FYS4150, Computational Physics
Project 5

Andreas Bjurstedt, Petter A. Langstrand & Eirik V. Winness

December 2019

Abstract

In this project we show the derivation of the acceleration components required to find the planet's orbits. Together with known initial position and velocity components of the planets included in our model, we can find a numerical approximation of the planet orbits by using a numerical method. We show the derivation and test two numerical methods: The Forward Euler method and the Velocity Verlet method. Initial tests on a simple system where the Earth is the only planet in orbit around the Sun, reveals that the Velocity Verlet method gives far better results than the Forward Euler method. Therefore the Velocity Verlet method is our method of choice when we also include Jupiter in our system. With Jupiter included, we see that letting the Sun be in a fixed position in origo of the coordinate system, instead of in its real position orbiting around the common mass center of the system, is a good approximation. The other planets in the Solar system is then included, and the visual result looks satisfactory. A simulation of the perihelion percession of Mercury is also included. When only the gravitational effect from the Sun is included, the observed perihelion percession over one century is 43" (arc seconds). In our simulation, with time step size $\triangle t = 10^{-6}$, we obtain 52 percent of this value.

The Python program developed for the system of the Sun, the Earth and Jupiter is based on classes with objects/instances. Therefore the inclusion of rest of the planets in the solar system didn't require much extra effort.

Introduction

With Newton's law of gravitation as a starting point, we are showing the derivation of the x and y component of the acceleration the Sun, planet Earth and Jupiter are exposed to due to the gravitational force from the other two objects. These acceleration terms reveal a pattern which makes it quite easy to include the rest of the planets too (and the plantet's moons if we want to), and also the z component of the acceleration. With the acceleration terms and with known initial position and velocity components in x and y direction of the planets, we have the information required to find the orbit of the planets numerically. For this task we show the derivation of and test the Forward Euler method and the Velocity Verlet method. The derivations of both the acceleration terms and the two numerical methods are extracted from [1], and the *Ordinary differential equations* link there. There we can also find the project decription which includes the masses in $[kg]$ of the Sun and all the planets of the solar system. The average distances in Astronomical Units $[AU]$ between the Sun and each of the planets are also included there.

We first test the two numerical methods in a simple model where the Sun is fixed in origo of the coordinate system, and where the Earth is the only planet in orbit around it. The tests reveal that The Velocity Verlet method is far better to use in finding numerical approximations of the Earth's orbits than the Forward Euler method. We see it in the plots of planet Earth's orbit, and the relative error terms of the conserved potential energy, kinetic energy and angular momentum reveals it. (In this simple solar system model, the orbit of the Earth is circular. Therefore both Potential energy and Kinetic energy are conserved, not just the total sum of energy.) Therefore, when we include Jupiter in our model of the solar system, and later the rest of the planets in the solar system too, we only use the Velocity Verlet method. When Jupiter is included, we also let our three solar system objects orbit around their common mass center, which is fixed in origo of the coordinate system. The test shows that the extent of the Sun's orbit is very small, so fixing the Sun in origo of our coordinate system is a good approximation. When the five remaining planets in the solar system, Mercury, Venus, Mars, Saturn, Uranus and Neptune, are included, we let them and the Sun orbit around their common Mass center. We also perform a simulation of the perihelion percession of Mercury.

The two numerical methods are programmed in Python. When Jupiter, and later the rest of the planets, are included in our model, we choose to leave our function based Python programs, developed for the first simple model, behind and develop a Python program based on classes and object/instances instead. Having one instance for each planet and for the Sun, makes it a lot easier to keep track of masses, initial conditions and lengthy acceleration terms.

In the first simple model, where the Earth is the only planet in a circular orbit around the Sun, we only use the Earth's average distance of 1 Asronomical unit ($[AU]$) from the Sun to etablish its initial position componets: $x_0 = 1AU$ and $y_0 = 0$. For the circular orbit of the Earth we then have the initial velocity components $v_{x0} = 0$ and $v_{y0} = 2\pi AU/year$. When Jupiter, and later the rest of the solar system planets, is included, we use the initial conditions found in Appendix B, which are extracted from [2].

Methods

We divide the methods section into three parts. In the first part we look at the gravitational force the Sun and the planets in the solar system are exposed to, and find an expression for the acceleration of the Sun and the planets due to the gravitational force. We also take a look at conservation of energy and angular momentum. A look at the solar systems mass center and momentum are also included. In addition the perihelion precession of Mercury is explained. In the second and third part we look at two methods which use the acceleration of an object in the solar system in order to find its orbit numerically. In part two we present the Forward Euler method and in part three the Velocity Verlet method is presented. The theory in this section is extracted from the *Ordinary differential equations* link in [1].

Part 1: The solar system

Newton's law of gravitation gives the force acting between two objects with mass M_1 and M_2

$$F = \frac{GM_1M_2}{r^2} \tag{1}$$

r is the distance between the objects and $G = 6.674\times10^{-11}N\cdot m^2\cdot kg^{-2}$ is the gravitational constant. If $M_1 = M_\odot$ is the mass of the Sun and $M_2 = M_E$ is the mass of planet Earth, equation (1) becomes

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

The x and y components of the gravitational pull acting on planet Earth from the sun are

$$\begin{aligned} F_x^E &= M_E \frac{d^2 x^E}{dt^2} = -\frac{GM_\odot M_E}{r^2} cos(\theta) \\ F_y^E &= M_E \frac{d^2 y^E}{dt^2} = -\frac{GM_\odot M_E}{r^2} sin(\theta) \end{aligned} \tag{2}$$

Now

$$\begin{aligned} r_x &= (x_E - x_\odot) = rcos(\theta) \\ r_y &= (y_E - y_\odot) = rsin(\theta) \end{aligned}$$

where

$$r = \sqrt{r_x^2 + r_y^2} = \sqrt{(x_E - x_\odot)^2 + (y_E - y_\odot)^2}$$

The distances $x_\odot$, $y_\odot$, x_E and y_E are relative to the mass center of the Sun and Planet Earth system. Equation (2) then becomes

$$\begin{aligned} F_x^E &= M_E \frac{d^2 x^E}{dt^2} = -\frac{GM_\odot M_E}{r^3} (x_E - x_\odot) \\ F_y^E &= M_E \frac{d^2 y^E}{dt^2} = -\frac{GM_\odot M_E}{r^3} (y_E - y_\odot) \end{aligned} \tag{3}$$

The acceleration components of planet Earth due to the gravitational pull from the Sun then is

$$\begin{aligned} a_x^E &= \frac{d^2 x^E}{dt^2} = -\frac{GM_\odot}{r^3} (x_E - x_\odot) \\ a_y^E &= \frac{d^2 y^E}{dt^2} = -\frac{GM_\odot}{r^3} (y_E - y_\odot) \end{aligned} \tag{4}$$

The average distance between the Sun and the Earth is $1AU = 1.5 \times 10^{11} m$, where AU means *Astronomical Unit*. For distances it is therefore more practical to use AU instead of meters $[m]$. It is also more convenient to use

Years in stead of seconds $[s]$, since years match the time evolution of the solar system better than seconds. In order to replace meters $[m]$ with AU, we assume Earth's orbit around the sun to be a perfect circle, where $r = 1$ AU. The gravitational pull on Earth from the Sun then is

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

where $a = v^2/r$ is the size of Earth's centripetal acceleration and v is its orbital speed. We then get

$$GM_\odot = v^2 r = v^2 AU$$

The Earth's orbital speed is

$$v = \frac{2\pi r}{year} = 2\pi \frac{AU}{year}$$

so

$$GM_\odot = v^2 r = v^2 AU = 4\pi^2 \frac{AU^3}{year^2} \tag{5}$$

By inserting equation (5) in to equation (4), we get

$$\begin{aligned} a_x^E &= \frac{d^2 x^E}{dt^2} = -\frac{4\pi^2}{r^3}(x_E - x_\odot) \\ a_y^E &= \frac{d^2 y^E}{dt^2} = -\frac{4\pi^2}{r^3}(y_E - y_\odot) \end{aligned}$$

Now

$$\begin{aligned} \frac{dx^E}{dt} &= v_x^E \\ \frac{dy^E}{dt} &= v_y^E \end{aligned}$$

so planet Earth's acceleration due to the gravitational pull from the Sun is

$$\begin{aligned} a_x^E &= \frac{dv_x^E}{dt} = -\frac{4\pi^2}{r^3}(x_E - x_\odot) \\ a_y^E &= \frac{dv_y^E}{dt} = -\frac{4\pi^2}{r^3}(y_E - y_\odot) \end{aligned} \tag{6}$$

We now let our system be a little bit more complicated by adding the gravitational pull on planet Earth from Jupiter in equation (3).

$$
\begin{aligned}
F_x^E &= M_E \frac{d^2 x^E}{dt^2} = -\frac{GM_\odot M_E}{r_{E\odot}^3}(x_E - x_\odot) - \frac{GM_J M_E}{r_{EJ}^3}(x_E - x_J) \\
F_y^E &= M_E \frac{d^2 y^E}{dt^2} = -\frac{GM_\odot M_E}{r_{E\odot}^3}(y_E - y_\odot) - \frac{GM_J M_E}{r_{EJ}^3}(y_E - y_J)
\end{aligned}
$$

where

$$
\begin{aligned}
r_{E\odot} &= \sqrt{(x_E - x_\odot)^2 + (y_E - y_\odot)^2} \\
r_{EJ} &= \sqrt{(x_E - x_J)^2 + (y_E - y_J)^2}
\end{aligned}
$$

We see that the term representing the gravitational pull from Jupiter is quite similar to the term representing the gravitational pull from the Sun. The distances x_E, y_E, $x_\odot$ and so on, are now relative to the mass center of the Sun, Earth, Jupiter system. We also see that it isn't difficult to include the third spatial dimension. In this project however we only look at the co-planar motions of the planets and therefore exclude the z coordinate.

Planet Earth's acceleration due to the gravitational pull from the Sun and Jupiter now becomes

$$
\begin{aligned}
a_x^E &= \frac{dv_x^E}{dt} = -\frac{4\pi^2}{r_{E\odot}^3}(x_E - x_\odot) - \frac{GM_J}{r_{EJ}^3}(x_E - x_J) \\
a_y^E &= \frac{dv_y^E}{dt} = -\frac{4\pi^2}{r_{E\odot}^3}(y_E - y_\odot) - \frac{GM_J}{r_{EJ}^3}(y_E - y_J)
\end{aligned}
$$

From equation (5) we get

$$
GM_J = GM_\odot \frac{M_J}{M_\odot} = 4\pi^2 \frac{M_J}{M_\odot}
$$

So Planet Earth's acceleration due to the gravitational pull from the Sun and Jupiter is

$$
\begin{aligned}
a_x^E &= \frac{dv_x^E}{dt} = -\frac{4\pi^2 M_\odot/M_\odot}{r_{E\odot}^3}(x_E - x_\odot) - \frac{4\pi^2 M_J/M_\odot}{r_{EJ}^3}(x_E - x_J) \\
a_y^E &= \frac{dv_y^E}{dt} = -\frac{4\pi^2 M_\odot/M_\odot}{r_{E\odot}^3}(y_E - y_\odot) - \frac{4\pi^2 M_J/M_\odot}{r_{EJ}^3}(y_E - y_J) \qquad (7)
\end{aligned}
$$

Notice that we also have included the term $M_\odot/M_\odot$, which makes it easier both to see the pattern and to perform the programming of the numerical algorithms in Python. By rearranging the indices in equation (7), we find Jupiters acceleration due to the gravitational pull from the Sun and planet Earth

$$\begin{aligned} a_x^J &= \frac{dv_x^J}{dt} = -\frac{4\pi^2 M_\odot/M_\odot}{r_{J\odot}^3}(x_J - x_\odot) - \frac{4\pi^2 M_E/M_\odot}{r_{JE}^3}(x_J - x_E) \\ a_y^J &= \frac{dv_y^J}{dt} = -\frac{4\pi^2 M_\odot/M_\odot}{r_{J\odot}^3}(y_J - y_\odot) - \frac{4\pi^2 M_E/M_\odot}{r_{JE}^3}(y_J - y_E) \end{aligned} \quad (8)$$

The acceleration of the Sun due to the gravitational pull from planet Earth and Jupiter is

$$\begin{aligned} a_x^\odot &= \frac{dv_x^\odot}{dt} = -\frac{4\pi^2 M_E/M_\odot}{r_{\odot E}^3}(x_\odot - x_E) - \frac{4\pi^2 M_J/M_\odot}{r_{\odot E}^3}(x_\odot - x_J) \\ a_y^\odot &= \frac{dv_y^\odot}{dt} = -\frac{4\pi^2 M_E/M_\odot}{r_{\odot E}^3}(y_\odot - y_E) - \frac{4\pi^2 M_J/M_\odot}{r_{\odot J}^3}(y_\odot - y_J) \end{aligned} \quad (9)$$

In equation (7), (8) and (9)

$$\begin{aligned} v_x^* &= \frac{dx^*}{dt} \\ v_y^* &= \frac{dy^*}{dt} \qquad \text{for } * = E, J \text{ and } \odot \end{aligned} \quad (10)$$

and

$$\begin{aligned} r_{E\odot} &= r_{\odot E} = \sqrt{(x_\odot - x_E)^2 + (y_\odot - y_E)^2} \\ r_{J\odot} &= r_{\odot J} = \sqrt{(x_\odot - x_J)^2 + (y_\odot - y_J)^2} \\ r_{JE} &= r_{EJ} = \sqrt{(x_E - x_J)^2 + (y_E - y_J)^2} \end{aligned} \quad (11)$$

We now see the pattern, so it not difficult to include the rest of the planets in the solar system.

The energy of an object in orbit around the Sun

The work the gravitational force from the Sun does on an object with mass

M in order to pull it from the distance $r = r_1$ to $r = r_2$ closer to the Sun $(r_2 < r_1)$ is

$$\begin{aligned} W &= \int_{r_1}^{r_2} \mathbf{F} \cdot \mathbf{dr} = \int_{r_1}^{r_2} -F\mathbf{i_r} \cdot \mathbf{i_r} dr = -\int_{r_1}^{r_2} F dr \\ &= -GM_\odot M \int_{r_1}^{r_2} \frac{1}{r^2} dr = GM_\odot M \left(\frac{1}{r_2} - \frac{1}{r_1} \right) \end{aligned}$$

The change in the potential energy of the object is then

$$\triangle E_p = E_p(r_2) - E_p(r_1) = -W = GM_\odot M \left(\frac{1}{r_1} - \frac{1}{r_2} \right)$$

Letting $r_2 = r$ and $r_1 \to \infty$, the potential energy of the object in a distance r from the Sun is

$$E_p(r) = -GM_\odot M \frac{1}{r} = -4\pi^2 M \frac{1}{r}$$

The kinetic energy of the object is

$$E_k(r) = \frac{1}{2} M v(r)^2$$

Since the object doesn't gain or lose any energy, its total energy is conserved.

$$E_p(r) + E_k(r) = \text{constant } < 0$$

We can now find out what speed the object needs to escape the gravitational force (gravitational field) of the Sun. E_k then has to be equal to the work which is required in order to lift the object out of the Sun's gravitational field. That means

$$\begin{aligned} E_p(r) + E_k(r) &= 0 \\ -GM_\odot M \frac{1}{r} + \frac{1}{2} M v_{\text{escape}}^2 &= -4\pi^2 M \frac{1}{r} + \frac{1}{2} M v_{\text{escape}}^2 = 0 \end{aligned}$$

which gives

$$v_{\text{escape}} = \left(\frac{8\pi^2}{r} \right)^{1/2} = 2\pi \left(\frac{2}{r} \right)^{1/2} \tag{12}$$

Angular momentum of an object in orbit around the Sun
The angular momentum (in two dimensions) of the object is

$$\mathbf{L} = \mathbf{r} \times m\mathbf{v} = (x\mathbf{i} + y\mathbf{j}) \times (v_x\mathbf{i} + v_y\mathbf{j}) = m(xv_y - yv_x)\mathbf{k}$$

Since the object doesn't gain or lose any energy, its angular momentum is conserved.

Mass center and momentum of the solar system
As for the acceleration expressions, we only include the Sun, planet Earth and Jupiter here. We then see the pattern, and it is easy to include the other planets. Only to dimensions are included.

Although it is a very good approximation, the planets actually doesn't orbit around the center of the Sun resting in origo. The planets and the Sun are orbiting around their common mass center, which is fixed in origo. When we know the initial position component x_0 to the Earth and Jupiter, we can find x_0 to the Sun relative to the mass center in the following way. We find the initial position component y_0 in exactly the same way.

$$M_\odot x_0^\odot + M_E x_0^E + M_J x_0^J = (M_\odot + M_E + M_J)x_{\text{center}} = 0$$

since $x_{\text{center}} = 0$. Then

$$x_0^\odot = -\left(\frac{M_E x_0^E + M_J x_0^J}{M\odot}\right) \tag{13}$$

Knowing that the system's momentum is zero, we can find the Sun's initial velocity component v_{x0} in the following way. We find the initial velocity component v_{y0} in the exact same way.

$$M_\odot v_{x0}^\odot + M_E v_{x0}^E + M_J v_{x0}^J = 0$$

then

$$v_{x0}^\odot = -\left(\frac{M_E v_{x0}^E + M_J v_{x0}^J}{M\odot}\right) \tag{14}$$

The perihelion precession of Mercury
The perihelion precession of Mercury is explained in the project description of project 5, see the *Project 5 Solar System* link in [1]. Mercury only approximately follows an elliptic orbit around the Sun. When the gravitational

pull from all the other planets in the Solar system is subtracted, the closest point to the Sun in Mercury's orbit, the perihelion, still doesn't occur in the same place in space in each orbit around the Sun. This movement is called *the perihelion precession of Mercury*. The angular movement of this point is 43" (43 arc seconds) per (planet Earth) century, and it can't be explained by Newton's law of gravitation alone. According to this law, the perihelion position of Mercury is fixed in space. A general relativistic correction has to be added, as explained in the description of project 5. Newton's law of gravitation (1) with this correction is

$$F = \frac{GM_{\odot}M_{ME}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right] \tag{15}$$

where

$$\begin{aligned} l &= |\mathbf{r} \times \mathbf{v}| = |(x\mathbf{i} + y\mathbf{j}) \times (v_x\mathbf{i} + v_y\mathbf{j})| \\ &= |(xv_y - yv_x)\mathbf{k}| = |xv_y - yv_x| \end{aligned}$$

is the magnitude of Mercury's orbital angular momentum per unit mass. $c = 299792.458$ km/s is the speed of light in vacuum. 1 Astronomical Unit [AU] is exactly 1 AU $= 1.495978707 \times 10^8$ km. The speed of light in vacuum can then be converted to

$$\frac{299792458}{1.495978707 \times 10^8} \times (3600 \times 24 \times 365.25) \text{ AU/year } = 63241.07708 \text{ AU/year}$$

Using the same method as described earlier in the method section, (15) can when the Sun is fixed in origo be rewritten as

$$\begin{aligned} a_x^{ME} &= \frac{dv_x^E}{dt} = -\frac{GM_{\odot}}{r^3}\left[1 + \frac{3l^3}{r^2l^2}\right]x_{ME} \\ a_y^{ME} &= \frac{dv_x^y}{dt} = -\frac{GM_{\odot}}{r^3}\left[1 + \frac{3l^3}{r^2l^2}\right]y_{ME} \end{aligned}$$

where

$$r = \sqrt{x_{ME}^2 + y_{ME}^2}$$

Using (5) we get

$$
\begin{aligned}
a_x^{ME} &= \frac{dv_x^E}{dt} = -\frac{4\pi^2}{r^3}\left[1+\frac{3l^3}{r^2l^2}\right]x_{ME} \\
a_y^{ME} &= \frac{dv_x^y}{dt} = -\frac{4\pi^2}{r^3}\left[1+\frac{3l^3}{r^2l^2}\right]y_{ME}
\end{aligned}
$$

where

$$
\begin{aligned}
v_x^{ME} &= \frac{dx^{ME}}{dt} \\
v_y^{ME} &= \frac{dy^{ME}}{dt}
\end{aligned}
$$

The perihelion precession of Mercury during one (planet Earth) century is simulated using the Velocity Verlet method. In order to get a representative result, the time resolution has to be sufficient. We use the time step $\triangle t = 10^{-6}$ year, which gives $n = 100 \times 10^6$ time steps over one century. In order to avoid difficulties with insufficient computer memory, the simulation is performed one (planet Earth) year at the time. For the current year, the coordinates for each perihelion are found before the simulation of the next year is performed by using the calculated position and velocity from the last discrete point in the current year as initial conditions.

Part 2: The Forward Euler method

The presentation of the method doesn't have a general form but emphasises the motion of objects for clarity. We start by Taylor expanding the x component of the position of an object, e.g a planet in the solar system. The x component of the object's velocity is v_x.

$$
\begin{aligned}
x(t+\triangle t) &= x(t) + \triangle t x'(t) + \frac{\triangle t^2}{2!}x''(t) + \frac{\triangle t^3}{3!}x^{(3)}(t) + \cdots \\
&= x(t) + \triangle t v_x(t) + \frac{\triangle t^2}{2!}x''(t) + \frac{\triangle t^3}{3!}x^{(3)}(t) + \cdots \\
&= x(t) + \triangle t v_x(t) + \triangle t\left(\frac{\triangle t}{2!}x''(t) + \frac{\triangle t^2}{3!}x^{(3)}(t) + \cdots\right)
\end{aligned}
$$

which can be written as

$$x(t + \triangle t) = x(t) + \triangle t v_x(t) + \triangle t O(\triangle t) \tag{16}$$

where $\triangle t O(\triangle t)$ is the error term. Now $\triangle t O(\triangle t) = O(\triangle t^2)$, which is the order of the local error for one step $\triangle t$ in the algorithm. But after running n steps, we get the total error, and the order of that error is

$$nO(\triangle t^2) = n\triangle t O(\triangle t) = O(\triangle t)$$

In the same way as for $x(t)$ we get

$$v_x(t + \triangle t) = v_x(t) + \triangle t v'_x(t) + \triangle t O(\triangle t) \tag{17}$$

where examples of the acceleration component v'_x is given in (7) (8) and (9).

We now want to find a numerical approximation of $x(t)$ and $v_x(t)$ in the time interval $t \in [t_0, T]$ based on (16) and (17). We divide this interval into n discrete time steps $\triangle t$, which gives us an approximate solution in the discrete grid points

$$t_0, t_1 \cdots, t_n = t_0, t_1 \cdots, T$$

where

$$t_i = t_0 + i\triangle t \qquad \text{and} \qquad t_{i+1} = t_i + \triangle t$$

In particular

$$t_n = t_0 + n\triangle t, \qquad \text{so} \qquad \triangle t = \frac{t_n - t_0}{n}$$

The numerical approximation of $x(t)$ and $v_x(t)$ are then

$$x_{i+1} = x_i + \triangle t v_{x,i}, \quad i = 0, 1, 2, \cdots n - 1$$

and

$$v_{x,i+1} = v_{x,i} + \triangle t v'_{x,i}, \quad i = 0, 1, 2, \cdots n - 1$$

We need an initial condition in order to start each of the discrete equations. The initial conditions are $x(t_0) = x_0$ and $v_x(t_0) = v_{x,0}$. Examples of $a_{x,i} = v'_{x,i}$ are given in (7), (8) and (9) for discrete values of x.

Including the y component of the object's position and velocity, we get the numerical approximations

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i + \triangle t v_{x,i} \\ y_i + \triangle t v_{y,i} \end{bmatrix}, \quad i = 0, 1, 2, \cdots n-1 \tag{18}$$

and

$$\begin{bmatrix} v_{x,i+1} \\ v_{y,i+1} \end{bmatrix} = \begin{bmatrix} v_{x,i} + \triangle t v'_{x,i} \\ v_{y,i} + \triangle t v'_{y,i} \end{bmatrix}, \quad i = 0, 1, 2, \cdots n-1 \tag{19}$$

with initial conditions

$$\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} v_x(t_0) \\ v_y(t_0) \end{bmatrix} = \begin{bmatrix} v_{x,0} \\ v_{y,0} \end{bmatrix} \tag{20}$$

In Python, the algorithm could look like this

```
for i in range(n):
        t[i+1] = t[i] + dt
        pos[i+1] = pos[i] + dt*vel[i]
        vel[i+1] = vel[i] + dt*f(pos,i)
```

where

$$\text{pos[i]} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}^T, \quad \text{vel[i]} = \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix}^T, \quad \text{and} \quad \text{f(pos,i)} = \begin{bmatrix} v'_{x,i} \\ v'_{y,i} \end{bmatrix}^T \tag{21}$$

f = f(pos,i), since we from e.g (8) have that the acceleration in a grid point i is dependent on the position there. For each grid point the number of Floating Point Operations (FLOPs) is

```
  1
+ 2 x (1 + 1)
+ 2 x (1 + 1) + number of FLOPs in f(pos,i)
= 9 + number of FLOPs in f(pos,i)
```

The Forward Euler algorithm is only used when we look at the orbit of planet Earth due to the gravitational pull from the Sun only.

Part 3: The Velocity Verlet method

As for the Forward Euler method, we start by Taylor expanding the x component of the position of an object.

$$
\begin{aligned}
x(t+\triangle t) &= x(t) + \triangle t x'(t) + \frac{\triangle t^2}{2!}x''(t) + \frac{\triangle t^3}{3!}x^{(3)}(t) + \cdots \\
&= x(t) + \triangle t v_x(t) + \frac{\triangle t^2}{2!}x''(t) + \frac{\triangle t^3}{3!}x^{(3)}(t) + \cdots \\
&= x(t) + \triangle t v_x(t) + \frac{\triangle t^2}{2!}x''(t) + \triangle t\left(\frac{\triangle t^2}{3!}x^{(3)}(t) + \cdots\right)
\end{aligned}
$$

which can be written as

$$
x(t+\triangle t) = x(t) + \triangle t v_x(t) + \frac{\triangle t^2}{2}v_x'(t) + \triangle t O(\triangle t^2) \tag{22}
$$

In the same way, the Taylor expansion of the object's x component v_x of its velocity is

$$
v_x(t+\triangle t) = v_x(t) + \triangle t v_x'(t) + \frac{\triangle t^2}{2}v_x''(t) + \triangle t O(\triangle t^2) \tag{23}
$$

The local error for each step in (22) and (23) is then $\triangle t O(\triangle t^2) = O(\triangle t^3)$ and the total error after running n steps is $n\triangle t O(\triangle t^2) = O(\triangle t^2)$. Rewriting the Taylor expansion for v_x a bit, we get

$$
v_x(t+\triangle t) = v_x(t) + \frac{\triangle t}{2}\left(2v_x'(t) + \triangle t v_x''(t)\right) + \triangle t O(\triangle t^2) \tag{24}
$$

For the object's x component $a_x = v_x'$ of its acceleration, the Taylor expansion is

$$
v_x'(t+\triangle t) = v_x'(t) + \triangle t v_x''(t) + O(\triangle t^2)
$$

which gives

$$
\triangle t v_x''(t) = v_x'(t+\triangle t) - v_x'(t) + O(\triangle t^2)
$$

Inserting this expression into (24), we get

$$
\begin{aligned}
v_x(t+\triangle t) &= v_x(t) + \frac{\triangle t}{2}\left(2v_x'(t) + v_x'(t+\triangle t) - v_x'(t) + O(\triangle t^2)\right) + \triangle t O(\triangle t^2) \\
&= v_x(t) + \frac{\triangle t}{2}\left(v_x'(t+\triangle t) + v_x'(t)\right) + \triangle t O(\triangle t^2) \qquad (25)
\end{aligned}
$$

We now want to find a numerical approximation of $x(t)$ and $v_x(t)$ in the time interval $t \in [t_0, T]$ based on (22) and (25). Once again we divide this interval into n discrete time steps $\triangle t$, which gives us an approximate solution in the discrete grid points

$$t_0, t_1 \cdots, t_n = t_0, t_1 \cdots, T$$

where

$$t_i = t_0 + i\triangle t \quad \text{and} \quad t_{i+1} = t_i + \triangle t$$

In particular

$$t_n = t_0 + n\triangle t, \quad \text{so} \quad \triangle t = \frac{t_n - t_0}{n}$$

The numerical approximation of $x(t)$ and $v_x(t)$ are then

$$x_{i+1} = x_i + \triangle t v_{x,i} + \frac{\triangle t^2}{2} v'_{x,i}$$

and

$$v_{x,i+1} = v_{x,i} + \frac{\triangle t}{2}\left(v'_{x,i+1} + v'_{x,i}\right)$$

The initial conditions are $x(t_0) = x_0$ and $v_x(t_0) = v_{x,0}$. Examples of $a_{x,i} = v'_{x,i}$ are once again given in (7), (8) and (9) for discrete values of x. Since these three expressions are dependent of x, we need to calculate x_{i+1} for all the three solarsystem objects before we can calculate $v_{x,i+1}$ for them. (The expression for $v_{x,i+1}$ is dependent of $v'_{x,i+1}$, which in turn is dependent of x_{i+1})

Including the y component of the object's position and velocity, we get the numerical approximations

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i + \triangle t v_{x,i} + \frac{\triangle t^2}{2} v'_{x,i} \\ y_i + \triangle t v_{y,i} + \frac{\triangle t^2}{2} v'_{y,i} \end{bmatrix}, \quad i = 0, 1, 2, \cdots n-1 \tag{26}$$

and

$$\begin{bmatrix} v_{x,i+1} \\ v_{y,i+1} \end{bmatrix} = \begin{bmatrix} v_{x,i} + \frac{\triangle t}{2}(v'_{x,i+1} + v'_{x,i}) \\ v_{y,i} + \frac{\triangle t}{2}(v'_{y,i+1} + v'_{y,i}) \end{bmatrix}, \quad i = 0, 1, 2, \cdots n-1 \tag{27}$$

with initial conditions

$$\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} v_x(t_0) \\ v_y(t_0) \end{bmatrix} = \begin{bmatrix} v_{x,0} \\ v_{y,0} \end{bmatrix} \tag{28}$$

In Python, the algorithm could look like this

```
for k in range(n):
        fpk = f(pos,k)
        t[k+1] = t[k] + dt
        pos[k+1] = pos[k] + dt*vel[k] + 0.5*dt**2*fpk
        vel[k+1] = vel[k] + 0.5*dt*(f(pos,k+1) + fpk)
```

where $pos[k]$, $vel[k]$ and $f(pos, k)$ are given in (21). Notice that $vel[k+1]$ is dependent of $pos[k+1]$. For each grid point the number of Floating Point Operations (FLOPs) is

```
  1
+ 2 + 2 x (1 + 1 + 1)
+ 1 + 2 x (1 + 1 + 1)
+ number of FLOPs in fpk = f(pos,k)
+ number of FLOPs in f(pos,k+1)
= 16 + number of FLOPs in fpk + number of FLOPS in f(pos,k+1)
```

For details in the Python programs, see Appendix A and [3].

Results

First we take a look at a simple system where the Sun's position is fixed in origo ($x(t) = y(t) = 0$) and the Earth is the only planet in orbit around it. We let the initial position components of the Earth be $x_0 = 1AU$ and $y_0 = 0$. In this position, the initial velocity components of the Earth are $v_{x0} = 0$ and $v_{y0} = 2\pi Au/year$. In the four figures below, we have runned the numerical analysis over a period of $T = 2years$, using both the Forward Euler method and the Velocity Verlet metod. The time period is divided into a succesive increasing number of time steps: $n = 20$, $n = 200$, $n = 2000$ and $n = 20000$. The orbit of the Earth is compared to a perfect circular orbit with radius $r = 1AU$ and with the Sun in its center.

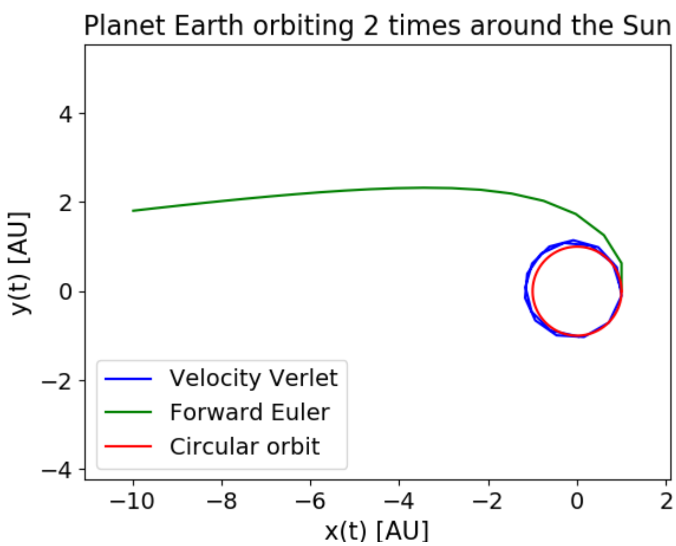

Figure 1: The orbit og planet Earth: T = 2 year long time period divided into n = 20 time steps.

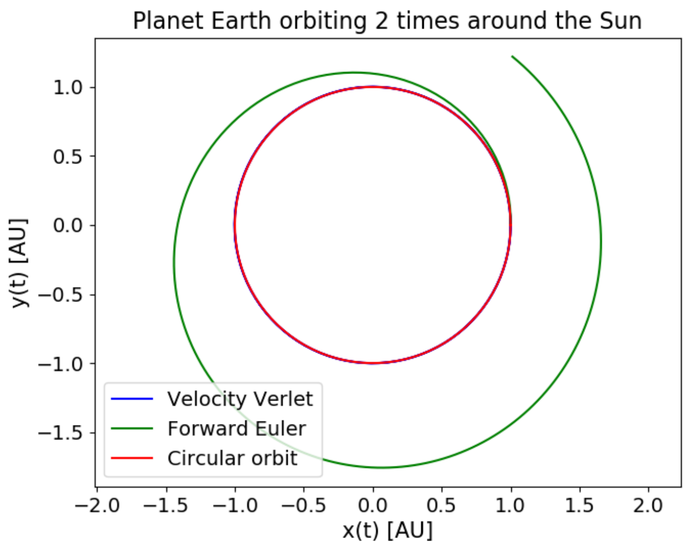

Figure 2: The orbit of planet Earth: T = 2 year long time period divided into n = 200 time steps.

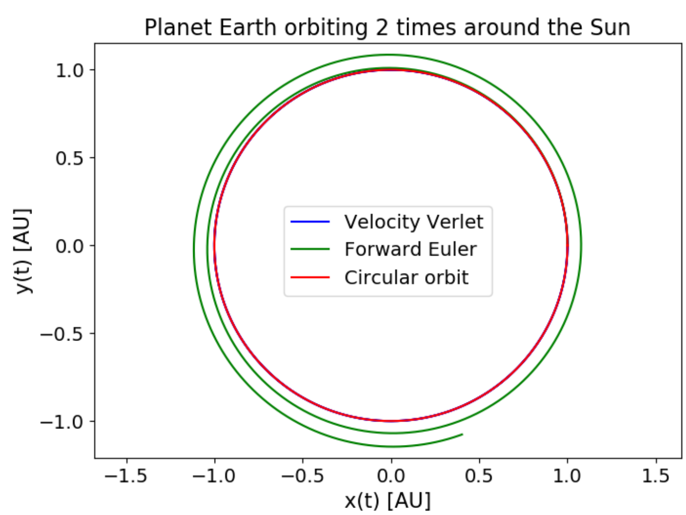

Figure 3: The orbit og planet Earth: T = 2 year long time period divided into n = 2000 time steps.

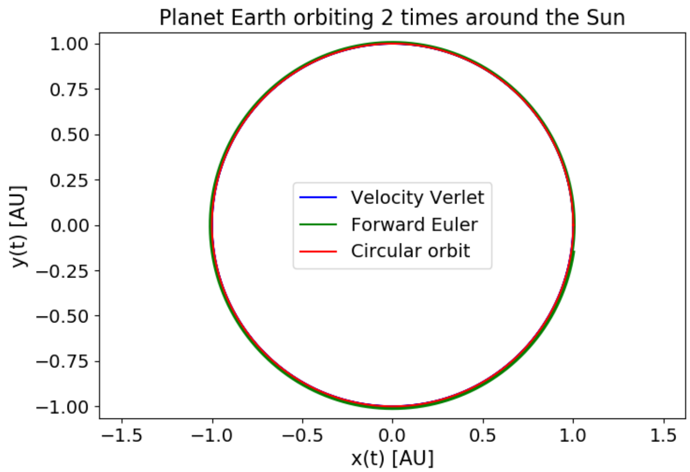

Figure 4: The orbit of planet Earth: T = 2 year long time period divided into n = 20000 time steps.

We see from the figures that the orbit gets more accurate when n increases. Up to $n = 2000$, the Verlocity Verlet method visually performs far better than the Forward Euler method. Just looking at the plots, we can see that the Velocity Verlet method is far better than the Forward Euler method in conserving the energy of the Earth.

In the Method section we have seen that the energy of planet Earth should be conserved since it is not gaining or losing any energy. Actually, since the orbit is circular here, both the potential energy and the kinetic energy should be conserved individually. The following two figures illustrate how well the potential energy and the kinetic energy is conserved for an increasing number of time steps n. We measure it by finding the maximum and the minimum calculated potential energy and kinetic energy respectively for one orbit around the Sun ($T = 1 year$). For the kinetic and potential energy we then find the difference between the maximum and the minimum energy value. In order to get the relative difference between the maximum and minimum energy value, we divide the energy difference with the average energy value for the whole time period.

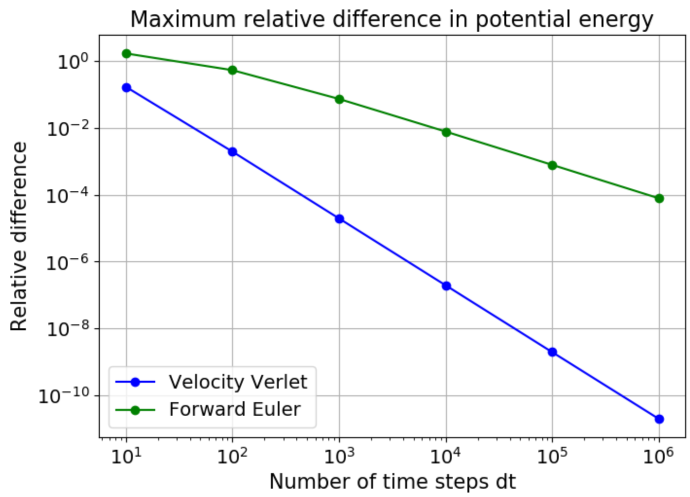

Figure 5: The orbit of planet Earth: Relative difference between maximum and minimum potential energy during a time period of T = 1 year.

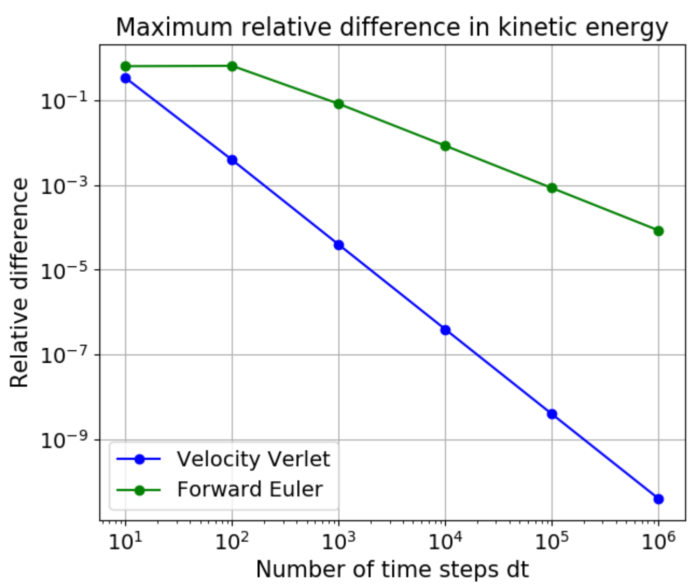

Figure 6: The orbit of planet Earth: Relative difference between maximum and minimum kinetic energy during a time period of T = 1 year.

From these two figures (figure (5) and (6)) we see that the Forward Euler method gives an error of order $O(\triangle t) = O(1/n)$ and that the Velocity Verlet method performs much better with an error of order $O(\triangle t^2) = O(1/n^2)$ as antisipated from the discussion in the Method section.

From the methods section we also know that the angular momentum of planet Earth should be conserved. Calculating the relative difference between maximum and minimum angular momentum during a time period of one year for an increasing number of time steps n (in the same way as for potential and kinetic energy), we get the results shown in the following figure (figure(7)). We see that the Velocity Verlet method performs far better than the Forward Euler method. Actually the results from the Velocity Verlet method is accurate and seems only to be slightly disturbed by numerical errors when the number of time steps n increases.

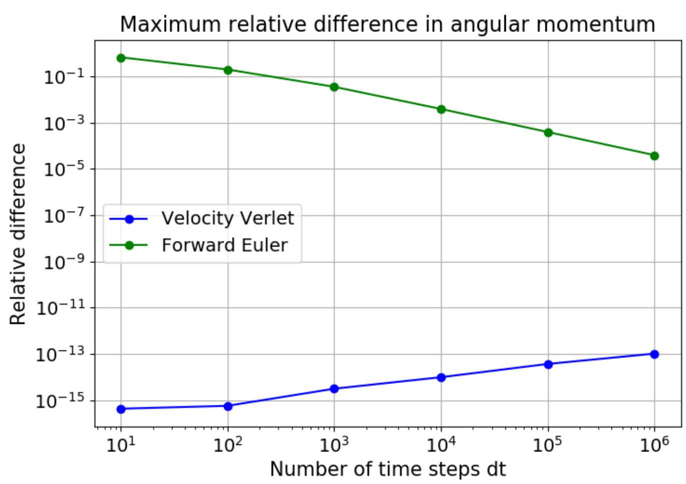

Figure 7: The orbit of planet Earth: Relative difference between maximum and minimum angular momentum during a time period of T = 1 year.

In the Methods section we discussed the number of FLOPs in one grid point for both the Forward Euler and the Velocity Verlet method. When planet Earth is the only planet which orbits around the Sun fixed in origo ($x(t) = y(t) = 0$), we count 11 FLOPs in the acceleration term which is programmed like this in Python for both methods in this simple special case:

```
def f(pos,i):
```

```
    f = np.array([-4*np.pi**2*pos[i,0],-4*np.pi**2*pos[i,1]])
    r = np.sqrt(pos[i,0]**2 + pos[i,1]**2)
    f = f/float(r**3)
    return f
```

For each grid point i we then count $9 + 11 = 20$ FLOPs for the Forward Euler method and $16 + 2 \times 11 = 38$ Flops for the Velocity Verlet method, which give a FLOP factor of $38/20 = 1.9$. This is not so far away from the time factors calculated from the elapsed CPU times measured for both methods for an increasing number of time steps n, see the following table.

n	Forward Euler	Velocity Verlet	Factor FE/VV
1E+01	1.344E-04	2.197E-04	1.635
1E+02	9.788E-04	1.746E-03	1.784
1E+03	1.060E-02	1.786E-02	1.684
1E+04	9.866E-02	1.756E-01	1.779
1E+05	9.840E-01	1.762E+00	1.790
1E+06	9.906E+00	1.757E+01	1.773

Table 1: Measured elapsed CPU time for an increasing number of time steps n.

When more of the planets are included in the system, the number of FLOPs in the acceleration term will increase.

In equation (12) we calculated the velocity required for an object to escape the Sun's gravitational field. We test this analytical value in our simple case with the Earth orbiting the Sun fixed in origo. Then $r = 1AU$ and $v_{\text{escape}} = 2\pi\sqrt{2}$ AU/year. The following four plots shows the Earth's trajectory for a succesive increasing initial velocity $\mathbf{v}_0$, up until the escape velocity. The figures shows that our numerical analysis using the Velocity Verlet method coincides with the analytical escape velocity.

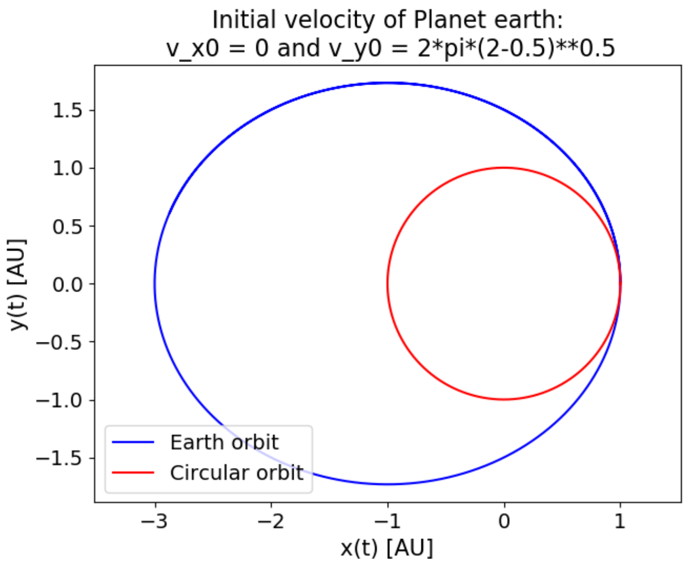

Figure 8: The orbit of planet Earth when the initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{y0} = 2\pi\sqrt{1.5}$ AU/year. $v_{x0} = 0$.

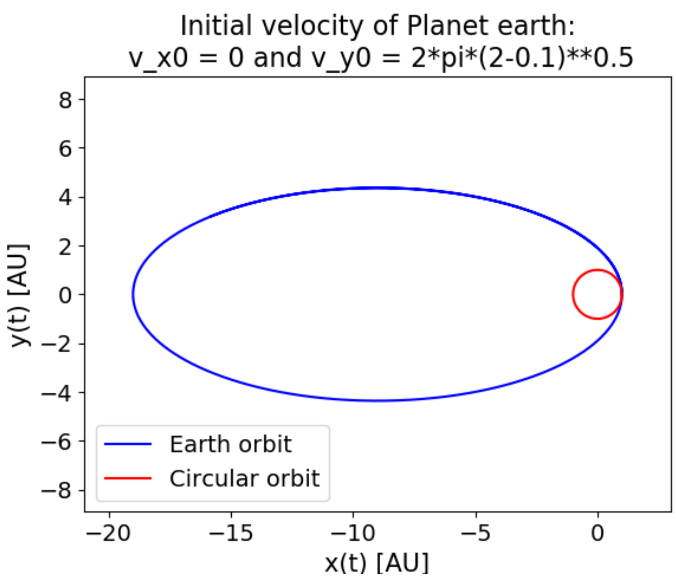

Figure 9: The orbit of planet Earth when the initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{y0} = 2\pi\sqrt{1.9}$ AU/year. $v_{x0} = 0$.

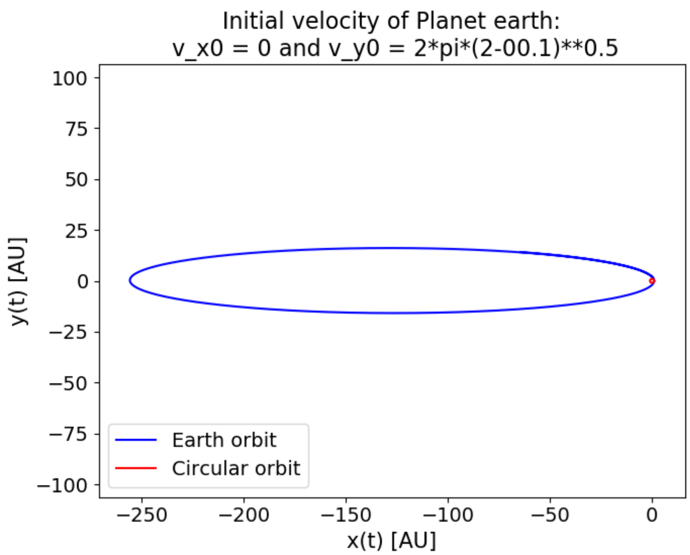

Figure 10: The orbit of planet Earth when the initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{y0} = 2\pi\sqrt{1.99}$ AU/year. $v_{x0} = 0$.

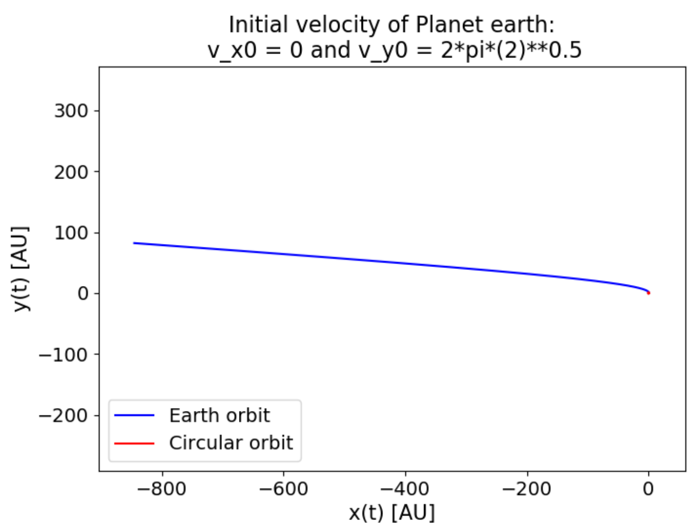

Figure 11: The trajectory of planet Earth when the initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{\text{escape}} = 2\pi\sqrt{2}$ AU/year. $v_{x0} = 0$.

We now investigate what happens if we change the denominator in Newton's law of gravitation (1) from r^2 to r^β, where $\beta \in [2,3]$ and β creeps towards 3. Our focus is still on the simple system where planet Earth is orbiting the Sun which is fixed in origo. We can then rewrite (1) to

$$F = \frac{4\pi^2 M_E}{r^\beta}$$

and the corresponding acceleration terms (6) for the Earth to

$$\begin{aligned} a_x^E &= \frac{dv_x^E}{dt} = -\frac{4\pi^2}{r^{\beta+1}}(x_E - x_\odot) \\ a_y^E &= \frac{dv_y^E}{dt} = -\frac{4\pi^2}{r^{\beta+1}}(y_E - y_\odot) \end{aligned}$$

where position of the Sun is fixed in $x_\odot = y_\odot = 0$. As long as the Earth follows the circular orbit with radius $r = 1AU$ the acceleration doesn't change with an increasing β, but for $r > 1AU$ the acceleration decreases with increasing β. We therefore can expect a lower escape velocity for the Earth if $\beta > 2$. In the following two figures we increase the Earth's initial velocity slightly from $v_{x0} = 0$ and $v_{y0} = 2\pi$ AU/year to $v_{x0} = 0$ and $v_{y0} = 2\pi\sqrt{2 - 0.999} = 2\pi\sqrt{1.001}$ AU/year and let $\beta = 2.99$ and $\beta = 3$ respectively. The analysis is runned over $T = 100 years$. We see that for $\beta = 2.99$, the Earth's orbit become unstable and for $\beta = 3$ the Earth escapes the gravitational field of the Sun.

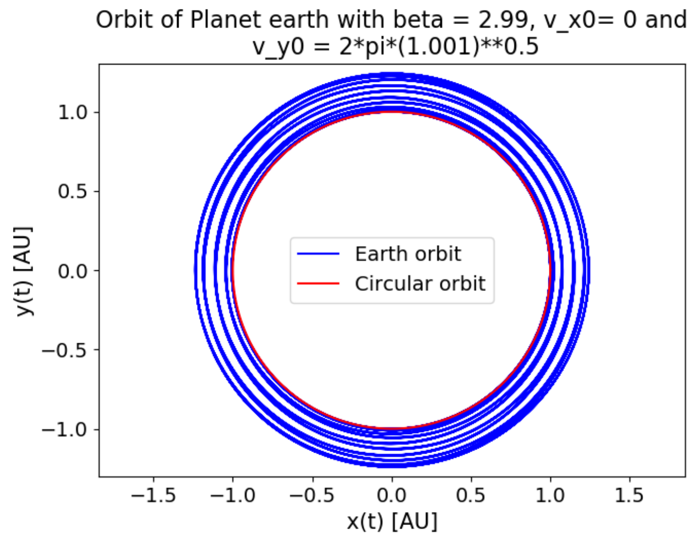

Figure 12: Planet Earth orbit: $\beta = 2.99$. The initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{y0} = 2\pi\sqrt{1.001}$ AU/year. $v_{x0} = 0$.

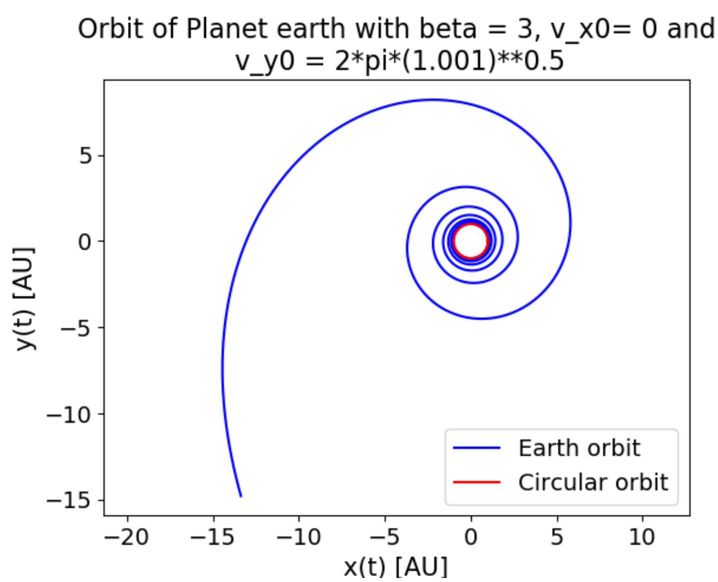

Figure 13: Planet Earth trajectory: $\beta = 3$. The initial velocity component v_{y0} has increased from $v_{y0} = 2\pi$ AU/year to $v_{y0} = 2\pi\sqrt{1.001}$ AU/year. $v_{x0} = 0$.

We include the planet Jupiter in our system, so now the Earth will experience a gravitational pull from both the Sun and from Jupiter. Jupiters mass are large, but the mass of the Sun is more than 1000 times larger. The masses in $[kg]$ of the Sun and all the planets in the solar system, and each planets average distance in $[AU]$ from the Sun are found in the project 5 description under the *Project 5 Solar System* link in [1]. We have already used planet Earth's average distance from the Sun, which is $1AU$, in our analysis which only included the Sun and planet Earth. Jupiters average distance from the Sun is $5.20AU$. However we will not use these average distances, but use positions and velocities found on the NASA site [2] as initial positions and velocities for the Earth and Jupiter instead. In Appendix B we have extracted a list from this site with position and velocity for all the planets in the solar system at 2019-Nov-29 00:00:00.0000. For now the Sun's position will still be fixed in origo.

Figure (14) shows the orbit of the Earth and Jupiter, for an analysis over $T = 12$ years (Jupiter uses in average 11.86 years to complete a single orbit around the Sun). We have used $n = 10000$ time steps. We have calculated the relative difference between maximum total energy and minimum total energy for the system of planet Earth and Jupiter based on the whole time period T to be $3.0509E - 06$. So the total energy is conserved. Since the planet orbits due to the interaction between them isn't perfect circles, neither the system's kinetic energy or the potential energy is conserved. But the sum of them, the total energy, is. The relative difference between maximum angular momentum and minimum angular momentum for the system is calculated to be $9.7038E-14$. So angular momentum is conserved.

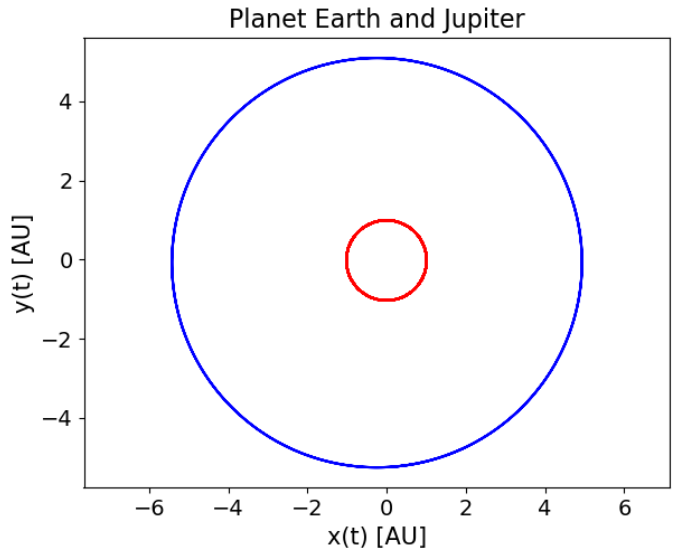

Figure 14: Planet Earth (red) and Jupiter orbiting the Sun, which is fixed in origo ($x = y = 0$).

We increase the mass of Jupiter by a factor of 10. Still the mass of the Sun is more than 100 times larger. Figure (15) shows that this mass increase doesn't have any effect on the orbit of planet Earth that we can discover visually.

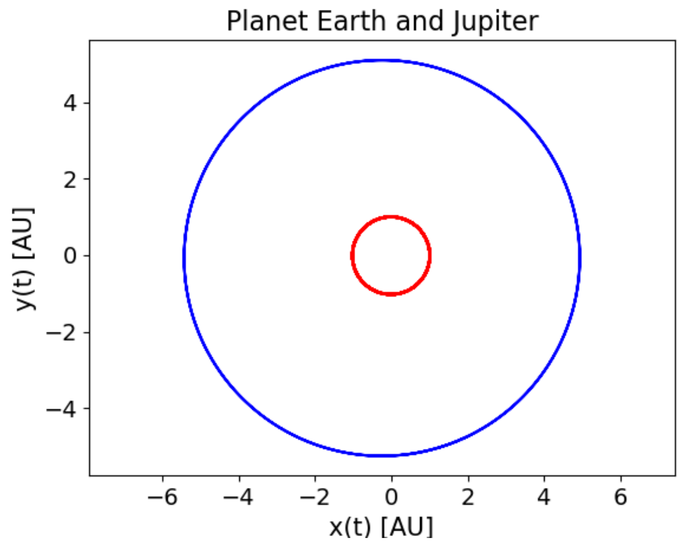

Figure 15: Planet Earth (red) and Jupiter orbiting the Sun, which is fixed in origo ($x = y = 0$). Jupiter's mass is increased by a factor of 10

When we increase the mass of Jupiter by a factor of 1000, Jupiter almost have the same mass as the Sun. It is like planet Earth is part of a system with two stars. In this situation it isn't a correct approximation to fix the position of the Sun in origo as we do. The Sun, planet Earth and Jupiter should really be orbiting around their common Mass center. It would also be important with initial velocities for the three objects suited for the situation, and which fulfill the requirement that the momentum of the system should be exactly zero.

We run three different analysis over a period of $T = 36$ years. Figure (16) shows the results when the date and time of the initial positions and velocities from [2] is 2019-Nov-29 00:00:00. The gravitational force from Jupiter makes Earth's orbit severly disturbed, before earth is sent in a straight line out of the Solar system. Increasing the number of time steps from $n = 10^6$ to $n = 3 \times 10^6$, doesn't have any significant effect on the result (that means a step size of $36/3 \times 10^6 = 1.2 \times 10^{-5}$ years.) Decrasing Jupiter's mass factor from 1000 down to 950 has a huge effect though, as figure(17) shows. Here only $n = 10^6$ time steps over $T = 36$ years are used. In figure(18) the initial conditions are dated 2019-Dec-31 00:00:00 instead. With $n = 3 \times 10^6$ time steps over $T = 36$ years, Earth still escapes from the Solar system, but the caotic orbit pattern before the escape has changed significantly from figure(16). This indicate that the initial conditions have a large impact on the result. Table(2) shows the relative difference between maximum total energy and minimum total energy and the relative difference between maximum and minimum angular momentum for these three analysis. We see that the angular momentum is conserved in the three analysis, but the conservation of energy is better when planet Earth don't escape from the Solar system than when it do so.

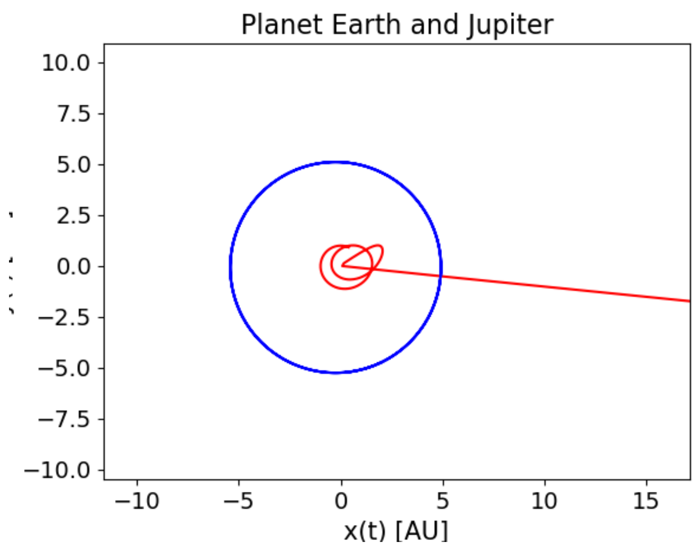

Figure 16: Planet Earth (red) and Jupiter orbiting the Sun, which is fixed in origo ($x = y = 0$). Jupiter's mass is increased by a factor of 1000. Initial conditions are dated 2019-Nov-29 00:00:00. $T = 36$ years and number of time steps are $n = 3 \times 10^6$.

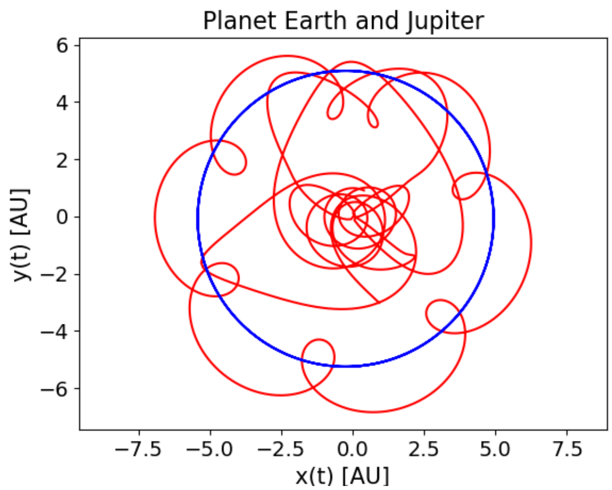

Figure 17: Planet Earth (red) and Jupiter orbiting the Sun, which is fixed in origo ($x = y = 0$). Jupiter's mass is increased by a factor of 950. Initial conditions are dated 2019-Nov-29 00:00:00. $T = 36$ years and number of time steps are $n = 10^6$.

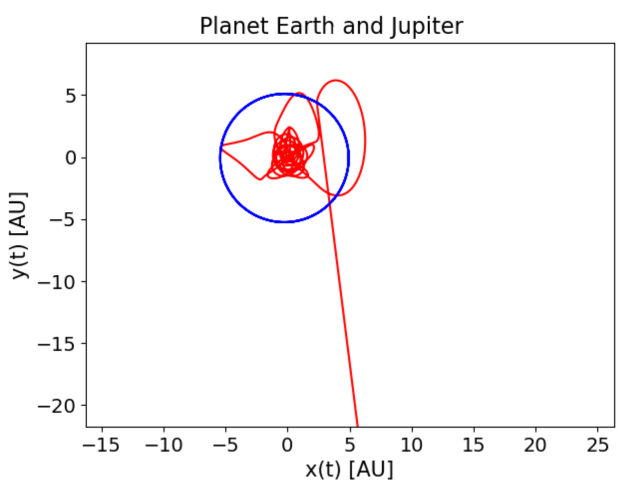

Figure 18: Planet Earth (red) and Jupiter orbiting the Sun, which is fixed in origo ($x = y = 0$). Jupiter's mass is increased by a factor of 1000. Initial conditions are dated 2019-Dec-31 00:00:00. $T = 36$ years and number of time steps are $n = 3 \times 10^6$.

Result from figure	n	Total Energy	Angular momentum
Figure 16	3E+06	2.397E-02	1.552E-13
Figure 17	1E+06	6.746E-04	9.358E-14
Figure 18	3E+06	2.499E-02	2.509E-13

Table 2: The relative difference between maximum and minimum total energy and between maximum and minimum angular momentum for the three analysis with graphical results shown in figure(16), figure(17) and figure(18).

We now find the initial position of the Sun relative to the common mass center of the Sun, planet Earth and Jupiter, with the method described in the Method section, see equation (13). The initial position and velocity of planet Earth and Jupiter are given in Appendix B. The initial velocity of the Sun is calculated from the requirement that the system's momentum is zero, see the Method section and equation (14). The analysis is performed over a time period of $T = 12$ years and with $n = 10000$ time steps. Total energy is not conserved very well, with the relative difference between maximum total energy and minimum total energy for the system being calculated to $9.9464E-02$, so including the motion of the Sun has a large impact on this result. The energy conservation doesn't improve by increasing the number of time steps. The angular momentum is conserved though, with the relative difference between maximum angular momentum and minimum angular momentum for the system being calculated to $1.4414E-14$. Figure(19) compares the orbits of planet Earth and Jupiter in the situation where the Sun orbits around the common Mass center with the situation where the Sun is fixed in origo. Visually it is not possible to spot any difference in the orbits of planet Earth and Jupiter when comparing the two situations. Figure(20) shows the Sun's orbit around the Mass center. As we see from the axes of the figure, the Sun's movements are very small, so letting the Sun's position being fixed in origo is a good approximation.

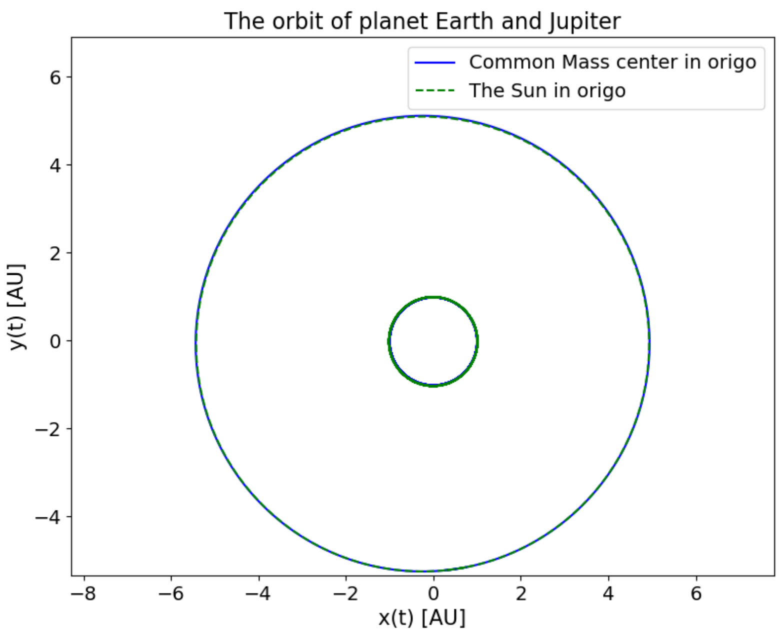

Figure 19: Comparing the orbit of planet Earth and Jupiter when the Sun is orbiting the system's common mass center, with the orbits of the two planets when the Sun is fixed in origo.

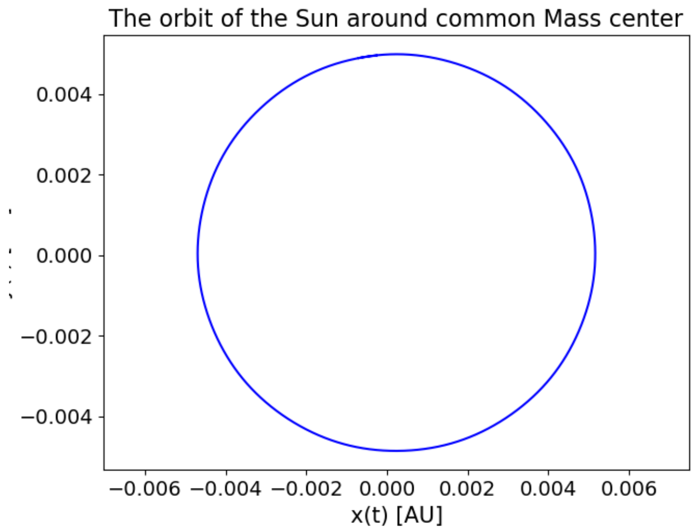

Figure 20: The Sun orbiting the Mass center of the Sun, Earth and Jupiter

Figure(21) shows the eight planets in orbit around the Mass center of the solar system: Mercury is the planet in orbit closest to the Sun. The second closest planet is Venus, then comes planet Earth (orbit marked in red), Mars, Jupiter, Saturn, Uranus and Neptune. In figure(21) we only get a proper view of the Jupiter, Saturn, Uranus and Neptune orbits. Figure(22) take a closer look at the Mercury, Venus, planet Earth and Mars orbits. Neptune uses 165 years on one orbit around the Mass center of the system. Mercury uses only 88 days, or 0.241 years. In our analysis $T = 165$ years, which is divided into $n = 150000$ time steps. Then Mercury will orbit $165/0.241 = 685$ times around the Mass center of the system during the analysis, which in turn only gives $n = 150000/685 = 219$ time steps for each Mercury orbit. Therefore the Mercury orbit is quite unclear in figure(22).

Mars uses 687 days, or 1.88 years, on one orbit around the Mass center of the solar system. In the analysis resulting in figure(23), which only shows the planets Mercury, Venus, planet Earth and Mars, we use $T = 2$ years, which is divided into $n = 150000$ time steps. During this time period Mercury orbits $2/0.241 = 8.3$ times around the Mass center of the system. With $n = 150000$ we then get $n = 150000/8.3 \approx 18000$ time steps for each Mercury orbit. The result is a much clearer picture of the Mercury orbit as the figure shows.

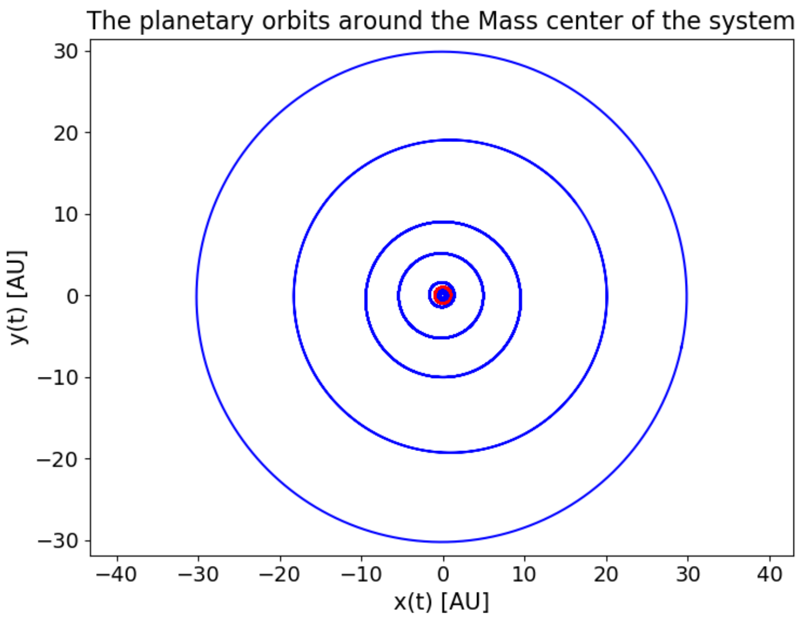

Figure 21: The eight planets of the solarsystem in orbit around the Mass center of the system. The orbit of planet Earth in red. $T = 165$ years, which is divided into $n = 150000$ time steps.

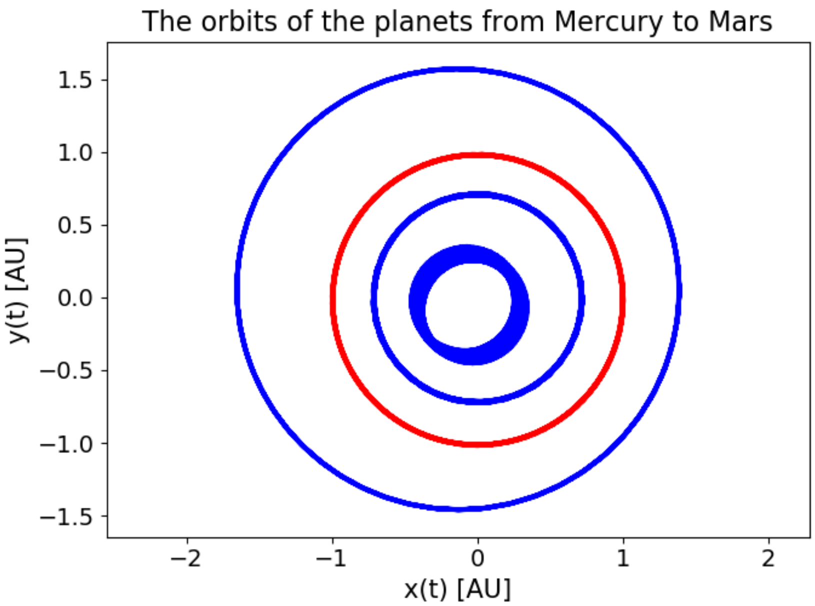

Figure 22: The four planets of the solarsystem closest to the Sun in orbit around the Mass center of the system. The orbit of planet Earth in red. $T = 165$ years, which is divided into $n = 150000$ time steps.

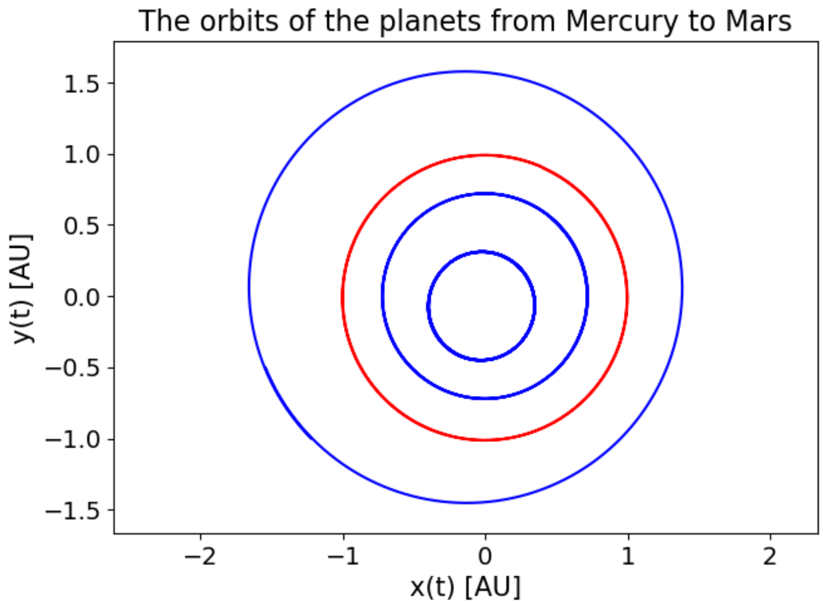

Figure 23: The four planets of the solarsystem closest to the Sun in orbit around the Mass center of the system. The orbit of planet Earth in red. $T = 2$ years, which is divided into $n = 150000$ time steps.

Figure(24) shows the development of Mercury's perihelion precession during one century, when only the gravitational impact from the Sun is included in the simulation. The figure also shows the result without the relativistic correction to the Newton gravitational force. The time step size in the simulation is $\triangle t = 10^{-6}$, so $n = 100 \times 10^6$ time steps are used in the simulation over one century. As the figure shows, both the result with and without the relativistic correction fluctuates. Without the relativistic correction, the result fluctuates around the theoretical value for this situation, which is zero degrees. With the relativistic correction included, the simulated perihelion precession fluctuates towards the observed perihelion precession value for one century, which is 43". In the end of our simulation over one century we only achieve a perihelion precession of 22.2" though, which is 52 percent of the observed value. We see the pattern, but the chosen time step size doesn't seem to be sufficient. Doubling the number of steps to $n = 200 \times 10^6$ over one century doesn't improve the result significantly.

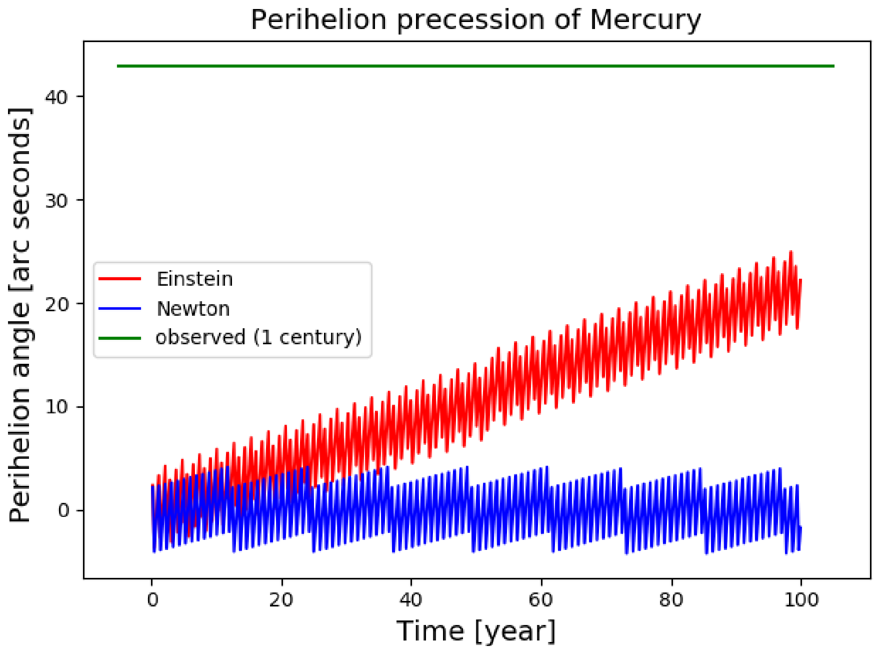

Figure 24: The observed perihelion precession of Mercury over one century when only the gravitational impact from the Sun is included is 43" (green). The simulated perihelion precession is shown in red. The simulation without the relativistic correction included is shown in blue.

Conclusion

With Newton's law of gravitation (1) as a starting point, we have shown the derivation of the x and y component of the acceleration the Sun, planet Earth and Jupiter are exposed to due to the gravitational force from the other two objects, see equation (7), (8) and (9). It is a clear pattern in these equations, so it is not difficult to expand them in order to include the other planets in the solar system, and actually the moons of the planets if we want too as well. The equations also show that including the z component in order to get a three dimentional model, is not difficult either. Including more solar system objects than the Sun and one or maybe two planets, gives many and lengthy acceleration terms to keep track of. To cope with this challenge, we uses classes in our Python programmes, when more planets than the Earth is included in our system. For the first System we investigate, which only include the Sun fixed in origo of our coordinate system and planet Earth, we just use Python functions.

We have shown the derivation of two numerical methods, the Forward Euler Method and the Velocity Verlet method. In testing both methods on the simple system with the Sun fixed in origo and only planet Earth orbiting around it, The Velocity Verlet method turns out to be a better numerical method than Forward Euler. The difference is visually clear in the plots of the Earth's orbit. In this simple system, where the Earth has a cirular orbit, potential and kinetic energy is conserved separately. For Potential and Kinetic energy, the relative error are reduced by an order of $1/n$ in the Forward Euler method and by an order of $1/n^2$ in the Velocity Verlet method, where n is the number of time steps in the analysis. This coincides with the order of the error terms in the derivation of the two methods. Angular momentum should also be conserved. Again the relative error is reduced by the order of $1/n$ for the Forward Euler method. The Velocity Verlet method conserves angular momentum. The result is only disturbed slightly by numerical errors as n increases. Based on these results, we conclude that the Velocity Verlet method is a far better method to use than Forward Euler for physical problems where energy and angular momentum are conserved. Using the Velocity Verlet method, our analyses also models the escape velocity of the Earth well. The result coincides with the analytical escape velocity.

When Jupiter is included in our system, we only use the Velocity Verlet method. We look at the system when the Sun is still fixed in origo of our coordinate system, and when the Sun, the Earth and Jupiter orbit around origo where their common Mass center is fixed. Since the extent of the Sun's orbit is so small, fixing the Sun in origo of our coordinate system is a good approximation. When the Sun is fixed in origo, we investigate what happens when the mass of Jupiter is increased by a factor of 1000. Then Jupiter almost has the same mass as the Sun, and the system becomes very unstable. We experience that the initial position and velocity of the two planets have a significant impact on planet Earths orbit, but Earth eventually escapes the Solar system in both our tests. In order for planet Earth to stay in the Solar system, we reduce the mass factor to 950. In our analysis where the Sun, the Earth and Jupiter orbits around their common mass center, we experience that the conservation of total energy become quite poor, with a relative error of 10 percent for $n = 10000$ time steps. Increasing the number of time steps n doesn't improve this result. The movement of the Sun around the common mass center of the system therefore seem to have a large impact of the conservation of energy in this analysis. We only look at the visual result when all the planets in the Solar system are included. The visual result looks reasonable.

The perihelion percession of Mercury over one century is also investigated by adding the general relativistic correction to the Newton gravitational force between the Sun and Mercury. The other planets in the Solar system are not part of the simulation. The observed value of the perihelion percession of Mercury over one century when the gravitational pull from the other planets are excluded, is 43" (arc seconds). Our simulation over one century with time step size $\triangle t = 10^{-6}$ year, only manage to obtain 52 percent of this value. Doubling the number of time steps over one century to 200×10^6 hardly improves the result. In order to get an improvement of any significance, we probably have to increase the number of time steps over one century by at least a factor of 10. Such a simulation will take quite some time. The results without the general relativistic correction looks reasonable though. The perihelion of Mercury then has small fluctuations around 0 degrees, which is the theoretical result with the classical Newton's law of gravitation (The perihelion of Mercury doesn't change its position in space when simulating it with Newton's classical law of gravitation).

References

[1] Morten Hjort-Jensen: *Course material in Computational Physics.* http://compphysics.github.io/ComputationalPhysics/doc/web/course

[2] NASA site for initial positions and velocities of the solarsystem planets: https://ssd.jpl.nasa.gov/horizons.cgi# results

[3] Andreas Bjurstedt: *Computational Physics, Project5.* https://github.com/abjurste/A19-FYS4150

Appendix A: Computer programs

The programming language used is Python3. For the simple system where the Earth is the only planet in orbit around the Sun, we have programmed the Forward Euler method and the Velocity Verlet method as functions in a program we call `Euler_and_Verlet_func.py`. This file is called by the Python scripts `Project5c.py` and `Project5d.py`, which deal with section 5c and 5d in the project description found in [1].

When Jupiter, and later the rest of the planets in the solar system, are included, we turn over to Python classes with objects/instances. The program `Project5_classes.py` contains 2 classes. The class called *Planet*, organizes one object/instance for the Sun and one object/instance for each planet included in the model. Each instance contains the mass and initial conditions for the planet/Sun which it represents. The class called *System*, calculates the acceleration components for the Sun and each of the included planets, and then uses the Velocity Verlet method in order to calculate the velocity and position components. This file is called by the Python script `Project5e.py`, which deals with section 5e in the project description, and `Project5f.py` and `Project5f_2.py` which deal with section 5f in the project description. In `Project5f.py`, the Sun, planet Earth and Jupiter orbit around their common mass center. In `Project5f_2.py`, the Sun and all the planets in the solar system are included and orbit around their common mass center. A separate script called `Project5g.py` simulates the perihelion precession of Mercury. All the programs/scripts described here are included in the GitHub repository [3].

Appendix B: Initial conditions for the planets extracted from [2]

```
----Mercury----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X =-3.089137495084154E-01 Y = 1.744886373010318E-01 Z = 4.167600354497743E-02
VX=-1.928258980107407E-02 VY=-2.350312105925493E-02 VZ=-1.520556440066312E-04
LT= 2.063169057337859E-03 RG= 3.572266485776222E-01 RR= 5.176804377659569E-03

----Venus----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 4.814605067455450E-01 Y =-5.345470370402726E-01 Z =-3.540916553726071E-02
VX= 1.493272115673404E-02 VY= 1.341199462523215E-02 VZ=-6.779258843704102E-04
LT= 4.159971440136809E-03 RG= 7.202767269378184E-01 RR= 6.133517258383193E-05

----Planet Earth----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 3.948527228009325E-01 Y = 9.100160380472437E-01 Z =-2.709495540997714E-05
```

```
VX=-1.603566066362973E-02 VY= 6.880628606437826E-03 VZ= 6.059839199455330E-07
LT= 5.729238098057788E-03 RG= 9.919868259914791E-01 RR=-7.080931502273646E-05

2458848.500000000 = A.D. 2019-Dec-31 00:00:00.0000 TDB
X =-1.528711301502854E-01 Y = 9.793926255065635E-01 Z =-1.893719660022524E-05
VX=-1.729804323045028E-02 VY=-2.680908671655300E-03 VZ= 7.650397297406155E-07
LT= 5.724991096969272E-03 RG= 9.912514805480407E-01 RR= 1.887434342451467E-05

----Mars----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X =-1.543208932754952E+00 Y =-5.042083307102040E-01 Z = 2.707009173557715E-02
VX= 4.926983218616618E-03 VY=-1.208451455394788E-02 VZ=-3.740559366586062E-04
LT= 9.377799730514502E-03 RG= 1.623715689632524E+00 RR=-9.363568647152507E-04

----Jupiter----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 2.771209156933313E-01 Y =-5.224508231691265E+00 Z = 1.546777941340911E-02
VX= 7.443826199722129E-03 VY= 7.587138148929383E-04 VZ=-1.696810240208194E-04
LT= 3.021679300686650E-02 RG= 5.231875525767454E+00 RR=-3.638640135743631E-04

2458848.500000000 = A.D. 2019-Dec-31 00:00:00.0000 TDB
X = 5.149322540479010E-01 Y =-5.194692596109877E+00 Z = 1.002424689480117E-02
VX= 7.417650642179744E-03 VY= 1.104560540984019E-03 VZ=-1.704718909239744E-04
LT= 3.014913887542619E-02 RG= 5.220161576030327E+00 RR=-3.677996789269148E-04

----Saturn----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 3.632628879585697E+00 Y =-9.348288811274543E+00 Z = 1.793542343960454E-02
VX= 4.891570166847385E-03 VY= 2.004764720827292E-03 VZ=-2.299017365589604E-04
LT= 5.792439433020694E-02 RG= 1.002929797918153E+01 RR=-9.731328567150615E-05

----Uranus----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 1.629688404988837E+01 Y = 1.128605542338266E+01 Z =-1.692115522793217E-01
VX=-2.268171563746997E-03 VY= 3.050128900966884E-03 VZ= 4.061969844748497E-05
LT= 1.144940404906019E-01 RG= 1.982402858413486E+01 RR=-1.284844029699959E-04
```

```
----Neptune----
$$SOE
2458816.500000000 = A.D. 2019-Nov-29 00:00:00.0000 TDB
X = 2.921750763559268E+01 Y =-6.461552366128481E+00 Z =-5.402832642395710E-01
VX= 6.568727514842341E-04 VY= 3.084041276878159E-03 VZ=-7.824234958842697E-05
LT= 1.728517517796348E-01 RG= 2.992835306898384E+01 RR=-2.316320393946492E-05
```