

COMPUTATIONAL PHYSICS - PROJECT 5

DAHL, JON KRISTIAN
 FLØISAND, JOHAN ANDREAS
 SAND, MATS OLA

Draft version December 19, 2019

ABSTRACT

We simulate different systems from our solar system and look at various different aspects of Newton's gravitational law as well as two different integration methods. We compare the numerical methods Forward Euler (FE) and Velocity Verlet (VV), where we find that only the latter is suitable for a gravitationally bound system like our solar system, since it is very good at conserving the total energy of the system. When simulating the Jupiter-Earth-Sun system with Jupiter's mass 1000 times its actual mass, VV had a maximum energy deviation of 8% from the initial energy during simulation, which can easily be improved by lowering the integration step size. VV is approximately 1.6 times slower than FE due to its 10 FLOPs per integration cycle compared to FE's 4 FLOPs. VV is still the preferred algorithm in this report since it delivers better performance than FE due to VV's lower errors at higher step sizes. We found that a gravitational force $F \propto r^{-\beta}$ gives closed orbits for $\beta = 2$ for the two-body problem, while for $\beta > 2$ the orbits are no longer closed. We have found that the escape velocity decrease for larger β . By considering the kinetic and gravitational potential energy of the Earth, we found that the escape velocity at a distance 1 AU from the Sun is 8.9 AU/yr. This is consistent with the analytic answer of ~ 8.89 AU/yr. We have seen that the presence of Jupiter alters the eccentricity of Earth's orbit. Simulations of the solar system using VV over 300 years with $dt = 10^{-5}$ yr, gave stable results for all bodies. By adding a relativistic term to Newton's law of gravity, we have calculated that the perihelion of Mercury's orbit precesses approximately $43.05''$ which is almost in agreement of the observed value of $42.98'' \pm 0.04''$ Rydin (2009). We suspect that the numbers will match better if the step size of the integration is lowered, but we have been unable to do this due to memory limitations in our computer.

1. INTRODUCTION

Newton's law of universal gravitation gives us a deterministic way of evaluating the world, but it becomes quite difficult to analytically solve a gravitationally bound system of several objects, since the two-body problem, and some cases of the three-body problem, are the only systems that give analytical answers. Our own solar system is an example of where it is impossible to compute the motion of all involving bodies analytically. We therefore turn to numerical methods to solve the coupled differential equations that govern the movement of the planets.

By treating the solar system as an isolated system, we know that properties like the total energy (kinetic + gravitational potential energy) and angular momentum are conserved values. Keeping these values conserved is a key element of simulating a gravitationally bound system. We will study the precision and performance of two algorithms for integrating ordinary differential equations, namely Forward Euler and Velocity Verlet. We will look at both the time consumption and the errors they produce.

We will study the gravitational force with different r -dependencies, to see how this affects an object's orbit in the two-body problem. We will also look at how the three-body system consisting of the Sun, Earth and Jupiter is affected by changing the mass of Jupiter. Finally we simulate the entire solar system, and we look at the relativistic correction to Newton's law of gravitation, and how this accounts for the missing precession of the perihelion of Mercury's orbit.

We begin by reviewing the theoretical aspects. The integration algorithms are presented, and the theory behind escape velocity and the perihelion precession is presented. Then, we continue with an overview of the methods used to produce the results. The results are presented with descriptions of plots and tables, with conclusions included.

All simulations are run with the g++ compiler on a computer with a Intel Core i7-8700K CPU, 48GB DDR4-3200 C16 RAM running Ubuntu 18.04.3 LTS.

All figures are gathered at the end of the report.

All code used to generate data for this report is available at the GitHub repository <https://github.com/johanaf1/FYS3150-4150/tree/master/project5>.

2. THEORY

2.1. *Gravitational force and Bertrand's theorem*

Assume we have two bodies; object 1 with mass m at position \mathbf{r}_1 and object 2 with mass M at position \mathbf{r}_2 . Newton's law of gravitation tells us that the gravitational force on object 2 from object 1 is given by

$$\mathbf{F} = \frac{GMm}{r^2} \frac{\mathbf{r}}{r}, \quad (1)$$

where G is the gravitational constant and $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$.

Assuming that the bodies we study have bound orbits (they do not escape the system), we can invoke Bertrand's theorem. Bertrand's theorem states that (Goldstein et al. 2002, chapter 3.7): "The only central forces that result in closed orbits for all bound particles are the inverse-square law and Hooke's law." Thus we ex-

pect a closed orbit for Newton's gravitational law. However, for a force

$$\mathbf{F} = -\frac{GMm}{r^\beta} \frac{\mathbf{r}}{r}, \quad (2)$$

we only expect closed orbits for $\beta = 2$.

2.2. Circular orbit

The centripetal acceleration of an object following circular motion is given by

$$\mathbf{a} = \frac{v^2}{r} \frac{\mathbf{r}}{r}.$$

From Newton's second law of motion we know that for a body with mass m and acceleration \mathbf{a} , the force on the object is given by $\mathbf{F} = m\mathbf{a}$. Assume now that we have a two-body system with object 1 at \mathbf{r}_1 with mass M and object 2 at \mathbf{r}_2 with mass m . If we assume that object 1 is at rest, while object 2 has a circular orbit around object 1 due to Newton's gravitational law, we find that

$$\begin{aligned} m \frac{v^2}{r} &= \frac{GMm}{r^2}, \\ \frac{v^2}{r} &= \frac{GM}{r^2}, \end{aligned}$$

where we have used that both forces point in radial direction, \mathbf{r}/r .

For the Earth-Sun system we have $m = M_{\text{Earth}}$, $M = M_{\odot}$ and $r = 1 \text{ AU}$. Assuming an orbital time of one year, and assuming a circular orbit, we get a initial velocity of $v = 2\pi \text{ AU/yr}$. This leads to the relation

$$GM_{\odot} = v^2 r = 4\pi^2 \text{AU}^3/\text{yr}^2.$$

2.3. Escape velocity

We now look at an object with mass m a distance $r = \sqrt{x^2 + y^2 + z^2}$ from the origin, where a point particle with mass M pulls on the object with a gravitational force given by equation (1). We assume the point particle is at rest. The total energy of the object is then

$$E_{\text{tot}} = K + U = \frac{1}{2}mv^2 - G \frac{Mm}{r},$$

where K and U are the kinetic and potential energy of the object and $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ is its velocity. In order for the object to escape the gravitational potential, its kinetic energy has to exceed the gravitational potential energy. If the kinetic energy is lower than the potential energy, the object will be bound to the gravitational pull from the point particle at the origin. We therefore define the escape velocity as the velocity which makes $K = U$:

$$\frac{1}{2}mv_{\text{esc}}^2 = \frac{GMm}{r}.$$

Solving for the escape velocity, yields:

$$v_{\text{esc}} = \sqrt{\frac{2GM}{r}}. \quad (3)$$

2.4. Taylor approximation

With the same premise as above (point particle with mass M at the origin and an object with mass m at a distance r). Combining equation (1) for the force \mathbf{F} with Newton's second law, the acceleration of the object \mathbf{a} is given by

$$m\mathbf{a} = m\ddot{\mathbf{r}} = \mathbf{F} = G \frac{Mm}{r^2} \frac{\mathbf{r}}{r}. \quad (4)$$

From this we can set up a system of coupled first order differential equation to solve:

$$\begin{aligned} \dot{x} &= v_x, & \ddot{x} &= \dot{v}_x = a_x, \\ \dot{y} &= v_y, & \ddot{y} &= \dot{v}_y = a_y, \\ \dot{z} &= v_z, & \ddot{z} &= \dot{v}_z = a_z, \end{aligned}$$

where x, y, z is position, v_i and a_i is velocity and acceleration in direction i respectively and the acceleration is computed from equation (4). This system of equation is not necessarily easy to solve analytically. We therefore Taylor expand the position and velocity to find a numerical algorithm:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \mathcal{O}(h^3), \quad (5)$$

$$v(t+h) = v(t) + h\dot{v}(t) + \frac{h^2}{2}\ddot{v}(t) + \mathcal{O}(h^3), \quad (6)$$

where $\dot{x}(t) = v(t)$ and $\ddot{x}(t) = \dot{v}(t) = a(t)$. From this we will derive the two following algorithms: forward Euler and velocity Verlet. We will derive these for one dimension, but generalising is not difficult.

2.4.1. Euler's method

Euler's method uses a first order approximation of the Taylor expansion. We discretize our time step $n = \frac{t_{\text{stop}} - t_{\text{start}}}{\Delta t}$, where $\Delta t = h$ in equations (5) and (6). For $i \in [0, n-1]$ Euler's method can be written as

$$x_{i+1} = x_i + v_i \cdot \Delta t, \quad (7)$$

$$v_{i+1} = v_i + a_i \cdot \Delta t. \quad (8)$$

This means that the error we make goes as $\mathcal{O}(\Delta t^2)$ for every new step.

2.4.2. Velocity Verlet

The velocity Verlet algorithm uses a second order Taylor approximation. From equation (6) we see that $\ddot{v}(t)$ is unknown. The first order approximation of $\ddot{v}(t)$ is

$$\begin{aligned} \ddot{v}(t) &= \frac{\dot{v}(t+h) - \dot{v}(t)}{h} + \mathcal{O}(h^2) \\ &= \frac{a(t+h) - a(t)}{h} + \mathcal{O}(h^2). \end{aligned} \quad (9)$$

To compute $\ddot{v}(t)$ from equation (9) we need the value of $a(t+h)$. In our case we can compute this from the position $x(t+h)$. We thus have

$$\begin{aligned} x(t+h) &= x(t) + h\dot{v}(t) + \frac{h^2}{2}a(t) + \mathcal{O}(h^3), \\ v(t+h) &= v(t) + h \frac{a(t+h) + a(t)}{2} + \mathcal{O}(h^3) \end{aligned}$$

In that specific order. We now discretize our time step $n = \frac{t_{\text{stop}} - t_{\text{start}}}{\Delta t}$, where $\Delta t = h$. Hence, for $i \in [0, n - 1]$, the velocity Verlet algorithm can be written as

$$x_{i+1} = x_i + v_i \cdot \Delta t + a_i \frac{(\Delta t)^2}{2} \quad (10)$$

$$v_{i+1} = v_i + \frac{(a_{i+1} + a_i)}{2} \Delta t \quad (11)$$

We here see that the error in each step goes as $\mathcal{O}(\Delta t^3)$.

2.5. Perihelion precession

The perihelion of an elliptical orbit is the shortest distance between the orbiting object and the centre of mass for the system. Newton's law of gravitation tells us that Mercury's and other planets' orbits around the Sun are precessing due to the fact that each planet pulls on every other planet. This results in a precession of each planets perihelion. Precession of planetary orbits due to gravitational pull by the remaining planets was already recognized by Isaac Newton [Harper \(2007\)](#); however, in 1859 Urbain Le Verrier found that the precession of Mercury's perihelion was moving too fast (39 arcseconds per Julian century [Rydin \(2009\)](#), which was later re-estimated to 43" by Simon Newcomb [Rydin \(2009\)](#)) for it to only be affected by the known planets in the Solar System (which gives a precession of 531.4 arcseconds/century [Rydin \(2009\)](#)). Le Verrier's solution to this problem was that it existed a planet yet to be discovered between Mercury and the Sun, which he named Vulcan. This was not the case, and it wasn't until 1916 that Albert Einstein found that the remaining arcseconds could be explained with his general theory of relativity [Einstein \(1916\)](#); he even wrote, and we quote: "These facts must, in my opinion, be taken as a convincing proof of the correctness of the theory" (translated from german).

As before, we look at a point particle with mass M at the origin and an object with mass m at a distance r from the origin. The ratio between the Newtonian and this new general relativistic term is expressed in different ways by different writers. We will in this report use the representation from [Goldstein et al. \(2002\)](#) (referring to the so-called Schwarzschild solution), which gives the GR potential

$$V = -\frac{GMm}{r} - \frac{h}{r^3},$$

where the force is found by integration:

$$\mathbf{F}_{GR} = \frac{GMm}{r^2} \frac{\mathbf{r}}{r} + \frac{3h}{r^4} \frac{\mathbf{r}}{r}.$$

(This is actually a Taylor approximation [Hansen \(2019\)](#).) The variable h is defined as ([Goldstein et al. 2002](#), equation (12.48))

$$h = GMm \frac{l^2}{c^2},$$

where $l = |\mathbf{r} \times \mathbf{v}|$ is the specific angular momentum and c is the speed of light. The relativistic term for the force then becomes

$$\mathbf{F}_{GR} = \frac{GMm}{r^2} \frac{\mathbf{r}}{r} + \frac{3GMml^2}{r^4c^2} \frac{\mathbf{r}}{r}.$$

For the two-body system Sun-Mercury with the Sun at rest at the origin, the force becomes

$$\mathbf{F}_{GR} = \frac{GM_{\odot}M_{\text{Mercury}}}{r^2} \frac{\mathbf{r}}{r} \left[1 + \frac{3l^2}{r^2c^2} \right]. \quad (12)$$

3. METHOD

3.1. Program

The code for solving the project has been constructed with object orienting in mind. We have modelled the program after `ODESolver.py` from the course IN1900 (formerly INF1100) ¹.

The general structure is a class `Solver` containing the framework for a general solver for ordinary differential equations. The `Solver` class contains no actual ODE solving algorithms, but takes care of the logistics, like the number of integration steps and the integration loop, setting the initial conditions, and writing the data to file.

All implementations of the ODE solving algorithms are implemented as subclasses of `Solver`, and the subclasses contain only a single method which advances the integration one step with the given algorithm. All other functionality needed is inherited from the `Solver` class. An advantage of this framework is the simplicity of adding a new algorithm to the system. Since `Solver` has all the logistics needed for the machinery to work, a new implementation needs only to contain the solving algorithm. A disadvantage of a system like this is that the generality and the unwillingness of C++ (compared to the ease of use by Python, where constructing this framework would literally only take a couple of hours) makes the implementation hard and time consuming. The two numerical methods Forward Euler and Velocity Verlet are implemented as two subclasses `ForwardEuler` and `VelocityVerlet`.

All the data for the solar system is contained in a class `SolarSystem`. The `SolarSystem` class allows adding celestial bodies with mass and initial x, y, z positions and velocities. When all celestial bodies are added, we must choose which acceleration implementation to be used. The `SolarSystem` class contains four acceleration implementations: 1: Two-body problem with a stationary object of one solar mass in the origin (1), 2: N-body problem with a stationary object of one solar mass in the origin, 3: the same as 2 but with a variable exponent of r in the gravitational acceleration (2), 4: N-body problem where all bodies are moving and the centre of mass is stationary in the origin. When the desired acceleration has been chosen, the `SolarSystem` class passes this function to the ODE solver. The system is then solved and the `SolarSystem` class has done its job.

All data generated in the C++ programs are saved as text files. The text files are then opened and analysed with Python.

3.2. Velocity Verlet vs Forward Euler

While the Forward Euler method is a conceptually simple method which is easy to implement, its efficiency and accuracy is exceeded by many other integration algorithms like the Velocity Verlet method. We measure

¹ <https://github.com/hplgit/scipro-primer/blob/master/src/ode2/ODESolver.py>

the accuracy of the integration algorithms by simulating a two-body problem consisting of the Earth in orbit around the Sun, fixed at the centre, for both the Forward Euler and Velocity Verlet algorithm. Both algorithms are run at the same step length, dt , and the same number of steps, N , to make the comparison equal and to make the simulated time the same. Two runs, one with $dt = 10^{-3}$ yr, $N = 10^5$ and one with $dt = 10^{-2}$ yr, $N = 10^6$, are performed for both of the algorithms to see how they cope with a change in the step length. These two step lengths are specifically chosen since they are of sizes which makes the change in dt and change in integration algorithm visually apparent. Two smaller values dt could just as well have been used, but they would not be particularly exciting to look at when visually represented as a plot of the planet trajectory.

The efficiency of the algorithms are considered by a combination of the number of floating point operations (FLOPs) needed for one iteration of the integration, and the total time it takes for the algorithms to complete the same system simulation. The timings are performed with the high precision timer `std::chrono`, and the number of FLOPs are simply counted in the C++ implementation of the two algorithms.

To find out which algorithm fits the system we want to simulate best, we must consider how long it takes for each of the algorithms to complete the simulation and we must consider the accuracy of the algorithms for a given dt . One of the algorithms could be much faster in completing the simulation, but at the same time have a much higher numerical error.

The accuracies (errors) of the algorithms are measured in three ways, all of them by running the Earth-Sun simulation for 100 years. We force the orbit of Earth to be circular by positioning the Earth at $\mathbf{r} = (1, 0, 0)$ AU, setting the initial velocity to $\mathbf{v} = (0, 2\pi, 0)$ AU/yr and setting $GM_{\odot} = 4\pi^2 \text{AU}^3/\text{yr}^2$. The methods for determining the accuracies are listed below.

- Drift in position: For every time the Earth passes $y = 0$ on the positive x side (that is, going from the fourth quadrant to the first quadrant in position space), the x position of Earth is measured. The accuracies are computed by how much the Earth's x position drifts per orbit, and how much it drifts in total.
- Conservation of energy: The total energy (kinetic and gravitational potential energy) is calculated at each time step. Since there are no external forces, the total energy of the system should be conserved. Any deviation from the initial energy E_0 will be due to numerical errors.
- Conservation of angular momentum: The angular momentum of the Earth is calculated at each time step. Since there are no external torques, the angular momentum should be conserved. Any deviation from the initial angular momentum L_0 will be due to numerical errors.

3.3. Escape velocity

We continue with the two-body system consisting of the Earth orbiting the Sun at 1 AU, where the Sun is a

rest in the origin. As described in section 2.3, we define the escape velocity as the velocity needed for the kinetic energy of the Earth to be equal to its gravitational potential energy. In other words, if the Earth's total energy (kinetic + potential) is greater than 0, then Earth is able to escape the Sun's gravitational pull.

Calculating the escape velocity of Earth is a very lightweight task computationally, since we only need to consider the total initial energy of the Earth, no integration needed. We find the escape velocity by calculating the total initial energy for a set of initial velocities. Once the total energy is positive, we have our escape velocity.

3.4. Two-body and variable β

Still using the Earth-Sun two body system, we aim to test Bertrand's theorem by using equation (2) where we can vary the exponent, β , thereby adjusting the strength and reach of the gravitational force.

3.5. Three-body and Jupiter's mass

Since Jupiter has the largest mass of all the planets in the solar system, it is by far the most gravitationally influential of the planets. This motivates us to see what happens to Earth's orbit if we were to adjust Jupiter's mass. Since Jupiter's mass is approximately one thousandth the mass of the Sun, we study the Jupiter-Earth-Sun system where Jupiter's mass is 1, 10, 100, and 1000 times its actual mass. We then see how Earth's orbit is affected by this drastic increase in mass. In this simulation, we keep the sun fixed in the origin.

3.6. Perihelion precession of Mercury's orbit

The relativistic contribution of the precession of Mercury's orbit is a very subtle effect which demands great precision to calculate. We start by using the modified gravitational force in equation (12) which incorporates an approximation to the relativistic contribution to the Newtonian gravitational force. As discussed in section 2.5 we expect to find precession close to the value $42.98'' \pm 0.04''$ given by Rydin (2009).

To measure the relativistic contribution to the precession, we simulate the Mercury-Sun system for 100 years with a time step of $dt = 10^{-7}$ years. We save data from only the first and last orbit since we are interested in the difference in precession from the first to the last orbit. We find the perihelion of the first orbit and use the relation

$$\tan \theta_P = \frac{y_P}{x_P}$$

to solve for the precession angle, θ_P . Here x_P, y_P are the x and y positions of Mercury at perihelion. We do the same for the final orbit and obtain a final precession angle. The difference in precession angle between the first and the last orbit is then compared. We use the initial conditions $v_y = 12.44$ AU/yr as the speed of Mercury and $x = 0.3075$ AU for the position. To see that the time step is sufficiently small, we check that the perihelion precession we get with a pure Newtonian force in the Mercury-Sun system is very close to zero.

4. RESULTS AND DISCUSSION

4.1. Algorithm timing

Timing data for the Earth-Sun system are displayed in table 1. Forward Euler uses 4 FLOPs per iteration and Velocity Verlet uses 10 FLOPs per iteration, which makes Forward Euler a less computationally heavy algorithm than Velocity Verlet. From the numbers in table 1 we can see that the Velocity Verlet algorithm is approximately 1.5 times slower than Forward Euler for both simulations. Time is however not everything, as we will see in the study of the numerical precision.

Method	$dt = 0.001 \text{ yr}, N = 10^8$	$dt = 0.01 \text{ yr}, N = 10^7$
Forward Euler	60.3105 s	6.31423 s
Velocity Verlet	96.2693 s	10.1983 s
Ratio, VV/FE	1.5962	1.6151

TABLE 1

THE COMPUTATION TIME OF FORWARD EULER AND VELOCITY VERLET FOR $dt = 0.001 \text{ yr}$ AND $dt = 0.01 \text{ yr}$. THE TWO TIME STEPS USE $N = 10^8$ AND $N = 10^7$ NUMBER OF STEPS RESPECTIVELY, WHICH YIELD A SIMULATED TIME OF $T = 10^5$ YEARS. THE RATIO BETWEEN VELOCITY VERLET AND FORWARD EULER IS DISPLAYED IN THE BOTTOM ROW.

4.2. Numerical precision

Figures 1 and 2 display the orbit of Earth for two different step lengths in the top two plots for velocity Verlet and forward Euler respectively. The bottom left plot show the absolute value of the relative total energy error, $\epsilon_{\text{rel}} = \left| \frac{E - E_0}{E_0} \right|$, of Earth as a function of integration steps. The bottom right plot show the x displacement of Earth's position from the initial position, at $y = 0$ as a function of the number of orbits. Figure 3 show the total angular momentum of Earth as a function of the number of time steps.

We can see that the Forward Euler method isn't a good method for a time evolving system bound by gravity, as it does not conserve the energy. The total energy and total angular momentum differs quickly from the initial values immediately after the system has started to evolve, and the displacement in position is evident after only one orbit for both $dt = 0.001 \text{ yr}$ and $dt = 0.01 \text{ yr}$. The relative energy error in the bottom left plot of figure 2 seems to converge for sufficiently many number of time steps, and we see a similar trend with the total angular momentum in figure 3. This is more or less consistent with what we expect since an object far away from a gravitational potential is barely affected by it, and the kinetic energy should therefore remain constant. Since the total angular momentum increases with time, we can conclude that the earth gains angular momentum. A way this can happen is if its orbit gets larger, for instance due to the numerical errors. This is confirmed by the position space plots in figure 2 where we can see that Earth's orbit gets larger for both values of dt .

For Velocity Verlet, we can see that the energy and angular momentum is conserved much better, with a relative energy error of maximum $2.5 \cdot 10^{-6}$ for $dt = 10^{-2} \text{ yr}$. The energy errors oscillate back and forth, so even though the energy deviates from one step to the next, it will quickly deviate in the opposite direction, back towards the correct value. Energy conservation makes the Velocity Verlet algorithm a much preferred integration algorithm for solving systems where energy conservation

is key. The same behaviour is seen in the position displacement in the bottom right plot in figure 1. There is a position displacement, but it too oscillates back and forth. For the total energy to be conserved we expect the position displacement to be close to zero, since a nonzero position displacement will result in a change in the potential energy, and the total energy has been shown to be nearly conserved.

4.3. Escape velocity

In figure 5 we can see the initial total energy for the Earth in the Earth-Sun system against Earth's initial velocity. From the plot we read that with an initial velocity of $\sim 8.9 \text{ AU/yr}$, the total energy of the system is greater than 0, which means that the kinetic energy is greater than the gravitational potential, and the Earth is able to escape the Sun's gravitational grasp. Solving equation (3) with $GM = 4\pi^2 \text{AU}^3/\text{yr}^2$ and $r = 1 \text{ AU}$, we find $v_{\text{esc}} \approx 8.89 \text{ AU/yr}$, which is very close to the numerical value.

4.4. Two-body and variable β

Figure 4 shows the orbital motion of the Sun-Earth system with the sun at rest, where the gravitational force varies with β (equation (2)). This plot was constructed using the velocity Verlet algorithm with initial velocity of the Earth as 6.7 AU/yr . This is a slightly higher initial velocity than the previously used value, $2\pi \approx 6.28318531 \text{ AU/yr}$. The higher initial velocity is used to quicker provoke changes to the system. For $\beta = 2$, which is the value in Newton's gravitational law (1), the Earth follows an elliptical motion with the Sun in one of the foci. When $\beta = \{2.33, 2.67\}$ we get a precession of the elliptic orbit for the Earth, and the larger of the two β values allows the Earth to move further away from the Sun in its orbit. This is consistent with Bertrand's theorem discussed in section 2.1. We only see closed orbits for $\beta = 2$. For $\beta = 3$ we see a different behaviour. The Earth starts on an orbital motion, but is thrown out of the system. A reason for this could be that as it moves further away from the Sun, the gravitational pull from the Sun gets so low that the kinetic energy of the Earth overcomes the potential energy. The Earth is no longer bound by the potential and escapes. Since the gravitational attraction weakens as β gets larger, the escape velocity of the Sun gets lower. That is why Earth is able to escape the Sun in the bottom right plot of figure 4 even though the initial velocity is the same for all the plots. When $\beta = 3$, the Earth starts on an orbital motion, but as the movement progresses, the Earth is flung away from the Sun. This could be due to the gravitational pull from the Sun getting too low so that the kinetic energy of the Earth overcomes the potential energy. The Earth is then no longer bound by the potential and escapes. Since the gravitational attraction weakens as β gets larger, the escape velocity of the Sun gets lower. That could be why Earth is able to escape the Sun in the bottom right plot of figure 4 even though the initial velocity is the same for all the plots.

Another reason could be numerical errors. The algorithm could calculate wrong values, which may lead to wrong trajectories and thus shooting the Earth out of the system.

4.5. The three-body system

We have simulated a three-body system consisting of the Sun, the Earth and Jupiter, where the Sun is fixed in the origin and Jupiter's mass is varying from its natural mass to a thousand times its natural mass, increasing with a factor of ten for each simulation. The initial values for position and velocity of the Earth and Jupiter are taken from NASA² with the sun at rest. All figures discussed in this section was created using the velocity Verlet algorithm. The motion of Jupiter and the Earth around the Sun can be seen in figure 6.

The top plots with one and ten times Jupiter mass shows fairly little alterations to the Earth's orbit for the Sun-Earth system (see figure 4 with $\beta = 2$). For Jupiter with 100 times its normal mass, Earth's orbit is clearly altered, as seen from the thicker line in the plot. Finally, when Jupiter's mass is 1000 times its normal mass, the mass of the Sun and Jupiter is almost the same, and we see that the Earth is not necessarily bound to any of them. The Earth starts orbiting the Sun, but is shortly bound to Jupiter before returning to an orbit around the Sun. We also see that the Earth passes very close to the origin, indicating that numerical errors are very likely since $1/r^2$ becomes very large, very fast for small values of r .

Another view of Jupiter's impact on Earth's orbit is seen in figure 7. The figure shows Earth's distance from the Sun as a function of time for the four different Jupiter masses. We can quite clearly observe a beat phenomenon in all plots where the Earth's orbit is bound to the Sun³. The presence of Jupiter changes the eccentricity of Earth's orbit around the Sun in an oscillating manner. Looking at the plots for $M_{\text{Jupiter}} = \{1, 10, 100\} M_{\text{Jupiter}}$ we see how much the eccentricity varies. A larger variation in distance means higher eccentricity and lower variation in distance means a lower eccentricity for the Earth's orbit. When Jupiter's mass is 1000 times its natural mass, it becomes meaningless to speak about the eccentricity of Earth's orbit since Earth never performs a stable and proper orbit. What we can see from the bottom right plot in figure 7 is Earth's initial oscillation around the Sun, which after approximately 5 years turn into oscillations around Jupiter.

The total energy as a function of time is visualised in figure 8 for the four different M_{Jupiter} . Ideally, the total energy should be conserved so any variation in the energy is due to numerical errors. The system has also been tested with a lower step length $dt = 10^{-7}$ to confirm that the energy variations are not caused by a too low resolution. When Jupiter has its normal mass, we see a simple sinusoidal oscillation in the total energy, with a period of approximately 12 years. The amplitude of the oscillation is relatively small, peaking at 1.00006 times the initial energy. For $M_{\text{Jupiter}} = \{10, 100\} M_{\text{Jupiter}}$ the total energy oscillates in the same manner, but with a beat-like⁴ addition to the oscillations. It seems that the

² <https://ssd.jpl.nasa.gov/horizons.cgi>

³ Beats are a phenomenon which occurs when two oscillations with almost equal frequencies are added, $\sin(\omega_1 t) + \sin(\omega_2 t)$. It is further described in Vistnes (2018).

⁴ We do not actually mean that we get beats, but the type of signal that occurs in figure 8 can be explained by yet again adding two signals very different frequencies. This ensures that destructive interference does not happen very close to each other. See Vistnes (2018) for further details.

change in the eccentricity of the Earth's orbit, as seen in figure 7, is linked with the oscillations in the total energy. These changes become more apparent at 100 times Jupiter's mass where the beat-like frequencies dominate over the base frequency, with an amplitude of ~ 1.0001 times the initial energy. The total energy of the system deviates more when Jupiter's mass increases. When Jupiter has approximately the same mass as the Sun, the total energy deviates from the initial energy by $\sim 8\%$ at the most and keeps oscillating around a deviation $\sim 2.5\%$ while the Earth is orbiting Jupiter. The Earth starts by orbiting the Sun with a total energy oscillating around the initial energy. The energy goes back to oscillating around the initial energy when the Earth is again orbiting the Sun. Note that choosing the Sun as the centre of mass is nonphysical for larger M_{Jupiter} . When Jupiter's mass is approximately the mass of the Sun, we are effectively looking at a binary star system where both stars orbit each other. Locking one of the stars in place is far from what happens in reality with such systems.

It is known that the orbital period around the Sun for Jupiter is approximately 11.8 years NASA (2018a). Looking at the frequencies observed in figures 7 and 8 we find that the shortest frequency is approximately $f_1 \approx 1$ yr, while the envelope frequency is approximately $f_2 \approx 12$ yr. It therefore seems safe to assume that the two frequencies correspond to the period of Earth and Jupiter, respectively, around the Sun. Computing the function

$$\psi(t) = 1 + \alpha \left[\sin\left(\frac{2\pi}{12}t\right) + \beta \cdot \sin\left(\frac{2\pi}{1}t\right) \right],$$

for variable parameters α and β , we clearly see the same tendencies as in figure 8. For the observed beats we know that it should follow a function

$$\phi(t) = 1 + \alpha [\sin(\omega_1 t) + \beta \cdot \sin(\omega_2 t)],$$

for variable parameters α and β . Unfortunately we were not able to analyse this thoroughly and it is thus a subject for further discussion in the future.

In figure 9 we have allowed the Sun to move, making the three celestial objects orbit around a joint centre of mass. In this simulation Jupiter has its actual mass. The system is simulated over 300 years and we can see that the Sun is now in movement. Even though the Sun is mostly influenced by the mass of Jupiter, we had to include all the other planets in the solar system for the sum of the momenta to be zero. When we ran the calculation with only the Sun, Earth, and Jupiter, the sum of the momenta was not zero, and the centre of mass had an initial velocity. The velocity of the centre of mass was constant, supporting the conservation of energy of the Velocity Verlet algorithm.

Consider now the movement of the Sun in figure 9. We can see that the Sun is at most ~ 0.01 AU away from the centre of mass in the origin. The centre of mass for the solar system can at times be located 700 000 km above the surface of the Sun Jose (1936). If we combine the maximum distance between the Sun and the centre of mass with the radius of the Sun, we get

$$700\,000 \text{ km} + R_{\odot} \approx 0.009 \text{ AU},$$

which is in very good agreement with the maximum dis-

tance of the sun in figure 9.

4.6. The Solar System

The solar system with all its planets as well as the dwarf planet Pluto can be seen in figure 10, which was created using the velocity Verlet algorithm. All the orbits that are clearly visible in the figure represent the gas planets, with the outermost being Pluto's orbit. It does look like Pluto crosses the trajectory to Neptune; this is however not the case as the trajectory of Pluto has an orbital inclination of 17.1° NASA (2019), while Neptune's orbital inclination is 1.77° NASA (2018b). Pluto does however reach closer to the Sun in its orbit where Pluto's perihelion is $4436.82 \cdot 10^6$ km NASA (2019) whereas Neptune's perihelion is $4444.45 \cdot 10^6$ km NASA (2018b).

The orbits for the terrestrial planets are enlarged in figure 11; both Mercury and Mars has a noticeable eccentricity, while Venus and Earth follows closer to circular orbits. Looking at figure 12, we can see the importance of choosing a small enough step length dt ; Mercury's orbit has a precession for $dt = 10^{-3}$ yr. This precession seems to largely be a consequence of numerical errors since the precession is barely noticeable in figure 11 where $dt = 10^{-5}$ is a hundred times smaller.

4.7. Perihelion precession

Effect	Expected precession [arcsec]	Calculated precession [arcsec]
Newtonian	0	-0.047
GR	42.98 ± 0.04	43.05

TABLE 2

SHOWS THE PRECESSION OF MERCURY'S PERIHELION IN MERCURY-SUN SYSTEM AFTER A CENTURY (100 JULIAN YEARS). HERE BOTH NEWTONIAN EFFECTS AND EFFECTS FROM THE GENERAL THEORY OF RELATIVITY IS INCLUDED. THE EXPECTED PRECESSION FOR GENERAL RELATIVITY IS TAKEN FROM RYDIN (2009).

In table 2 we see the results from the simulated Mercury-Sun system with the general relativity term included in the gravitational force (equation (12)). We can also see a visualisation of the precession in figure 13, where the perihelion of the initial orbit is marked in red, and the perihelion of the final orbit after 100 years of simulated time is marked in green.

As described in section 2.5, there are $42.98'' \pm 0.04''$ missing from the precession of the perihelion of Mercury's orbit in the Newtonian calculations after 100 years of orbit Rydin (2009). The missing arcseconds are indeed accounted for when including the relativistic term. The simulation yielded a precession of $43.05''$ which is very close to the expected value, but not within the error of $\pm 0.04''$. Since the precession is very small compared to the orbit of Mercury, the numerical integrator demands very high precision when calculating the precession. We experienced that Mercury's orbit had a large precession due to numerical errors, noticeable down to at least $dt = 10^{-5}$ years. The precession due to numerical errors is seen in figure 12 where a step length of $dt = 10^{-3}$ has been used. The precession calculations are done with $dt = 10^{-7}$ years but we suspect that we can get better results still by lowering the step length. As of now, our numerical implementation in C++ is too

memory hungry to run the system for even lower step lengths. At $dt = 10^{-7}$ years the program demands 46.5 GB of memory which is just 1.5GB shy of the capacity of our workstation. If we wish to lower the step length, we must modify the program to only store data for the first and last orbit, while throwing away all other data points on the fly. As of now, this is not implemented in our code, but it is a proposal for future improvements.

The fact that the ~ 43 missing arcseconds per century were accounted for with the relativistic correction to Newton's force of gravity was something Albert Einstein felt was a proof that his general theory of relativity was true. Relativity is a subject very hard to grasp and to understand. To see a low velocity real life example where general relativity is noticeable is not something you see every day and it makes this calculation unique in its own way.

5. CONCLUSION

Our simulations has shown that Velocity Verlet is indeed far superior to Forward Euler when it comes to simulating celestial mechanics; it does have more FLOPs per integration cycle, and the computation time for Forward Euler is $\sim 40\%$ lower than that of Velocity Verlet, but the latter has a lower relative error in energy which is crucial when simulating systems where Newton's laws are involved. Velocity Verlet has shown to conserve energy to a very good precision, where the biggest deviation was when Jupiter's mass got large. The energy deviation was at its largest 8% from the initial energy. We have seen that the energy conservation can easily be improved by lowering the step length. The Velocity Verlet algorithm is a fairly lightweight algorithm which runs very fast in C++. The biggest bottleneck to the simulations proved to be storing the data when the simulation time got large and the step size small, but lowering the step size further still is not a problem computation time wise.

Considering the Earth-Sun system, we have found that the escape velocity of the Sun is achieved when the kinetic energy of the Earth is larger than the gravitational potential energy. We have numerically found the escape velocity to be $v_{\text{esc}} \approx 8.9$ AU/yr which is in good agreement with the analytical result of ≈ 8.89 AU/yr.

By adjusting the exponent of r , denoted β , in the gravitational force, we have seen results in agreement with Bertrand's theorem. When β is increased above 2, the Earth no longer form closed orbits, and at $\beta = 3$ the gravitational grasp of the Sun is too weak to keep the Earth in orbit. We have confirmed that this behaviour is not due to numerical errors, since we see the same trend for very low step lengths down to 10^{-7} years.

We have seen that introducing Jupiter to the Earth-Sun system slightly alters the orbit of the Earth. Jupiter induces a variation in the eccentricity of Earth's orbit, and we see a larger variation in eccentricity when Jupiter's mass is increased. When Jupiter's mass is increased with a factor of one thousand, its mass is very close to one solar mass and the Earth is no longer exclusively bound to orbit the Sun. The Jupiter-Earth-Sun system acts now as a binary star system, where we no longer can keep the Sun fixed in the origin.

When no longer keeping the Sun fixed in the origin, we have seen that its orbital movements around the centre of mass is in very good agreement with observational

data. The observational data states that the solar system's centre of mass is at most located 700 000 km outside the surface of the Sun, where our calculations report $\approx 725\ 669$ km.

The precession of the perihelion of Mercury's orbit was simulated to be $43.05''/\text{century}$ due to Schwarzschild-like general relativity effects. This implies that the observed

precession missing from the Newtonian effects of Mercury's perihelion is well explained by the general theory of relativity, as Einstein stated in his article [Einstein \(1916\)](#).

All code used to generate data for this report is available at the GitHub repository <https://github.com/johanaf1/FYS3150-4150/tree/master/project5>.

REFERENCES

- Einstein, A. 1916, Annalen der Physik, 354, 769,
doi: [10.1002/andp.19163540702](https://doi.org/10.1002/andp.19163540702)
- Goldstein, H., Poole, C. P., & Safko, J. L. 2002, Classical Mechanics, 3rd edn. (Addison Wesley).
<https://books.google.no/books?id=tJCuQgAACAAJ>
- Hansen, F. K. 2019, AST2000 Lecture Notes, Part 2C, General Relativity: Basic principles, https://www.uio.no/studier/emner/matnat/astro/AST2000/h19/undervisningsmateriell_h2019/forelesningsnotater/part2c.pdf
- Harper, W. 2007, Philosophy of Science, 74, 932,
doi: [10.1086/525634](https://doi.org/10.1086/525634)
- Jose, P. D. 1936, Popular Astronomy, 44, 542
- NASA. 2018a, Jupiter Fact Sheet. <https://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html>
- . 2018b, Neptune Fact Sheet. <https://nssdc.gsfc.nasa.gov/planetary/factsheet/neptunefact.html>
- . 2019, Pluto Fact Sheet. <https://nssdc.gsfc.nasa.gov/planetary/factsheet/plutofact.html>
- Rydin, R. A. 2009, The General Science, 13
- Vistnes, A. I. 2018, Physics of Oscillations and Waves (Springer verlag)

APPENDIX

FIGURES

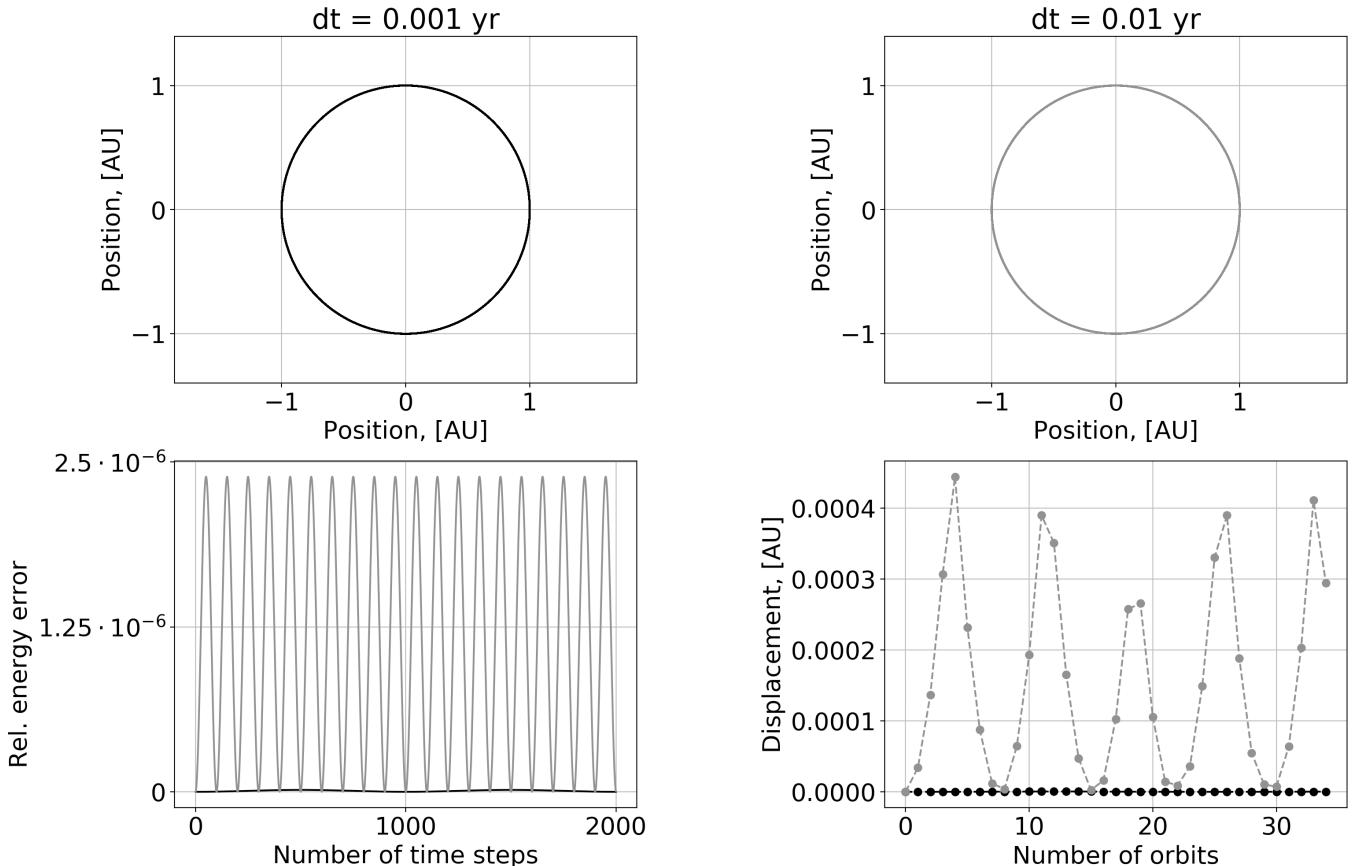


FIG. 1.— The top two plots show Earth's orbit produced with the Velocity Verlet method at step lengths $dt = \{0.001, 0.01\} \text{ yr}$. The bottom left plot show the relative energy error $\epsilon_{\text{rel}} = |\frac{E - E_0}{E_0}|$ at each time step for both step lengths, as indicated by the color of the graphs. The bottom right plot show the displacement of the x coordinate from the initial position, at $y = 0$ (only for positive values of x).

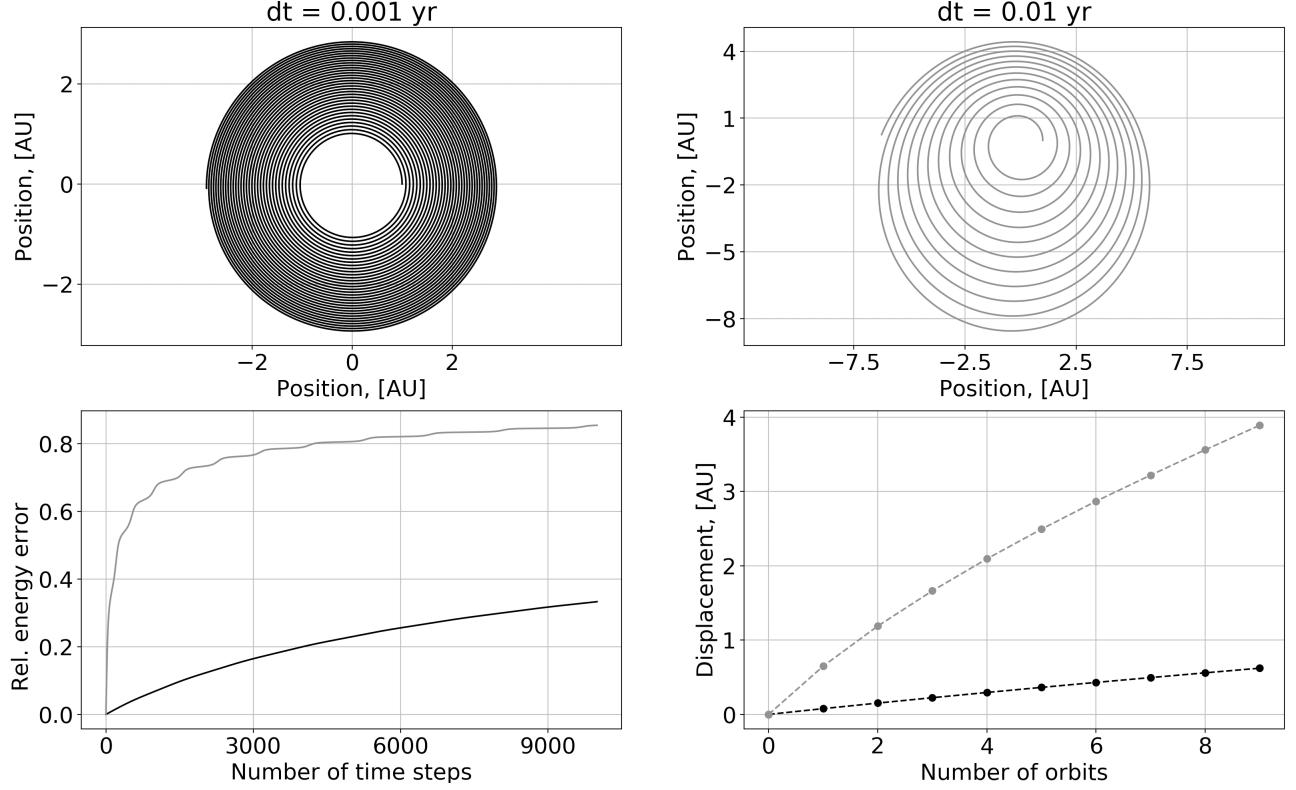


FIG. 2.— The top two plots show Earth's orbit produced with the Forward Euler method at step lengths $dt = \{0.001, 0.01\}$ yr. The bottom left plot show the relative energy error $\epsilon_{\text{rel}} = |\frac{E - E_0}{E_0}|$ at each time step for both step lengths, as indicated by the color of the graphs. The bottom right plot show the displacement of the x coordinate from the initial position, at $y = 0$ (only for positive values of x).

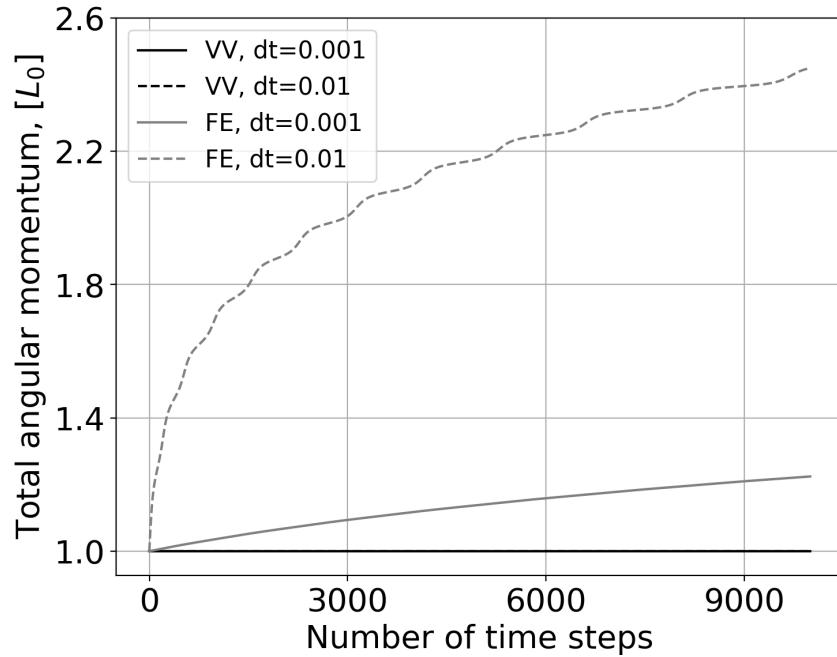


FIG. 3.— The total angular momentum at each time step in units of the initial angular momentum, for Velocity Verlet and Forward Euler at step lengths $dt = \{0.001, 0.01\}$. Since the total angular momentum should be conserved, any deviation from 1 is the result of numerical errors induced by the integration algorithms.

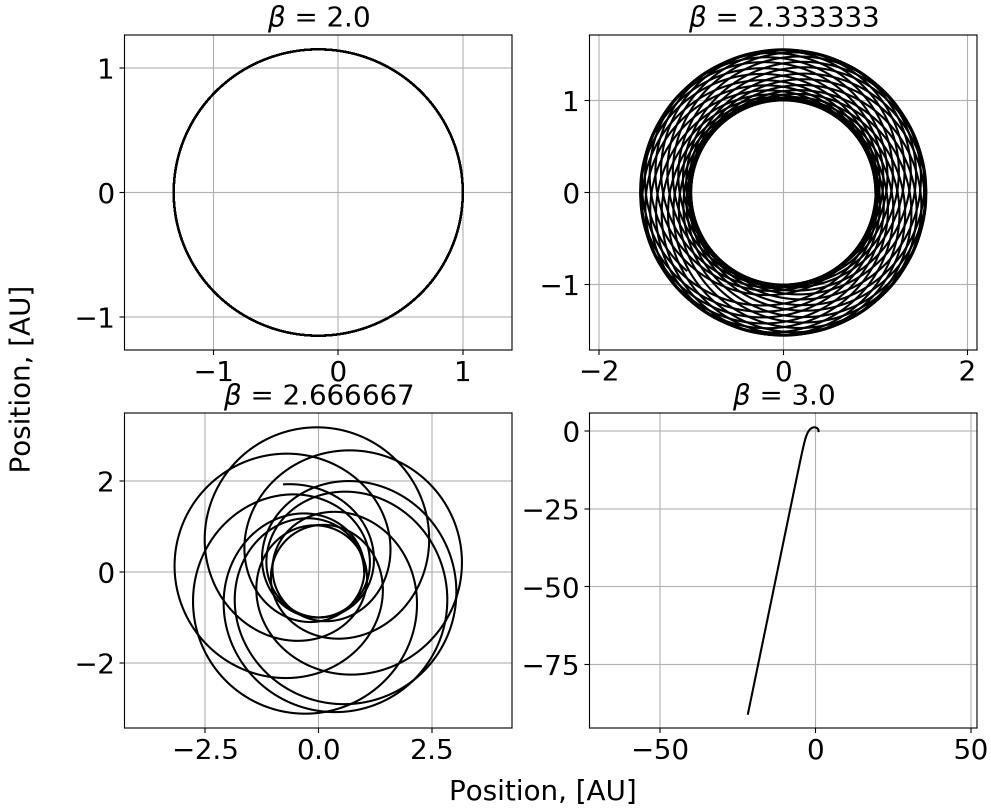


FIG. 4.— Earths orbit where the gravitational force is given by equation (2). $\beta = \{2, 2.33, 2.67, 3\}$ have been used and show that the orbit is dramatically changed when β changes. We see for $\beta > 2$ the bound orbits are no longer close, just as Bertrand's theorem states. The plot was created using the velocity Verlet algorithm. Earth initial velocity: 6.7 AU/yr, which is larger than its actual initial velocity to provoke faster changes. Simulated time: 40 yr, $dt = 10^{-5}$ yr.

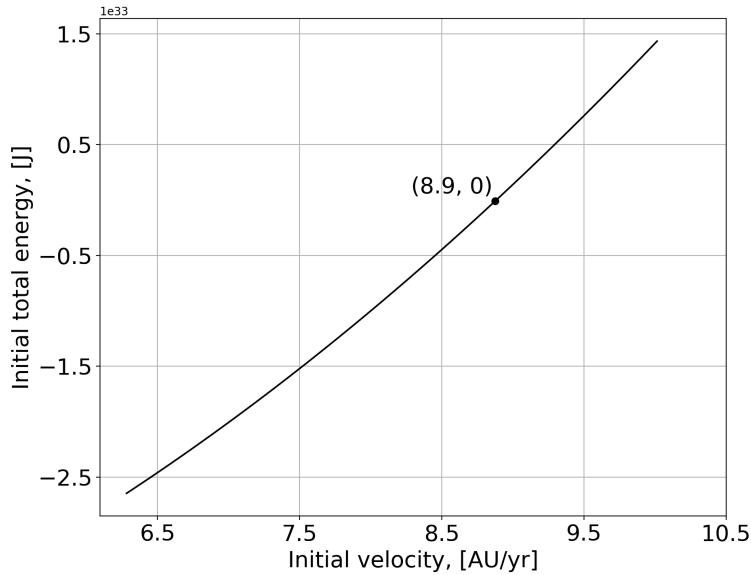


FIG. 5.— Starting at a distance of 1 AU from the Sun, the Earth has a total energy (kinetic + gravitational potential) as indicated on the y-axis. The initial velocity in units of AU/yr is indicated on the x-axis. At an initial velocity of approximately 8.9 AU/yr, the total energy is positive (kinetic energy larger than the gravitational potential energy) and Earth is able to escape the gravitational attraction of the Sun. The y-axis is scaled with 10^{33} .

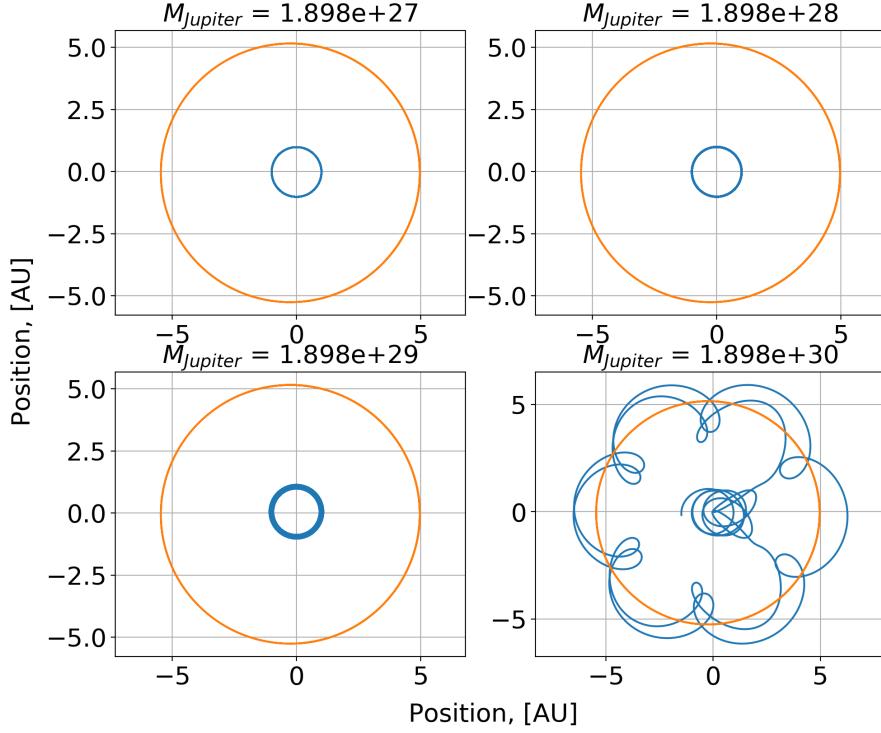


FIG. 6.— Earth and Jupiter’s motion around the Sun at rest in the origin, where Earth follows the blue line and Jupiter follows the orange. Jupiter’s mass varies from plot to plot between $1M_{\text{Jupiter}}$ to $1000M_{\text{Jupiter}}$ with a factor of 10 separating each plot. Simulated time: 30 yrs with $dt = 10^{-5}$, created using the velocity Verlet algorithm.

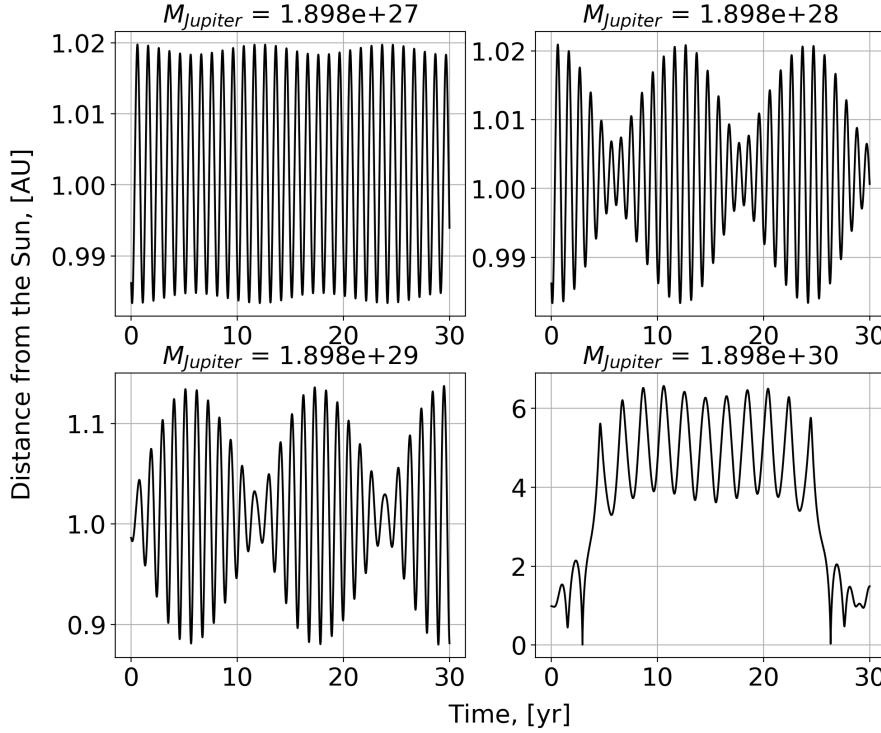


FIG. 7.— Shows the Earth’s distance to the Sun at rest in the origin and how it’s affected by the presence of Jupiter with Jupiter’s mass varying as in figure 6. Simulated time: 30 yrs with $dt = 10^{-5}$, created using the velocity Verlet algorithm.

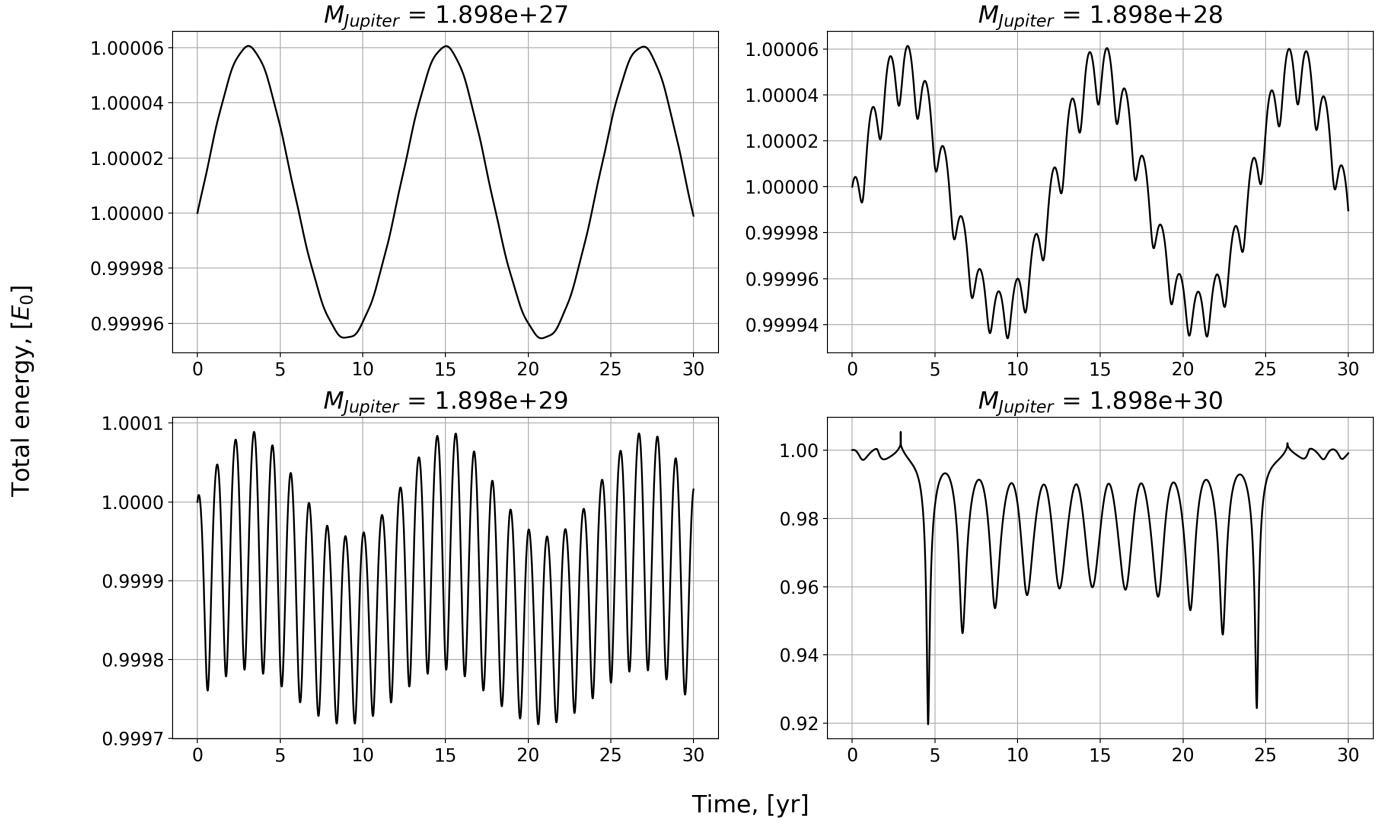


FIG. 8.— Shows the total energy of the three-body system Sun-Earth-Jupiter with the Sun fixed at the origin, for varying Jupiter mass (same as in figure 6). Simulated time: 30 yrs with $dt = 10^{-5}$, created using the velocity Verlet algorithm.

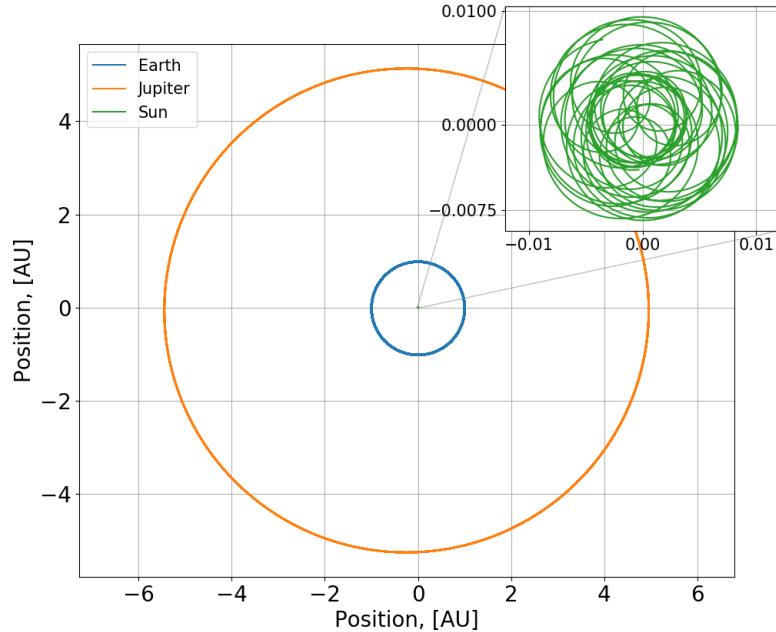


FIG. 9.— The plot visualises the motion of the Sun, the Earth and Jupiter orbiting the centre of mass for the system. Simulated time: 300 years with $dt = 10^{-5}$ years, created using the velocity Verlet algorithm. The orbit of the Sun is enhanced to visualise the orbital motion of the Sun about the centre of mass at the origin. Only the Sun, the Earth, and Jupiter is visible, but the simulation is run with all the planets in the solar system and the dwarf planet Pluto to conserve momentum.

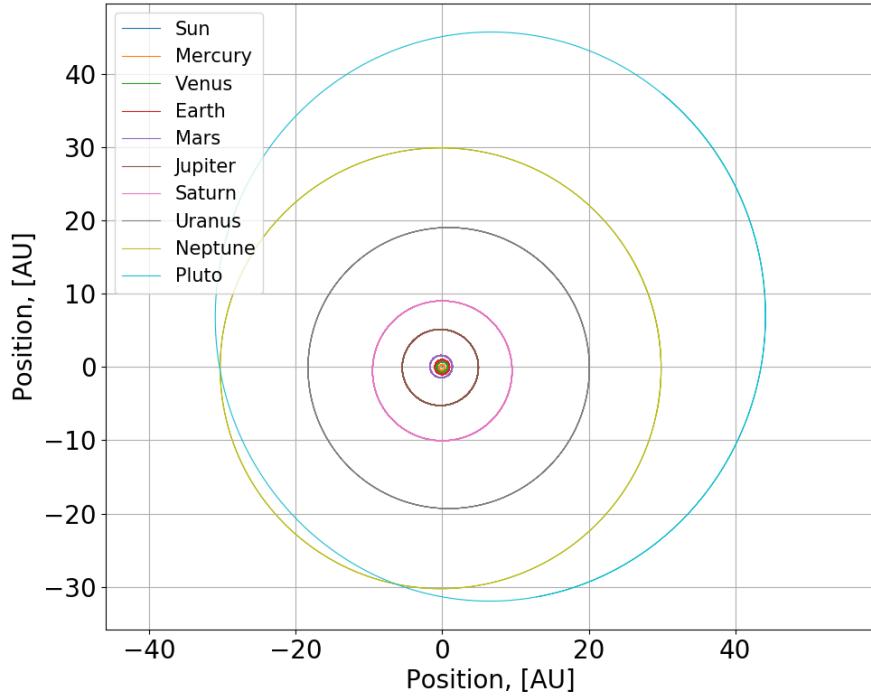


FIG. 10.— The Solar System with all planets as well as our little dwarf Pluto included. Simulated time: 300 years with $dt = 10^{-5}$ years, created using the velocity Verlet algorithm.

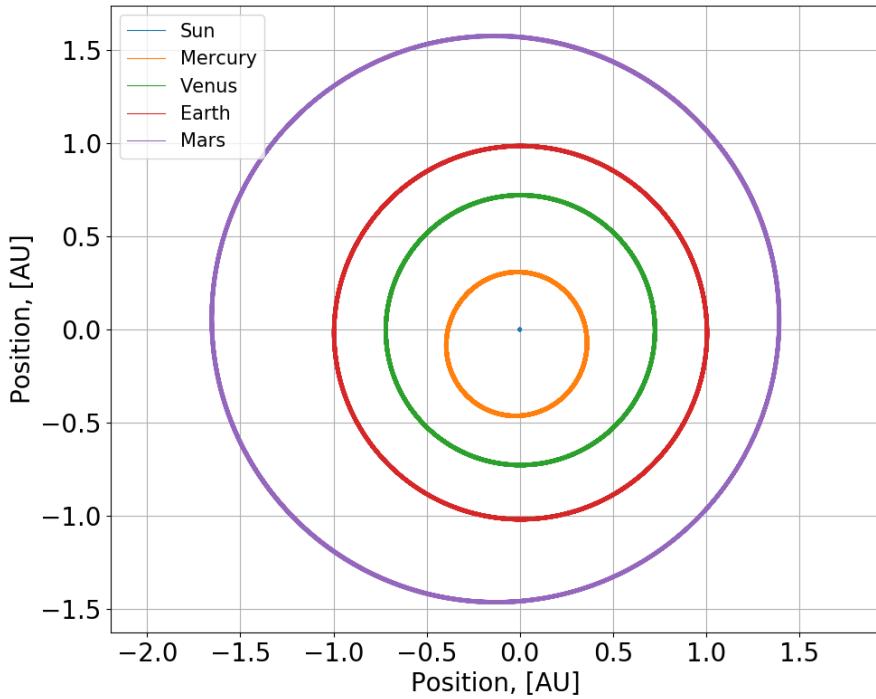


FIG. 11.— The Solar System from figure 10, still with all planets + Pluto, but zoomed in on the terrestrial planets of the Solar System. Simulated time: 300 years with $dt = 10^{-5}$ years.

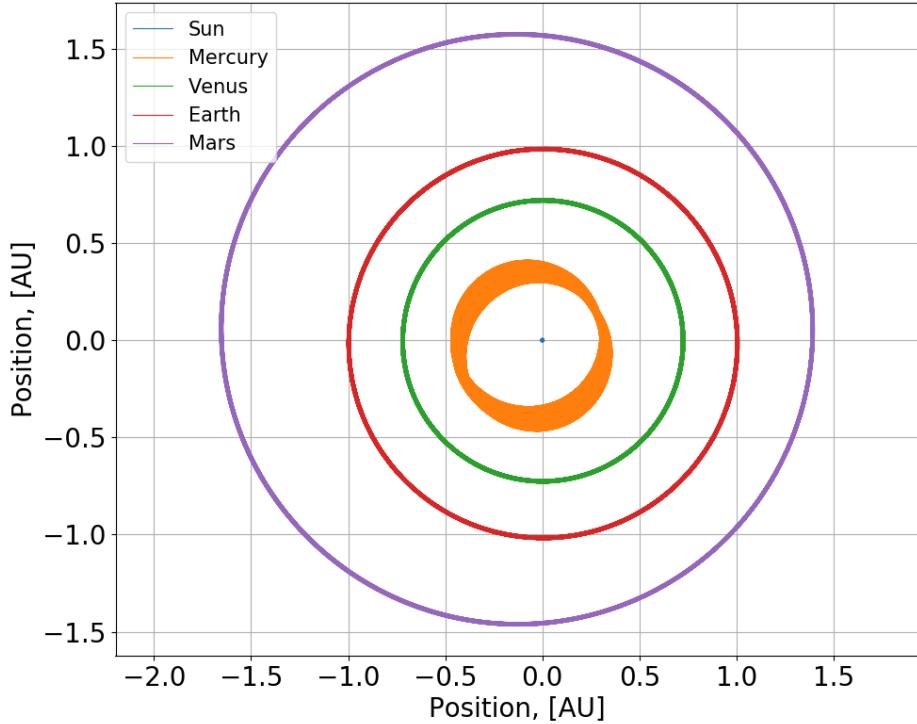


FIG. 12.— The Solar System, again still with all planets + Pluto, but zoomed in on the terrestrial planets of the Solar System, now with another dt . Simulated time: 300 years with $dt = 10^{-3}$ years.

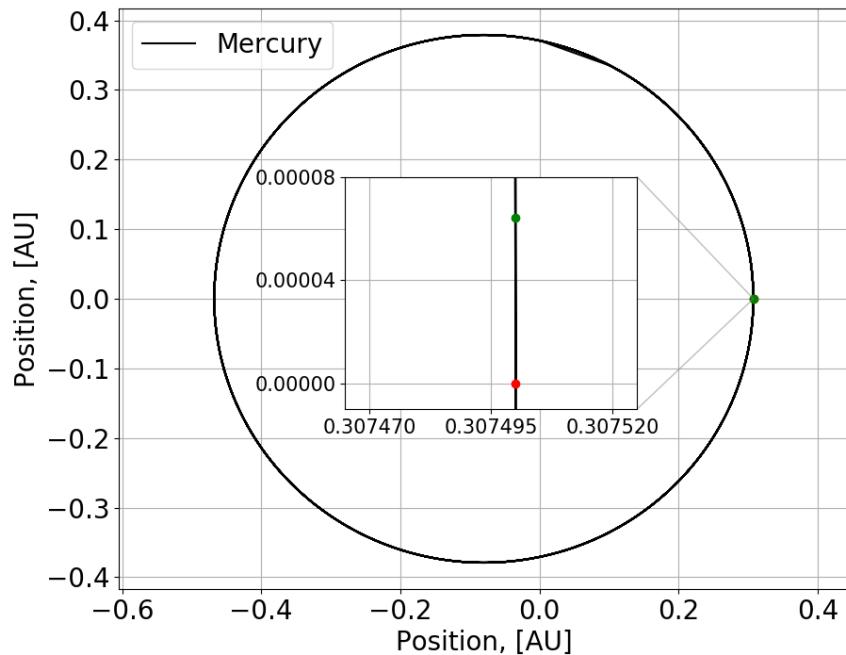


FIG. 13.— The precession of the perihelion of Mercury's orbit due to the relativistic term in the gravitational force (equation (12)). Only the first and last orbit is plotted and it is therefore possible to see that the curve is not circular in the upper right corner. The red dot indicates the perihelion of the first orbit, while the green dot indicates the perihelion of the last orbit after 100 years of simulated time. The system is solved with the Velocity Verlet algorithm, and a step size of $dt = 10^{-7}$ has been used. The Sun is fixed in the origin.