Solving ODEs with numerical algorithms

Philip Hoel, Elakkyen Pathmanathan
and Theo Griffin Halvorsen

October 26, 2020

UiO : Universitetet i Oslo

Contents

1 Abstract

In this project, we develop a code that simulates the solar system using two methods; the velocity Verlet method and the Euler forward method, to solve a coupled differential equation. We find out that the Verlet velocity method gives mathematically fewer errors and gives us more acceptable approximations.

2 Introduction

In mathematics, we have some equations called differential equations. These are some of the essential types of equations for understanding the world around us. They can be found almost everywhere, from modeling the dynamics of a chemical solution to the financial market and frequently within physical systems. A specific type of differential equation is called ordinary differential equations (or ODEs for short). An example of ordinary differential equations is Newton's motion equations, which will be used in this project. The project will, through modeling the solar system, be solving ODEs numerically. Newton's equations of motion is an approximation for movement in a gravitational field. All the planets in our solar system revolve around the sun because of the sun's gravitational pull. To describe this motion, we solve the equation for Newton's second law.

We will be using the numerical algorithms called the Euler method (after Leonard Euler) and the Velocity Verlet method (after being rediscovered by Loup Verlet). Our tools for this project will be the programming language of C++. C++ will be used for the heavy numerical computations. Furthermore, Python, which will be used for plotting our results.

The project report will start with some mathematical theory of the systems. We start by focusing only on the earth-sun system before adding more planets. We will then explain the algorithms' implementations before moving on to the project's results and conclusion. There will be a short comment on the software for those interested in the end.

3 Theory

As mentioned in the introduction, we will be modeling the solar system consisting of eight planets. We start with a quick summary of what a differential equation is.

A differential equation is an equation that relates one or more functions and their derivatives. There are many different types, but we will be focusing on ordinary differential equations. These are on the form

$$\dot{y}(x) + q(x)y(x) = f(x) \tag{1}$$

and

$$\ddot{y}(x) + p(x)\dot{y}(x) + q(x)y(x) = f(x) \tag{2}$$

(1) We call a *First Order Ordinary differential Equation* and (2) is a *Second Order Ordinary Differential Equation*. Here $y(x)$ is the unknown function we want to find, the dotted $\ddot{y}(x)$ are the second derivative, and $\dot{y}(x)$ the first derivatives of $y(x)$. The functions $p(x)$ and $q(x)$ are known functions. The variable x is also known, for some interval.
For a more formal introduction to differential equations, see a textbook on the subject [3].

Newton's second law is stated as

$$F = ma \tag{3}$$

where F is the force, m is the mass and a is the acceleration. Or stated with words

"*The force on an object is equal to its mass times its acceleration*" (4)

As it stands now, it is just a standard equation. However, if the acceleration is not constant and depends on a variable t(for time), and we know that acceleration is the second derivative of position, then the equation becomes a second-order ODE.

$$F(t) = m \cdot a(t) = m\ddot{x}(t) \tag{5}$$

or

$$\ddot{x}(t) = \frac{F(t)}{m} \tag{6}$$

3.1 Earth-Sun system

Now that we have the gravitational equation, let us look at the earth-sun system. The equation will look like this for the earth-sun system

$$F_G = \frac{GM_{\odot}M_E}{r^2} \tag{7}$$

where G is the gravitational constant, $M_\odot$ is the solar mass, M_E is the earth's mass, and r is the distance between the sun and the earth. We rewrite the equation using vector notation, and the ODE will look like

$$\frac{d^2\tilde{\mathbf{r}}}{dt^2} = \frac{F_G}{M_E} \tag{8}$$

Here $\tilde{\mathbf{r}}$ is the position vector. We know that $F_G = -\frac{GM_\odot M_E}{r^2}\tilde{\mathbf{r}}$. If we now rewrite again, into two first order ODEs, we get

$$\begin{aligned} \frac{d\tilde{\mathbf{v}}}{dt} &= -\frac{GM_\odot}{r^2}\tilde{\mathbf{r}} \\ \frac{d\tilde{\mathbf{r}}}{dt} &= \tilde{\mathbf{v}} \end{aligned} \tag{9}$$

The distances we deal with in space are considerably vast. We, therefore, use astronomical units, written AU. We will also switch mass for solar mass and seconds for years.

$$\begin{aligned} 1\ AU &= 1.5 \cdot 10^1 1\ m\ (\textit{Distance between earth and sun}) \\ 1\ M_\odot &= 2 \cdot 10^3 0\ kg \\ 1\ yr &= 31557600\ s\ =\ 1\ T\ (\textit{period}) \end{aligned} \tag{10}$$

Now, we assume that the orbit of the earth around the sun is a perfect circular motion. It gives us

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

$$v^2 r = GM_\odot \tag{12}$$

Since we know that

$$\frac{v^2}{r} = \frac{4\pi^2 r^2}{T^2} \tag{13}$$

$$GM_\odot = \frac{4\pi^2 AU^3}{yr^2} \tag{14}$$

and since the solar mass is 1

$$G = \frac{4\pi^2 AU^3}{yr^2} \tag{15}$$

3.2 Adding the rest of the planets

Let us look at an expansion of the solar system by adding more planets. When we add more planets, the complexity increases. With more planets, we must consider more forces, forces that act on all the planets. Nevertheless, the simple principle above still applies. Instead of calculating the forces between two objects, we do it for several. Then we get an equation that takes into account all the forces acting on a given planet, namely

$$F_i = -G\sum_{j=1}^{s} \frac{M_j}{|\mathbf{r_i} - \mathbf{r_j}|^2}(\mathbf{r_i} - \mathbf{r_j}), \qquad for\ j \neq i \tag{16}$$

where s is the number of bodies. Here, the sun is counted as one of the bodies. To describe the equation above intuitively, we can imagine ourselves standing in the middle of a tug-of-war with several ropes pulling in each direction, and we want to find where we end up.

3.3 Conservation of momentum and energy

The gravitational force is a conservative force where an object's total energy is constant and independent of the path taken. The total energy is defined by the sum of kinetic and potential energy. Kinetic energy is defined by the centripetal force, which describes the curved path of the object, and the gravitational force defines the potential energy. The following two equations define the kinetic energy and potential energy

$$E_k = \frac{1}{2}mv^2 \tag{17}$$

$$E_p = \frac{GMm}{r} \tag{18}$$

E_k is the kinetic energy, and E_p is the potential energy. G is the gravitational constant, v the velocity, r the distance between the two objects with respectively mass M and m. Because we have assumed that we have a circular orbit, both the equations above will be zero, and hence the total energy will be zero.

3.4 Escape velocity

The escape velocity is the minimum speed required to escape the gravitational field of an object. By using the principle of conservation of energy, we can find the escape velocity. If an object successfully escapes, the total energy at some point reaches 0. We can deduce what is required to get the escape velocity

$$E_k + E_p = 0 \tag{19}$$

$$\frac{1}{2}mv_e^2 + \frac{GMm}{r} = 0 \tag{20}$$

$$v_e = \sqrt{\frac{2GM}{r}} \tag{21}$$

v_e is the speed needed to escape the gravitational field of a celestial body with mass M and from a distance r. The expected escape velocity for Earth at a distance of 1 AU from the Sun with mass 1 is

$$v_e = \sqrt{\frac{2G1}{1}} = 8.886\text{AU/yr} \tag{22}$$

4 Methods

We have rewritten the second-order differential equation in two first-order differential equations and scaled it with astronomical units. To further solve our coupled differential equation, we need to discretize it. The basic strategy for defining a differentiated method is to evaluate a function at a few points, find the polynomial that interpolates the function at these points, and use the derivative of this polynomial to approximate the derivative of the function. We are looking at two widely used methods, the Euler Forward method and the Verlet method. We start looking at the Euler Forward method. Assume that we have a basic differential equation

$$\dot{y} = f(t, y) \tag{23}$$

and an interval from $[a, b]$, that we discretize into $t_0, t_2, ..., t_n$ points. That gives us approximations $y_0(= c), y_1(t_1), ..., y_n(t_n)$, with step size $h = (t_n - t_0)/n$ and a intial condition c. To construct the numerical method, we use Taylor expansion

$$y(t + h) = y(t) + h\dot{y}(t) + \frac{h^2}{2}\ddot{y}(t) + O(h^3) \tag{24}$$

We can get the forward Euler method by rewriting the Taylor expansion, truncating it to the first derivative, and write it in an iterative notation, namely

$$y(t_{i+1}) = y(t_i) + h\dot{y}(t_i) \tag{25}$$

then replace $\dot{y}$ with f

$$y(t_{i+1}) = y(t_i) + hf(t_i, y_i) \tag{26}$$

We have now derived the forward Euler method [2]. The same yields for our case with the coupled differential equation. We consider our coupled differential equations $\frac{dr}{dt}$ and $\frac{dv}{dt}$, and make a Taylor expansion of v and r truncated to the first derivative. With $r_{i+1} = r(t_{i+1}) = r(t + h)$

$$r_{i+1} = r_i + h\dot{r}_i \tag{27}$$

$$v_{i+1} = v_i + h\dot{v}_i \tag{28}$$

We recognize $\dot{r}$ as the velocity v, and $\dot{v}$ as the acceleration a. The acceleration do we know from via Newton's second law $a = F/m$. We rewrite our equations with the mentioned identities

$$r_{i+1} = r_i + h\dot{v}_i \tag{29}$$

$$v_{i+1} = v_i + h\dot{a}_i \tag{30}$$

With a_i as

$$a_{i+1} = \frac{F_i}{m} \tag{31}$$

Now let us look at the second numerical method we are gone use, the Verlet velocity method [2]. We start just like earlier with a Taylor expansion.

$$r(t+h) = r(t) + h\dot{r}(t) + \frac{h^2}{2}\ddot{r}(t) + O(h^3) \tag{32}$$

$$v(t+h) = v(t) + h\dot{v}(t) + \frac{h^2}{2}\ddot{v} + O(h^3) \tag{33}$$

As mentioned, the second derivative is know from via Newton's second law $\ddot{r}(t) = \dot{v}(t) = a(r,t)$, and the first derivative as the velocity $\dot{r} = v$. However, we now have an expression $\ddot{v}$, which need to define. We define $\ddot{v}_i$ as

$$\ddot{v}(t) = \frac{\dot{v}(t+h) - \dot{v}(t)}{h} \tag{34}$$

If we now insert the previous identities we get

$$r(t+h) = r(t) + hv(t) + \frac{h^2}{2}a(t) + O(h^3) \tag{35}$$

$$v(t+h) = v(t) + ha(t) + \frac{h}{2}(\dot{v}(t+h) - \dot{v}(t+h)) + O(h^3) \tag{36}$$

We rewrite this in a iterative way with $r(t+h) = r_{i+1}$, and $v(t+h) = v_{i+1}$ which gives us

$$r_{i+1} = r_i + hv_i + \frac{h^2}{2}a_i \tag{37}$$

$$v_{i+1} = v_i + ha_i + \frac{h}{2}(\dot{v}(t+h) - \dot{v}(t+h)) \tag{38}$$

with a_i as

$$a_i = \frac{F_i}{m} \tag{39}$$

We have now derived the Verlet velcoity method.

5 Implementation

Github link to code

The full implementation lies on the Github repo: https://github.com/philhoel/Computational-Physics

Most of how things work can be found through the implementations. However, I will give a brief look at the main algorithms with some pseudo code

5.1 Euler method

Algorithm 1: Forward Euler algorithm

for $t = 0;\ t < time;\ t{+}{+}$ **do**
  for $p = 0;\ p < planets;\ p{+}{+}$ **do**
    $\mathrm{a}^p_{t+1} = \frac{f^p_t}{m}$;
    $\mathrm{v}^p_{t+1} = v^p_t + a^p_t * h$;
    $\mathrm{r}^p_{t+1} = r^p_t + v^p_t * h$;
  end
end

It is in place with a comment, as we can see the outer for-loop iterates through time steps, and the inner for-loop iterates through the different planets. If we do not update all the planets before moving on to the next time step, we will get wrong forces, which gives wrong positions. In the entrails, we see the calculation and update of the acceleration, velocity, and position. A remark is the exalted p; it marks the planet we are on and not the power of some variable x.

5.2 Velocity Verlet method

Algorithm 2: Forward Euler algorithm

for $t = 0;\ t < time;\ t{+}{+}$ **do**
  for $p = 0;\ p < planets;\ p{+}{+}$ **do**
    $\mathrm{r}^p_{i+1} = r^p_i + v^p_i * h + \frac{h^2}{2} * a^p_i$;
    $\mathrm{a}^p_{t+1} = \frac{f^p_t}{m}$;
    $\mathrm{v}^p_{t+1} = v^p_t + \frac{a^p_t + a^p_{t+1}}{2} * h$;
  end
end

The velocity Verlet method is relatively similar to the Euler method with the for-loops and the update of the position, acceleration, and velocity. However, the order is different. The velocity, in the Verlet method, is pending on acceleration, which is pending on position. Therefore, we must be careful about which ones we evaluate first, which gives us an order of position, acceleration, and velocity.

Tests

It is essential to test the methods, both to make them safe and for further analysis. To test if our program is accurate enough to use for scientific purposes, we need to test if the laws of physics holds in our calculations as well. What we will test for is energy conservation. To do this, we make a method calculating the potential energy of the earth in the Earth-Sun system. These values should be the same the whole way around. In reality, the earth will move a small amount towards the sun and back again. This will create a wave-like oscillation. When we have calculated the potential energy for a time period, we split the period into two intervals. Then we find the maximum value in the two intervals and check if they are the same, within a certain tolerance. If they are, we have energy conservation.

6 Results

First thing first, we consider a system of sun and earth.

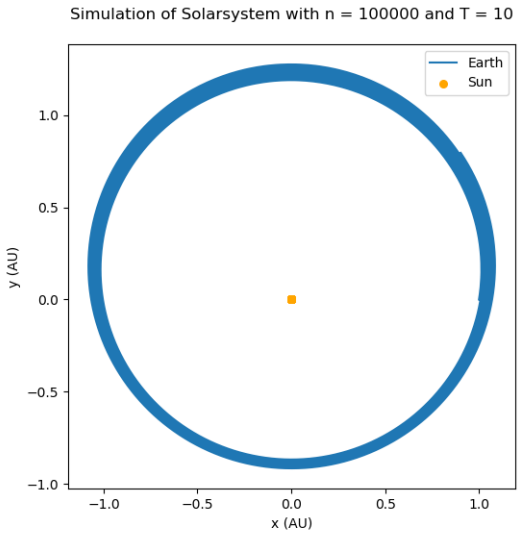

Figure 1: System of sun and earth, evaluated with Euler forward method over a period of 10 years

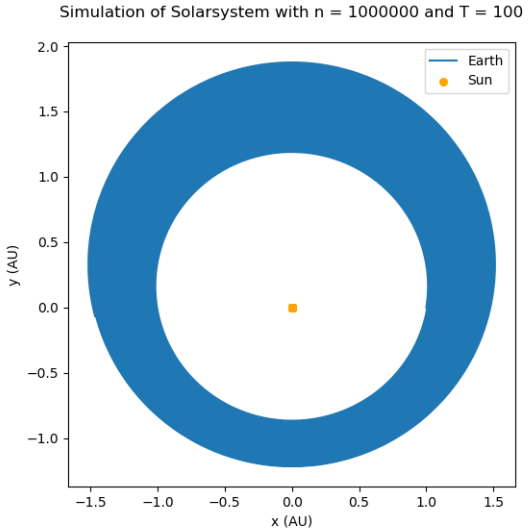

Figure 2: System of sun and earth, evaluated with Euler forward method over a period of 100 years

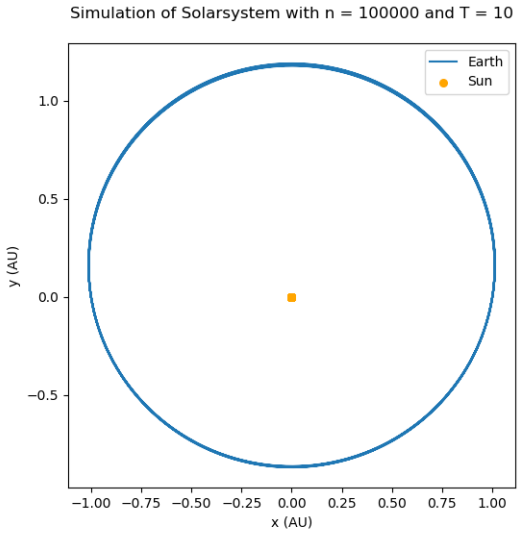

Figure 3: System of sun and earth, evaluated with Verlet velocity method over a period of 10 years

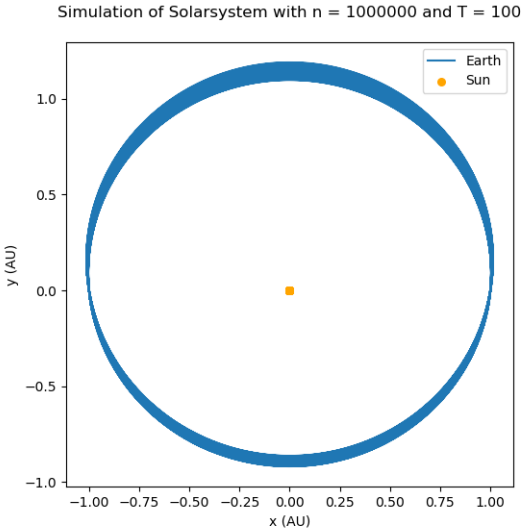

Figure 4: System of sun and earth, evaluated with Verlet velocity method over a period of 100 years

The first four plots plot Earth's orbit around the sun for different time intervals and methods. We can see from the first four plots that our variable time changes the result considerably. The first two plots are produced with the forward Euler method, and we can see that the cursor from the orbit enlarges. The same yields for the two plots below, using the Verlet velocity method. A larger cursor means that a planet slips out of orbit. If we compare the two methods, we see that the orbit is different. We see that the Verlet method produces more refined lines than the forward Euler method. The next three plots plot a system with the Sun, Earth, and Jupiter evaluated with the Verlet velocity method. However, it also displays how different masses of Jupiter affects the system.

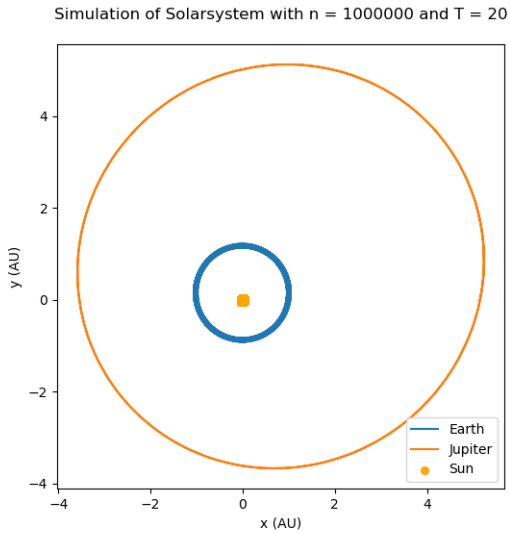

Figure 5: System of Sun, Earth, and Jupiter, evaluated with Verlet velocity method, and with Jupiter mass x10

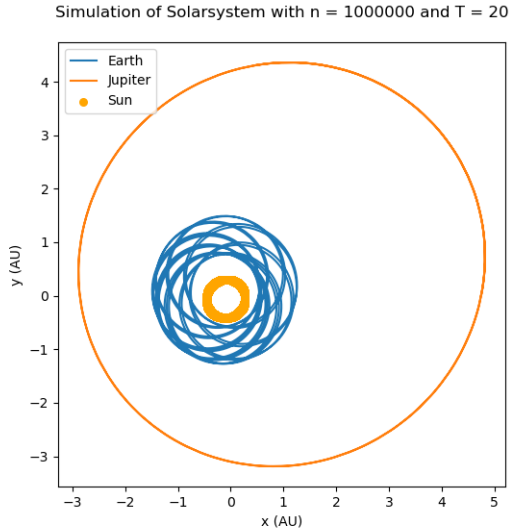

Figure 6: System of Sun, Earth, and Jupiter, evaluated with Verlet velocity method, and with Jupiter mass x100

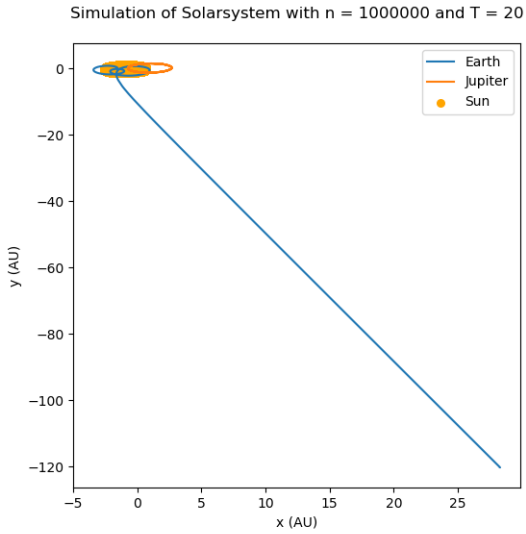

Figure 7: System of Sun, Earth, and Jupiter, evaluated with Verlet velocity method, and with Jupiter mass x1000

We see that both Sun, Earth, and Jupiter are relatively stable with a circle-like orbit for the first plot. However, for the second plot, we see that some odd things happen. The sun starts to orbit, and the Earth makes some beautiful art, an orbit shape of a torus. For the third plot, the only thing we can see is that Earth gets slung out. Subsequent plots show what happens when we change the beta in the following equation changes

$$F_G = \frac{GM_{sun}M_{earth}}{r^{\beta}} \tag{40}$$

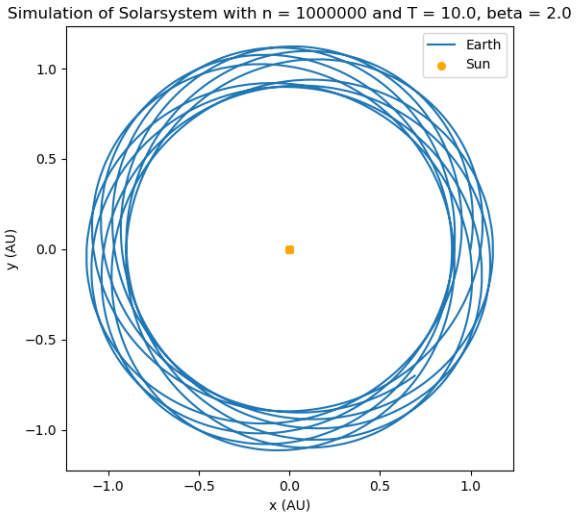

Figure 8: System of Sun, and Earth evaluated with Verlet velocity method, and with beta = 2

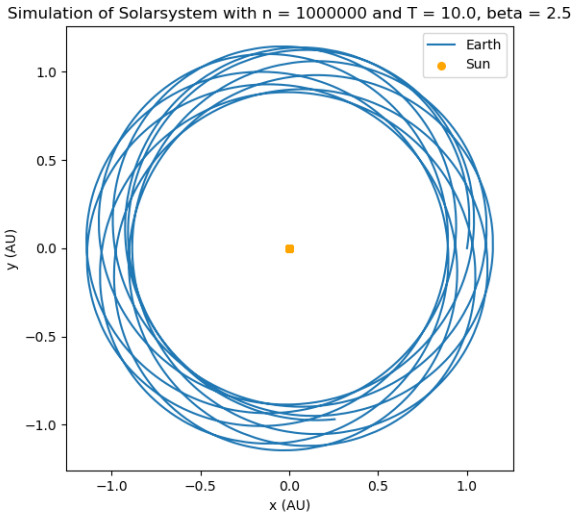

Figure 9: System of Sun, and Earth evaluated with Verlet velocity method, and with beta = 2.5

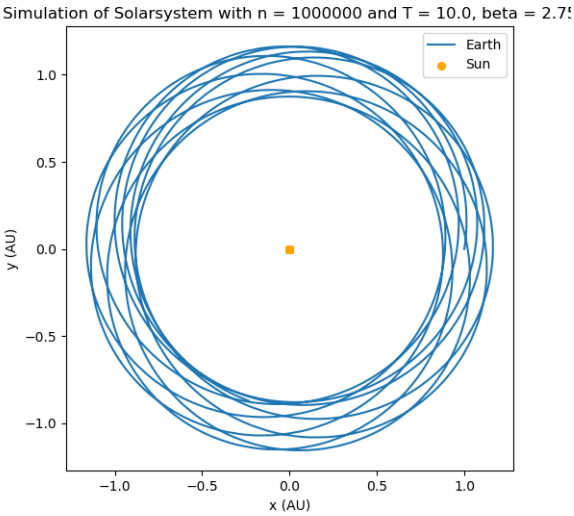

Figure 10: System of Sun, and Earth evaluated with Verlet velocity method, and with beta = 2.75

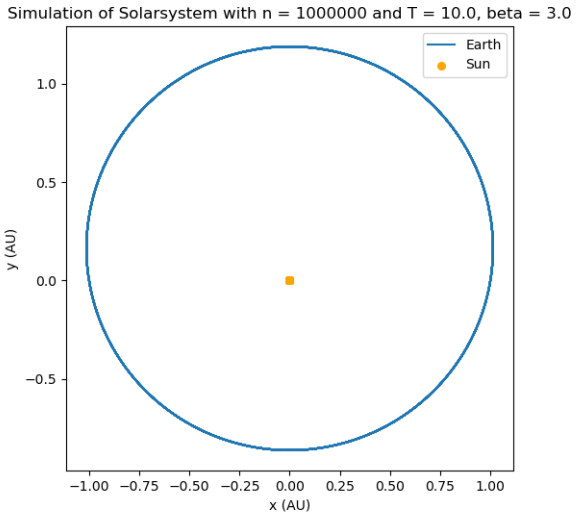

Figure 11: System of Sun, and Earth evaluated with Verlet velocity method, and with beta = 3

From the plots with different beta's we see that the Earth's orbit is changing. It has to do with the fact that we affect the gravitational field. Next plot is a 3D plot of all the planets with real values from NASA [1].

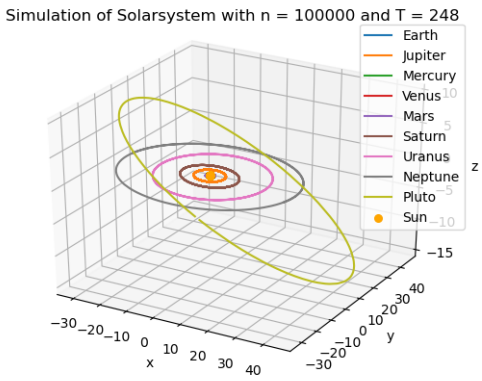

Figure 12: plot of all planets in the solar system, evaluated with Verlet velocity method and Initial values from NASA[1]

7 Discussion

We first consider the difference in results between the Euler and the Verlet method. Since we have assumed that we work with circular orbits, there should be no difference in the path from one period to another. Therefore, we see that the two methods have a clear difference in precision because the plotted orbits' thickness is different. Let us take a closer look at why. One possible explanation may lie in the local errors inherent in the methods. From earlier, when we made a Taylor expansion to derive the different methods, we got an approximate error, where Euler had $O(h^2)$ and Verlet $O(h^3)$, which is called truncation error. We see that the truncation error is proportional to h and therefore tends to zero when h tends to zero [3]. Then we see that the Verlet velocity method's error tends to zero faster than the Euler forward method, giving higher accuracy. It may seem like we can push down the size of h as much as we want, but unfortunately, we can not. We have another error due to round-off, which hinders us. The round-off error is proportional to $1/h$ and therefore becomes larger when h tends to zero. As we can see, our errors are mostly rooted in the time steps. If we look at how h is defined, $h = T/n$, where n is the number of integration points and T is the time period in years. We see that it depends on the number of points and the desired time we want to run. Since it depends on the number of integration points and the number of years, we can see a correlation from previous plots. If we look back at our first four plots, we see that the orbits become thicker for a more extensive T and the same n, which defines the step length.

The plots concerning the different masses of Jupiter can be explained intuitively. As the mass of Jupiter increases, so will the force of attraction. We can see this from the equation.

$$F_{earth-Jupiter} = \frac{GM_{Jupiter}M_{earth}}{r^2_{Earth-Jupiter}} \tag{41}$$

It will then affect the planets around, in this case, the Earth and the sun. In other plots with Jupiter, we see that the Earth is pulled back and forth, and the sun gets a larger orbit. Finally, we see that the Earth is thrown out when we increase Jupiter's mass times 1000.

8 Conclusion

In this project, we have numerically solved the coupled differential equations that rules the solar system and successfully made a program to model it. From the discussion, we see that the choice of method has a lot to say. The velocity Verlet method has an inherently smaller mathematical error, which means that the total error becomes smaller. We also see that the choice of stride length has a lot to say for possible errors. We have also seen how much the mass of a planet can change the trajectory of the other planets.

9 Comment on software

For readability and reusability, the code was written in object-oriented design.

What is object-oriented design? Object-oriented programming (OOP) is about making objects for more comfortable and safer interaction between different code parts. The variables can be stored as class variables, where they will be locked to a specific object. This means the value is taken from within the class, and we do not need to be afraid of accidentally making a copy of our variable.

References

[1] Alan B. Chamberlin. "HORIZONS Web-Interface". In: (October 26 2020), mainpage. URL: https://ssd.jpl.nasa.gov/horizons.cgi#top.

[2] Morten Hjort-Jensen. "Computational physics lectures 2015". In: (August 2015), pp. 240–273. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[3] Knut Mørken. "Numerical Algorithms and Digital Representation". In: (September 2017), pp. 321–354. URL: https://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h17/kompendiet/matinf1100.pdf.