Project3: Simulation of the solar system with numerical methods

FYS4150: Computational Physics

Sajjad Ahmadigoltapeh, Einar Aurbakken, Bastian Skjelstad, Per Harald Barkost

sajjadah@uio.no, eanorway@gmail.com, bastiabs@uio.no, ph.barkost@gmail.com

Link to program: `https://github.com/einaur/fys4150/tree/master/Project3`

Abstract

In this study, two algorithms, Forward Euler and Velocity Verlet, are used to examine the celestial bodies of the solar system. Several situations are analyzed, such as circular orbiting, escape velocity of the circular orbit, non-circular orbits and many-body systems with non-circular orbits. For investigating the motion of Mercury and its perihelion precession, the relativistic effect is taken into account as well. Furthermore, the choice of time step, dt, and how it affects the results is discussed.

It was found that the Forward Euler algorithm does not conserve energy and angular momentum. The Velocity Verlet algorithm, on the other hand, conserves both the energy and angular momentum. Therefore, Velocity Verlet was used throughout the study.

Utilizing the Velocity Verlet algorithm, the circular motion is reproduced accurately and the determined escape velocity is in good correspondence with the theoretical value. Furthermore, the perihelion precession found for Mercury is in good agreement with the value given in the literature [1].

Contents

1 Introduction

The aim of this work is to analyze the motion of some of the solar system's celestial bodies, resulting from the gravitational pull by the other bodies. To successfully predict their trajectory, a reliable numerical integration method is required. This is a problem which not only applies when determining the orbits of astronomical objects, but also in many other fields of research. One instance where this problem of numerical integration appears may be the prediction of the trajectory of a macromolecule over time, in a so-called molecular dynamics (MD) simulation, to examine the behavior of the macromolecule and to find its stable conformation(s).

In this study, the Forward Euler algorithm and the Velocity Verlet algorithm are utilized to reproduce the celestial bodies' motion. Furthermore, the precision and run-time of these algorithms will be discussed. The more favorable method will then be used to analyze different astronomical systems such as two-body systems, three-body systems and many-body systems, and some of their properties. These properties that will be investigated are the reproduction of a perfectly circular orbit, the determination of escape velocity, the variation of the distance-dependent gravitational force, the effects of two similar masses affecting an object, and the relativistic effects accounting for observed perihelion precession.

2 Theory

2.1 Newton's law of gravitation

Newton's law of gravitation states that the force acting from The Sun on Earth is

$$F_{21} = -Gm_1m_2\frac{r_{21}}{|r_{21}|^3}, \tag{1}$$

where G is the gravitational constant, m_1 and m_2 the Sun and Earth masses, respectively, and r_1 and r_2 the corresponding position vectors.$r_{21} = r_2 - r_1$ is the distance vector from Earth to The Sun. As is observed, the gravitational force between two celestial bodies is proportional to the product of masses and the inverse of the distance, r_{21}. Resulting acceleration due to the force from the Sun onto the Earth is:

$$a_{21} = -Gm_1 \frac{r_{21}}{|r_{21}|^3}.$$

Considering Newton's third law, the same force acts from the Earth onto the Sun, but with opposite direction. So, by the same method the acceleration due to the force from the Earth onto the Sun could be calculated. In this case study, it is assumed that the mass of the sun is much larger than the other planets, $m_\odot \gg m_{\text{earth}}$. Therefore, the movement of the Sun is approximated to zero.

To extend a two-body system to a three-body system, we add Jupiter to the current system. If the acceleration of the Sun is a_1, the acceleration of The Earth is a_2, and the acceleration of Jupiter is a_3, the following may be written:

$$a_1 = a_{12} + a_{13} = -Gm_2 \frac{r_{12}}{|r_{12}|^3} - Gm_3 \frac{r_{13}}{|r_{13}|^3},$$

$$\begin{aligned} a_2 = a_{21} + a_{23} &= -Gm_1 \frac{r_{21}}{|r_{21}|^3} - Gm_3 \frac{r_{23}}{|r_{23}|^3} \\ &= +Gm_1 \frac{r_{12}}{|r_{12}|^3} - Gm_3 \frac{r_{23}}{|r_{23}|^3}, \end{aligned}$$

$$\begin{aligned} a_3 = a_{31} + a_{32} &= -Gm_1 \frac{r_{31}}{|r_{31}|^3} - Gm_2 \frac{r_{32}}{|r_{32}|^3} \\ &= +Gm_1 \frac{r_{13}}{|r_{13}|^3} + Gm_2 \frac{r_{23}}{|r_{23}|^3}. \end{aligned}$$

The above equations can be extended for an m-body system as below:

$$a_i = \sum_{j \neq i} a_{ij} = \frac{d^2 r_i}{dt^2} = \sum_{j=1}^{i-1} Gm_j \frac{r_{ji}}{|r_{ji}|^3} + \sum_{j=i+1}^{m} -Gm_j \frac{r_{ij}}{|r_{ij}|^3}. \tag{2}$$

The approach of this study is modelling of 3-dimensional solar system. Therefore, the following differential equations must be solved for each planet:

$$\frac{d^2 r_x}{dt^2} = \frac{F_x(x,y,z)}{m}, \tag{3}$$
$$\frac{d^2 r_y}{dt^2} = \frac{F_y(x,y,z)}{m}, \tag{4}$$
$$\frac{d^2 r_z}{dt^2} = \frac{F_z(x,y,z)}{m}. \tag{5}$$

In order to solve the obtained 2nd order ordinary differential equations (ODEs), with the aim of calculating r and v, one can convert eq. (3), 4 and 5 to a system of ODEs including first-order ODEs as it is illustrated in eq. (6) and (7).

$$\begin{aligned} \frac{\mathrm{d}r_x}{\mathrm{d}t} &= v_x(t) \\ \frac{\mathrm{d}v_x}{\mathrm{d}t} &= \frac{F_x(t)}{M} = a_x(t) \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\mathrm{d}r_y}{\mathrm{d}t} &= v_y(t) \\ \frac{\mathrm{d}v_y}{\mathrm{d}t} &= \frac{F_y(t)}{M} = a_y(t) \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\mathrm{d}r_z}{\mathrm{d}t} &= v_z(t) \\ \frac{\mathrm{d}v_z}{\mathrm{d}t} &= \frac{F_z(t)}{M} = a_z(t) \end{aligned} \tag{8}$$

2.2 Relativistic equation

Newton's law of gravity approximate the solar system with a suitable accuracy, but a planet like Mercury is seen as an exception. Mercury moves faster (particularly in perihelion) than Newton's prediction. The perihelion of Mercury's orbital path around Sun changes 43 arc second per century, which Newton's gravitational law can not predict. To explain this we need to take into account the general theory of relativity.

The Newtonian gravitational force with a factor correcting for the general relativity for the Sun-Mercury system is given by:

$$F_G = \frac{GM_{\text{Sun}}M_{\text{Mercury}}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right], \tag{9}$$

where F_G is the force, G is the gravitational constant, M_{Sun} and M_{Mercury} are the masses of the Sun and Mercury, respectively, r is the distance separating the objects, l is the angular momentum, and c is the speed of light.

2.3 Taylor expansion

After establishing a system of 1st-order ODEs, derived equations can be approximated using Taylor method to expand $\vec{r}$ and $\vec{v}$ with respect to t, using h, time step, as the variable:

$$\begin{aligned} r(t+h) &= r(t) + r'(t)h + \frac{r''(t)}{2}h^2 + \mathcal{O}(h^3) \\ &= r(t) + v(t)h + \frac{a(t)}{2}h^2 + \mathcal{O}(h^3) \end{aligned} \tag{10}$$

$$\begin{aligned} v(t+h) &= v(t) + v'(t)h + \frac{v''(t)}{2}h^2 + \mathcal{O}(h^3) \\ &= v(t) + a(t)h + \frac{a'(t)}{2}h^2 + \mathcal{O}(h^3) \end{aligned} \tag{11}$$

Then, **Forward Euler** and **Velocity Verlet** methods are applied separately for approximation of r and v. It is worth to note that for both methods, the time is discretized to N integration points i.e. $\Delta t = \frac{t_f - t_0}{N}$ so that:

$$t \rightarrow t_i = t_0 + i\Delta t$$

where $i = 0, 1, \ldots, N$. which gives:

$$r(t) = r(t_i) \rightarrow r_i$$

and the same dicretizing strategy for velocity and acceleration gives: $v(t_i) \rightarrow v_i$ and $a(t_i) \rightarrow a_i$. Details of Forward Euler and Velocity Verlet will be described in the next sections.

2.4 Forward Euler method

Euler method is a basic numerical method to solve ODEs. Using the first Taylor expansion in eq. (10) and eq. (11) and removing the terms with order, higher than one gives the equations for the Euler method:

$$
\begin{aligned}
r_x(t+h) &= r_x(t) + v_x(t)h + \mathcal{O}(h^2) \\
v_x(t+h) &= v_x(t) + a_x(t)h + \mathcal{O}(h^2)
\end{aligned}
$$

The same approximation for $v_y(t)$, $v_z(t)$, $r_y(t)$ and $r_z(t)$ could be written easily as shown above. Generally, discretizing form of approximated r and v are given as:

$$
\begin{aligned}
r_{i+1} &= r_i + v_i h \\
v_{i+1} &= v_i + a_i h
\end{aligned}
$$

It is worth to note that error of Euler method is $\mathcal{O}(h^2)$ per each iteration. Therefore, after N iteration the error will be $\mathcal{O}(h)$.

2.5 Velocity Verlet method

In order to derive Verlet method, it is required to write discretized form of Taylor expansion for r and v as below:

$$
\begin{aligned}
\text{Taylor:} \quad r_{i+1} &= r_i + hr'_i + \frac{h^2}{2!}r''_i + \mathcal{O}(h^3) \\
v_{i+1} &= v_i + hv'_i + \frac{h^2}{2!}v''_i + \mathcal{O}(h^3)
\end{aligned} \tag{12}
$$

approximation of a by first order Taylor expansion gives:

$$
\text{First order Taylor:} \quad a_{i+1} = a_i + ha'_i + \mathcal{O}(h^2)
$$

factorizing of a' term gives:

$$
\begin{aligned}
ha'_i &= a_{i+1} - a_i + \mathcal{O}(h^2) \\
\frac{h^2}{2!}a' &= \frac{h}{2}(a_{i+1} - a_i) + \mathcal{O}(h^3)
\end{aligned} \tag{13}
$$

Now, plugging eq. (13) into eq. (12) gives:

$$\begin{aligned} v_{i+1} &= v_i + ha_i + \frac{h}{2}(a_{i+1} - a_i) + \mathcal{O}(h^3) \\ &= v_i + \frac{h}{2}(a_{i+1} + a_i) + \mathcal{O}(h^3) \end{aligned} \tag{14}$$

Finally Velocity Verlet method could be written as:

$$\begin{aligned} r_{i+1} &= r_i + hv_i + \frac{h^2}{2}a_i \\ v_{i+1} &= v_i + h(\frac{a_{i+1} + a_i}{2}) \end{aligned} \tag{15}$$

As it could be seen, in this method a_{i+1} depends on r_i. It is important to note that in eq. (15) the error for r and v is $\mathcal{O}(h^3)$. However, according to eq. (13) error of a is $\mathcal{O}(h^2)$. Since the dominant error per iteration is $\mathcal{O}(h^3)$, after N iteration the accumulated error will be $\mathcal{O}(h^2)$.

2.6 Circular orbit of the earth around a stationary

From Newton gravitational law the force between earth and the sun can be expressed as

$$F_e = \frac{Gm_e m_\odot}{r^2}. \tag{16}$$

By applying Newton's second law, the acceleration of the earth can be written as

$$a_e = \frac{F_e}{m_e} \tag{17}$$

with direction directly towards the sun. Assuming that the earth moves in a circular orbit with radius r around the Sun the velocity is tangential to the circular path. And the absolute value of the velocity will be constant. The acceleration can be written in terms of the absolute value of the velocity v_{cric} and radius r:

$$a_e = \frac{v_{circ}^2}{r} = \frac{Gm_\odot}{r^2} \tag{18}$$

Further, we can solve eq. (18) with respect to v.

$$v_{circ}^2 = \frac{Gm_\odot}{r} \Rightarrow v_{circ} = \sqrt{\frac{Gm_\odot}{r}} \tag{19}$$

with direction tangential to the circular orbit around the Sun.

2.7 Escape velocity

Conservation of energy can be used to estimate the escape velocity of earth in the earth-sun system. The escape velocity is reached when the kinetic energy is equal or greater than the potential energy due to gravity, that is

$$\frac{1}{2}m_e\mathbf{v_e}^2 = \frac{Gm_em_\odot}{r}.$$

Solving for the velocity gives the escape velocity

$$v_e = |\mathbf{v_e}| = \sqrt{\frac{2GM_\odot}{r}} \tag{20}$$

For a more general centrosymmetric force of the form where we assume $\beta > 1$

$$F_e = \frac{Gm_em_\odot}{r^\beta} \tag{21}$$

the potential energy is given by

$$U = \int_0^r \frac{Gm_em_\odot}{r^\beta}dr = \frac{1}{\beta - 1}\frac{Gm_em_\odot}{r^{\beta-1}} \tag{22}$$

Equating with the kinetic energy and solving for velocity again, gives the escape velocity

$$v_e = \sqrt{\frac{2}{\beta - 1}\frac{GM_\odot}{r^{\beta-1}}} \tag{23}$$

2.8 Programming

2.8.1 Initial conditions

All of the computation code is written in C++. The position data for each object (planets, moons and the sun) is written to files. Plotting is done with

Python. In order to execute the code in C++, it is required to assign initial conditions. The reliable initial conditions are obtained from NASA website [2].

2.8.2 Object oriented program

Object orient technique in programming is used in this study. The program has three classes:

Lin: This class provides vector operations for three-dimensional vectors, like addition, subtraction scalar multiplication and cross-product.

Planet: Contains information about a planet (position and velocity at a particular time, and its mass). This is therefore also used to set the initial condition.

Solver: This class contains numerical algorithms for Euler and Velocity Verlet that are used for solving the derived ODEs. It also has functions for computation of forces and writing to file.

2.8.3 Unit test

Unit tests are implemented using catch2. In our model of the solar system the planets are simplified as point masses which means the planets will not have no spin along its own axis. This means that the energies in our model consists of potential and transnational energy from the planets. The gravitational force is conservative, thus, the energy of our model can be written as:

$$(K + U)_0 = (K + U)_1 = constant$$

A unit test of the sun-earth system for the Velocity Verlet solver is implemented to test the conservation of energy. The initial energy is compared with the final, and must be within a relative error ϵ to pass.

$$E_{initial} = (K_{initial} + U_{initial})$$
$$E_{final} = (K_{final} + U_{final})$$

E_{final} must not deviate from $E_{initial}$ more than $\pm\ \epsilon\ E_{initial}$ to pass.

Since, there is no external torque, angular momentum should be conserved (Newtons third law for rotation). This fact allow the initial angular momentum of the model to remain unchanged during the simulations. Total angular momentum of the model can be written as a summation of angular momentum of each planet around the center of mass:

$$L = \sum_i \boldsymbol{r_i} \times \boldsymbol{p_i} = \sum_i m_i(\boldsymbol{r_i} \times \boldsymbol{v_i})$$

Where $\boldsymbol{p_i}$ is momentum vector, $\boldsymbol{v_i}$ velocity vector and $\boldsymbol{r_i}$ is a vector showing distance from center of mass. Center of mass is defined as following:

$$R = \frac{\sum_i m_i \boldsymbol{r_i}}{M}$$

Where $\boldsymbol{r_i}$ position vector and $\boldsymbol{m_i}$ is mass of each planet and M is the total mass of the system. The initial angular momentum is compared with the final angular momentum in a unit test in the same manner as the energy.

The initial energy and angular momentum is also compared with a precalculated values.

3 Algorithm

Prior to write an understandable algorithm, some notation such as N, m, r, v and a are defined as number of integration points, number of celestial bodies, position, velocity and acceleration respectively. Furthermore, updated values for position, velocity and acceleration through iteration is shown as r_{i+1}, v_{i+1} and a_{i+1} letters.

3.1 Acceleration

To have an efficient algorithm, instead of summing up the contribution of all bodies into final acceleration, we utilize this fact that a_{ij} and a_{ji} have the same value and different sign. γ is defined as below:

$$\gamma = G\frac{r_i - r_j}{|r_i - r_j|^3}$$

then

$$a_{ij} = -m_j \gamma_{ij} \tag{24}$$

and

$$a_{ji} = m_i \gamma_{ij} \tag{25}$$

Then, for computing the acceleration for the first planet i.e. a_1, it is required to compute every single a_{1j} and summing them to get a_1.

simultaneously, a_{j1} could be calculated and stored in a_j . Afterward, for the second planet, it is clear that a_{21} was computed already and now a_{2j} should be computed. This algorithm continues til last step

which is planet number m and at this step a_m is computed. Suppose, using non-efficient method i.e. summing up all contributions, to compute final acceleration. In this case two loops with m and $m-1$ iterations are required that totally gives $m(m-1)$ loop. However, for the efficient method which is summarized above, only $\frac{m(m-1)}{2}$ loops are required which gains around fifty percent of FLOPS.

Algorithm 1 Acceleration

procedure ACC($m, r, v, \Delta t$) ▷ m → number of integration points
  for $i \leftarrow 1, \ldots, m$ **do**
    $a_i \leftarrow 0$
  end for
  for $i \leftarrow 1, \ldots, m-1$ **do**
    for $i \leftarrow i+1, \ldots, m$ **do**
      $\gamma = G\frac{r_i - r_j}{|r_i - r_j|^3}$
      $a_i = a_i + m_j\gamma$
      $a_j = a_j + m_i\gamma$
    end for
  end for
end procedure

Algorithm 2 Forward Euler Method

procedure EULER($N, r, v, \Delta t$) ▷ N → number of integration points
  for $i \leftarrow 0, \ldots, N-1$ **do** ▷ Accel=$\frac{GM}{r_{ij}^2}$
    $a_i \leftarrow$ ACCEL($m, r_i, v_i, \Delta t$) ▷ m → number of planets
    $v_{i+1} \leftarrow v_i + a_i\Delta t$
    $r_{i+1} \leftarrow r_i + v_i\Delta t$
  end for
end procedure

According to Velocity Verlet method which summarized as eq. (15), it is realized that acceleration is being updated by known r_{i+1} and v_{i+1}. But, in case of dependence of a_{i+1} to v_{i+1} (such as relativistic correction), it is not possible to calculate a_{i+1} and v_{i+1} simultaneously. The solution for this, is using Euler method to compute v_{i+1} and then plugging this value into eq. (15) to calculate a_{i+1}. Once more, the eq. (15) is summarized as below:

$$r_{i+1} = r_i + hv_i + \frac{h^2}{2}a_i$$
$$v_{i+1} = v_i + h(\frac{a_{i+1} + a_i}{2})$$

Actually, it could be said that the V_n is calculated via Euler method and time step of $t_i + \frac{h}{2}$.

Algorithm 3 Velocity Verlet Method

procedure VERLET($N, r, v, \Delta t$) ▷ N → number of integration points
  for $i \leftarrow 0, \ldots, N-1$ **do** ▷ Accel=$\frac{GM}{r_{ij}^2}$
    $a_i \leftarrow (m, r_i, v_i, \Delta t)$ ▷ M → NUMBER OF PLANETS
    $r_{i+1} \leftarrow r_i + v_i \Delta t + \frac{1}{2} a_i \Delta t^2$
  end for
  for $i \leftarrow 0, \ldots, m-1$ **do**
    $a_{i+1} \leftarrow$ ACCEL($m, r_i, \Delta t$)
    $v_{i+1} \leftarrow v_i + \frac{a_{i+1}}{2} \Delta t$
    $r_{i+1} \leftarrow r_i + v_{i+1} \Delta t + \frac{a_i + a_{i+1}}{2} \Delta t^2$
  end for
end procedure

Therefore, Verlet could be reformulated as:

$$
\begin{aligned}
r_{i+1} &= r_i + h v_{i+\frac{1}{2}} \\
v_{i+1} &= v_{i+\frac{1}{2}} + (\frac{h}{2}) a_{i+1} \\
v_{i+\frac{1}{2}} &= v_i + (\frac{h}{2}) a_i
\end{aligned}
$$

Algorithm 4 Classical acceleration

function CLASSICAL($m, r, v, \text{target}, \Delta t$) ▷ Estimate acceleration of target
  $a_i \leftarrow \mathbf{0}$ ▷ zero vector
  $\gamma = \frac{G}{r_{ij}^2}$
  for $i \leftarrow 0, \ldots, m-1$ **do** ▷ m → number of planets
    $a_i = \gamma m_i$
    if $i \neq$ target **then** ▷ repeating interaction forces are avoided to keep FLOPS low
      $dist \leftarrow r_{\text{target}} - r_i$
      $d \leftarrow \|dist\|$
      $a \leftarrow a - \frac{G m_i \cdot dist}{d^3}$ ▷ d^3 due to using unit vector
    end if
  end for
  return a
end function

Algorithm 5 Relativistic acceleration

function RELATIVISTIC($m, r, v, \text{target}, n, \Delta t$) ▷ Estimates the acceleration of target
  $a \leftarrow \mathbf{0}$ ▷ zero vector
  $\gamma = \frac{G}{r_{ij}^2}$
  for $i \leftarrow 0, \ldots, m-1$ **do** ▷ m → number of planets
    $a_i = \gamma m_i$
    if $i \neq \text{target}$ **then**
      $dist \leftarrow r_{\text{target}} - r_i$
      $d \leftarrow \|dist\|$
      $l \leftarrow \|r_{\text{target}} \times v\|$
      $a \leftarrow a - \frac{Gm_i dist}{d^3}\left[1 + \frac{3l^2}{d^2c^2}\right]$ ▷ d^3 due to using unit vector
    end if
  end for
  return a
end function

3.2 Astronomical Unit consistency

Apparently, metric unit is not suitable for evaluation of solar system with huge dimensions.For astronomical systems, it is more efficient to use AU, years, and solar mass instead of length, time (second or hour) and mass, respectively. Hence, the gravity constant should become consistent with astronomical dimensions.As it was illustrated in eq. (19), linear velocity is shown as:

$$v = \sqrt{\frac{Gm_{\odot}}{r}} = \sqrt{\frac{Gm_{\odot}}{1AU}}$$

where

$$G = \frac{1AU}{m_{\odot}}v^2$$

plugging the astronomical units into formula, gives:

$$v = \frac{2\pi \cdot 1AU}{1year}$$

Therefore, G with consistent unit to the a solar system is written as following equation. As it is illustrated here, the constant number is $4\pi^2$:

$$G = \frac{2\pi \cdot 1AU}{1year} \cdot \frac{1AU}{1m_\odot} = 4\pi^2 \cdot AU^3 \cdot m_\odot{}^{-1} \cdot (year)^{-1} \tag{26}$$

3.3 Comparison of floating points and errors

Optimization of the code is a significant strategy that allows a code developer to conserve time and memory during code execution. In Euler method, there are 4 FLOPS per iteration for position and velocity. By Considering 4 FLOPS for 3-dimension it becomes 12 and finally for for N iteration it will be $12N$ FLOPS. It is important to note that, interaction forces between different planets introduce some FLOPS, but they are the same for Euler and velocity Verlet method. That's why these FLOPs will not be counted in comparison of velocity Verlet and Euler method. In velocity Verlet method it is decided to pre-calculate $\Delta t^2/2$ which provides lower FLOPS. By this approach, estimated FLOPS for velocity Verlet is 7 per iteration and for 3-dimension it becomes $21N$. On the other side, Euler method has error of $\mathcal{O}(h^2)$ per iteration. Therefore after N iteration accumulated error will be $\mathcal{O}(h)$. However, velocity Verlet at each iteration has error of $\mathcal{O}(h^3)$, therefore, accumulated error for velocity Verlet after N iteration will be $\mathcal{O}(h^2)$. Consequently, velocity Verlet method is more precise than Euler although, Euler method has lower FLOPS. Evidently a method with lower FLOPS requires shorter run time.

4 Results and discussion

4.1 The Earth-Sun system

4.1.1 Circular Orbit

The velocity for circular motion is determined by eq. (19) to be

$$v_{circ} = 2\pi \approx 6.2831853071795862 \text{ AU/year}.$$

when the mass of the sun i set as 1, and the mass of the other planets are scaled accordingly.

Figure 1 shows a plot of the earth-sun system utilizing the forward Euler algorithm with time steps 10^{-3} and 10^{-4} years. The initial position of the earth is 1 AU in the x-direction, and the initial velocity is v_{circ} in the y-direction. The simulation time is 100 yeas, and the sun position is fixed at the origin. Simulation with the same setup using the velocity Verlet algorithm, with time steps 10^{-1} and 10^{-3} years is shown in Figure 2. From Figure 1 it is clear that the Euler algorithm does not reproduce a circular motion. During the simulation, earth has drifted out to a distance of approximately 3 AU and 1.5 AU with time steps 10^{-3} and 10^{-4} years, respectively. This suggests that the potential energy decreases, and the kinetic energy increases, during the simulation, as can be seen in Figure 3. Also, it is seen that angular momentum is not well conserved. The problem with non-conserved energies and angular momentum is reduced by decreasing the time step, but the drifts in the values are still severe with dt = 10^{-4}.

As can be seen in Figure 2, the velocity Verlet algorithm reproduces well the expected circular motion with dt = 10^{-3}. At dt = 10^{-1}, the position fluctuates significantly about the ideal circle, but even at this large time step, it does not have the same drifting problem as the Euler method. This is also illustrated in Figure 4, where it can be seen that both kinetic and potential energy fluctuates, but the average does not change throughout the simulation. Also, the angular momentum is well conserved with both time steps. A unit test on the sun-earth system with time step 10^{-3} in a 100 years simulation passed with allowed relative error, ϵ, of 10^{-7} in both energy and angular momentum.

The CPU time were measure with the clock() function in C++ for both Euler and Verlet. Writing to file is not included in the timing. The average CPU time of ten runs with Euler was 423 271 and for the Velocity Verlet it was 667 026. This is in accordance with the higher number of FLOPs needed using the Velocity Verlet algorithm.

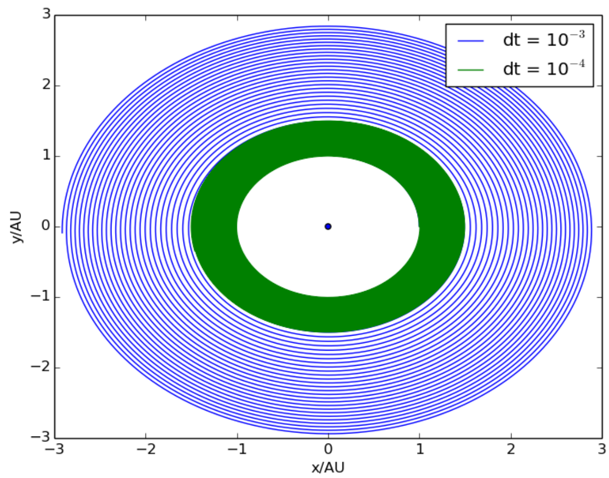

Figure 1: Trajectory plot utilizing the Forward Euler algorithm for the sun-earth system with time-steps, dt, 0.001 and 0.0001 years. The sun position is fixed, and the simulation ran for 100 years.

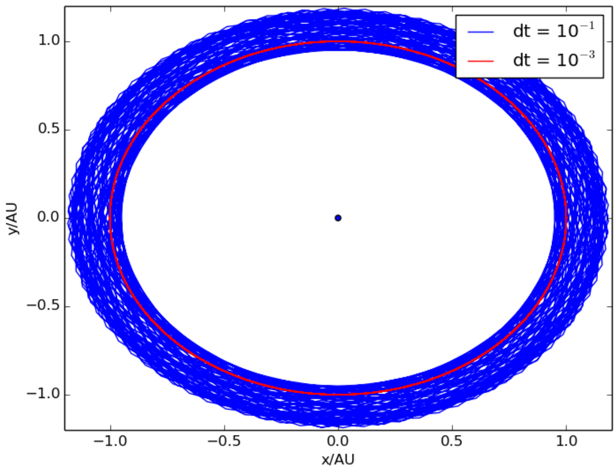

Figure 2: Trajectory plot utilizing the velocity Verlet algorithm for the sun-earth system with time-steps, dt, 0.1 and 0.001 years. The sun position is fixed and the simulation ran for 100 years.

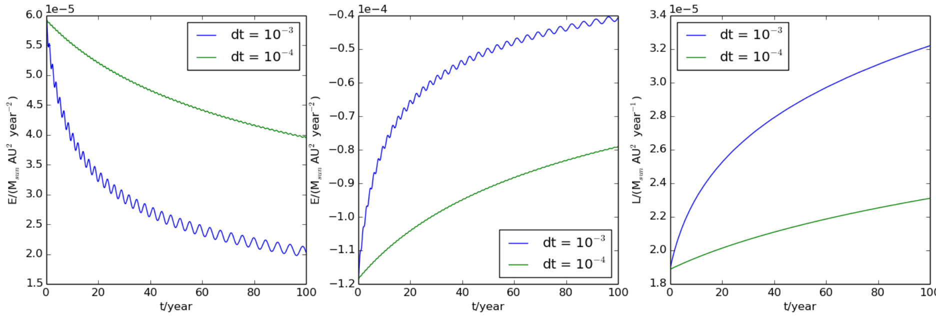

Figure 3: Plot of kinetic energy (left), potential energy (middle) and angular momentum (right), using the Forward Euler algorithm for the sun-earth system with time-steps, dt, 0.1 and 0.001 years. The sun position is fixed and the simulation ran for 100 years.

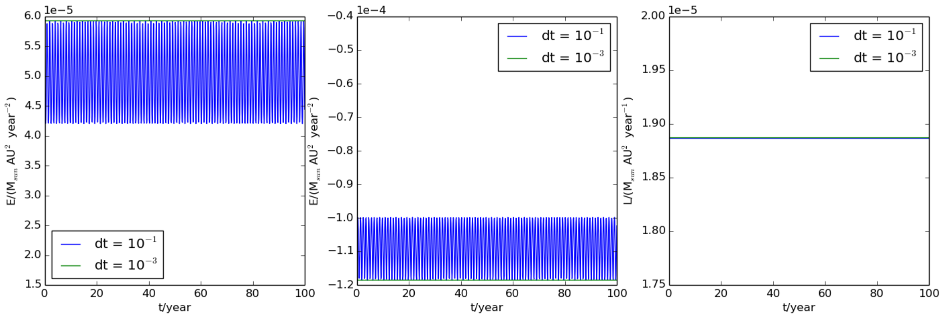

Figure 4: Plot of kinetic energy (left), potential energy (middle) and angular momentum (right), using the velocity Verlet algorithm for the sun-earth system with time-steps, dt, 0.1 and 0.001 years. The sun position is fixed and the simulation ran for 100 years.

4.1.2 Escape velocity

The escape velocity is a velocity in which, the kinetic energy is larger than the absolute value of the potential energy. Thus, to study the escape velocity, it is required to choose an algorithm that conserves total energy such as Velocity Verlet. According to eq. (20) and value of G from eq. (26), it is possible to calculate escape velocity. It is important to note that, for calculating escape velocity, the distance between Earth and Sun is assumed 1 AU. Figure 5 illustrates a plot of orbit in which potential energy is slightly larger than kinetic energy (velocity of 8.8 AU/year), furthermore a trajectory in which kinetic energy is slightly larger than potential energy (velocity of 8.9 AU/year). Evidently, the escape velocity is located in-between 8.8 - 8.9 AU/year. As it is expected, the former when potential energy is slightly higher than kinetic energy the trajectory is a ellipse, and by escape velocity the planet is disconnected.

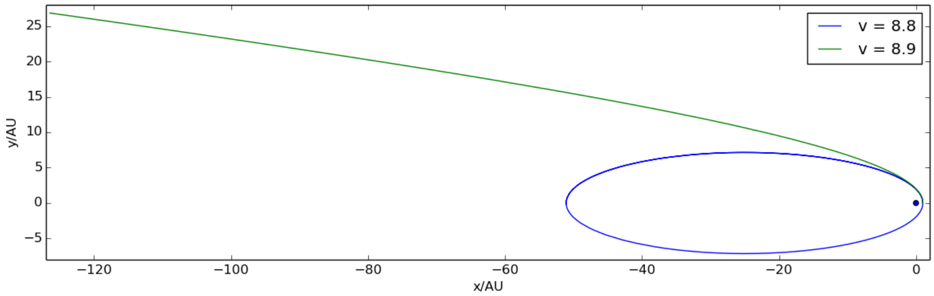

Figure 5: Plots show Earth with and without escape velocity. Blue curve belongs to condition that velocity of Earth is 8.9 (AU/year). Green curve belongs to condition that velocity of Earth is 8.8 (AU/year). Considered time-step is 10^{-3} years.

by plugging the numerical values into eq. (20), the theoretical escape velocity is is given as:

$$v_{esc} = 8.8857658763166842 \left[\frac{\text{AU}}{\text{year}}\right],$$

which well fits to the results of numerical simulations.

4.1.3 Distance-dependency of the gravitational force

It is an well known fact that the Newtons gravitational law is a so called inverse squared law. But what happens if we modify the gravitational force to move from a inverse squared law to a inverse cubic law? The model was modified such that the gravitational force was proportional to $r^{-\beta}$, where $\beta \in [2, 3]$. Initial velocity corresponding to circular motion were used.

As seen in Figure 6, when β is increased from 2 to 2.9, the orbit stays circular. This is because the numerical value of the force does no change since the distance is 1 AU (except for small numerical deviations). So for the scaling used, at this distance the velocity corresponding to a circular orbit does not change when β is increased. However, as also seen in Figure 6, when β is pushed very close to 3, Earth starts to spiral outwards. Putting $\beta = 3$ into eq. 23, an escape velocity for that system is given by

$$v_{esc} = \sqrt{\frac{GM_\odot}{r^2}}.$$

With r = 1 AU, this is exactly equal to the circular orbit velocity. Thus, the solar system model gets unstable as β gets close to 3.

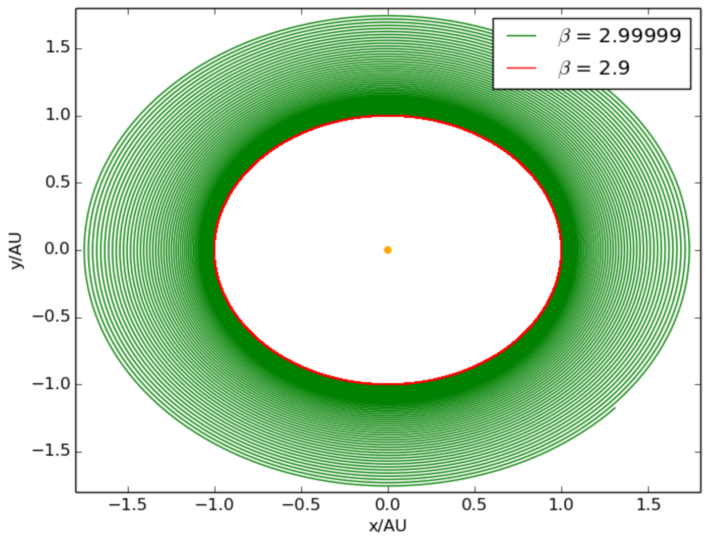

Figure 6: Trajectory plot of the sun-earth system where the initial sun-earth distance is fixed at 1 AU, and the initial velocity is that of a circular orbit. The distance dependence of Newtons law of gravity is generalized to being proportional to $r^{-\beta}$. Time step is 10^{-3} years

4.2 Three body system

Figure 7 shows the orbits of the Earth and Jupiter around a fixed sun. It also shows the final position of earth with and without Jupiter and time steps 10^{-3} were used. The simulations were done ignoring the z component of position and velocity. A plot of a similar simulation with the mass of Jupiter increased by a factor of 10 is shown in Figure 8. As can be read from table 1, the presence of Jupiter delays earths position with approximately 3.1° in 100 years. If the Jupiter mass is multiplied by 10, Earths position is delayed by about 30°.

In the simulations shown in Figure 9, the Jupiter mass is increased by a factor of 1000, making it almost as massive as the sun. Two simulations were done, one with dt = 10^{-3} years and the other with dt = 10^{-4} years. As seen to the left in Figure 9, with dt = 10^{-3} years, earth escapes from the Sun-Jupiter system after what seems to be a very close impact with

Jupiter. Earth also escapes in the simulation with dt $= 10^{-4}$ years, but this time after a close impact with The Sun. The two orbits with different time steps are quite different, which suggest that at least the simulation with the higher time step is not reliable. Figure 10 shows that the total energy has an abrupt change during the simulations, which suggest nonphysical behaviour, and that similar simulations on this system is unreliable.

Table 1: The distances and angle difference between Earths position with and without Jupiter present, at the end of a 100 years simulation. Values are given for a simulation using the real Jupiter mass, M_J, and $10M_J$. Angles were computed with the assumption that the orbits were circular and a radius unaffected by Jupiter's presence.

Mass of Jupiter	Linear distance/AU	Angle/degree
M_J	0.0547	3.14
$10M_J$	0.520	29.8

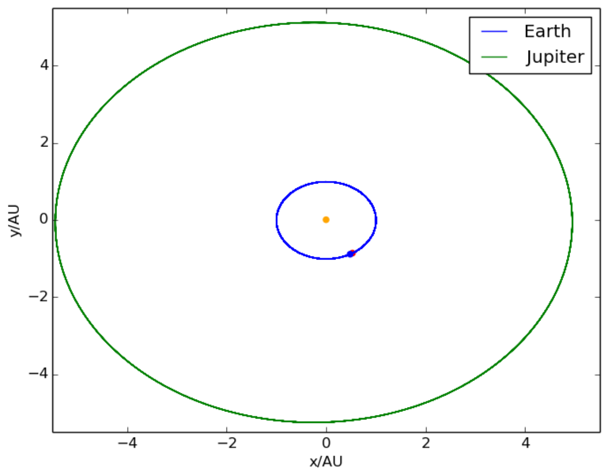

Figure 7: Trajectory plot of the sun-earth-jupiter system with time-step, dt, 0.001 years. The sun position is fixed and the simulation ran for 100 years. The blue dot is the final position of the Earth, while the red dot is the final position of Earth in the same simulation without Jupiter. Algorithm used is the velocity Verlet.

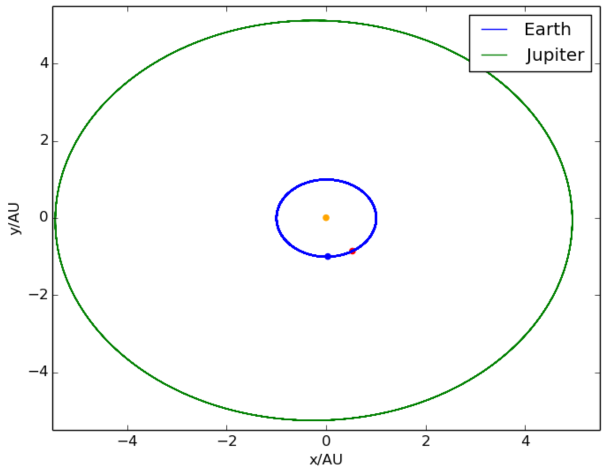

Figure 8: Trajectory plot of the Sun-Earth-Jupiter system where the Jupiter mass is increased with a factor of 10. Time-step, dt, used is 0.001 years. The sun position is fixed and the simulation ran for 100 years. The blue dot is the final position of the Earth, while the red dot is the final position of Earth in the same simulation without Jupiter. Algorithm used is the velocity Verlet.

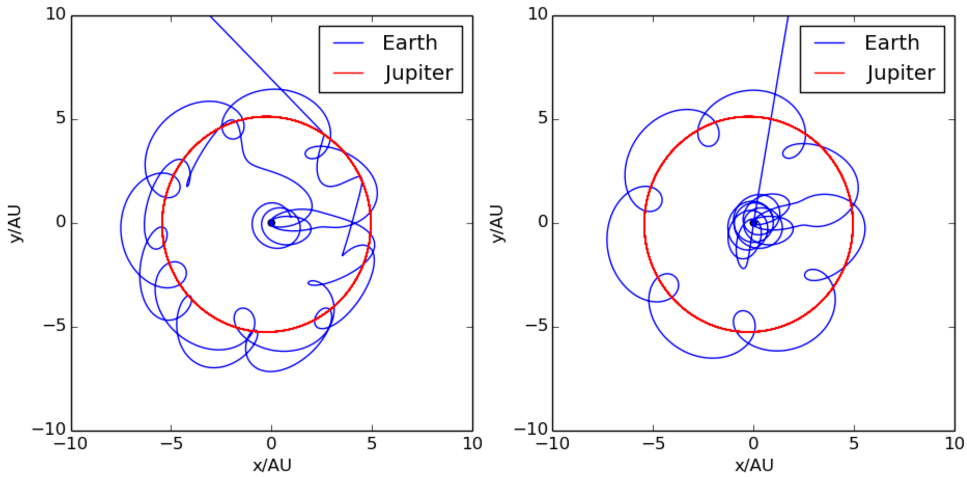

Figure 9: Trajectory plot of the sun-earth-jupiter system where the Jupiter mass is increased with a factor of 10. Time-steps, dt, 0.001 years (left) and 0.0001 years (right). The sun position is fixed and the simulation ran for 100 years. Algorithm used is the velocity Verlet.

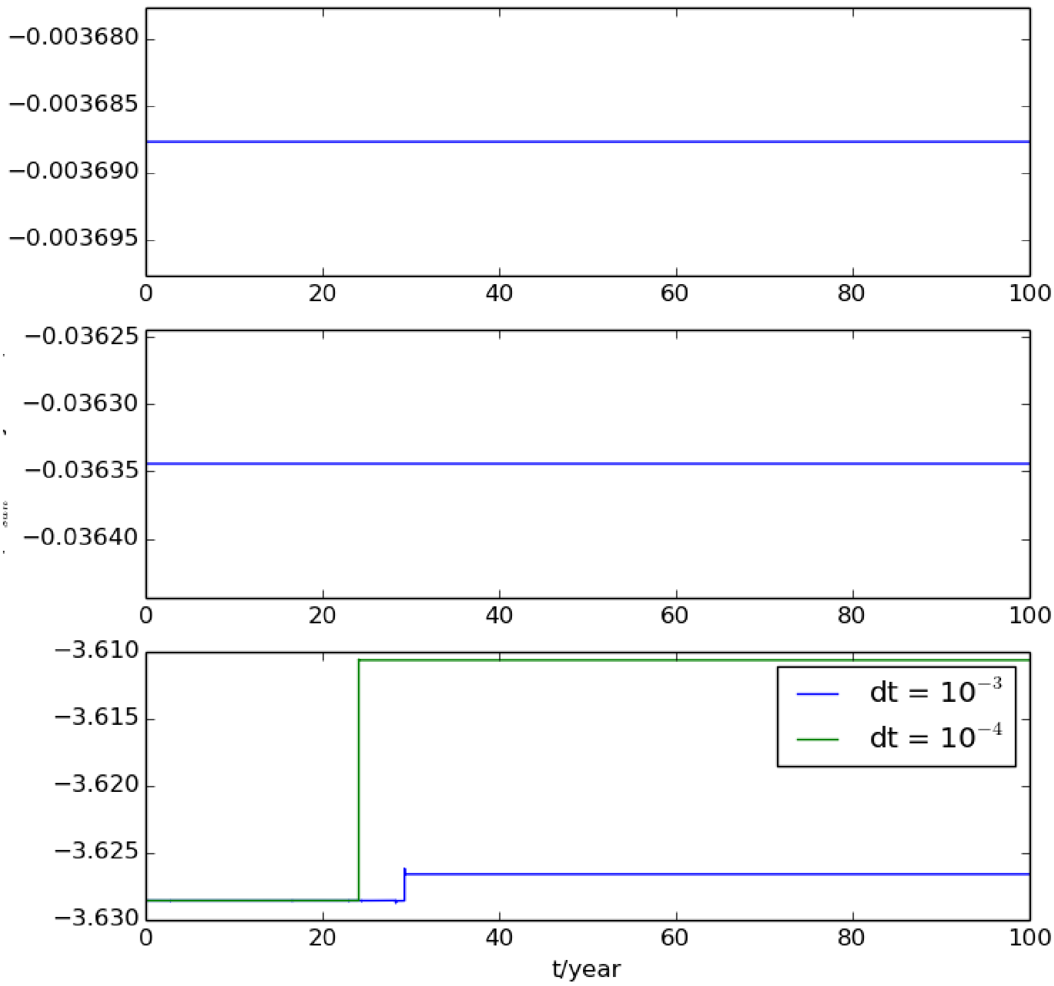

Figure 10: Plot of the total energy of the sun-earth-jupiter system with Jupiter masses M_J (top), $10M_J$ (middle) and $1000M_J$ (bottom), where M_J is the real Jupiter mass.

4.3 The solar system

Figure 11 shows the model of a real solar system including Mercury, Venus Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, Titan, Europa, Luna and Sun. For ease of modeling, object oriented programming technique was applied with simulation time of 500 years and time step of 5×10^{-4}. Initial values were obtained from NASA website.

Furthermore, outcome of program for total energy and angular momentum are plotted in Figure 12 for 500 years and time step of 5×10^{-4}. As it is observed, both plots keep parallel to horizontal axis which presents stability of the model.

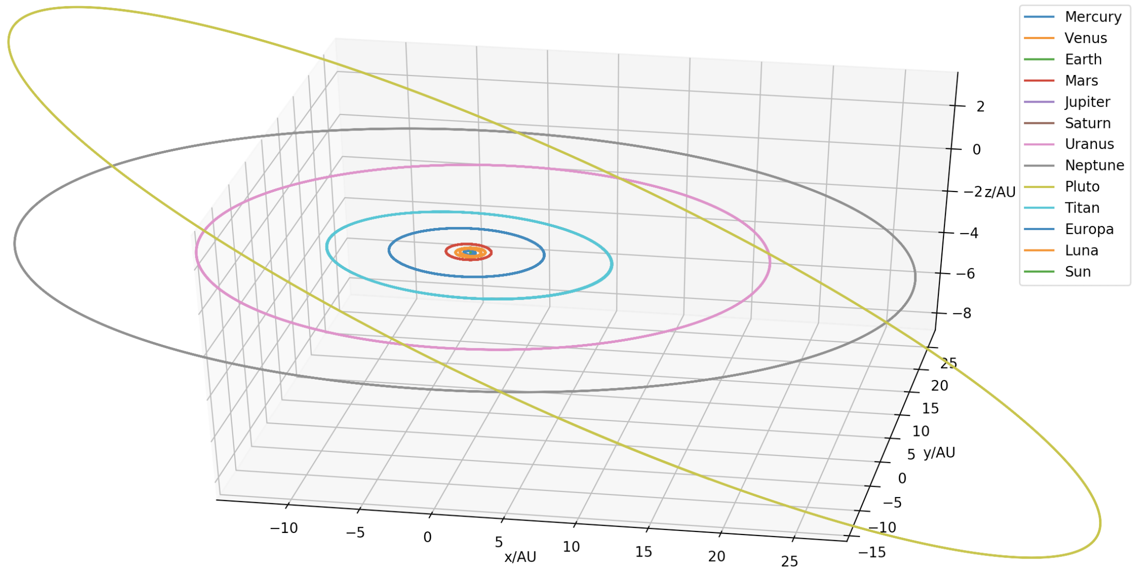

Figure 11: Trajectory plot of the entire solar system with some of the orbital moons. Time step, $dt = 5 \times 10^{-4}$, the simulation time was 500 years.

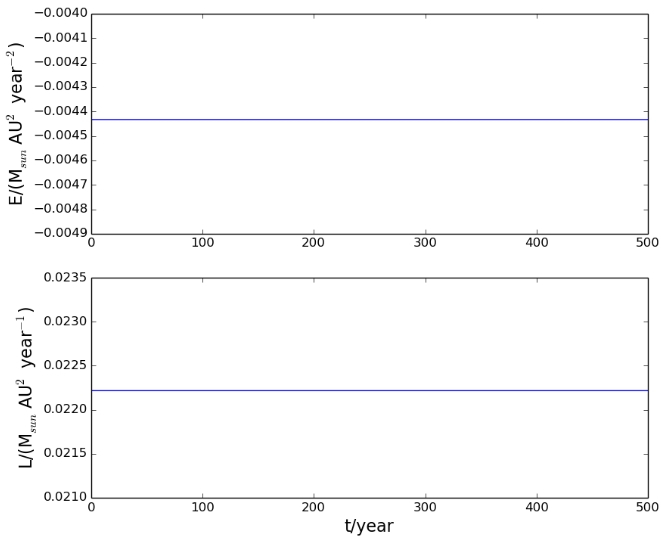

Figure 12: Plot of the total energy (upper plot) and the absolute value of the angular momentum (lower plot) of the solar system. From the plot we can observe that both the angular momentum and energy remain relative constant.

4.4 The perihelion precession of Mercury

A test of general theory of relativity is to compare its precession of Mercury to the observed value. If we only evaluate the Sun - Mercury system (a two body system with no other gravitational forces from other planets), we expect from observations that the value of perihelion precession are 43″ (43 arc seconds) per century. In this section we will use our model to observe this perihelion precession of Mercury.

The sun have much larger mass than the Mercury therefore we will assume that the Sun is stationary. Since the precession is only 43″ (2.085×10^{-4} radians) per century we need a small time step to be able observe the precession. In our simulation we chose the time step to be 10^{-7} and the simulation time of 100 years which results 10^9 iterations. To save computation time only the first and last orbit are stored to file. Figure 13 shows the how the orbit have changed after one century.

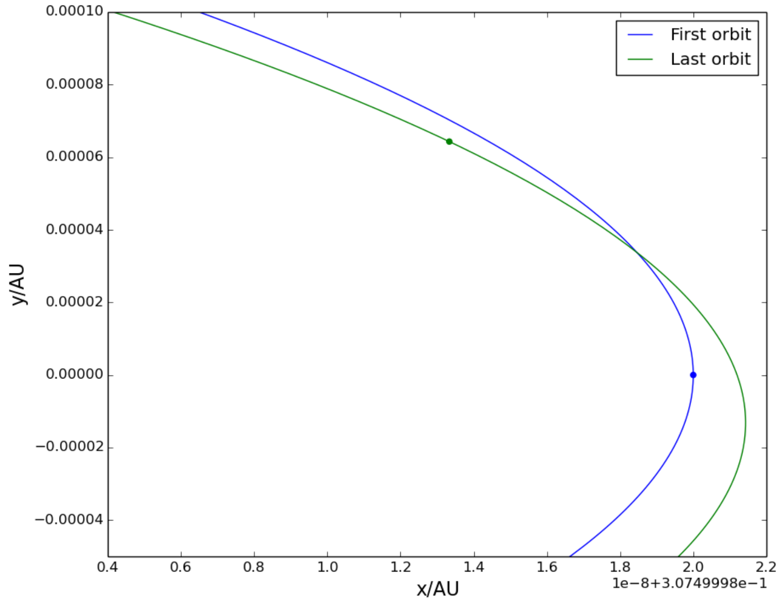

Figure 13: Trajectory plot of the first and last orbit of a 100 years simulation of the sun-mercury system close to the perihelion. The dots in the plots shows the perihelion at the different orbits. Time step, dt, 10^{-7}, were used

Calculation of the angle between perihelion of the first year and last year gives

$$\theta = 43'',$$

which is in correspondence with the experimental value given in the literature.

5 Conclusion

As shown, the velocity Verlet method conserves energy and angular momentum when analyzing The Earth-Sun system, but Euler method was not stable. Only the Verlet method was able to correctly simulate a circular orbit. The velocity Verlet method demands more FLOPS than Euler method, however accuracy of Velocity Verlet is higher than Euler.

By trail and error we found that the escape velocity of the earth from the sun were close to the theoretical value we derived in eq. (20) which further verify that the verlet method is suitable to describe our model of the solar system.

As it was seen for the three body system (Sun(fixed) - Earth - Jupiter), adding Jupiter too the two body system we observed that Jupiter delays the Earths orbit around the Sun. When increasing the mass of Jupiter significantly, the celestial objects deviates from a orbital path way. Observing the energies and angular momentum in these simulations suggests that our model is not stable for a fixed sun, and a large mass of Jupiter.

When using the Verlet method we added all the planets in the solar system to the model. Retrieving the initial conditions of the planets from NASA we obtained stable simulations. Which means that the energy and angular momentum were conserved during our simulation of the solar system.

Finally, by adding the relativistic coefficient to the gravitational force, we observed that Mercury's perihelion angle follows the theoretical expectation value when using the Verlet method.

References

[1] M. Hjorth-Jensen. *Project 3*. Department of Physics, University of Oslo, Norway, 2018. URL: `http://compphysics.github.io/ComputationalPhysics/doc/Projects/2018/Project3/pdf/Project3.pdf` (visited on 10/24/2018).

[2] NASA. *HORIZONS*. Jet Propulsion Laboratory, California Institute of Technology. URL: `https://ssd.jpl.nasa.gov/horizons.cgi#top` (visited on 10/10/2018).