases. The orbit of Jupiter is omitted from the figures. In (a) Jupiter is 10 times its normal mass. In (b) Jupiter is 10 times its normal mass and the sun is fixed at the origin. In (c) Jupiter is 1000 times its normal mass. In (d) Jupiter is 1000 times its normal mass and the sun is fixed at the origin.

of magnitude as the Sun's mass. This is shown in figure 3d. A smaller step size is needed to capture the complicated motions of the Earth when the Sun is fixed. If the gravitational forces acting on the Sun is considered the solution again becomes stable, as seen in figure 3c. The Sun and Jupiter start orbiting the system's new center of mass and the Earth follows in orbit around the Sun.

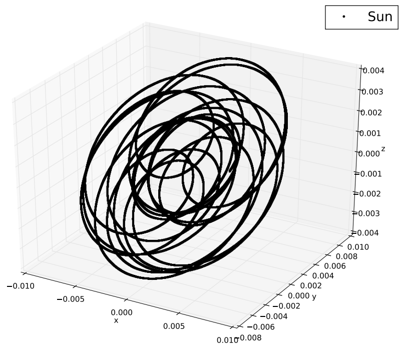
Finally, adding all the planets in the solar system the problem becomes a 9-body problem. The planet furthest away from the Sun is Neptune with a orbital period of approximately 164 years. To get a full revolution of all the orbits a solar system simulation is run with a duration of 200 years and with time step 0.001. The results are shown in 4b. Even with this crude time step and 9 bodies the Velocity Verlet algorithm produces stable orbits for all the celestial bodies. In figure 4c the influence of the other planets on the Sun can be seen but compared to the radius of the Sun at approximately 0.0047AU the motion is not very large. In figure 4d the precession of Mercury's orbit due to the gravitational force of the other planets can be seen.



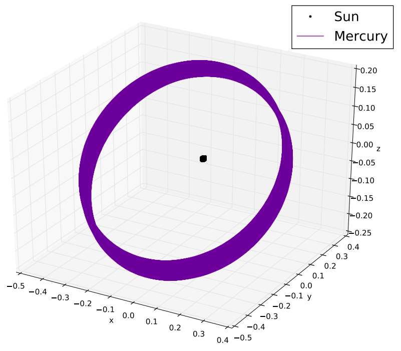
(a) Three-body problem



(b) 9-body problem



(c) 9-body problem



(d) 9-body problem

Figure 4: N-body system solved with the Velocity Verlet algorithm with initial position and velocities from [5]. Gravitational forces on the Sun is included. In (a) the time step is set to 0.001 years and the simulation is run for 10 years. In (b) the time step is set to 0.001 years and the simulation is run for 200 years. (c) shows the orbit of the Sun from the solution in (b) and (d) shows the orbit of Mercury from the solution in (b).

3.3 Mercury's perihelion precession

Removing the effect on Mercury's orbit from the gravitational force from the other planets there is still some part of the observed precession that is unaccounted for. By using the modified Newtonian gravitational force in 17 and solving the two-body problem of the Mercury-Sun system this effect can be seen. Starting with Mercury at perihelion at t_0 and with $\boldsymbol{x}_0 = (0.3075AU, 0, 0)$ and $\boldsymbol{v}_0 = (0, 12.44AU/year, 0)$, the perihelion angle can be calculated as $\theta_p = \arctan{(y_p/x_p)}$, where x_p and y_p is the x and y coordinate of the perihelion. This motion is 2-dimensional and the z-coordinate is ignored. With $r^2 = x_p^2 + y_p^2$, finding the perihelion positions is the problem of finding all the minimum values of r over the duration of the simulation. This is done and figure 5 shows how the perihelion angle changes over time. The simulation is run with a time step of $h = 10^{-6}$ and a duration of 100 years. The time step is not small enough to determine the perihelion position with high precision, which causes the fluctuations seen in the figure, but it is small enough to see the trend. The theory of general relativity predicts a perihelion precession of $43''$ per century, as described in section 2.6, which is approximately 0.01194 degrees. The simulation clearly shows a linear trend in the perihelion precession and the value over one century is close to $43''$.

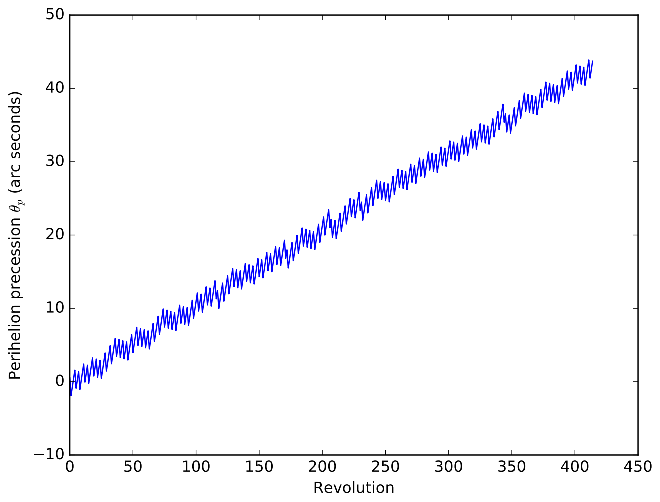


Figure 5: Angular change in arc seconds in Mercury's perihelion for each revolution around the Sun for a simulation run over 100 years.

4 Conclusions

For simulating the orbital motion of celestial bodies in the solar system the Velocity Verlet algorithm is superior to the Forward Euler algorithm. The Forward Euler algorithm is slightly faster and easier to implement, but the Velocity Verlet algorithm produces more stable orbits for step size $h = 10^{-2}$ than the Forward Euler algorithm does for step size $h = 10^{-7}$.

The Sun is the by far most important contributor to the planets orbits, but the planets also have an effect on the motion of the Sun. The Sun orbits the solar system's center of mass, and the orbital motion is of the same order of magnitude as the Sun's radius. The planets also have a smaller effect on each other, with the most visible effect on Mercury. Mercury's precessing orbit seems to be a combination of the gravitational effect from the other planets and the theory of general relativity.

References

[1] Hjort-Jensen, M., 2015. Computational physics. Available at https://github.com/CompPhysics/ComputationalPhysics/

[2] Young, Freedman, Sears and Zemansky's University Physics with Modern Physics, 11th edition, Addison-Wesley, 2004

[3] https://github.com/mortele/solar-system-fys3150. Framework for implementation

[4] http://ssd.jpl.nasa.gov/horizons.cgi Jet Propulsion Laboratory - Horizons web interface

[5] ftp://ssd.jpl.nasa.gov/pub/eph/planets/bsp Jet Propulsion Laboratory - Ephemerides files

[6] https://pypi.python.org/pypi/jplephem - Parser for ephemerides files