# Solar System model - solving ordinary differential equations

Project 5, FYS4150
Autumn 2019

Jowita Borowska

## ABSTRACT

In the following project we solve the system of ordinary differential equations, coupling the acceleration (given by Newton's second law of motion), velocity and position of celestial bodies in their periodic movement around the Sun. We consider two solvers - the Euler's method and velocity Verlet method. We compare their performance on the example of a simple Sun-Earth system, concluding that the velocity Verlet method is much more accurate (in creating closed orbits predicted by the Newtonian gravitational force) and conserves the constants of motion (total energy and angular momentum) much better than the Euler's method. It is, however, associated with longer CPU time. Thereafter, we determine the escape velocity of Earth numerically to be 8.9 AU/yr, which is in agreement with the exact value, $v_e = 2\pi\sqrt{2} \approx 8.886$. We investigate also the impact of decreasing the gravitational force between the Earth and Sun on Earth's orbit ($\beta$ parameter). Next, we proceed to the Sun-Mercury system, determining its perihelion precession to be $39.6''$ per century, which is close to the observed value when all the classical effects are removed ($43''$). Thereafter, we consider the system with Sun, Earth and Jupiter, altering Jupiter's mass. Distinct distortions of Earth's orbit can be observed as Jupiter's mass increases. Finally, we include all 8 planets, Sun and Pluto in the simulation and plot the movement of these celestial bodies over 150 years.

## 1 Introduction

Observations of the night sky have always been an important factor in learning about our place in the Universe. Throughout the history of the human kind people were trying to understand the movement of celestial bodies around us - the Solar System. From the Ptolemaic model with Earth in the center, through the heliocentric idea of Copernicus and the discoveries made by Galileo, we were getting closer to unravel the truth about the structure and kinematics of the Solar System. Finally, the gravitational law, introduced by Newton, succeeded in not only explaining, but also predicting the motion of the planets. It is determined by the universal gravitational force between each object, proportional to the product of their masses and inversely proportional to the square of the distance between them. Today the problem

comes down to solving a set of differential equations, coupling the acceleration, velocity and position as

$$\frac{\sum \vec{F_G}}{M} = \vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2}, \tag{1}$$

where we have used Newton's second law in the first equality - sum of all gravitational forces acting on a given object divided by its mass gives acceleration.

In the following project, the above differential equations will be solved. First, the system containing only Earth and Sun will be considered, with use of both Euler's method and the velocity Verlet method. We will discuss the accuracy and computational efficiency of both algorithms, in addition to how well they conserve the total energy and angular momentum of the system. Thereafter, we will focus on the notion of escape velocity and slightly modify the gravitational force. Then, we will add Jupiter, investigating the behaviour of the system after altering its mass. Finally, the kinematics of the Sun, all the eight planets and Pluto will be simulated.

For most objects in the Solar System the classical treatment is sufficient. However, the Newtonian gravitational force cannot account for some observed effects, like perihelion precession of Mercury. Therefore, we will introduce a general relativistic correction to the force in the Sun-Mercury system in order to investigate this phenomenon.

The physics behind all of the concepts, as well as the details of algorithms and the numerical implementation are thoroughly described in the Methods section. In the next part, we present and discuss the results produced by the programs, followed by some general conclusions.

The code is based on the concept of object-oriented programming, making it possible to reuse methods for many celestial bodies involved. The main program (SolarSystemModel.cpp) is written in C++ and together with some selected results, vec3 class and the Python code (project5_read_plot.py), used to read the output files produced by the main program, process and plot the data, available in the following Github repository:
github.com/jowborowska/CompPhysics

# 2 Methods

## 2.1 Algorithms

We want to solve a set of coupled ordinary differential equations (Eq. 1). The two algorithms that will be employed are Euler's and velocity Verlet method. Their description follows closely that of Hjorth-Jensen (p. 245-250) and Nikolic.

Both algorithms are finite difference, one-step methods. In both cases, we start with the initial vectors of the position, $\vec{x}_0 = \vec{x}(t = t_0)$, and velocity, $\vec{v}_0 = \vec{v}(t = t_0)$. Thereafter, we can calculate the fixed time-step,

$$\Delta t = \frac{t - t_0}{N}, \tag{2}$$

where $t_0$ is the initial time, $t$ is the final time and $N$ is the number of used data-points (time-steps). With this step and derivative of the function, we can construct the next value at $\vec{x}_1 = \vec{x}(t_1 = t_0 + \Delta t)$ (analogously for velocity). In general, the next value of some function, $y_{i+1}$, in terms of the previous step, $y_i$, can be written as

$$y_{i+1} = y(t = t_i + \Delta t) = y(t_i) + \Delta t \left( y'(t_i) + ... + y^{(p)}(t_i)\frac{\Delta t^{p-1}}{p!} \right) + \mathcal{O}(\Delta t^{p+1}), \tag{3}$$

where we Taylor expanded the function $y$ and $\mathcal{O}(\Delta t^{p+1})$ represents the truncation error.

**Euler's method**

If we truncate the above equation at first derivative, we will get

$$y_{i+1} = y(t_{i+1} = t_i + \Delta t) = y(t_i) + \Delta t y'(t_i) + \mathcal{O}(\Delta t^2), \tag{4}$$

which gives the Euler method. Even though every step we make an approximation error of order $\mathcal{O}(\Delta t^2)$, the total error is the sum over all steps and goes as $N\mathcal{O}(\Delta t^2) \approx \mathcal{O}(\Delta t)$ (since $N \propto \Delta t^{-1}$). We can then increase the precision by decreasing $\Delta t$ (increasing $N$).

Back to our case, with position, velocity and acceleration. Using the appropriate notation for Eq. 4, we have that Euler's algorithm goes as

$$\vec{x}_{i+1} = \vec{x}_i + \Delta t \vec{\dot{x}}_i = \vec{x}_i + \Delta t \vec{v}_i,$$

$$\vec{v}_{i+1} = \vec{v}_i + \Delta t \vec{\dot{v}}_i = \vec{v}_i + \Delta t \vec{a}_i,$$

where acceleration, $\vec{a_i}$ can be found from Newton's second law and we repeat the procedure for each point in time, $i \in [0, N-1]$, and each celestial body in the system.

**Velocity Verlet method**

If we truncate Eq. 3 at second derivative, we will get

$$y_{i+1} = y(t_{i+1} = t_i + \Delta t) = y(t_i) + \Delta t y'(t_i) + \frac{\Delta t^2}{2} y''(t_i) + \mathcal{O}(\Delta t^3). \tag{5}$$

Again, this translates to

$$\vec{x}_{i+1} = \vec{x}_i + \Delta t \vec{\dot{x}}_i + \frac{\Delta t^2}{2} \vec{\ddot{x}}_i = \vec{x}_i + \Delta t \vec{v}_i + \frac{\Delta t^2}{2} \vec{a}_i.$$

If we add the above equation to the corresponding Taylor expansion for $\vec{x}_{i-1}$, we obtain

$$\vec{x}_{i+1} = 2\vec{x}_i - \vec{x}_{i-1} + \Delta t^2 \vec{a}_i + \mathcal{O}(\Delta t^4). \tag{6}$$

We see that the velocity is not included. However, it can easily be calculated through

$$\vec{v}_i = \frac{\vec{x}_{i+1} - \vec{x}_{i-1}}{2\Delta t} + \mathcal{O}(\Delta t^2). \tag{7}$$

Equations 6 and 7 together consitute the original Verlet algorithm. Because this method is not self-starting, another algorithm must be used to obtain the first few terms. Additionally, the new velocity in Eq. 7 is found by computing the difference between two quantities of the same order of magnitude, which can lead to a substantial loss of numerical precision. We will derive a mathematically equivalent version of this method - velocity Verlet algorithm. First, we see from Eq. 7 that $\vec{x}_{i-1} = \vec{x}_{i+1} - 2\Delta t \vec{v}_i$, which can be substituted into the Eq. 6 to obtain

$$\vec{x}_{i+1} = 2\vec{x}_i - (\vec{x}_{i+1} - 2\Delta t \vec{v}_i) + \Delta t^2 \vec{a}_i \Rightarrow 2\vec{x}_{i+1} = 2\vec{x}_i + 2\Delta t \vec{v}_i + \Delta t^2 \vec{a}_i$$

$$\Rightarrow \vec{x}_{i+1} = \vec{x}_i + \Delta t \vec{v}_i + \frac{\Delta t^2}{2} \vec{a}_i. \tag{8}$$

Moreover, based on Eq. 7, we have

$$\vec{v}_{i+1} = \frac{\vec{x}_{i+2} - \vec{x}_i}{2\Delta t}, \tag{9}$$

and based on Eq. 6,

$$\vec{x}_{i+2} = 2\vec{x}_{i+1} - \vec{x}_i + \vec{a}_{i+1}\Delta t^2. \tag{10}$$

Substituting Eq. 10 into Eq. 9 gives

$$\vec{v}_{i+1} = \frac{2\vec{x}_{i+1} - \vec{x}_i + \vec{a}_{i+1}\Delta t^2 - \vec{x}_i}{2\Delta t} = \frac{\vec{x}_{i+1} - \vec{x}_i}{\Delta t} + \frac{\vec{a}_{i+1}\Delta t}{2}. \tag{11}$$

Now, we can subsitute Eq. 8 into Eq. 11, which yields

$$\vec{v}_{i+1} = \frac{\vec{x}_i + \Delta t\vec{v}_i + \frac{\Delta t^2}{2}\vec{a}_i - \vec{x}_i}{\Delta t} + \frac{\vec{a}_{i+1}\Delta t}{2} = \vec{v}_i + \frac{\vec{a}_i\Delta t}{2} + \frac{\vec{a}_{i+1}\Delta t}{2}$$

$$\Rightarrow \vec{v}_{i+1} = \vec{v}_i + \frac{\Delta t}{2}(\vec{a}_i + \vec{a}_{i+1}). \tag{12}$$

Equations 8. and 12. represent the velocity Verlet algorithm. The new acceleration, $\vec{a}_{i+1}$, can also be obtained via Newton's second law, using the updated position, $\vec{x}_{i+1}$. Again, the process is repeated for each point, $i \in [0, N-1]$, and each celestial body.

### 2.2 Gravitational force and units

The Newton's law of gravitation is given by

$$\vec{F}_{AB} = \frac{GM_AM_B}{r^2}\hat{r}, \tag{13}$$

where $\vec{F}_{AB}$ is the force acting on object A exerted by object B, $G$ is the gravitational constant, $M_A$, $M_B$ are masses of the objects, $r = |\vec{r}|$ is the distance between them and $\hat{r}$ is the normalized position vector, $\hat{r} = \frac{\vec{r}}{|\vec{r}|}$. The position vector must be defined as $\vec{r} = \vec{r}_B - \vec{r}_A$, so that the force is directed from the object it is applied to. With this definition, it is clear that $\vec{F}_{AB} = -\vec{F}_{BA}$. The total net force acting on object A (that will be used in Newton's second law to calculate its acceleration) is the sum of the gravitational forces exerted by all the other bodies in the system,

$$\vec{F}_A = \sum_{B \neq A} \vec{F}_{AB}. \tag{14}$$

We will use solar masses as mass units for all the celestial bodies, where one solar mass is equal $M_\odot = 2 \times 10^{30}$ kg. For distance, astronomical units will be used, where 1 AU = $1.5 \times 10^{11}$ m is the average distance between Sun and Earth. The time will be measured in years, since it is most suitable for time evolution of the Solar System. The value of gravitational constant, $G$, in these units can be found by treating the gravity as centripetal force, making the Earth's orbit circular,

$$\frac{M_\oplus v^2}{r} = \frac{GM_\odot M_\oplus}{r^2} \tag{15}$$

$$\Rightarrow G = \frac{rv^2}{M_\odot} = \frac{1\text{AU} \cdot \left(\frac{2\pi \cdot 1\text{AU}}{1\text{yr}}\right)^2}{M_\odot} = 4\pi^2 \frac{AU^3}{\text{yr}^2\, M_\odot},$$

where we have also shown that Earth's speed (assuming circular orbit) is $v = 2\pi\frac{\text{AU}}{\text{yr}}$.

### 2.3 Conserved quantities

One of the features that can test the stability of the algorithms is how well they conserve the total energy and angular momentum. We do not introduce any external forces (only a conservative force - gravity, acting between the objects in the system), so that the total mechanical energy should be conserved over time. Moreover, just as linear momentum is conserved when there are no external forces, the angular momentum remains constant with the net torque equal zero, which is true in our case.

#### Energy

We will treat all the celestial bodies as point particles, neglecting rotational kinetic energy. Then, the total kinetic energy is

$$E_k = \sum_i \frac{1}{2} M_i v_i^2, \tag{16}$$

where we sum over all bodies $i$ with mass $M_i$ and translational velocity $v_i$. The total potential energy is

$$E_p = \frac{1}{2} \sum_i \sum_{j \neq i} \frac{-GM_i M_j}{r}, \tag{17}$$

where $\frac{1}{2}$-factor accounts for the fact that we use each pair of bodies twice (numerically this problem is fixed by correctly employing for-loops, starting the inner loop with index $j = i+1$). The total mechanical energy is then the sum, $E_{tot} = E_k + E_p$.

#### Angular momentum

The total angular momentum is defined as

$$\vec{L} = \sum_i \vec{L}_i = \sum_i \left( \vec{r}_i \times m_i \vec{v}_i \right), \tag{18}$$

where $\vec{r}_i$ and $\vec{v}_i$ are respectively the position and velocity vectors of body $i$ with respect to the origin, placed in the center of the mass (initially taken as the position of the Sun). In three dimensions, we have $\vec{r}_i = x\hat{i} + y\hat{j} + z\hat{k}$ and $\vec{v}_i = v_x\hat{i} + v_y\hat{j} + v_z\hat{k}$, so that

$$\vec{L}_i = L_x\hat{i} + L_y\hat{j} + L_z\hat{k} = (x\hat{i} + y\hat{j} + z\hat{k}) \times m_i(v_x\hat{i} + v_y\hat{j} + v_z\hat{k}).$$

Using the fact that cross product of basis vectors fulfills the following relations in a right-hand coordinate system,

$$\begin{aligned} \hat{i} \times \hat{i} = \hat{j} \times \hat{j} = \hat{k} \times \hat{k} = \vec{0}, \\ \hat{i} \times \hat{j} = -(\hat{j} \times \hat{i}) = \hat{k}, \\ \hat{j} \times \hat{k} = -(\hat{k} \times \hat{j}) = \hat{i}, \\ \hat{k} \times \hat{i} = -(\hat{i} \times \hat{k}) = \hat{j}, \end{aligned}$$

we can easily show that

$$\vec{L}_i = L_x\hat{i} + L_y\hat{j} + L_z\hat{k} = m_i(yv_z - zv_y)\hat{i} + m_i(zv_x - xv_z)\hat{j} + m_i(xv_y - yv_x)\hat{k}. \tag{19}$$

The value of the angular momentum of body $i$ is then the length of this vector,

$$|\vec{L}_i| = \sqrt{L_x^2 + L_y^2 + L_z^2} = m_i\sqrt{(yv_z - zv_y)^2 + (zv_x - xv_z)^2 + (xv_y - yv_x)^2}$$

(analogously for the total angular momementum, being the sum of $\vec{L}_i$'s).

### 2.4 Escape velocity and force modification

Escape velocity is the minimal speed needed in order to free a body from the gravitational influence of another massive body. We will try to determine its value for Earth numerically, performing the simulation with different values of its initial velocity. In non-relativistic regime, we can use conservation of energy in order to find the value of escape velocity analytically. There must be equality between the intial sum of kinetic energy and gravitational potential energy and the sum evaluated at infinity, which is zero,

$$E_k + E_p = 0. \tag{20}$$

In general, the gravitational potential, $V$, is the gravitational energy per unit mass, which at a distance $r$ from mass $M$ can be defined as the work that needs to be done by an external force to bring the unit mass from infinity to that point,

$$V(\vec{r}) = \frac{1}{m}\int_{\infty}^{r} \vec{F}\cdot d\vec{r}. \tag{21}$$

We introduce the modification to the Newtonian gravitational force between the Sun and the Earth in the form

$$\vec{F}_G = \frac{GM_\odot M_\oplus}{r^\beta}\hat{r}, \tag{22}$$

where we let $\beta \in [2,3]$ ($\beta = 2$ corresponds to the classical gravitational force). The gravitational potential is then

$$V = \frac{1}{M_\oplus}\int_{\infty}^{r} \frac{GM_\odot M_\oplus}{r^\beta}dr = -\frac{1}{\beta-1}\frac{GM_\odot}{r^{\beta-1}}. \tag{23}$$

Using this form of the potential to find the gravitational binding energy, $E_p = VM_\oplus$, as well as the conservation of energy (Eq. 20), we get

$$\frac{1}{2}M_\oplus v_e^2 - \frac{1}{\beta-1}\frac{GM_\odot M_\oplus}{r^{\beta-1}} = 0 \tag{24}$$

$$v_e = \sqrt{\frac{2GM_\odot}{(\beta-1)r^{\beta-1}}}. \tag{25}$$

We see that in general, the higher the value of $\beta$, the smaller the escape velocity - gravitational force gets smaller more abruptly as a body drifts away from the central mass, so that the orbit becomes unstable. This will be checked numerically. For Newtonian gravitational force, $\beta = 2$, with previously defined units, we have

$$v_e = \sqrt{\frac{2\left(4\pi^2\frac{AU^3}{yr^2 M_\odot}\right)M_\odot}{AU}} = \sqrt{8\pi^2\frac{AU^2}{yr^2}} = 2\pi\sqrt{2}\frac{AU}{yr}, \tag{26}$$

where $v_e$ is the escape velocity.

### 2.5 General relativistic correction

At the beginning of XX-th century Einstein proposed the general theory of relativity, being a revolution of our understanding of space and time. In this treatment mass determines the curvature of spacetime and that curvature, in turn, controls the movement of objects. In case of a small mass, the predictions of general relativity must agree with Newton's law of gravitation. Differences between the models are subtle. However, they are possible to detect.

Of all the planets in the Solar System, Mercury orbits the closest to the Sun and is therefore most affected by the distortion of spacetime produced by the star. According to classical, Newtonian approach, the gravitational forces exerted by other planets would cause Mercury's perihelion (the point of orbit closest to the Sun) to change position by about $531''$ per century. Nevertheless, it was observed that the actual advance is $574''$ per century. This discrepancy (43 arc seconds) is predicted by the general theory of relativity (lumen, Astronomy).

We will test the above by removing all the classical effects from other planets (considering Sun-Mercury system only) and introducing a general relativistic correction to the Netwonian gravitational force, so that

$$F_G = \frac{GM_\odot M_{\text{Mercury}}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right], \tag{27}$$

where $c$ is the speed of light in vacuum (in our units, $c = 63239.7263$ AU/yr) and $l = |\vec{r} \times \vec{v}|$ is the magnitude of Mercury's orbital angular momentum per unit mass, $\vec{l} = \vec{L}/M_{\text{Mercury}}$. We will run a simulation of the system containing only the Sun and Mercury, initializing the planet at x-axis. Thereafter, we will check the perihelion angle,

$$\theta_p = \arctan\left(\frac{y_p}{x_p}\right), \tag{28}$$

where $x_p$, $y_p$ are coordinates of the position of Mercury at perihelion. The initial $\theta_p$ is then zero, and we will compare it to the value after 100 years of simulation.

### 2.6 Initial conditions and considered systems

Throughout the whole project, we treat all the celestial bodies as point masses. First of all, we consider the system with Sun and Earth only. We keep the Sun at rest (by not updating its position and velocity, which is initialized to be zero). Sun is positioned at the origin and, simultaneously, the mass-center of the system. We start the Earth at $\vec{x}_0 = (1, 0, 0)$ AU with the circular velocity, $\vec{v}_0 = (0,2\pi,0)$ AU/yr. This configuration will be used for the comparison between velocity Verlet and Euler's methods in the first part of Results section. Thereafter, we change the initial velocity of Earth in order to find the escape velocity. Finally, we alter the value of $\beta$-exponent (in the modified gravitational force), initializing the Earth with $\vec{v}_0 =$ (0,7,0) AU/yr.

Next, we investigate the perihellion precession of Mercury, again keeping the Sun at rest as the center of the mass of the Sun-Mercury system. Mercury is initialized with $\vec{x}_0 = (0.3075, 0, 0)$ AU and $\vec{v}_0 = (0,12.44,0)$ AU/yr.

Therafter, we examine the system containing Sun, Earth and Jupiter, while changing Jupiter's mass. The Sun, fixed at the origin, again is considered to be the mass-center of the system. The initial positions of Jupiter and Earth are collected using NASA's HORIZONS Web-Interface (see references) for the day 12 December 2019. As these data contain positions

with respect to the barycenter of the Solar System, we adjust them accordingly, placing the Sun as the center instead. Next, we investigate the behaviour of the same system, but now with all the bodies in movement around the center of mass, including the Sun.

Finally, we add remaining planets and Pluto. All the initial positions and velocities are taken from HORIZONS Web-Interface for 12th of December 2019.

### 2.7 Numerical implementatation

The code is based on the exemplary set of classes written by Hjorth-Jensen (see references for the github link), yet significantly modified and developed for our purposes and applications. We have focused on the object-orientation of the program, so that the most possible parts can be re-used and applied to different celestial bodies. There are several classes - for creating an object, calculating physical quantities associated with its motion and gravitational pull (forces, energies, angular momentum), writing data of interest to an output file, and finally, performing algorithms that solve the set of coupled ordinary differential equations in order to simulate the motion of the system. The whole code is available in the SolarSystemModel.cpp file. We also use an external class for handling vectors, vec3, as well as the python program to read the output files, produce all the necessary figures and perform simple calculations associated with finding the position of Mercury's perihelion, project5_read_plot.py.

## 3 Results and discussion

### 3.1 Euler vs velocity Verlet algorithm

#### Accuracy

We treat both Earth and Sun as point particles here, keeping the Sun at rest throughout the whole process. Initializing Earth's position 1 AU away from the Sun (at x-axis), with circular velocity, $\vec{v}_0 = 2\pi\hat{j}\ \left[\frac{\text{AU}}{\text{yr}}\right]$, should give a circular orbit that closes after one year time (Earth comes back to its original position). One of the factors that can assess the stability of algorithms is how well they produce a closed circular orbit after one year of simulation. We test that for both algorithms, as a function of number of steps, $N = \frac{1\text{yr}}{\Delta t}$. The results, giving the distance between the initial and the final position of Earth after one year, are shown in Figure 1. Obviously, the precision of both algorithms increases with the number of steps, N. However, it is clear that the velocity Verlet algorithm is much more accurate (or needs far less data-points to produce results with the same accuracy as Euler's method). We have also plotted the position of Earth orbiting the Sun during one year of simulation, using velocity Verlet method with $N = 10^5$ time-steps, which produces a closed circular orbit (see Figure 2).

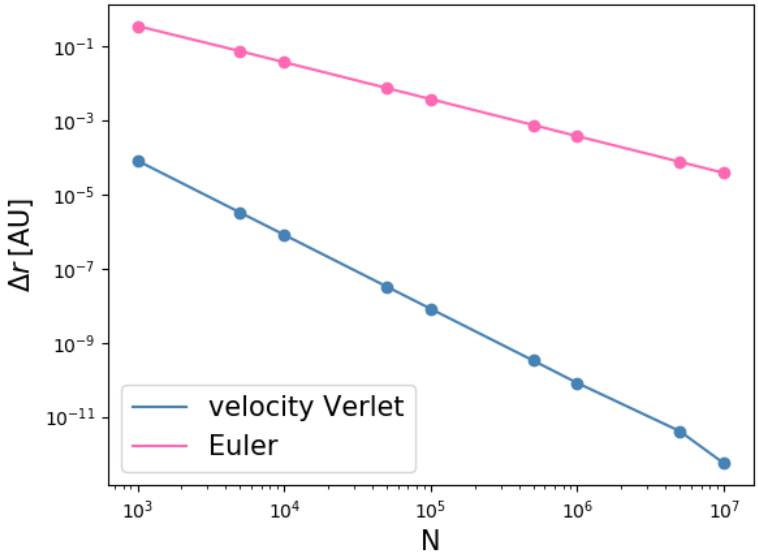


Figure 1: Distance between the initial and the final position of Earth after one year of simulation, $\Delta r$, as a function of number of time-steps, N. Smaller separation corresponds to a better accuracy in creating a stable (closed, circular) orbit. For both algorihtms, more data-points (smaller time-steps) give better accuracy, but clearly, the velocity Verlet algorithm is more precise.

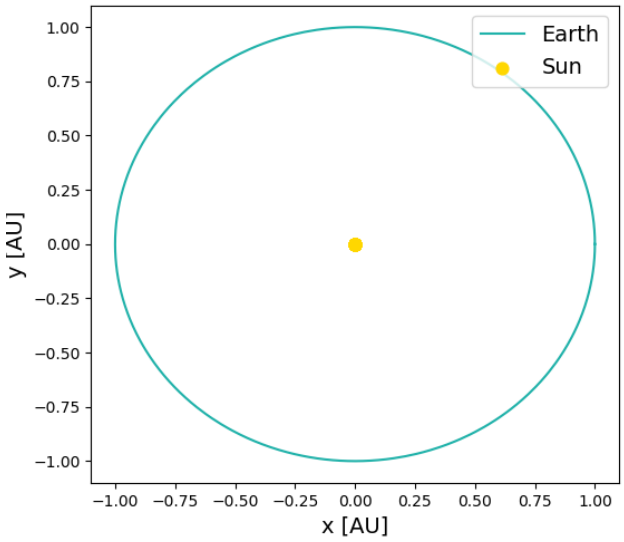


Figure 2: Position of Earth and Sun over one year period, produced by the velocity Verlet algorithm with time-step $\Delta t = 10^{-5}$ yr. Sun is kept at rest throughout the whole simulation time and the initial velocity of Earth is $v_0 = 2\pi \frac{\text{AU}}{\text{yr}}$, which clearly results in a closed circular orbit.

### Energy and momentum conservation

The next aspect of both algorithms, that we are interested in comparing, is how well they conserve the energy and angular momentum. Figure 3 shows the total kinetic energy and total potential energy of the system as a function of time over 10 years of simulation. We can clearly see that both types of energy are very well conserved by the velocity Verlet algorithm, whereas the Euler's mathod is quite unstable. Here, the kinetic energy drops and the potential energy increases by approximately 0.7 % of their initial values. We plot also the absolute relative change in the total angular momentum and total energy, defined as $\Delta E = \left|\frac{E-E_0}{E_0}\right|$, where $E_0$ is the initial energy (analogously for angular momentum), which can be seen on Figure 4. The logarithmic scale shows the difference in precision of both algorithms of many orders of magnitude. Velocity Verlet method is very accurate, with the associated error of order $10^{-13}$. However, Euler's algorithm does not conserve the constants of motion precisely (as in the case of potential and kinetic energy, the error is of order $10^{-3}$). If the Sun is not fixed or the time-step is higher, both quantities oscillate a bit more for the velocity Verlet algorithm, but its accuracy is still much better compared to the Euler's method.

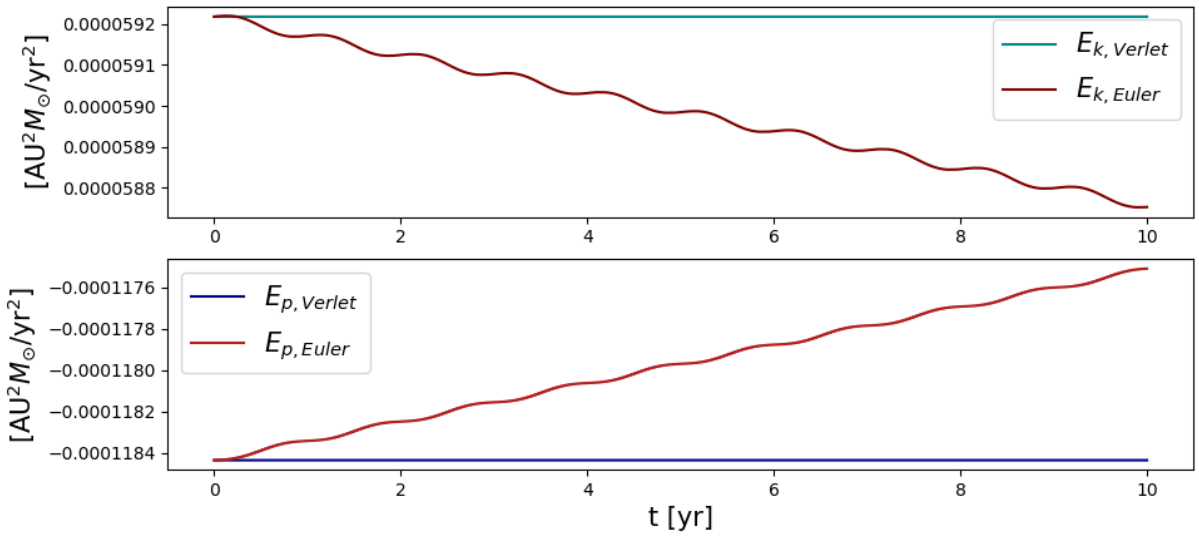


Figure 3: The total kinetic energy (upper panel) and potential energy (lower panel) for 10 years of simulation of the Sun-Earth system, with Sun kept at rest as the mass-center of the system. The time-step used is $\Delta t = 10^{-5}$ yr. We can clearly see that the velocity Verlet algorithm conserves energy much better than Euler's method, which is associated with the relative error of order 7 %.

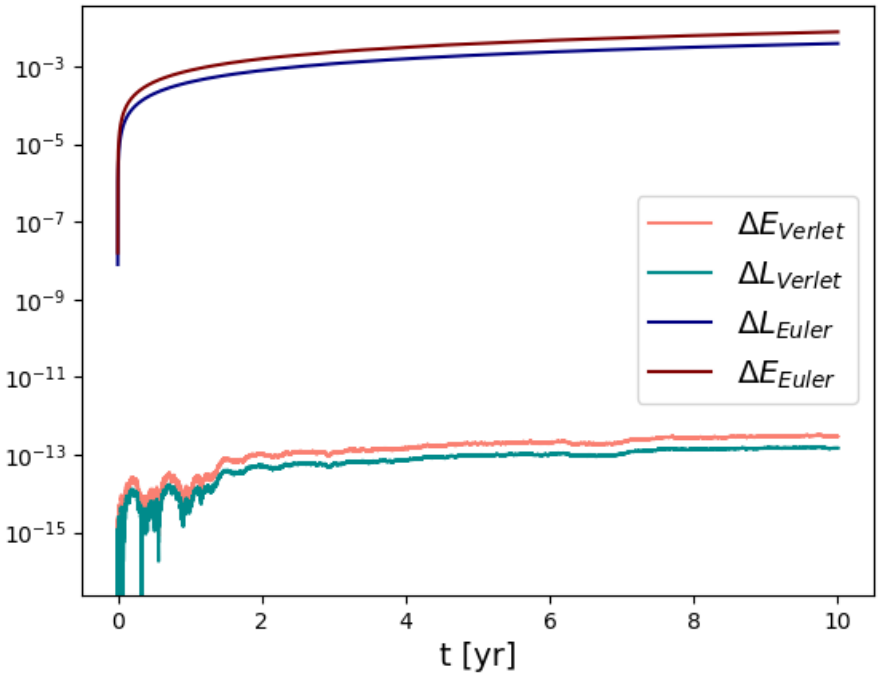


Figure 4: Absolute relative change in the total energy and angular momentum over 10 years of simulation (Sun-Earth system, Sun at rest). The time-step used is $\Delta t = 10^{-5}$ yr. y-axis is logarithmic to show the difference of orders of magnitue between results produced by the two algorithms. The velocity Verlet solver conserves constants of motion much better than Euler's method.

### Time efficiency

Finally, we want to look at the computational time needed in order to perform calculations using both algorithms. The Euler's method is a quite simple approach to solving ordinary differential equations and requires performing 4N floating point operations (two for updating the position and two for velocity), where N is the number of data-points used. For our case, this is in addition to calculating the forces and acceleration, using Newton's second law, for each data-point. However, velocity Verlet method is much more complicated, as it requires 9N FLOPS (see Eq. 8 and 12), as well as determining the acceleration twice. This should correspond to a longer time of computations. We measure the CPU time (average over 3 consecutive runs of the program, to minimize uncertainties) needed in order to simulate Earth's motion in the Sun-Earth system (where Sun is kept at rest) for different numbers of data-points, $N$. The results are gathered in Table 1. It is indeed clear that the velocity Verlet algorithm requires longer computations than the Euler's method for equal final times.

| | CPU time [s] | |
|---|---|---|
| N | Euler | velocity Verlet |
| $10^4$ | 0.005 | 0.012 |
| $10^5$ | 0.040 | 0.124 |
| $10^6$ | 0.530 | 0.856 |
| $10^7$ | 3.467 | 8.599 |
| $10^8$ | 34.275 | 138.025 |

Table 1: The CPU time elapsed while performing the compuations with use of the velocity Verlet and Euler's algorithms for different number of time-steps, $N$, during one year long simulation of the Sun-Earth system (where Sun is kept at rest by not updating its position). The velocity Verlet algorithm is clearly more "time-consuming".

### 3.2 Escape velocity and gravity modification

We have performed the simulation of Earth's motion with different values of its initial velocity (all along y-direction) and Sun kept at rest in order to numerically determine the escape velocity. It is not a trivial task - the notion of escape velocity implies "never coming back" and we cannot simulate eternity. However, we will extend the simulation time to 100 years and assume that this time period is long enough (10000 years gives similar results, but 100 year-plot is included below, since longer simulations focus mostly on the part of trajectory after the escape). The results are shown on Figure 5. By trial and error, we estimate the escape velocity to be somewhere between 8.8 AU/yr (orbit is closed) and 8.9 AU/yr, where the Earth is able to overcome the gravitational influence of the Sun and escape. This is in a very good agreement with the analytically derived value, $v_e = 2\pi\sqrt{2} \approx 8.886$ AU/yr.

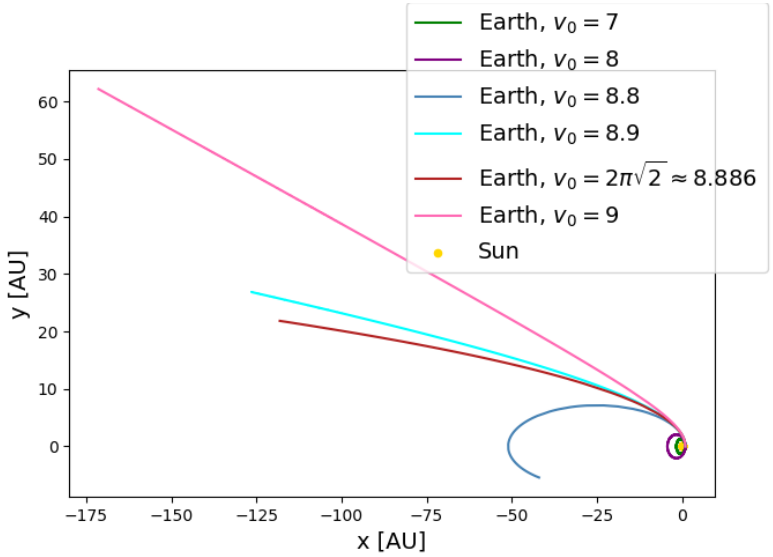


Figure 5: The simulation of the Sun-Earth system over 100 years, with various values of initial velocity (in AU/yr) of the Earth and the Sun kept at rest. The computations are performed using velocity Verlet algorithm with $N = 10^6$ data points (which gives the time-step equal $\Delta t = 10^{-4}$). We can clearly see that for velocities $\geq v_e = 2\pi\sqrt{2}$ AU/yr Earth escapes the gravitational influence of the Sun and leaves a closed orbit.

Next, we simulate the motion of Earth, changing the gravitational force through $\beta$-parameter. As $\beta$ increases, the gravitational pull gets weaker (see Eq. 22) and the orbit becomes unstable (not-closed). The Earth seems to escape the gravitational influence of the Sun completely as $\beta$ creeps towards 3.

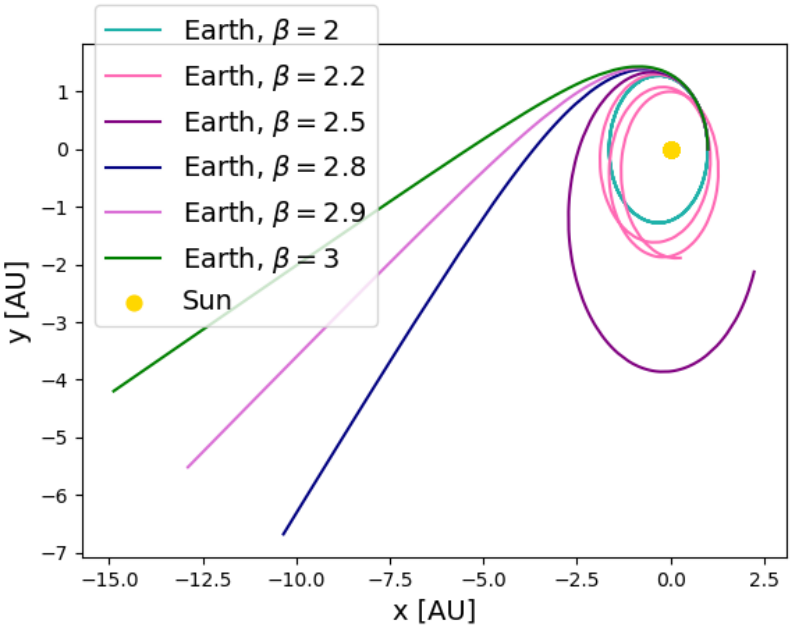


Figure 6: The simulation of Sun-Earth system over 5 years with various values of $\beta$-exponent, modifying the gravitational force. Here, the velocity Verlet algorithm with $5 \cdot 10^5$ time-steps ($\Delta t = 10^{-5}$ yr) is used and the initial velocity of Earth $v_0 = 7$ AU/yr, while the Sun is kept at the origin. Higher $\beta$-value corresponds to a weaker gravitational binding, hence less stable orbit, as can be observed.

### 3.3 Mercury precession

Next, we consider the system with Sun and Mercury only in order to determine the change in position of Mercury's perihelion after one century of simulation. Obviously, due to a finite precision of the algorithm, there will be some "numerical precession" arising from the fact that the orbit is not perfectly closed after one period of motion around the Sun, even without the general relativistic correction implemented to the force. We minimize this inaccuracies by using a small time-step ($\Delta t = 10^{-6}$ yr). Therafter, we find the minimal distance between Mercury and Sun (0.3075 AU) and the corresponding coordinates. After calculating the angle through Eq. 28, converting radians to arc seconds and subtracting the precession resultant from numerical uncertainties, we get $39.6''$, which is in quite good agreement with the observed value, $43''$ per century.

### 3.4 Sun-Earth-Jupiter

Now, we consider the system containing the Sun, Earth and Jupiter. We alter the mass of Jupiter in order to examine its impact on Earth's orbit, which is shown on Figure 7. We see that orbits are stable for the system with the true mass of Jupiter, $M_J = 9.5 \cdot 10^{-4} M_\odot$ (also, the constants of motion are conserved). Therefore, we can assume that any instabilities arise from the mass alterations, not from inaccuracies of the numerical algorithm we use (velocity Verlet). We can observe that as $M_J$ increases, the Earth's orbit becomes unstable (Earth does not come to the same position after one period of motion) - we see thicker lines on the 100 times higher mass-plot instead of overlapping orbits. For the mass that is 1000 times higher, we see that the Earth is gravitationally attracted by the heavy Jupiter and orbits around Jupiter's orbit rather than around the Sun.

So far we have kept the Sun at rest. However, studying the system with Sun in motion around the true mass-center of the system gives very similar results. This implies that our assumption of fixed Sun, being the center of mass, is very good for studying such systems. In reality, the barycenter of the Solar System can range from being near the center of the Sun to being close to its surface. Therefore, especially since we treat the Sun as a point-mass, it is indeed a good approximation to place the mass-center inside the Sun with no translational motion.

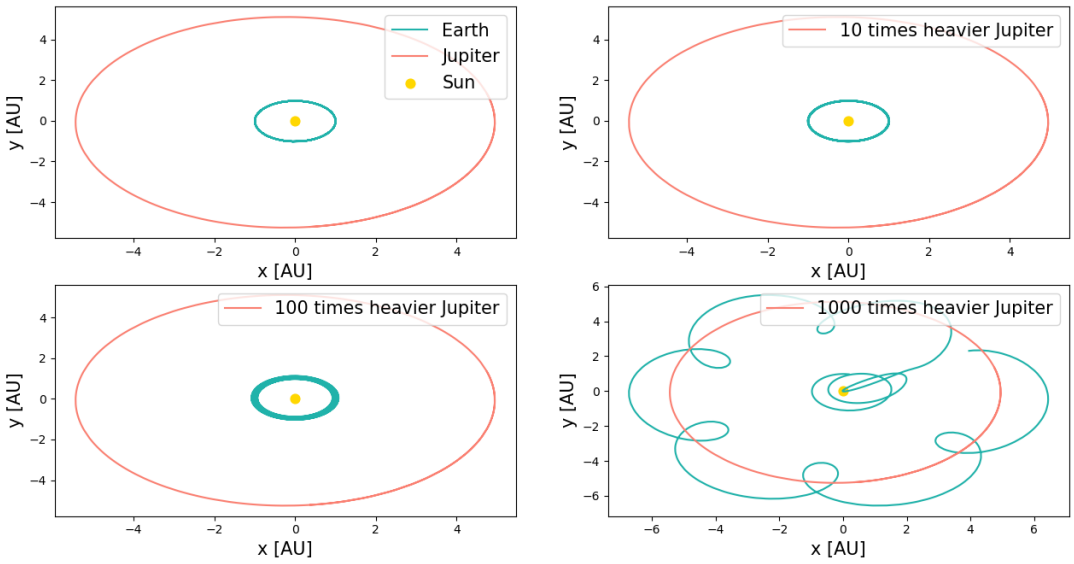


Figure 7: The simulation of Sun-Earth-Jupiter system over 15 years, with various alterations to Jupiter's mass (respectively 10, 100 and 1000 times larger). The results are produced with use of the velocity Verlet algorithm and time-step $\Delta t = 10^{-5}$ yr ($N = 1.5 \cdot 10^6$ datapoints). The Sun is kept at rest as the center of mass of the system (at the origin) and the initial positions of Earth and Jupiter are adjusted accordingly. We see distinct disturbations of Earth's orbit for the case with heaviest Jupiter.

### 3.5 Solar System model

Figure 8 shows motion of the celestial bodies in the Solar System when we include the Sun, all the planets and Pluto. The plots are both in two- and three dimensions. We can observe that not all the orbits are entirely closed due to a relatively short simulation time. For instance, Neptune's period of motion is 165 years, while we have performed the simulation over 150 years only. Moreover, orbits of all the planets are co-planar (lie in the xy-plane), whereas Pluto's orbit is significantly tilted with respect to the xy-plane and lies further away. Its period of motion about the barycenter of the Solar System is 248 years, which is also not completed in our simulation. However, we can see that Pluto is about to cross the orbit of Neptune, which is in agreement with true observations.

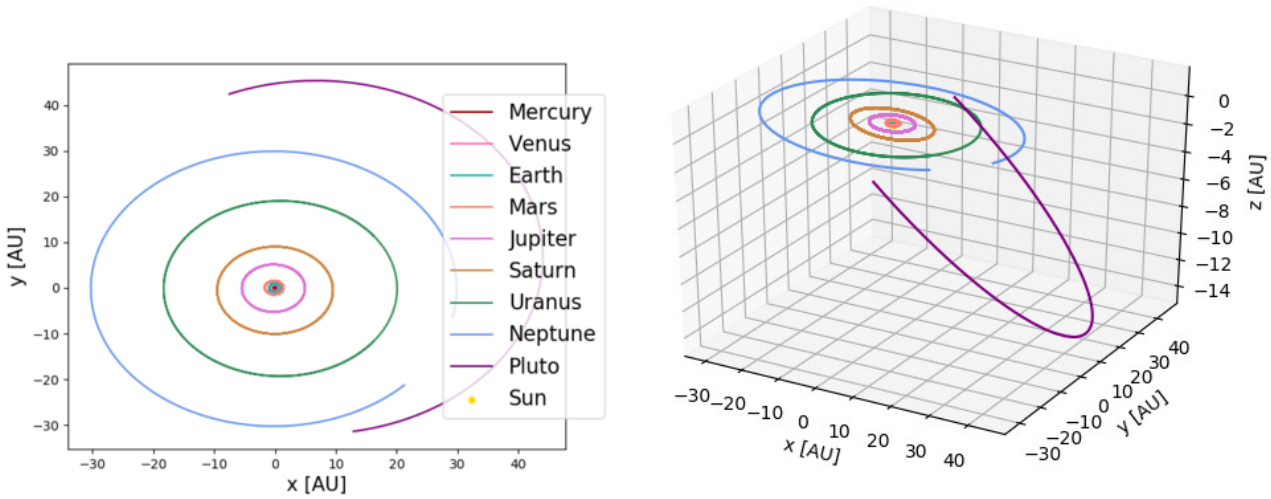


Figure 8: The model of Solar System in 2D (left panel) and 3D (right panel), containing the Sun, 8 planets and Pluto. It was simulated over 150 years, with use of the velocity Verlet algorithm and $1.5 \cdot 10^6$ time-steps ($\Delta t = 10^{-4}$).

## 4 Conclusions

We have seen two different methods for solving a coupled set of ordinary differential equations - velocity Verlet and Euler's algorithm. The Euler solver is a very straightforward approach, not requiring a long computational time. However, the velocity Verlet method is far more accurate and conserves the constants of motion (energy, angular momentum) much better. Therefore, we have chosen to use this method to perform all of the parameter changes throughout the rest of the project.

All of the analyzed features of motion under the influence of gravitational force unraveled some interesting facts about the Solar System. We have tested the prediction of general theory of relativity and learned what would have happened if Earth sped up significantly or Jupiter was much heavier. Additionally, we have tested our assumptions of treating all celestial bodies as point-masses and the Sun being fixed as the center of the mass of the system. These approximations worked very well for our purposes.

# REFERENCES

Hjorth-Jensen, M. (2015) *Computational Physics - Lecture Notes*. Retrieved from: github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf

Hjorth-Jensen, M. (2018) *Class example*. Retrieved from: github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Programs/OOExamples/solar-system

lumen, Astronomy. *Tests of General Relativity*. Retrieved from: courses.lumenlearning.com/astronomy/chapter/tests-of-general-relativity/

NASA, HORIZONS Web-Interface: ssd.jpl.nasa.gov/horizons.cgi#top

Nikolic, Branislav K. (2003) *Computational Methods of Physics - Verlet method*. Retrieved from: physics.udel.edu/ bnikolic/teaching/phys660/numerical_ode/node5.html