FYS3150/4150 - Project 3
Simulating solar system using numerical methods for solving ordinary differential equations

Aram Salihi[1], Adam Niewelgowski[1]
[1]Department of Physics, University of Oslo, N-0316 Oslo, Norway

October 25, 2018

Abstract

In this project we have solved ordinary differential equations using two different algorithms to solve the N-body problem - our solar system using real data retrieved from NASA. We have chosen to compare Forward Euler (FE) and Velocity Verlet (VV) algorithms by calculating different quantities like conservation of energy and angular momentum. In addition we have checked the stability of earths orbit when using these two algorithms. The result showed us that the Velocity Verlet is superior when it comes to presicision compared with Forward Euler. We also found out that Velocity Verlet did conserve fundemental quantities such as energy, linear and angular momentum which Forward Euler failed to do. In addition we also found out that Velocity Verlet produced good result for $N = 10^5$ gridpoints. Unfortunately we did not manage to reproduce the predicted perihelion precission, 43", the possible sources for error of this calculation is disccused in this article.

Contents

1 Introduction

In physics, the N-body problem occours almost everywhere, and a analytical solution is highly desired. Due to its complexity, most of the time an analytical solution is unattainable, therefore numerically methods are required. We will be investigating a planetary N-body initial value problem with Newton's law of gravity. In this project we are using two different numerical methods to solve the Earth-Sun problem and the entire solar system using real data from NASA. The main goal is to compare stability, precision and conservation of different quantities, such as energy and angular momenta. We have chosen to compare Velocity Verlet (VV) and Forward Euler's (FE). In this article we will first present a theoretical background to all the equations we wish to solve, along with an introduction to the numerical algorithms. The results will be presented and discussed. We will also be looking at the perihelion precession predicted by General Relativity by using a modified Newton's law of gravity (relativistic force).

2 Theoretical Model

In this section we will present the mathematical equations needed and algorithms in order to simulate our planetary system. We will also discuss the choice of physical units used in our calculations.

2.1 Newton's law of gravity

Newton's universal law of gravitaty states that every point particle with mass M, attracts other point masses with a gravitational force, $\mathbf{F}_G$, given by

$$\mathbf{F}_G = -G\frac{m_1 m_2}{r_{12}^2}\frac{\mathbf{r}}{r_{12}} \tag{1}$$

Where G is the universal gravitational constant, m_i are the masses of the point particles, r is the relative distance between the point particles, and $\mathbf{r}$ is the vector pointing from m_2 to particle with mass m_1. Using Newton's law of gravity in Newton's second law of motion, we achieve a second-order differential equation on the form:

$$m_1\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} = \mathbf{F}_G \tag{2}$$

For the sake of simplicity this equation can be reduced to a set of first order differential equations. By introducing a velocity vector, $\mathbf{v}$, this equation can be rewritten to:

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v}(t) \quad \wedge \quad \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{\mathbf{F}_G}{m_1} \tag{3}$$

This law can also be used for a set of N interacting point particles, Newton's law of gravity can be written as:

$$\mathbf{F}_G = -\sum_{i<j}\sum_{j} G\frac{m_i m_j}{r_{ij}^2}\frac{\mathbf{r}}{r_{ij}} \tag{4}$$

Where the sum iterates through all the bodies in the system with $i \neq j$.

2.1.1 Initial conditions

This N-body problem is a initial value problem, this means that we need to apply initial conditions in order to find a unique solution. we will be using two different set of initial values: We will first use $\mathbf{r} = (1, 0, 0)$ and $\mathbf{v} = (0, 2\pi, 0)$ (units in AU and AU/yr) [1] for the sun-earth system, this will give a circular orbit. This conditions are satisfied by:

$$v_0 = \sqrt{\frac{GM_\odot}{r}} \tag{5}$$

Where $M_\odot$ is the mass of the sun and r is the distance between the sun and earth. Afterwards we will use real initial conditions gathered from NASA for solving our solar system. Since the sun is the heaviest object in the system it will contribute most to the center of mass, but due to the forces applied on the sun, the sun will accelerate and start to move away. This means that the center of mass will also follow the same path as the sun. In order to ensure that this does not happen, we must consider that the total linear momentum is zero at all time:

$$\sum_i \mathbf{P}_i = \sum_i m_i \mathbf{v}_i = 0 \tag{6}$$

Defining the first index to be the sun, we have:

$$M_\odot \mathbf{v}_\odot + \sum_{i=2} m_i \mathbf{v}_i = 0 \tag{7}$$

Solving with respect to the suns velocity,we have:

$$\mathbf{v}_\odot = -\sum_{i=2} \frac{m_i \mathbf{v}_i}{M_\odot} \tag{8}$$

This ensures us that center of mass is always at the origin. When plotting the system and zooming at the sun we will notice a flower pattern, this is a consequence of this result. This means that the suns orbit around a center of mass centered in origin.

2.2 Discretization

To be able to solve given set of differential equation numerically, we are required to rewrite our equations and variables to discretize form

$$\mathbf{r}_i = \mathbf{r}(t_i), \qquad \mathbf{v}_i = \mathbf{v}(t_i), \qquad \mathbf{a}_i = \mathbf{a}(r_i) \tag{9}$$

Where t_i is defined as

$$t_i = t_{start} + ih \qquad \text{where } i \in [0, N] \tag{10}$$

For a given time ,t_{start}, we want to end the simulation for a time t_{final} for a given points N. The timestep is in our simulation is then derived as:

$$h = \frac{t_{end} - t_{start}}{N} \tag{11}$$

2.3 Forward Euler

In Forward Euler algorithm we want to find an expression which describes the updated position. More precisely, $\mathbf{r_{i+1}}$, in terms of previous position, $\mathbf{r}_i$. From Taylors expansion theorem we have

$$\mathbf{r}_{i+1} = \mathbf{r}(t_i + h) = \mathbf{r}(t_i) + h\left(\mathbf{r}'(t_i) + \ldots + \mathbf{r}^p(t_i)\frac{h^{p-1}}{p!}\right) + O(h^{p+1}) \tag{12}$$

As we mentioned previously we want to discretize our variables and equation, and we want to define $\mathbf{r}'(t_i) = \mathbf{v}(t_i, r_i)$ to $\mathbf{r}'_i = \mathbf{v}_i$, and we get

$$\mathbf{r}(t_{i+1}) = \mathbf{r}(t_i) + \mathbf{v}(t_i)h + O(h^2) \to \mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_i h + O(h^2) \tag{13}$$

[1] These choice of units will be disccused later.

Notice we also need to update the velocity for each timesteps, to do this consider following expression

$$\mathbf{v}(t_{i+1}) = \mathbf{v}(t_i) + \mathbf{a}(t_i, \mathbf{r}_i)h + O(h^2) \rightarrow \mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i h + O(h^2) \tag{14}$$

Which is also derived from Taylors expansion theorem, where $\mathbf{v}_i' = \mathbf{a}_i$. The Forward Euler algorithm becomes

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_i h \tag{15}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i h \tag{16}$$

$$t_{i+1} = t_i + h \tag{17}$$

Notice that this algorithm does not update the velocity first but the position, because of this the energy will not be conserved. If we switch the order the algorithm, calculating the velocity vector first then the position vector, then the energy is conserved (this is Euler-Cromer, but we will not use this algorithm in this project). Details about this method is found here [Michael, 4 15] and also in [Hjorth-Jensen, 2015].

If we consider only the basic arithmetic operation, such as addition and multiplication when we calculate number of FLOPs, we get 4FLOPs per time step. The main loop is calculating updatefunctions N times, so we get $4N$ FLOPs.

2.4 Verlet alogrithm

The standard Verlet method is mostly interested in position. We will be using the alternative algorithm called Velocity Verlet which is conserving energy. We use the Taylor expansion again

$$\mathbf{r}(t \pm h) = \mathbf{r}(t) \pm h\mathbf{r}'(t) + \frac{h^2}{2}\mathbf{r}''(t) + O(h^3) \tag{18}$$

we now define

$$\mathbf{r}_{i \pm h} = \mathbf{r}(t_i \pm h) \tag{19}$$

$$\mathbf{r}_i'' = \mathbf{r}''(t_i) = \mathbf{a}(t_i, r_i) \tag{20}$$

$$\mathbf{r}_i' = \mathbf{r}'(t_i) = \mathbf{v}(t_i, r_i) \tag{21}$$

$$\tag{22}$$

and get

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h\mathbf{v}_i + \frac{h^2}{2}\mathbf{a}_i + O(h^3) \tag{23}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + h\mathbf{a}_i + \frac{h^2}{2}\mathbf{v}_i'' + O(h^3) \tag{24}$$

since we do not have any expression for $\mathbf{v}_i''$ in terms of already defined variables we will expand $\mathbf{v}_i'$ and keep only the first terms

$$\mathbf{v}_{i+1}' = \mathbf{v}_i' + h\mathbf{v}_i'' + O(h^2) \tag{25}$$

solving for $\mathbf{v}_i''$ gives us

$$\mathbf{v}_i'' = \frac{\mathbf{v}_{i+1}' - \mathbf{v}_i'}{h} = \frac{\mathbf{a}_{i+1} - \mathbf{a}_i}{h} \tag{26}$$

Velocity Verlet algorithm becomes as follows

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \frac{h}{2}(\mathbf{a}_{i+1} + \mathbf{a}_i) + O(h^3) \tag{27}$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h\mathbf{v}_i + \frac{h^2}{2}\mathbf{a}_i + O(h^3) \tag{28}$$

where we need to update $\mathbf{r}_{i+1}$ first, so we can find $\mathbf{a}_{i+1} = \mathbf{a}(t_{i+1}, r_{i+1})$ so we can update $\mathbf{v}_{i+1}$. Comparing this algorithm with Forward Euler, we notice that we update the velocity before the position. Since we are using convserved forces (conserved forces satisifies $F = -\nabla V$ where V is a potential field) to calculate the acceleration. The velocity updated from this will ensure us that we retrieve a correct position vectors we can use throughout the integration, in order to achieve conservation of energy and other quantities, see [Anna Lincoln,] and [Hjorth-Jensen, 2015] for more explaination.

Compared with FE, this algorithm costs a few more FLOPs. We are not taking calculation of a_{i+1} into account, because compared with method above this calculation does not increase the FLOPs number. Velocity Verlet method is using 12FLOPs per each time step. Again, looping over update functions N times, in total gives $12N$FLOPs.

2.5 Units used

Simulating system this large, needs a wise choice of units, which will result in simplifying some of the constants. A natural unit of length for a solar system is an average distance between Earth and Sun, astronomical unit, AU. In SI-units 1AU corresponds to approximately 1.496×10^{11}m. Since Sun is the most massive object in this system, we scale every other mass object according to this value, $\mathrm{M}_{\odot} \approx 1.9891 \times 10^{30}$kg. Time plays essential role in this model, we will be using one Julian year, an astronomical unit which is defined to be 365.25days = 86400s.

With units described above the universal gravitational constant becomes

$$G = 4\pi^2 \mathrm{AU}^3 \mathrm{yr}^{-2} \mathrm{M}_{\odot}^{-1} \tag{29}$$

and the speed of light, needed in relativistic part of our model, becomes

$$c = 63239.73\ \mathrm{AU/yr} \tag{30}$$

2.6 Quantities check

To make sure that our system is modelled correctly we will check if some of the main properties are conserved and if the system follows other well known quantities we can calculate analytically.

2.6.1 Conservation

We will check if the total mechanical energy and the angular momentum is conserved, this means we check if these are constant throughout the simulation.

Energy Since we do not have any external forces acting on our system, the total energy, which consists of potential and kinetic energy should be conserved.

The kinetic energy is given by

$$E_{kin,sys} = \sum_i \frac{1}{2} m_i |\mathbf{v}_i|^2 \tag{31}$$

Potential energy, for the whole system is given by

$$U_{sys} = -\sum_{i<j} \sum_j \frac{G m_i m_j}{|\mathbf{r}_{ij}|} \tag{32}$$

Where $|\mathbf{r}_{ij}|$ is the relative distance of two bodies, and $|\mathbf{v}|$ is lenght of a velocity vector. Sums run through all the objects in our model. Second loop in potential energy runs only for $i < j$ to avoid doubling of energy. Total energy is a simply sum of these two

$$E_{tot} = \sum_i \frac{1}{2} m_i |\mathbf{v}_i|^2 + \left(-\sum_{i<j} \sum_j \frac{G m_i m_j}{|\mathbf{r}_{ij}|} \right) \tag{33}$$

Angular momentum Again, no external force acting on our system should result in preserved angular momentum for our system. Angular momentum for entire solar system is given by

$$\mathbf{l}_{tot} = \sum_i \mathbf{r}_i \times \mathbf{v}_i \tag{34}$$

where we sum angular moemntum for every planet. Here, $\mathbf{r}_i$ is a position vector for a given planet in relationship to the center of mass for the system, and $\mathbf{v}_i$ is its velocity vector.

2.6.2 Escape velocity

An additional check of behavior of our model is to inspect if analytical escape velocity matches the nummerical result when increasing the initial velocity of the planet. Calculating escape velocity for n-body system, with $n > 2$ is tedious and demanding, so we will apply this check only on Sun-Earth system. The analytical expression for escape velocity is

$$|\mathbf{v}_{escape}| = \sqrt{\frac{2G\mathrm{M}_\odot}{r}} \tag{35}$$

With our units and initial position described earlier, (choosing $\mathbf{r} = (1, 0, 0)$ with $M_\odot = 1$) we have

$$|\mathbf{v}_{excape}| = \sqrt{8\pi^2} = 2\sqrt{2}\pi\mathrm{AU/yr} \approx 8.89\mathrm{AU/yr} \tag{36}$$

2.6.3 Adding β factor to gravitational force

We want to investigate how a slightly changed gravitational force from equation (1) will act on our model. We introduce β factor as follows

$$\mathbf{F}_G = -G\frac{m_1 m_2}{r_{12}^\beta}\frac{\mathbf{r}}{r_{12}} \tag{37}$$

where $\beta \in [2, 3]$.

2.7 Relativistic/modified Newtons law of gravity

When looking at mercury's orbit we notice a small prescision (described in next sub section) for every century. This is due to speed and the warping of space time around mercury is higher compared with other objects in our solarsystem. We must then use General Relativity to describe this oribt, since its a accelerating system. From General Relativity or peturbate Newtons law of gravity, one can show that the force applied on mercury are mathematically derived as:

$$\mathbf{F} = -G\frac{m_{sun}m_{mercury}}{r^2}\left[1 + \left(\frac{3l^2}{r^2c^2}\right)\right]\frac{\mathbf{r}}{r} \tag{38}$$

where $l = |\mathbf{l}| = |\mathbf{r} \times \mathbf{p}|$ where $\mathbf{l}$ is the angular momentum of mercury, $\mathbf{r}$ is the relative position vector (sun-mercury) and $\mathbf{p}$ is the linear momentum of mercury.

2.7.1 Perihelion precision

The perihelion precision is a phenomena which occours when the perihelion point of the objects orbit changes by some magnitude. For each century the perihelion point of mercury orbit change by $43''$ arcseconds. Since Mercury is the closest planet to the sun, the velocity and its acceleration is significantly large and General Relativity must be used in order to get correct results. In this project we will use the expression in previous sub section to calculate this angle, when doing this we will neglect all other planets and only look at mercury sun system.

3 Method

3.1 Model

In this section we are going to present four types of solar models. We will first present a simplified model which consist of the sun and earth, then we will add jupiter which is the second heaviest object in the system. The third model we will simulate our entire solar system just by adding the remaining planets.

3.1.1 Sun-Earth system

In this model we will look at the earth orbiting around the sun. We are going to use the algorithms derived in section (2.3) and (2.4) with the initial conditions presented in the begining of section (2.1.1). Keep in mind that we are using Newton's law of gravity to find the acceleration for earth. The system is going to be simulated for 10 years. Further the result produced from these algorithms are presented in the result and compared later in the disccusion section.

3.1.2 Three-body system

As described above we will now add jupiter to the existing system to see how the system changes. We will now use the original mass of jupiter, and then inscrease it by 10 and 1000 to see how this affect Earths orbit (We will do this for fixed and unfixed center of mass). Because of this, we want to use a algorithm which is precise and efficient to avoid nummerical errors. We therefore want to use Velocity Verlet. We will now use NASA initial condition for earth, jupiter and $\mathbf{r}_\odot = \mathbf{v}_\odot = (0, 0, 0)$. This system is going to be simulated for 14 years with 10^5 points.

3.1.3 Entire solar system

We will now add the remaining bodies neglected previous in to our solar system. We will now use the initial conditions described in the end of section (2.1.1). We will also be using Velocity Verlet to produce the data. The system is simulated for 250 years with $N = 10^5$.

3.1.4 Sun-Mercury system

We will now investigate the Mercury Sun problem. In this part we will be using the Newtonian gravitational force and modified Newtonian force (derived from General Relativity). We will be comparing the result produced when using these two different forces. Further we will be also looking at the perihelion prescision. The initial condition used is gathered from NASA. Keep in mind we are still going to use Velocity Verlet and not Forward Euler.

3.2 The structure of the code used

For this project we have used the programming language C++ to produce the result. This is due to the speed. The code consist of five classes. These are described down below:

- body.cpp: The idea behind this class is to make an object for each body initialized in main. These object contains information about the body, such as position, velocity and different form of energies.
- solver.cpp: This class consist of one method called "integrator". This method consist of two main "if tests", instead of making two methods for nummerical algorithm used, we just pass a string argument to the method to choose which one we want to use. This method calls a function called calculate_system() and calculate_potential() which are method in "quantities.ccp".
- store_values.cpp: This class stores the information of each objects and then stores them into a ".txt" file.

- quantities.cpp: This class contains all information to update and calculate different quantities (such as energy, linear moementum, forces, etc..) in the solar system.
- vec3.cpp: This class is a three dimensional vector class. In this class different types of mathematical and programming operators are defined. Our idea and structure of the code is based on the idea behind this class. As one will see, we have not used armadillo library.

For plotting and analyzing the data produced from this code we have used python. The python programmed used are called "plotting.py".

3.3 The presicision of FE and VV

3.3.1 Stability

In order to check the stability and compare these two algorithms, we must check conservation of energy and the difference in the start and end position of the object orbiting. For this we have chosen earth and sun. The main idea is, if the algorithm is stable then the energy is conserved and the difference star and end position is minimal for different Δt. Thus:

$$|\mathbf{r}_{start} - \mathbf{r}_{end}| \leq \epsilon \tag{39}$$

Where $\epsilon = 10^{-5}$. When decreasing Δt the result should converge to a certain value. The difference in (39) is then plotted against Δt.

3.4 Perihelion angle

In order to calculate the perihelion point we must chose a distance d_i, this distance must be less than d_{i-1} and d_{i+1}. In other words we must find the minimum. If the minimum is found the angle is calculated

$$\theta_p = \arctan\left(\frac{y_p}{y_p}\right) \tag{40}$$

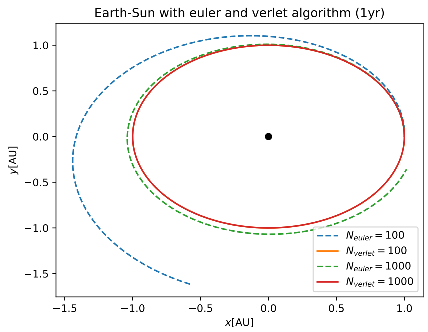
Where x_p and y_p are the perihelion point in x, y direction. The position of mercury and the sun are saved as *mercury.txt* and *sun.txt*, we then use plotter.py to find minimum and then plot the perihelion angle.

4 Results

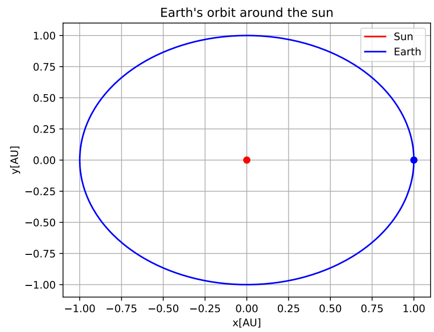
In this section we present all figures produced, we will discuss the result in the next section. In this project we have used three dimentional vectors in our calculations, but for the sake of simplicity we have only included two dimensional figures of the problem. In addition we also show graphs explaining different quantities in the system, such as energy.

4.1 Results for Sun-Earth system

To compare the stability of FE and VV methods, we plot earth's orbit as a function of meshpoints N (in other words Δt) to see how it converge towards a circular orbit.

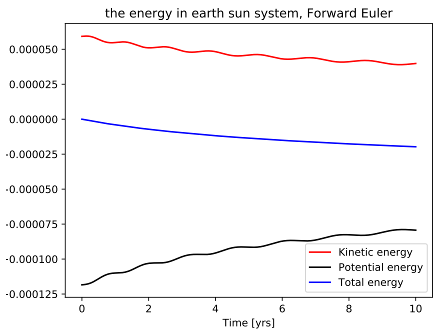


(a) Sun-Earth system plotted with two different algorithms (FE and VV) for two different values of N showed in the figure above. This is done for one earth year.

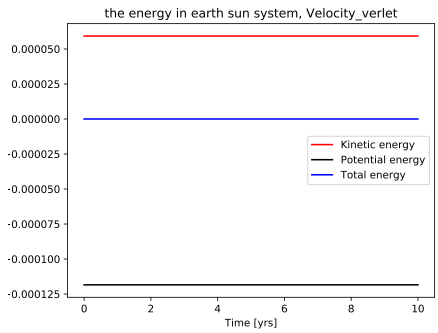


(b) The sun earth system produced by VV with $N = 100$, this is shown to highlight the accuracy of the VV algorithm. The same inital values are used as in the left figure.

The energy produced by FF and VV Another test to check the stability in the system is to look at the total energy, angular and linear momentum. The energy figures produces are presented down below:

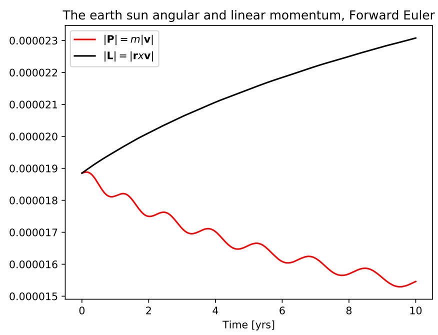


(a) Energy produced by FE method, $N = 10^5$, for 10 years.

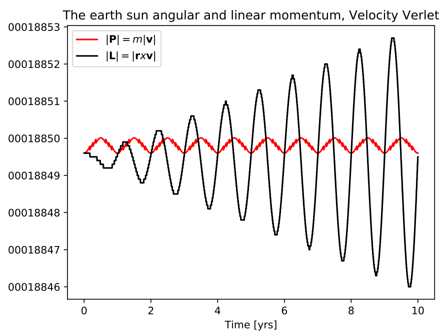


(b) Energy produced by VV method, $N = 10^5$, for 10 years.

Angular and linear momentum The linear and angular momentum figures produced by FE and VV are presented down below:



(a) Angular and linear momentum produced by FE method, $N = 10^5$, for 10 years.



(b) Angular and linear momentum produced by VV method, $N = 10^5$, for 10 years.

4.1.1 Benchmarking: efficiency and precission

The table down below shows:

- computing time for the main loop
- absolute error in position calculated with start and end position of earth

	Time [s]		Relative error in position	
Δt	Forward Euler	Velocity Verlet	Forward Euler	Velocity Verlet
1×10^{-1}	9.00×10^{-5}	1.11×10^{-4}	4.51×10^{1}	1.85×10^{-1}
1×10^{-2}	5.05×10^{-4}	1.07×10^{-3}	1.43×10^{0}	1.53×10^{-5}
1×10^{-3}	7.91×10^{-3}	8.93×10^{-3}	4.92×10^{-1}	6.78×10^{-8}
1×10^{-4}	5.24×10^{-2}	9.71×10^{-2}	7.32×10^{-2}	8.73×10^{-8}
1×10^{-5}	5.33×10^{-1}	7.86×10^{-1}	7.80×10^{-3}	8.88×10^{-8}
1×10^{-6}	5.17×10^{0}	7.73×10^{0}	7.82×10^{-4}	8.88×10^{-8}
1×10^{-7}	5.80×10^{1}	8.84×10^{1}	7.83×10^{-5}	8.88×10^{-8}

Table 1: Efficiency and precission as a function of Δt for both FE and VV methods

4.1.2 Escape velocity

Earth's motion around the sun with different initial velocities to investigate the behaviour of the orbit after 14 earth years.

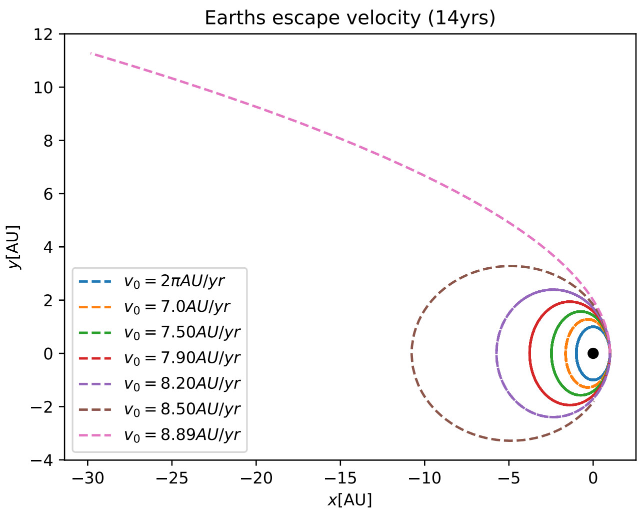
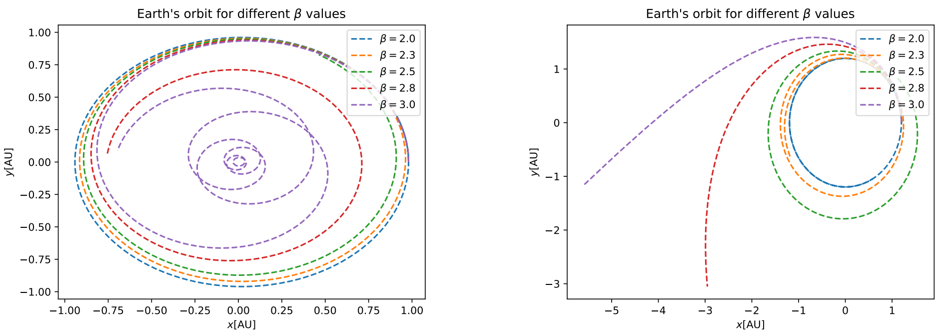


Figure 4: Sun-earth system with different initial velocity for earth for the period 14 earth years

4.1.3 Introducing varying β factor in Newtons law of gravity



(a) The initial position and velocity used is $\mathbf{r} = (0.98, 0, 0)$ and $\mathbf{v} = (0, 2\pi, 0)$

(b) The initial position and velocity used is $\mathbf{r} = (1.1, 0, 0)$ and $\mathbf{v} = (0, 5.98, 0)$

Figure 5: Sun-Earth system with chaning exponent in Newtons law of gravity. As we in the first figure, earth will collapse into to the sun with increasing β and thrown away in the second figure.

Keeping the initial position as $(1, 0, 0)$ for earth, such that distance to the sun is exactly 1AU from the sun, together with $\mathbf{v} = (0, 2\pi, 0)$ which gives circular orbit, we will observe same orbit as for different β values

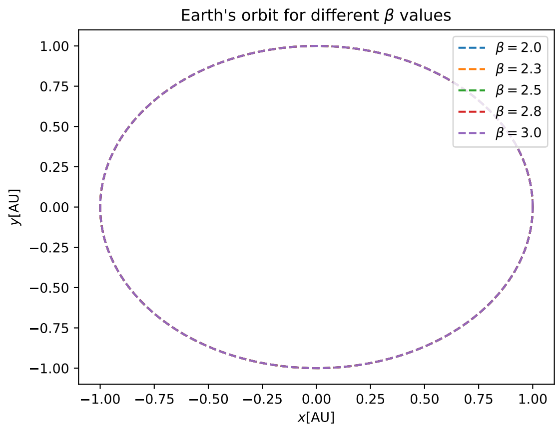
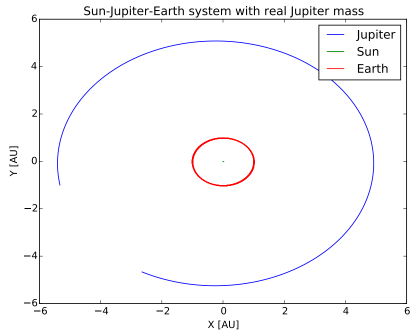


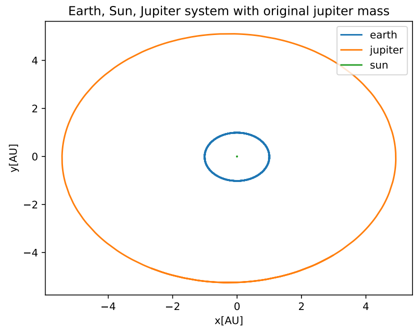
Figure 6: The same motion for different β.

4.2 Results for Sun-Jupiter-Earth system, with varying mass of Jupiter

Adding Jupiter to our system. All simulations are runned with $N = 10^5$ gridpoints.



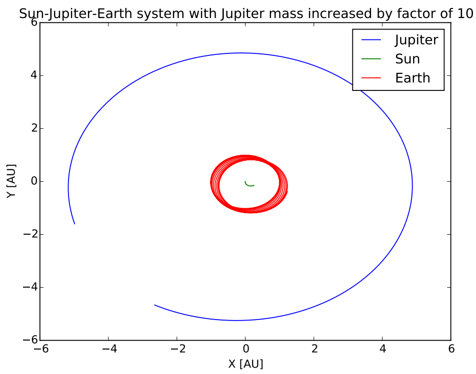
(a) Sun-Jupiter-Earth system with original mass of Jupiter. Center of mass is not fixed at origin.



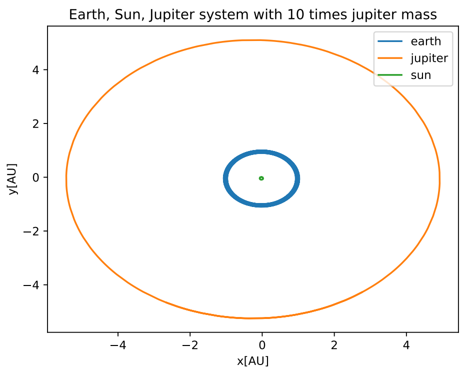
(b) Sun-Jupiter-Earth system with original mass of jupiter. Center of mass is fixed at origin.

Figure 7

We will now increase the mass of jupiter by 10 and 1000 to analyze and observe the systems behaviour.

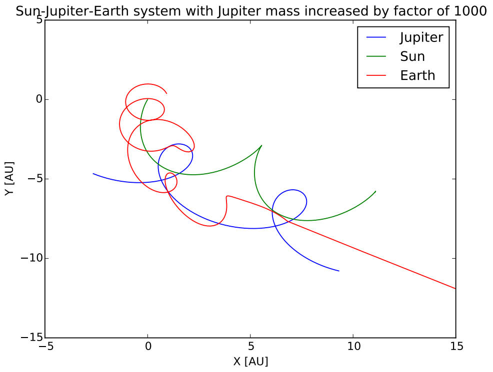


(a) Jupiter mass x10, center of mass is not fixed at origin.

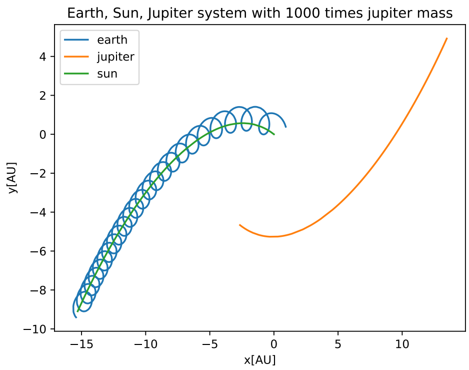


(b) Jupiter mass x10, center of mass is fixed at origin.

Figure 8



(a) Jupiter mass x1000, center of mass is not fixed at origin.

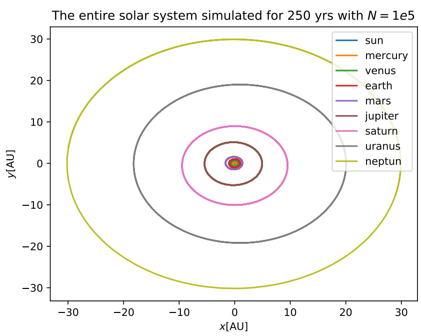


(b) Jupiter mass x1000, center of mass is fixed at origin.

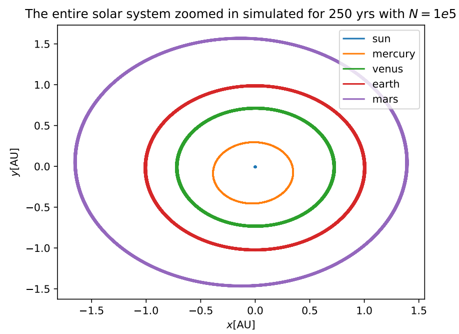
Figure 9

4.3 Results for entire solar system

We will now introduce figures of the entire solar system (excluding Pluto) with fixed center of mass, together with a zoomed version of the inner planets (up to Saturn)



(a) Entire solar system, excluding Pluto



(b) Zommed version of entire solar system, inner planets up to Saturn

Figure 10: Entire solar system (and its zoomed version). Plotted over 250 years with 10^5 mesh-points

4.3.1 Energy and angular momentum

Down below we have presented energy diagrams and diagram over the total angular momentum in the entire solar system (this is runned for $N = 10^5$ for 250 years)

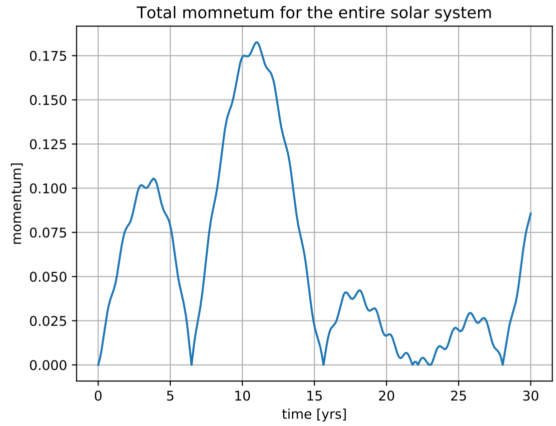
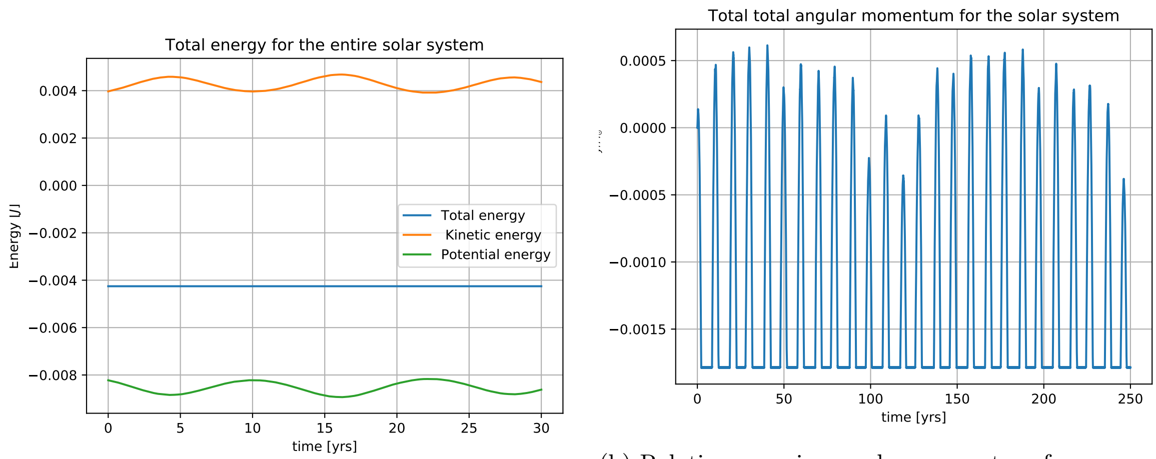


Figure 12: Total momentum in the entire solar system.



(a) Entire solar system's energy plot over 30 years

(b) Relative error in angular momentum for entire system plotted against the time over 30 years (wrong title)

Figure 11

4.4 Results for Sun-Mercury system

5 Discussion

In this section we will discuss the results presented in the previous section.

5.1 Sun-earth system

5.1.1 Comparing algorithms

As we can see in section ((2.3) and (2.4)), the error generated in Velocity Verlet algorithm goes as $O(h^3)$ and in Forward Euler as $O(h^2)$. This is confirmed in figure (figure 1a). Even with use of relative large timesteps (relative small N), we clearly see that earth's orbit deviates much more when applying FE compared with VV method.

Figure 13: The perihelion angle measured during the course of 100 years with $N = 10^9$ gridpoints.

For futher comparison we have benchmarked the presicision and efficiency as a function of Δt for these two algorithms, this table (1) is found section (4.1.1). Here we have taken the time when decreasing Δt, and looked the absolute error in start and final position of earth around the sun simulated for 1 earth year. As we analyze this table,we will already see for Δt of order 10^{-2}, that there is a relatively large different of these two algorithms. Futhermore, we can see that the absolute error in start and final position produced by VV method convergence to $\epsilon \approx 10^{-8}$ for $\Delta t \approx 10^{-4}$. Comparing this with FE, FE keeps decreasing when decreasing Δt. For $N = 10^7$ the absolute error is $\epsilon_{FE} \approx 7.83 \cdot 10^{-5}$ meanwhile VV is $\epsilon_{VV} \approx 8.88 \times 10^{-8}$. This clearly shows that VV outperforms FE in presicision where both algorithm have approximately the same time consumption.

In physics not only is presicision important, but also conservation of different quantities such as energy, angular and linear momentum. As we analyze the figure (**??**) and (**??**) produced from VV , we oberserve that all the quantities are conserved as expected. "The reason why these quantities are conserved is if the acceleration indeed results from the forces in a conservative mechanical or Hamiltonian system, the energy of the approximation essentially oscillates around the constant energy of the exactly solved system, with a global error bound again of order one for Forward Euler" (REF). Comparing these result with the result produced from FE, consider figure (3a) and (2a), we will observe that the system looses energy and gaining linear and angular momentum over time. This will give a unstable system. Using these oberservation, we have therefore used the algorithm VV with $\delta t = 10^{-5}$ throughout this project.

In the end of section ((2.3) and (2.4)) we have showed that FE is using less floating point operations per time step compared with VV. Thus FE is a bit fastet than VV. Thus the FLOPS calculated agrees with table (1). All of the facts discussed above, justifies our choice of numerical algorithm for the rest of the calculations done.

5.1.2 Escape velocity in sun-earth system

When increasing the initial velocity of earth, the orbit goes from a almost perfect circular orbit to a elliptical orbit. As the oribt becomes more and more elliptic and elongated, the force of gravity will have a weaker pull, this behaviour is quite noticeable in figure (4). When earth reaches its escape velocity, the planet is thrown out of earths gravitational field. Thus our numerical calculations agrees with the theoretical work done in section (2.6.2).

5.1.3 β factor

By introducing a β factor, we want to investigate the behavior of earth orbit in sun's gravitational field. As shown in equation (37), if β varies between $[2, 3]$ we can see different results. This also depend on what initial position and velocity value earth have. As we found out, if the distance is smaller than 1, $|\mathbf{r}| < 1$, we get a scenario where earth circulates towards the sun. This totally make sense, since the gravitational force exerted on earth becomes greater for increasing β. This scenario is showed in figure (5a).

We have also considered the scenario when the distance is greater than 1 AU, $|\mathbf{r}| > 1$. This scenario is quite similar the behaviour we observed, when increasing the initial velocity in figure (4). Thus increasing β leads to a force being decreased, what then leads to a less circular orbit. We will notice that for, $2.5 \leq \beta \leq 3.0$, earth will escape the sun graviational field resulting in earth traveling into deeper space.

When the distance between these two objects is exactly 1 AU, $|\mathbf{r}| = 1$, and the initial velocity is matching the circular orbit velocity (36). We get the same circular orbit for different β values, consider figure (6). These observation shows that Newtons law of gravity must goes as $\beta = 2$ in order to keep circular motion, otherwise we will have a unstable orbital motions.

5.2 Sun-Jupiter-Earth system

Jupiter's original mass In figure (7) we have two scenarios. In figure (a) we have system where the center of mass is not fixed, and figure (b) the center of mass is fixed at origin. Comparing these two scenarios, we will see a more or less pretty stable solar system. Thus our model is pretty is consistent and stable for not a fixed center of mass.

10x Jupiter's original mass When the mass of Jupiter is increased by factor of 10, we can clearly see that the orbit of Earth gets affected more than in previous model. The force from Jupiter becomes more significant, this leads to a more unstable orbit for earth - wobbling orbit. Comparing the case with fixed and unfixed center of mass, we notice that the total system in figure (8(a)) is more unstable. Earth is wobbling towards to the side and the sun is moving in a less circular orbit, this is due to jupiters tidal forces. In figure (8(b)), the system is more stable compared to the previous, we will also notice that the sun is having a circular orbit around the center of mass. This means when sligthly increasing the mass of the jupiter, the approximation of not having the center of mass fixed will affect the solar system more. Thus for more precise result it is better to have fixed center of mass.

1000x Jupiter's original mass Finally, increasing Jupiter's mass by factor of 1000 leads to very unrealistic for both fixed and unfixed center of mass in origin. This is because of jupiters mass is factor 10 less than sun mass. consider figure (9a). As we analyze the figure we see a chaotic three-body system, where earth's orbit seems to be changing from orbiting the Sun first, then shifting to orbit Jupiter and crashing. In the case with center of mass being fixed in the origin (see figure (9b)) we do not have chaotic system as previous. Earth is orbiting around the Sun, while the sun is moving away unrealistically. We noticed that we did not calculate the center of mass correctly here, since the mass of jupiter is a tenth of the mass of the sun, we cannot assume that the sun contributes almost entirely to the center of mass. So the expression in the end of section (2.1.1) is not valid for this case.

5.3 Entire solar system

5.3.1 Orbits of the planets

Figure (10) shows the motion of all the planet in our solar system, notice that we have excluded pluto. This simulation is runned for 250 earths years with, $N = 10^5$, gridpoints. Notice that, even for 250 years with our gridpoints, the solar system remains stable. The zoomed version (10b), shows that the inner planets have some deviations, note the ticker lines. This is due to stronger forces from heavier planets in the system and some minimal numerical error.

5.3.2 Energy and angular momentum

Here we are discussing diagrams generated with $N = 10^5$ over 30 years period.

Energy In figure (11a) we have plotted kinetic, potential and total energy for the entire solar system. As described in section (2.6.1), we expect total energy to be conserved what is the case shown with the horisontal line in the figure (11a).

As we further analyze this figure, we notice that average kinetic energy is two larger than the potential energy (ignoring the negative sign, this is due to the negative sign in newtons law of gravity)

$$\sum_i \langle K_i \rangle = -\sum_i \frac{1}{2} \langle V_i \rangle \tag{41}$$

Where $\langle K_i \rangle$ and $\langle V_i \rangle$ is the average kinetic and potential energy. Notice that the total energy is satisfying Virials theorem, which is a good indication that our solar system is stable and the energy is conserved. This means when we include all planet into our system, the system does not loose or gain energy from the envirorment, thus total energy is conserved.

Angular momentum Figure (11b) shows the angular momentum in the entire solar system. Notice the scale of the axis, further notice that the angular momentum is oscillating between two values (skriv hva verdiene er), we do see some variation (not stable maximum points) at some points in the figure due to some noise in the result. This noise can be numerical error generated and increased for each timesteps, but it is minimal due to previous result. Also keep in mind that the data retrieved from NASA considers all external forces in the galaxy, meanwhile we only look at the internal forces in our system. This can also be possible outcome. Even though, neglecting this small variation, the angular momentum is pretty stable and conserved.

Linear momentum: Since the center of mass is fixed at the origin as a result of setting the total linear momentum equal to zero. We will get following figure (12). Again as we analyze the figure we will oberserve a oscillating behaviour in the momentum in system. Again the momentum tend to oscillate, but again the minimas and maximas varies alot. As mentioned in previous in angular momentum, this is not entirely obvious why we have this behaviour. As discussed, this can be a result from numerical errors in our code or because of NASA data. Since the Kinetic energy is conersved, thus the linear momentum must also be peserved. Another a factor could also be a bug in the program we used. After analyzing this figure, the linear momentum is not conserved (but should be).

5.4 Sun-Mercury system

As we know from other experiments done, the mercurys orbit is changing with $43''$ arcsec per century. Unfortunately, we did not manange to produce this result, the result produced were (inser value). One of the reason why we failed could be due to numerical error/bug in program even though we used $N = 10^9$ for 100 years. Another possible error could be the algorithm we used explained in section (3.4), the possible outcome of this error could be miss implementing the algorithm. Notice we did not calculate the angle in C++, did was done with a python script. We used a saved positions (sampled with per 100 iteration then saved for 10^9 steps), this could also be potential error.

6 Conclusion

In this project we have shown that Velocity Verlet has an significantly better presicision than forward euler. Using this method we have managed to create a very stable solar system, where different quantities such as energy are conserved. This analysis have shown that choice of numerical method is crucial when solving complex systems, in order to achieve stability and keep fundemental quantities conserved. Further notice that we managed to prove Virial theorem with this Velocity Verlet. Even though we observed some odd behaviour in angular and especially in linear momentum, but a simple explaination of this behaviour as explained the section above can be numerical error or data retrieved from Nasa. The reason why this simple explaination make sense is because of the observed quantities for the two body system (sun earth). Since these fundemental quantities is conserved for the two body, it should also conserve for N-body. For the two body system the numerical oberservation such as escape velocity and variation of β parameter satisifies the theoretical background presented in the begining. When it comes to perihelion precission, we did not manage to reproduce the predicted 43" arcsecond result. This could be due to numerical error or a bug in the code.

References

[Anna Lincoln,] Anna Lincoln, Arnero, B. B. B. C. M. https://www.saylor.org/site/wp-content/uploads/2011/06/ma221-6.1.pdf.

[Hjorth-Jensen, 2015] Hjorth-Jensen, M. (Fall 2015). Computational physics fys3150/fys4150, lecture notes.

[Michael, 4 15] Michael, Z. (1998-04-15). http://web.mit.edu/10.001/web/course$_{n}otes/differential_{e}quations_{n}otes/$