Numerical Simulation of The Solar System

Steinn Hauser and Simen Håpnes
(Dated: February 9, 2019)

A numerical integrator solution to an n-body coupled differential equation system is presented as well as an additional study of Schwarszchild-like general relativity effects on the orbit of Mercury. Euler's method and the velocity Verlet method were compared and the latter was found to be more suitable as it conserved the energy and angular momentum of the system. A three body system consisting of the Sun, Earth and Jupiter was simulated and studied for several variations. A 10 body solar system consisting of the Sun, all eight planets and Pluto was also simulated. The behaviour of n-body systems was found to be complex, as the interplay of a large number of gravitational forces affected the motions of the bodies. The complexity of the system gave rise to an orbital precession which was most significant for the inner planets. Mercury's orbit was found to have a rate of precession over time of 42.98 $\pm$ 0.06 $''$/100yr (arc seconds per Julian century) due to general relativity.

I. INTRODUCTION

The interplay between the stars and planets of our cosmos produces complex and hard to predict motions; building a simulation depicting our solar system in a way which is consistent with observations gets progressively more difficult with increasing detail. Even when disregarding all meteors and moons, an n-body system only has an analytic solution in the case of two bodies. There is no analytic solution for three bodies, and the problem becomes exponentially more complicated to solve analytically when more bodies are introduced. The complex nature of the problem arises from the increasing number of gravitational forces in play between the celestial bodies, as for n planets there are $((n-1)+(n-2)+\ldots+2+1)$ gravitational forces affecting each body. The Solar system is therefore best simulated using numerical tools.

There are various numerical methods to solve coupled ordinary differential equations. Two such numerical integration methods are presented and used: *Euler's method* and the *velocity Verlet* method. Euler's method is the simplest of the two and will act as good basis for the concept of numerical integration. The velocity Verlet method utilizes an algorithm similar to Euler's, but with some advantageous deviations. The numerical approach is to write an object oriented code programmed in C++ which can both solve an initial condition system numerically using the *velocity Verlet* method, and also write out the simulated data to be animated in Python for analyses. The final aim of the study is to simulate the solar system in three dimensions including the Sun, all eight planets and Pluto: a total of ten different bodies exerting gravitational forces on one and other.

An additional study will go into detail on the observed behaviour of Mercury and how general relativity affects its orbit around the Sun. This phenomenon is known as the perihelion precession of Mercury, and was a perplexing mystery to earlier astronomers.

The research is presented in several sections which brief the reader on the theory and method utilized to produce results. Further discussions and conclusions which the results raise are gone into detail subsequently. The project is a cooperative effort by Steinn Hauser and Simen Håpnes and the code developed is included in the following shared GitHub repository: `https://github.com/steinnhauser/FYS3150/tree/master/Project3`

II. THEORY

A. Cosmic Correlations

There is one major force which dictates the motion of the celestial bodies of the solar system, namely the gravitational force. Newton's equation for this force is remarkably still applicable to this day. For the special two-body case of the Sun and the Earth, the Sun will be approximately static due to the fact that $M_\odot \gg M_{Earth}$. In other words, the Sun is affected so little by the gravitational pull of the Earth that it can be disregarded. Newton's law of gravitation states that the gravitational force exerted by the Sun on the Earth is:

$$\vec{F} = -G\frac{M_\odot M_{Earth}}{r^2}\vec{u} \tag{1}$$

G is the gravitational constant ($G \approx 6.67 \cdot 10^{-11} m^3 kg^{-1} s^{-2}$), $M_\odot$ is the mass of the Sun ($M_\odot \approx 2 \cdot 10^{30} kg$), M_{Earth} is the mass of the Earth ($M_{Earth} \approx 6 \cdot 10^{24} kg$), r is the absolute distance between the two bodies in meters, and the vector $\vec{u}$ is a unit vector ($|\vec{u}| = 1$) which points from the Sun to the Earth. This allows us to calculate the acceleration of the Earth, assuming this is the only force acting upon it. Newton's second law states that:

$$\Sigma\vec{F} = m\vec{a} \tag{2}$$

Inserting $m = M_{Earth}$ and $\vec{a} = -a \cdot \vec{u}$ (the minus comes from the acceleration pointing towards the Sun) produces the following acceleration expression for the Earth:

$$-M_{Earth} a \cdot \vec{u} = -G\frac{M_\odot M_{Earth}}{r^2}\vec{u} \tag{3}$$

The acceleration (as a scalar) of the Earth around the Sun (assuming a two-body system) is therefore expressed as:

$$a = G\frac{M_\odot}{r^2} \tag{4}$$

This expression can be scaled to more properly fit the cosmological scale which the research theme asserts; expressing cosmological phenomena in meters and seconds produces numbers unnecessarily large which can become difficult to handle. Assuming the Earth has a circular orbit helps us in this regard, as the relation for centripetal acceleration is now included:

$$a = \frac{v^2}{r} = G\frac{M_\odot}{r^2} \tag{5}$$

Introducing the astronomical unit $AU \equiv 149\,597\,870.7 km$ as a unit of distance and Julian year, $a \equiv 365.25$ days (hereby utilized as the definition of a year, $a = yr$) as the unit of time produces the following new expression for the Earth's velocity around the Sun (by insertion into equation 5):

$$v = \frac{2\pi \cdot 1AU}{1yr} \Rightarrow \left(2\pi\frac{AU}{yr}\right)^2 \frac{1}{r} = G\frac{M_\odot}{r^2} \tag{6}$$

The distance $r = 1AU$ can be assumed to be constant (as is the case for circular orbits), and the equation is then appropriately scaled to cosmological units:

$$GM_\odot = 4\pi^2\frac{AU^3}{yr^2} \tag{7}$$

This relation allows us to rewrite the force relation (from equation 1) to a more generalized acceleration expression for one planet orbiting another with mass M:

$$a = 4\pi^2\left(\frac{M}{M_\odot}\right)\frac{1}{r^2} \tag{8}$$

The radius r is given in AU, and supplying the mass of the other planet in solar masses $M/M_\odot$ produces the acceleration expression which will be utilized in the simulation. Note that the mass of the planet being accelerated is not included, only the mass of the other planet (which applies the force) is relevant. Although this approximation involves the assumption that all orbits are circular, it can be very convenient in numerical calculations; this is due to the fact that parameters can often produce *overflow* and *underflow* when using SI-units in cosmology. These are problems a computer encounters when calculating with numbers which are either too large or too small. The conversion into cosmological units presented is a decent solution to these numerical concerns.

B. Escape velocities and Newtonian gravity

Consider a celestial body orbiting a central mass. If the orbiting body has an increasing mechanical energy, it will eventually escape the central mass' gravitational pull. For this to happen, the kinetic energy of the body must be larger or equal to the potential energy of the central mass. Using the limit of $r \to \infty \Rightarrow E_P \to 0$ and $E_K \to 0$, then the body will need an *escape velocity* v_e of at least

$$E_{K,0} + E_{P,0} \geq E_{K,1} + E_{P,1} \tag{9}$$

$$E_{K,0} \geq -E_{P,0} \tag{10}$$

$$\frac{1}{2}mv_e^2 \geq \frac{GMm}{r_0} \tag{11}$$

$$\Rightarrow v_e \geq \sqrt{\frac{2GM}{r_0}} \tag{12}$$

to escape the gravitational pull of the central mass. G is Newtons gravitational constant, M is the mass of the central mass which the satellite orbits, and v_e is the escape velocity at a distance of r_0 (absolute distance from the central mass to the satellite). It turns out that inserting the parameters for the Earth-Sun system produces an escape velocity of v_e at $r_0 = 1AU$ is $v_e = 2\sqrt{2}\pi AU/yr \approx 8.8858 AU/yr$. This is the minimum speed required (at distance 1 AU) for the Earth to escape the gravitational pull of the Sun.

However, the force model between two bodies (as described for the Earth-Sun system in equation 1) is a purely mathematical model dictated by the inverse-square law (intensity $\propto$ 1/distance^2). This is not perfectly consistent with observations of the bodies closest to the Sun. A study of Earth's orbit is therefore conducted where the factor $1/r^2$ is varied. Consider now a new gravitational force model which is given by the relation:

$$F = G\frac{Mm}{r^\beta} \tag{13}$$

The exponent of the distance r between the masses M and m is now dictated by the parameter β. Several different cases can be tested with this parameter but this research will study β the domain of $\beta \in [2, 3]$.

Varying the parameter β for the force will also affect the potential energy expression. This change in potential energy comes from the conservative force relation

$$\vec{F} = -\nabla V, \tag{14}$$

which implies that the following is valid for the potential energy expression V:

$$F \propto 1/r^\beta \Rightarrow V \propto 1/r^{\beta-1} \tag{15}$$

Integrating F (equation 13) with regards to r produces the following potential energy (as a function of β) expression:

$$V = -\frac{1}{(\beta-1)}\frac{GMm}{r^{\beta-1}} \tag{16}$$

This new expression for the potential energy causes the parameter β to have an influence on the escape condition described in equation 10. This means that keeping the kinetic energy constant and increasing the parameter β may cause the orbiting body to escape depending on its initial position r_0 and velocity v_0.

C. The Perihelion Precession of Mercury

The perihelion of an elliptical orbit is the point closest to the center of mass. For Mercury's orbit, observations reveal that the perihelion precesses around the Sun. Arbitrarily one could also say that the entire elliptical orbit is rotating, resulting in a flower-shaped trajectory pattern.

The predictions from classical physics for this precession were roughly $43''/100yr$ (arc seconds/Julian century) less than what is observed. Albert Einstein proposed in 1916 that his theory of general relativity explains this additional precession of $43''/100yr$[1].

This effect can be explained by Schwarzschild's solution to the general relativity equations which applies to spherical mass objects such as the Sun. The gravitational force F (as proposed by Schwarzschild) becomes far more intricate, but can be simplified (using reasonable approximations for the Sun [3]) to:

$$F \approx G\frac{Mm}{r^2}\left(1 + 3\frac{l^2}{c^2r^2}\right) \tag{17}$$

Where M, m are the masses of the two objects, G is Newton's gravitational constant, r is the distance between the objects, c is the speed of light in a vacuum, and l is the angular momentum per mass ($l = r \times v$). The force is dependent on angular momentum.

Newer and more precise theoretical estimations have been done by (Park et. al. 2017), which concluded that the estimation for the perihelion precession of Mercury is $(575.3100 \pm 0.0011)''/100yr$, including $(42.9799 \pm 0.0009)''/100yr$ which is the contribution from Schwarzschild like general relativity effects[2].

III. METHOD

A. Numerical integration methods

1. Euler's Method

Euler's method is a well known rudimentary algorithm for simulating objects in motion such that if the initial position and velocity are well defined, the method can approximate the object's trajectory for any given acceleration expression. The method involves discretizing space into a grid of N integration points x_i (where $i \in [0, N]$) with a step size h (where $x_i + h = x_{i+1}$) and utilizing the tangent of the curve x_i' to approximate the next point x_{i+1}. The step size h is related to the number of integration points, N, and the boundary conditions x_0 and x_N (initial and final values):

$$h = \frac{x_N - x_0}{N+1} \tag{18}$$

In the case of a solar system simulation, the discretized axis is *time*. In this case, the step size Δt will be given by the total simulation time T and the number of integration points N:

$$\Delta t = T/(N+1) \tag{19}$$

For each small step size Δt, the acceleration is used to approximate the evolution of the velocity, which is then used to approximate the evolution of the trajectory (due to the relation $a{=}v'{=}r''$) by the following algorithm:

$$r_{i+1} = r_i + \Delta t \cdot v_i \tag{20}$$

$$v_{i+1} = v_i + \Delta t \cdot a_i \tag{21}$$

r is the position in space, v the velocity of the object, and a is its acceleration. This is the essence of Euler's method. This algorithm must be run for each coordinate, depending on the number of dimensions simulated. In three dimensional Cartesian coordinates, the positions are updated for

x, y, and z according to their corresponding decomposed velocities v_x, v_y, v_z. The acceleration vector is given by

$$\vec{a}_i = -\frac{a_i}{|\vec{r}_i|}\vec{r}_i, \tag{22}$$

where the acceleration a_i is expressed by equation 8 (in the case presented) and $\vec{r}_i = (x_i, y_i, z_i)$ is the position vector. If many planets are included, the positional coordinate $\vec{r}_i$ is relative depending on the positions of two celestial bodies. I.e to apply equation 22 on the force on Earth from Jupiter, the following expression is applied:

$$\vec{a}_i = -\frac{a_i}{|\vec{r}_{diff}|}\vec{r}_{diff}, \tag{23}$$

where

$$\vec{r}_{diff} = \vec{r}_{Jupiter} - \vec{r}_{Earth} \tag{24}$$

$$|\vec{r}_{diff}| = \sqrt{(x_E - x_J)^2 + (y_E - y_J)^2 + (z_E - z_J)^2} \tag{25}$$

2. Velocity-Verlet Method

Similarly to *Euler's method*, the *Velocity-Verlet method* utilizes the slope of a curve to approximate the subsequent discretized points. However, with *Euler's method* comes a consequence: For some curves, the numerical error of one iteration can be added to the previous one, and after many iterations the approximation becomes practically incorrect. This error is proportional to the step size Δt; a very small step size will minimize the error of each calculation, but after thousands of iterations the error becomes noticeable again. In other words (in the case of astrophysics), the trajectory of the Earth produced by the simulation will stray from its trajectory and not follow its typical elliptical orbit. This is a big problem as the energy of a system must be conserved. If Earth's trajectory is not a closed elliptical orbit (due to numerical errors), then the total energy (the sum of the kinetic and potential energies) of the Earth will increase or decrease accordingly. This means that the algorithm is not *symplectic*, meaning that it will not take into consideration the conservation of potential and kinetic energy.

The *Velocity-Verlet method* is introduced for precisely this reason. This method is *symplectic* in that any numerical errors which arise will not have an influence on the total energy of the simulated body (however the total energy might oscillate around a value). The method uses the following iterations to calculate the evolution of the positional coordinate x_i and velocity v_i at step $i \in [0, N-1]$ (see appendix A for derivation):

$$x_{i+1} = x_i + h \cdot v_i + \frac{h^2}{2}a_i + \mathcal{O}(h^3) \tag{26}$$

$$a_{i+1} = a(x_{i+1}, t_{i+1}) \tag{27}$$

$$v_{i+1} = v_i + \frac{h}{2}(a_{i+1} + a_i) + \mathcal{O}(h^3) \tag{28}$$

The position (in the next time step) is approximated using a second order Taylor polynomial, which is the used in the calculation for the acceleration (at the next time step). At last the velocity is updated using both accelerations (step i and $i+1$). This integration method is implemented in the same fashion as *Euler's method*, where an iteration calculates each dimension $[x, y, z]$ independently using the corresponding decomposed acceleration $\vec{a}_i$ described in equation 22.

B. Object oriented programming and test functions

The mantra of computational simulations is *write once, use many times*. This very applicable to the research in question as the simulation script can either be highly specified to the case presented, or it can be generalized to be used for other planet compositions or in subsequent projects. Generalizing the code enriches the research as planets and other celestial objects can be included with little effort. This makes it possible to simulate any cosmic n-body system, where the parameters need only be set to fit the system in question (e.g. the masses and orbital radii of the planets in our solar system, or the special case of Jupiter and its chaotic moons). If a code is generalized such that it can be used in subsequent research, it must first meet a standard of result certainty. One incorrect statement in a function could cause errors in the results of research, so the program must be tested and validated to be viable.
Test functions are implemented into scripts for precisely this reason, they verify that the components and functions of the script are working as they should by comparing them to the expected results. If a component produces incorrect results (within a margin of error), then the test function of that component should raise a calculation error.

C. Simulation of the Solar System

Simulation of the solar system was done using object oriented programming in C++. The program is initialized by reading a text file containing the initial values for all celestial bodies (The Sun, all eight planets, and of course Pluto). The units used were AU for distance, AU/yr for speed and $M_\odot = 10^{30}kg$ for mass. The solar system barycenter was used as the origin for the coordinate system, and all initial values were obtained from NASA's *Horizons* Web-Interface: `https://ssd.jpl.nasa.gov/horizons.cgi#top`. Note: the initial conditions obtained from Nasa list the Solar System barycenter as origin. The time and date of these initial values is October 4th 2018, 00:00:00.0000 TDB (also referred to as Barycentric Dynamical Time). The units were AU for distance, AU/day for velocities, and various units for mass. The masses were all converted to units $10^{24}kg$, and a conversion for velocities to AU/yr and mass to solar masses is done automatically in the script.

1. Escape Velocity and Newtonian gravity

As derived in the theory section (by the condition 10), the theoretical value for Earth's escape velocity at distance $1AU$ from the Sun is $v_e \approx 8.8858AU/yr$. The Earth is given initial velocities around the theoretical value v_e at distance $1AU$, and simulated (only Earth-Sun system) using the *velocity Verlet method*. The aim of these simulations is to study consistency between the theory and simulations and to see if Earth escaped the gravitational pull of the Sun (within a limited time frame, as the theoretical value is for infinite time and distance).

Afterwards, the factor β is implemented (as described in equation 13) in the *velocity Verlet method* and varied to multiple values in the domain $\beta \in [2, 3]$. The simulation involved only the Earth and Sun, where the Sun is fixed in place and the aim is to look at the shape of Earth's orbit.

2. Planet Class

Every celestial body included in the initializer file is saved as an object of type planet (even for Pluto and the Sun). This class contained information of various characteristics of the planet such as its current position, velocity and mass. The class had functions for finding the distance between the current planet and another, as well as the acceleration relative to another planet (from the scaled acceleration equation 8). A beneficial result of initializing the simulation in this fashion is that planets can be added or removed by simply editing the initial values text file. Many such simulations were of course conducted out of interest of watching the coupled body simulation in action.

3. Solver Class

An instance of the solver class is constructed with four arguments: Time step (Δt), total simulation time (T), a vector containing all planet objects and a boolean variable ($sunfixed$) which dictates whether or not the Sun should be fixed at the origin. When using initial conditions from NASA with the barycenter at origin the Sun should not be fixed. However, by utilizing the option to keep the Sun fixed, the Sun itself should be placed at the origin (the code also makes sure that the Sun does not move when this option is chosen).

The solver class has two functions, one of which is used to calculate the sum of accelerations for each planet due to the forces from all the others. This is a tricky function to format as all the planets assert a_x, a_y and a_z acceleration components on each other but it is solved using a simple $3 \times n$ matrix containing all the accelerations (x-y-z components for all n planets). The second function is a solver which loops through time and solves the velocity Verlet method equations using the acceleration matrix.

All positions are stored in a three dimensional $3 \times N \times n$ tensor (utilizing the cube package from the *Armadillo* library) where the dimensions are:

- Space dimension, $d_i \in [x, y, z]$ (numerically $[0, 1, 2]$)
- Time dimension, $t_i \in [0, T]$ (numerically $[0, N-1]$, $N = T/\Delta t + 1$ which includes the last time T)
- Planet index, $p_i \in [1, n]$ (numerically $[0, n-1]$)

For example, the y-position of planet number four after a time $8 \cdot \Delta t$ is indexed as $Tensor[1, 8, 3]$). This positional tensor is very useful when compressing the method of writing out the data to an independent file for subsequent analyses in Python.

4. Flow of data

The flow of data in the solar system simulation program can be simplified to the following chronological steps:

- Call the initializing planet list function (from the script initializer.cpp) to read initial values and create a corresponding vector containing all planet objects.
- Create an instance of the solver class which declares the simulation time T, the time step Δt and a positional tensor for the system which corresponds with the number of planets n and integration points N chosen.
- Send these values to the velocity Verlet solver, which simulates the trajectories of all objects for the entire time span.
- Write the values calculated in the positional tensor into a file for each planet. The files are designed in the following format: First few lines contains information about the planets name, mass, and column headers x,y,z and t. The remaining lines is data describing the planet's position x_i, y_i, z_i at time t_i.
- Analyze the data in Python using a 3D animation tool from the *matplotlib* library.

For any good simulation there is often a large number of data points, so it is practical to write the files using every 10th or every 100th data point or so. This speeds up the analyses program significantly.

5. Three body problem

The three body system consisting of the Sun, Earth and Jupiter is simulated for various masses of Jupiter ($1M_{Jupiter}$, $10M_{Jupiter}$ and $1000M_{Jupiter}$). This is done for two situations where the Sun is kept fixed in the origin, and where Sun is allowed to move with the system's barycenter at the origin. The stability of the velocity Verlet method is then also studied by calculating the initial and final total mechanical energy $E = E_K + E_P$ for all the system variations.

D. Mercury's perihelion precession

To study at which rate the angle of Mercury's perihelion position over time, a large accuracy is required. The analyses is done in one C++ program which is highly optimized for the aim of the research. The Sun is kept fixed at the origin, and Mercury's orbit is simulated in only two dimensions. The positions are not stored this time, as this would require far more space than a normal laptop has available for the desired number of iterations. The flow of data in the perihelion precession program can be summarized as:

- Initialize all values: time step, total time, initial conditions of Mercury.
- Loop over time, and use the velocity Verlet algorithm to calculate the planet's trajectory.
- Store the distances at the current time step as well as the two previous time steps.
- Verify whether or not the previous time step was a perihelion event by checking if the distance is at a minimum (in relation to the one before that and the current).
- Save two lists including the angles and the times of the perihelion events.
- Write the angles and time values to a file.
- Plot the data and a linear regression (by least squares method) using Python.

This produces a graph where the slope of the linear regression is the rate of precession over time.

Mercury's orbit was simulated around a fixed Sun at the origin, with initial conditions $x_0 = 0.3075AU$ and $v_{y,0} = 12.44AU/yr$. The simulation was conducted with and without the general relativity term (described in equation 17). The time duration of the simulations is 100 years, with a time step of 10^{-7} years. During the simulation, events when Mercury was at the Perihelion were recorded with current angle and time as described above. The rate of precession over time was found as a linear fit using the least squares method. The linear fit was found using the *Numpy* library (*polyfit* and *polyval* methods). An uncertainty for the slope of the line is also calculated. See appendix B for derivation of how the uncertainty is calculated.

IV. RESULTS

Following is a table comparing the two numerical methods:

Characteristic	Euler	Verlet
Runtime	$2631.9ms$	$4099.3ms$
Error $\mathcal{O}$	$\propto h^2$	$\propto h^3$
FLOPS	22	44
Conserved E and l?	No	Yes

TABLE I. Comparison between the two numerical tools utilized for a simulation with $N = 10^7$ integration points. The runtimes were calculated using the average of 10 runs.

The results of the simulations are split in four categories:

- Earth-Sun system
- Three body system (the Sun, Earth and Jupiter)
- Full solar system
- Perihelion precession of Mercury's orbit

A. Earth-Sun system

For the Earth-Sun system, the results will be presented in the following order:

- Position, energy and angular momentum of the Earth, first using Euler's method and then the velocity verlet method. Both used the same initial conditions and number of integration points.
- Escape velocity v_e of the Earth found by trial and error.
- Various plots with different exponents β of the distance r from equation 13.

1. *Euler's Method*

Figure 1, 2 and 3 shows the position, energy and angular momentum of the Earth, respectively. The time duration for these simulations were 10 years with a time step of 0.001 year. The Earth was orbiting a fixed Sun in the origin, and the initial conditions of the Earth were $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year (this should result in a circular orbit).

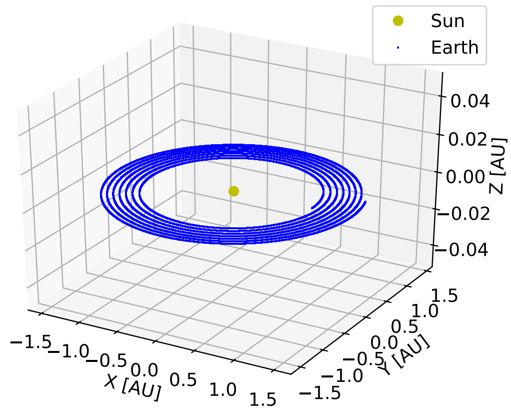

FIG. 1. Euler's method: Position of the Earth in three dimensions. Total simulation time of 10 years with a time step of 0.001 year. Initial conditions of the Earth were: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

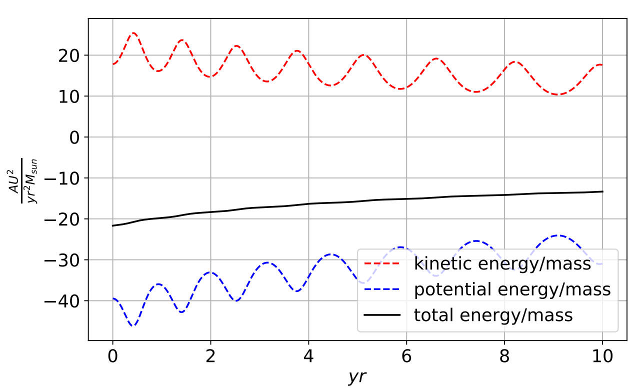

FIG. 2. Euler's method: The kinetic, potential and total energy per mass of the Earth. 10 year simulation time with time step of 0.001 year. Initial conditions: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

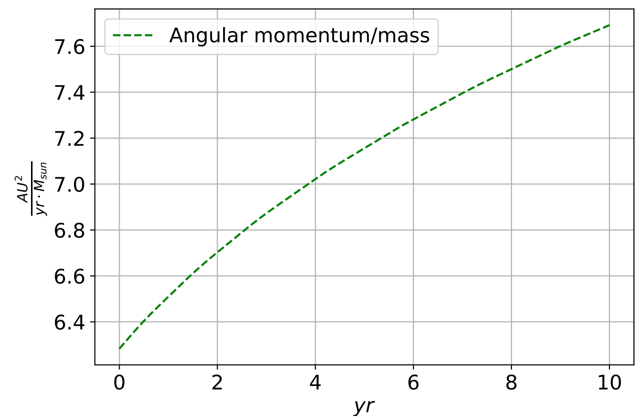

FIG. 3. Euler's method: Angular momentum per mass ($\vec{r} \times \vec{v}$) for the Earth. 10 year simulation time with time step of 0.001 year. Initial conditions: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

2. *Velocity Verlet method*

Figure 4, 5 and 6 shows the position, energy and angular momentum of the Earth, respectively. These simulations had the same initial conditions as for Euler's method, but were implemented with the velocity Verlet method.

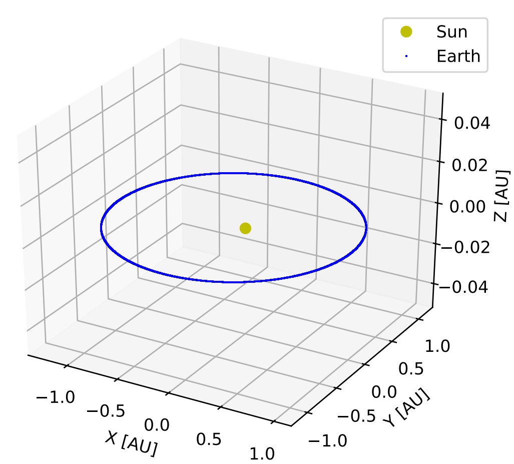

FIG. 4. Velocity Verlet method: Positions of the Earth. 10 year simulation time with time step of 0.001 year. Initial conditions: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

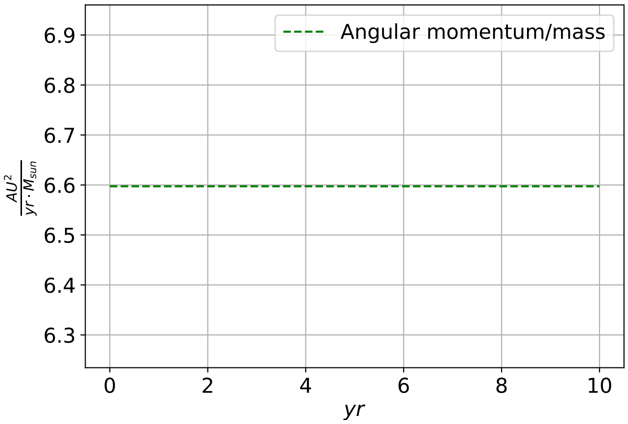

FIG. 6. Velocity Verlet method: Angular momentum per mass ($\vec{r} \times \vec{v}$) for the Earth. 10 year simulation time with time step of 0.001 year. Initial conditions: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

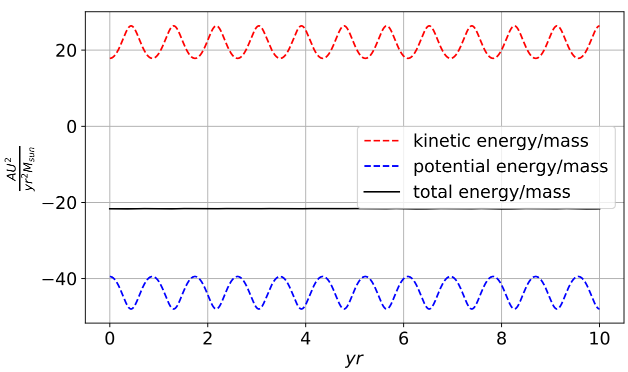

FIG. 5. Velocity Verlet method: The kinetic, potential and total energy per mass of the Earth. 10 year simulation time with time step of 0.001 year. Initial conditions: $x_0 = 1$ AU and $V_{y,0} = 2\pi$ AU/year.

3. *Escape Velocity*

Figure 7 shows Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 8.88$ AU/year. Figure 8 shows Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 8.89$ AU/year. The velocity Verlet method was used as algorithm for the simulations, and the time duration for the simulations were 10 years, with a time step of 0.001 year.

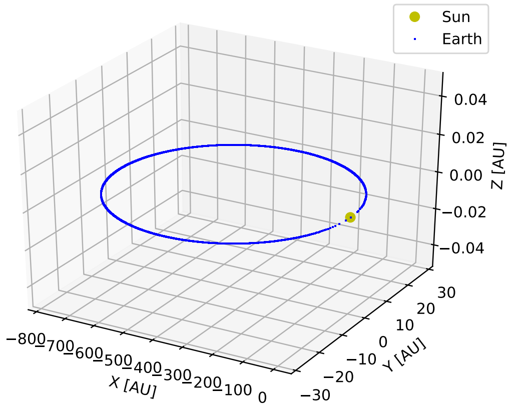

FIG. 7. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 8.88$ AU/year. Total time of simulations: 10 years, time step 0.001 year.

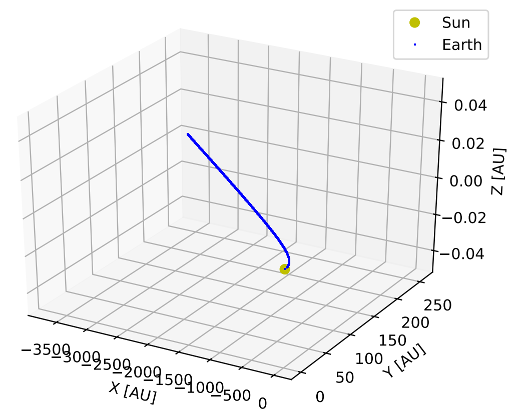

FIG. 8. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 8.89$ AU/year. Total time of simulations: 10 years, time step 0.001 year.

4. *Different exponents of r in Newton's law of gravitation*

The velocity Verlet method was used for these simulations with a simulation time of 10 years and a time step of 0.001 year. The initial conditions were $x_0 = 1$ AU and $v_{y,0} = 2.1\pi$ AU/year. Figure 9 shows Earth's trajectory for $\beta = 2.01$.

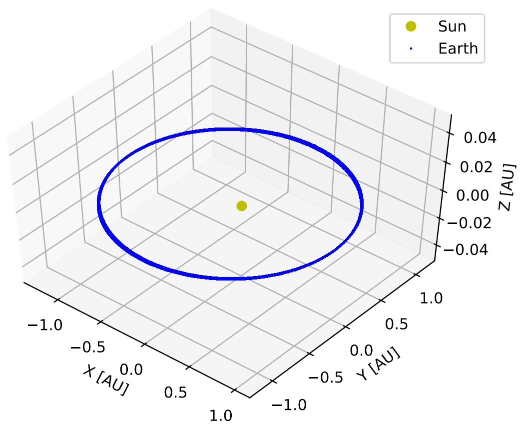

FIG. 9. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 2.1\pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.01$

Figure 10 shows Earth's trajectory for $\beta = 2.1$.

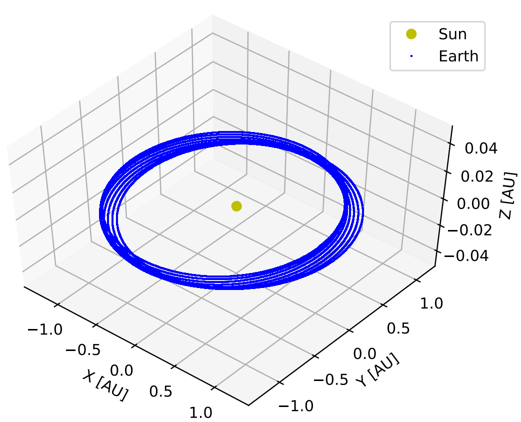

FIG. 10. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 2.1\pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.1$

Figure 11 shows Earth's trajectory for $\beta = 2.436$.

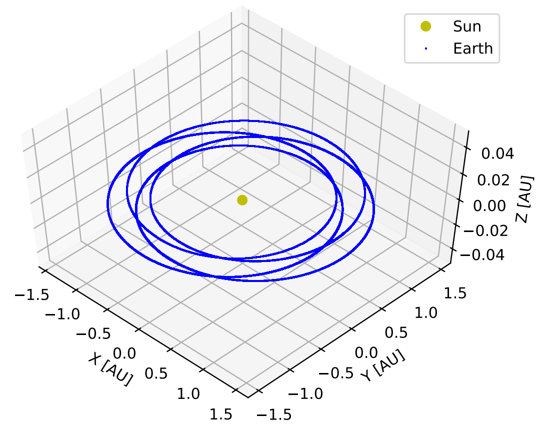

FIG. 11. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 2.1\pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.436$

Figure 12, shows Earth's trajectory for $\beta = 2.666$.

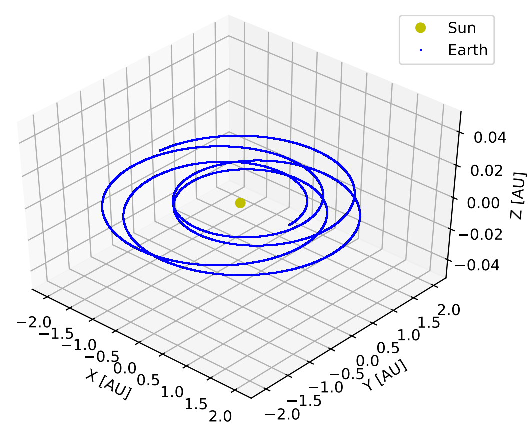

FIG. 12. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 2.1\pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.666$

Figure 13, shows Earth's trajectory for $\beta = 2.99$.

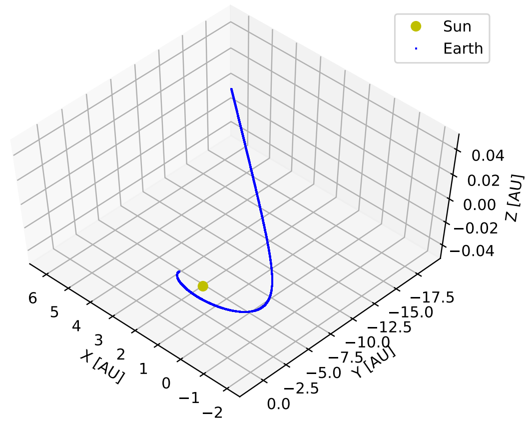

FIG. 13. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 2.1\pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.99$

Figure 14, shows Earth's trajectory for $\beta = 2.99$ and initial velocity $1.9\pi AU/yr$.

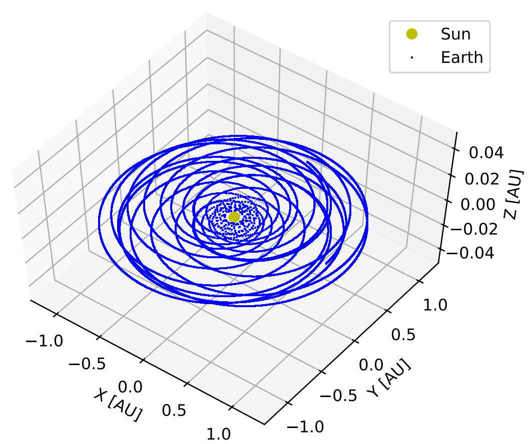

FIG. 14. The velocity Verlet method: Earth's positions with initial conditions $x_0 = 1$ AU, $v_{y,0} = 1.9pi$ AU/year. Total time of simulations: 10 years, time step 0.001 year. $\beta = 2.99$

B. Three body system, with the Sun, Earth and Jupiter

The following simulations were done with three celestial bodies: the Sun, Earth and Jupiter. For all simulations, the velocity Verlet method was used. All simulations were done with a time duration of 15 years and a time step of 0.001 year. The results for the three body system is split in two categories: the Sun is kept fixed at the origin, and the Sun is allowed to move. The mass of Jupiter was varied to be $1M_{Jupiter}$, $10M_{Jupiter}$ and $1000M_{Jupiter}$. Initial conditions were obtained from NASA, at Ocotber 4th.

1. *Sun fixed at the origin*

Figure 15 shows positions of the three bodies, with Sun fixed.

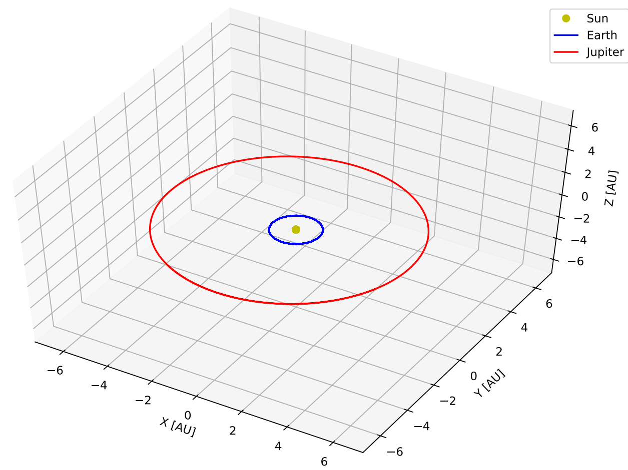

FIG. 15. The velocity Verlet method: three body system with the Sun fixed at the origin. Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $1M_{Jupiter}$.

Figure 16 shows positions of the three bodies, with Sun fixed. The mass of Jupiter is increased to $10M_{Jupiter}$.

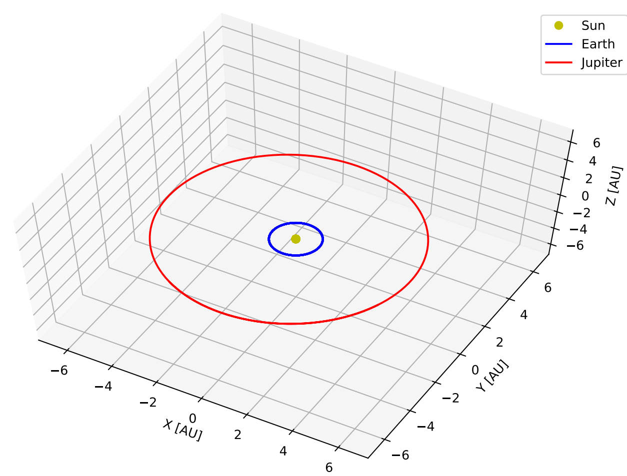

FIG. 16. The velocity Verlet method: three body system with the Sun fixed at the origin. Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $10M_{Jupiter}$.

Figure 17 shows positions of the three bodies, with Sun fixed. The mass of Jupiter is increased to $1000M_{Jupiter}$.

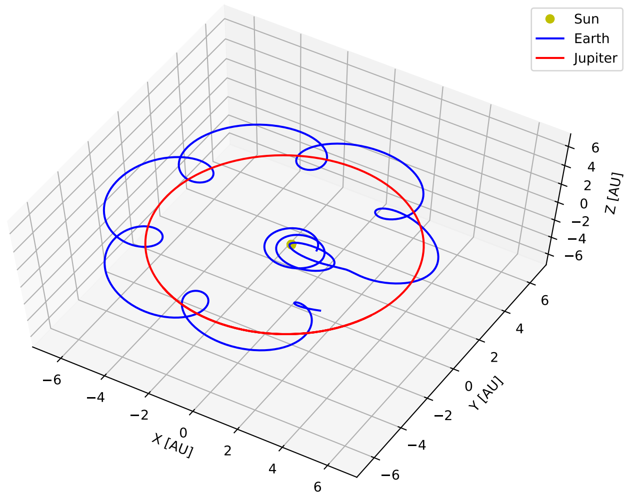

FIG. 17. The velocity Verlet method: three body system with Sun fixed at the origin. Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $1000M_{Jupiter}$.

	Total energy $[AU^2/(yr^2 M_\odot)]$		Relative error [%]
Jupiter mass	Initial	Final	
$1M_J$	−0.01081	−0.01136	5.1
$10M_J$	−0.1065	−0.1120	5.2
$1000M_J$	−10.63	−11.18	5.2

TABLE II. Table listing the initial and final total energies $E = E_K + E_P$ in the initial values and at the final stage of the simulation. The simulation lasted 15 years with a time step of 0.0001 years and the sun was not allowed to move.

2. *Sun allowed to move*

Figure 18 shows positions of the three bodies and the Sun is allowed to move.

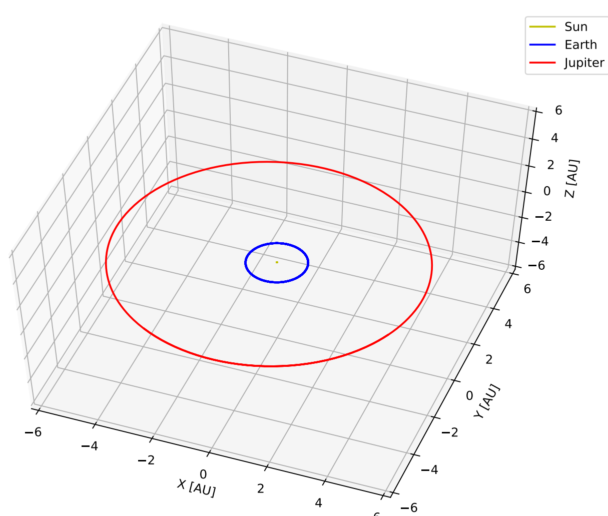

FIG. 18. The velocity Verlet method: three body system (Sun allowed to move). Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $1M_{Jupiter}$.

Figure 19 shows positions of the three bodies and the Sun is allowed to move. The mass of Jupiter is increased to $10M_{Jupiter}$.

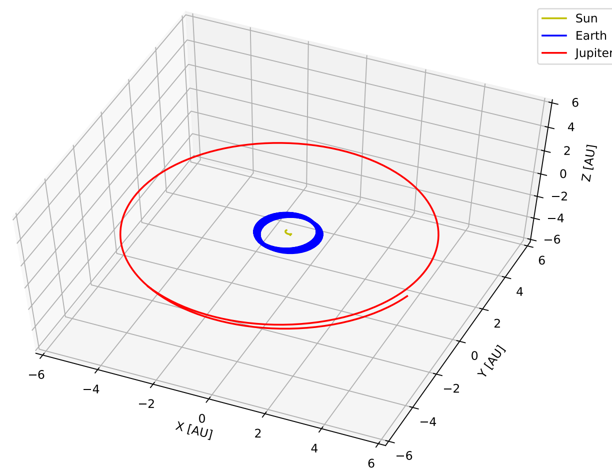

FIG. 19. The velocity Verlet method: three body system (Sun allowed to move). Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $10M_{Jupiter}$.

Figure 20 shows positions of the three bodies and the Sun is allowed to move. The mass of Jupiter is increased to $1000M_{Jupiter}$.

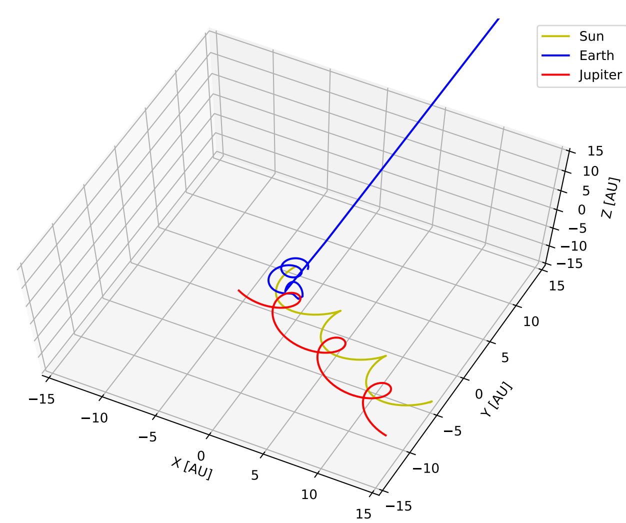

FIG. 20. The velocity Verlet method: three body system (Sun allowed to move). Total time of simulations: 15 years, time step 0.001 year. Mass of Jupiter: $1000M_{Jupiter}$.

	Total energy $[AU^2/(yr^2M_\odot)]$		Relative error [%]
Jupiter mass	Initial	Final	
$1M_J$	-0.01079	-0.01131	4.8
$10M_J$	-0.1064	-0.1125	5.7
$1000M_J$	-10.62	-10.90	2.6

TABLE III. Table listing the initial and final total energies $E = E_K + E_P$ in the initial values and at the final stage of the simulation. The simulation lasted 15 years with a time step of 0.0001 years and the sun was allowed to move. The unit of the energies listed is in $AU^2/(yr^2M_\odot)$

C. Full solar system

The full solar system was simulated using the velocity Verlet method with initial conditions obtained from Nasa at October 4th. There are 10 Celestial bodies in the plots: the Sun, the eight planets and Pluto. Figure 21 shows the full simulation of the solar system for a time duration of 248 years, with a time step of 0.001 year.

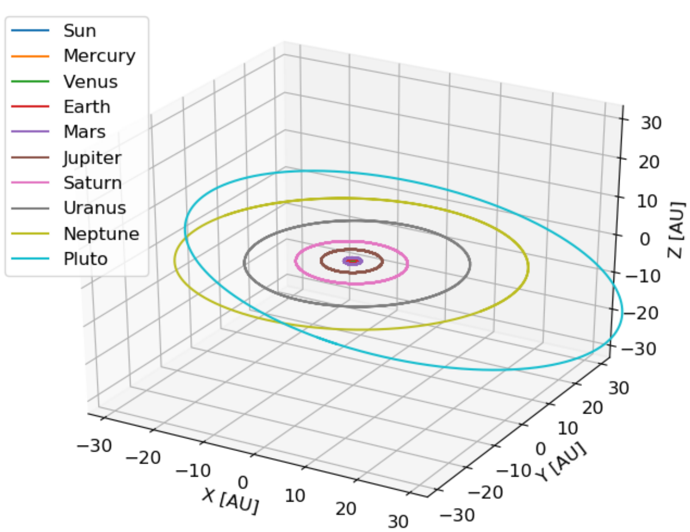

FIG. 21. Full solar system simulated using the velocity Verlet method for a time duration of 248 years (roughly one Plutonian year) with a time step of 0.001 year.

Figure 22 shows a simulation of the full solar system, but only the Sun and the inner four planets are showns (Mercury, Venus, Earth and Mars). This simulation was done for a time duration of 2 years with a time step of 0.0001 year.

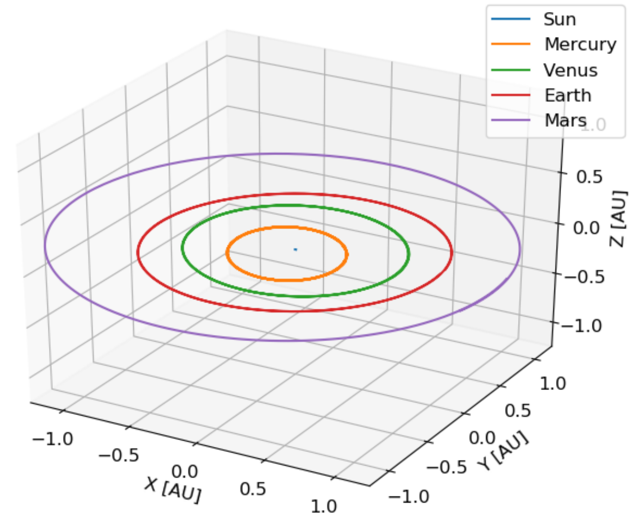

FIG. 22. Full solar system simulated, but only the Sun and the four innermost planets are shown. The simulation was done using velocity Verlet method for a time duration of 2 years with a time step of 0.0001 year.

D. Perihelion precession of Mercury's orbit

Figure 23 shows all angles and times when Mercury was at perihelion for two separate simulations with and without general relativity effects on the gravitational force.

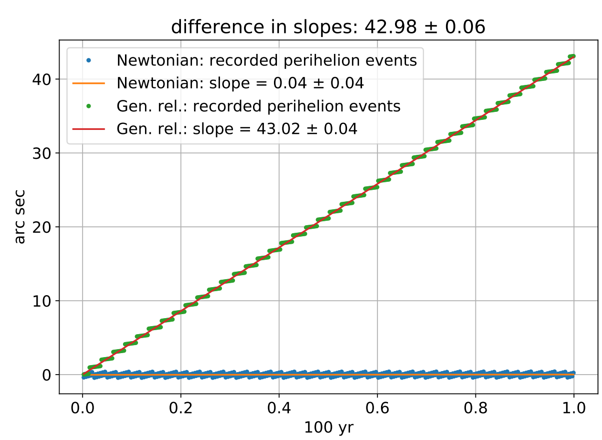

FIG. 23. Data points of angle of Mercury's position (taken from x-axis) and times at perihelion events for two simulations of pure Newtonian gravity and general relativity. A linear fit by least squares method for both data sets. There were 415 perihelion events for this simulation over 100 years, with a time step of 10^{-7} years.

The perihelion precession of Mercury's orbit (taken as the differences of the slopes with and without general relativity) around the Sun is $42.98 \pm 0.06''/century$ due to general relativity effects.

V. DISCUSSION

A. Earth-Sun system

The comparison between the two numerical methods was conclusive: In exchange for more floating point operations, the velocity Verlet method promises a more accurate result. Neither the total energy nor angular momentum was conserved using Euler's method. These values deviated from their initial state with time as illustrated by figures 2 and 3, which is a crucial error in simulating astronomical systems. This was met by the impressively consistent velocity Verlet method, as was illustrated by the energy and angular momentum graphs in figures 5 and 6. Although both had minor numerical deviations in the short time domain, the initial parameters were conserved in the long run.

The numerical escape velocity was found by trial-and-error to be in the range of $v_e \in [8.88, 8.89] AU/yr$. The Earth clearly does not escape for an initial velocity of $8.88 AU/yr$, but seems to escape for $8.89 AU/yr$ (as shown in figures 7 and 8). However, this is a limited analysis due to the simulation duration being only 10 years (it may return after a longer time). The dramatic difference between Earth's orbit in figure 7 and 8 indicates that the numerical escape velocity is between $8.88 AU/yr$ and $8.89 AU/yr$, which is consistent with the previously derived theoretical value of $v_e = 8.8858 AU/yr$.

Varying the values of β in the gravitational force (equation 13) produced different trajectories of Earth's orbit. Recalling the criteria for escape described in equation 10, the energy needed to escape should decrease for increasing values of β. In other words, the consequence of increasing β whilst keeping the initial velocity v constant is that kinetic energy E_K might be large enough for the satellite to escape.

The initial velocity used for these simulations was $2.1\pi AU/yr$, which in the Newtonian case would produce a closed elliptical orbit. For $\beta = 2.01$, as illustrated by figure 9, there is a slight precession of Earth's orbit to be seen. The precession then becomes more noticeable as β increases, as shown in figure 10 (where $\beta = 2.1$). A special case is illustrated by figure 11 (where $\beta = 2.436$): The orbit seems to precess with a rate such that after four orbits, the Earth is roughly back at it's starting point.

For larger values of β (figures 12 and 13), the force of gravity clearly diminishes in strength. In fact, for $\beta = 2.99$ all initial velocities greater or equal to 2.1π cause the earth to escape the gravitational pull of the Sun. This is not a surprising result considering that the criteria for escape velocity changes with β as mentioned previously. However, for $\beta = 2.99$ and an initial velocity $v_0 = 1.9\pi AU/yr$, the Earth does not escape (illustrated by figure 14). This can be explained by the new expression for potential energy 16, where the escape velocity at $1AU$ approaches $2\pi AU/yr$ as $\beta \to 3$. When β is higher than 2 and the initial velocity is less than escape velocity, the behaviour of Earth's trajectory can be summarized as forming a disc (due to precession over time). This disc will have a thickness described as difference between perihelion and aphelion. The thickness of this disc increases as β creeps towards 3.

B. Three body system

1. *Fixed Sun*

The behaviour of the three body system consisting of the Sun, Earth and Jupiter seems to be consistent with observations. The orbits of both planets are closed and approximately circular, which is to be expected of the simulation as it implies that both the total energy and angular momentum is conserved for both planets. Manipulating the mass of Jupiter caused the system to behave in interesting manners. The case of $10 \cdot M_{Jupiter}$ produced no significant differences, but $1000 \cdot M_{Jupiter}$ caused the Earth to be pulled from its orbit as illustrated in figure 17. The Earth followed an unstable trajectory around the Sun due to the gravitational pull from Jupiter, and after a few years, Earth's trajectory is close enough to the Sun to be slingshotted towards Jupiter. The Earth is subsequently caught into a fascinating orbit around Jupiter for a couple of years. The orbit resembles that of a moon in an elliptical orbit around the central planet orbiting a central star: A motion which is chaotic and cannot be predicted analytically.

2. *Sun allowed to move*

Allowing the Sun to move (no longer fixed in the origin) did not have any significant impact on the simulation, as illustrated by the trajectories of the three bodies in figure 18. This is good to see as it means that the previous approximation of fixing it in place was appropriate to the case. However, the Sun does start moving for larger Jupiter masses $10 \cdot M_{Jupiter}$, as can be seen in figure 19. This is to be expected as Jupiter now has a significantly stronger gravitational influence on the bodies surrounding it. This change also causes both the Earth and Jupiter to have disturbances in their orbits (no longer closed). This is taken to the extreme for the next case of $1000 \cdot M_{Jupiter}$. The dominating motion in this situation is the Sun and Jupiter orbiting each other (as their masses now are very close to each other). Earth however, is a much lighter object and did at one point pass close to Jupiter. This resulted in the Earth being slingshotted with a tremendous velocity away from the system. The Sun and Jupiter continue to oscillate in an interesting pattern away from the origin.

3. *Stability of velocity Verlet method*

The initial total mechanical energy of the velocity Verlet simulation was found to not be equal to the final.

The total mechanical energy of the three body system when the sun was fixed had a consistent relative error of 5.1 to 5.2% as shown in table II. Likewise, the total mechanical energy of the three body system when the sun was allowed to move had a relative error between 2.6% and 5.7% as shown in table III.

These results are likely a consequence of numerical errors due to an insufficiently small time step. Another explanation is that the initial and final energy values were unlucky samples of the short time domain energy oscillations described earlier.

C. Full solar system

The simulation of the full solar system produced results which are accurate with astronomical observations. All of the planet's trajectories as shown in figure 21 behaved as expected. Most of the orbits were closed elliptical orbits in one geometrical plane, except for Pluto's orbit which is slightly tilted. The innermost planets such as Mercury did precess over time, which is a consequence of the other planet's gravitational forces. Figure 22 shows the four innermost planets and the Sun. This plot was included to easier study these orbits as they are very small in the scale of the entire solar system. Figure 22 also shows that the orbits of Mercury and Mars have an ellipse eccentricity which is easily seen, whereas Earth and Venus have close to circular orbits. Finally, the motion of the Sun due to the other planetary forces is unnoticeable in these illustrations.

D. Perihelion precession of Mercury's orbit

The pure Newtonian gravity force simulation gave a small perihelion precession, as illustrated by the Newtonian slope in figure 23. Theoretically, it is expected to be a closed elliptical orbit for this system (no precession), but the small perihelion precession is likely a result of systematical numerical errors.

There is reason to expect the same systematical error for the general relativity case. One approach to solving this problem is to simply take the difference between the general relativity and the pure Newtonian simulation results.

Hopefully, this solution removes the systematical Newtonian errors (leaving only random numerical errors) in the final general relativity value. This is not certain as the precession in the Newtonian case might have been due to random errors. The result is then $42.98 \pm 0.06''/100yr$. This is very consistent with (Park et. al.)[2], which estimates that general relativity effects (Schwarzschild like) causes a precession of $42.9799 \pm 0.0009''/100yr$. The differences in observations and Newtonian theory (which differed with $43''/100yr$), indicates that these observations are well explained by the theory of general relativity. One last notable mention is that this is precisely what was first proposed by Einstein in 1916 [1].

VI. CONCLUSION

The velocity Verlet integration method was found to be superior to Euler's method in that the conservation of energy and angular momentum was taken into account at the cost of more FLOPS and almost double the run-time. The velocity Verlet method was therefore a suitable algorithm for simulations of the three body problem and full solar system. The behaviour of the three-body system of the Earth, Sun and Jupiter produced trajectories (and relative errors less than 5.7%) which affirm that the simulation script was behaving properly. The perihelion precession of Mercury's orbit was found to be $42.98 \pm 0.06''/100yr$ due to Schwarzschild like general relativity effects, which indicates that the observed precession is well explained by the theory of general relativity (along with classical physics).

REFERENCES

[1] Einstein, A. (1916). Die grundlage der allgemeinen relativitätstheorie. *Annalen der Physik*, 354(7):769–822.

[2] Park, R. S., Folkner, W. M., Konopliv, A. S., Williams, J. G., Smith, D. E., and Zuber, M. T. (2017). Precession of mercury's perihelion from ranging to the messenger spacecraft. *The Astronomical Journal*, 153(3):121.

[3] Soldovieri, T. and Muñoz, (2003). Obtaining the gravitational force corresponding to arbitrary spacetimes. the schwarzschild's case. 49:271–275.

[4] Squires, G. L. (2001). *Practical Physics*. Cambridge University Press, 4 edition.

APPENDIX A: VELOCITY VERLET DERIVATION

Following are the Taylor-expansions for a positional function $x(t)$ and a corresponding velocity function $x'(t) = v(t)$ for a small step size $(t+h)$:

$$x(t+h) = x(t) + h \cdot x^{(1)}(t) + \frac{h^2}{2!} \cdot x^{(2)}(t) + \mathcal{O}(h^3) \quad (29)$$

$$v(t+h) = v(t) + h \cdot v^{(1)}(t) + \frac{h^2}{2!} \cdot v^{(2)}(t) + \mathcal{O}(h^3) \quad (30)$$

The error here has an order of magnitude proportional to the step size $\mathcal{O}(h^3) \propto h^3$, and the indexing $x^{(n)}$ indicates the n-th derivative (in relation to time) of the function x. Discretization $x(t_i \pm h) = x_{i\pm1}$, where $t_i = t_0 + i \cdot h$, and including the velocity and acceleration indexing (using the relations between position, velocity and acceleration $a = v^{(1)} = x^{(2)}$) returns (not including the error):

$$x_{i+1} = x_i + h \cdot v_i + \frac{h^2}{2!} \cdot a_i \quad (31)$$

$$v_{i+1} = v_i + h \cdot a_i + \frac{h^2}{2!} \cdot v_i^{(2)} \quad (32)$$

The expression for $v_i^{(2)}$ is unknown, and can be expressed by use of another Taylor expansion (now expanding the expression of $v_i^{(1)} = a_i$):

$$v_{i+1}^{(1)} = v_i^{(1)} + h \cdot v_i^{(2)} + \mathcal{O}(h^2) \quad (33)$$

$$v_i^{(2)} = (a_{i+1} - a_i - \mathcal{O}(h^2))/h \quad (34)$$

Inserting this expression into 32 produces the following expression for the velocity:

$$v_{i+1} = v_i + h \cdot a_i + \frac{h^2}{2!} \cdot (a_{i+1} - a_i - \mathcal{O}(h^2))/h \quad (35)$$

The error $\mathcal{O}(h^2)$ included in equation 35 is multiplied with h^2/h, so it becomes proportional to $\mathcal{O} \propto h^3$, the same error proportionality described in equations 29 and 30. This means that it is redundant and can be combined with the previous error expression.
The *Velocity Verlet* algorithm is then given by:

$$x_{i+1} = x_i + h \cdot v_i + \frac{h^2}{2} a_i + \mathcal{O}(h^3) \quad (36)$$

$$v_{i+1} = v_i + \frac{h}{2}(a_{i+1} + a_i) + \mathcal{O}(h^3) \quad (37)$$

APPENDIX B: UNCERTAINTY OF LINEAR FIT BY LEAST SQUARES METHOD

For a data set of n x and y data points, the least squares method for a first degree polynomial is given by:

$$y_{fit} = mx + c \quad (38)$$

To find the uncertainties Δm and Δc, the scalars D, E and F are introduced:

$$D = \sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2 \quad (39)$$

$$E = \sum x_i y_i - \frac{1}{n}\sum x_i \sum y_i \quad (40)$$

$$F = \sum y_i^2 - \frac{1}{n}\left(\sum y_i\right)^2 \quad (41)$$

Where (x_i, y_i) are the data points. These parameters allow us to calculate the uncertainties Δm and Δc in the following fashion:

$$(\Delta m)^2 \approx \frac{1}{n-2}\frac{DF - E^2}{D^2} \quad (42)$$

$$(\Delta c)^2 \approx \frac{1}{n-2}\left(\frac{D}{n} + \bar{x}^2\right)\frac{DF - E^2}{D^2} \quad (43)$$

n is the number of data points and $\bar{x}$ is the mean of all x_i values. For the data set (time, angle) of perihelion events of Mercury's orbit, the slope m will be the rate of precession over time with uncertainty Δm. This uncertainty is a statistical derivation method obtained from (Squires, G, L 2001)[4].