

FYS3150 Report

Project 3 : Gravitational N -Body Simulation

Nils Johannes Mikkelsen
(Dated: October 24, 2018)

Gravitational N -body systems are analysed using the initial condition numerical integration methods Forward-Euler and Velocity-Verlet. The Velocity-Verlet method is found easily superior to the low-order Forward-Euler method. The systems studied include the Newtonian systems: *Earth-Sun*, *Earth-Jupiter-Sun* and the complete *Solar System*, and the relativistic system *Mercury-Sun*. The focus lies on circular orbits, perturbations of circular orbits as well as escape velocity, numerical stability and perihelion precession due to relativistic effects.

ABOUT PROJECT 3

This is a report for Project 3 in FYS3150 Computational Physics at UiO, due October 24st, 2018. [2] The project description was accessed October 15th, 2018 with the following web address:

<http://compphysics.github.io/ComputationalPhysics/doc/Projects/2018/Project3/pdf/Project3.pdf>

All material written for this report can be found in this GitHub repository:

<https://github.com/njmikkelsen/comphys2018/tree/master/Project3>

I. INTRODUCTION

The aim of this project is to use numerical integration in order to study the gravitational N -body problem, in particular our Solar System. The problem will be dissected into smaller components in order to isolate the effects between the gravitational objects. The first system (or component) to be studied is the isolated Earth-Sun system: it serves mainly as a classification of the numerical methods and to study how gravity behaves with respect to numerical algorithms. Next, Jupiter is added to the Earth-Sun system so that the perturbations of planetary orbits may be studied in isolation. Then, finally all planets are added to the system and simulated with all planet-planet gravitational effects included.

As a final challenge, a relativistic correction is added to the gravitational acceleration such that the observed perihelion precession of Mercury may be studied. The perihelion precesses by as little as 43" per century, meaning this system serves as a good estimator for whether the numerical algorithms are stable in the long run.

II. THEORY

A. On The Physics of Gravitational Systems

The main focus in Project 3 lies on 4 different gravitational systems: Three Newtonian systems with 2, 3 and N bodies, and one relativistic system with 2 bodies.

1. The general problem statement

Assuming an inertial frame of reference with J rigid objects not in contact, the net force on object $j \in \{1, \dots, J\}$ may be decomposed into gravitational effects and all other effects:

$$\mathbf{F}_{\text{net}}^j = \sum_{k \neq j}^J \left(\mathbf{G}^{kj} + \mathbf{F}_{\text{add}}^j \right) \quad (1)$$

where \mathbf{G}^{kj} is the gravitational force from object k on j and $\mathbf{F}_{\text{add}}^j$ includes all additional forces on object j such as electromagnetic radiation etc. The contributions from $\mathbf{F}_{\text{add}}^j$ are miniscule compared to gravity and will therefore be ignored in this report.

Using Newtonian gravity, the gravitational force \mathbf{G}^{kj} is

$$\mathbf{G}^{kj} = GM_j M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (2)$$

where G is the *universal gravitational constant*, M_j and M_k are the masses of objects j and k respectively, and $\delta \mathbf{r}^{kj} = \mathbf{r}^k - \mathbf{r}^j$ is the position of object k relative to the position of object j . Following Newton's second law, the gravitational acceleration \mathbf{g}^{kj} experienced by object j due to \mathbf{G}^{kj} is simply

$$\mathbf{g}^{kj} = \frac{\mathbf{G}^{kj}}{M_j} = GM_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (3)$$

Therefore in a Newtonian system, the net gravitational acceleration experienced by object j due to the other objects is given by rewriting equation (1) using Newton's second law and inserting (3):

$$\mathbf{g}^j = G \sum_{k \neq j}^J M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (4)$$

Invoking General Relativity, the concept of a gravitational force is replaced by the curvature of spacetime. However, the Newtonian jargon may be continued by introducing a relativistic correction to the Newtonian expression for gravitational acceleration:

$$\mathbf{g}_{\text{rel}} = \mathbf{g}_{\text{N}} \left[1 + 3 \frac{|\mathbf{r} \times \mathbf{v}|^2}{|\mathbf{r}|^2 c^2} \right] \quad (5)$$

where \mathbf{g}_{rel} and \mathbf{g}_N is the relativistic and Newtonian acceleration, \mathbf{r} and \mathbf{v} are the position and velocity, and c is the speed of light. Hence, the relativistic version of equation (4) is

$$\mathbf{g}^j = G \left[1 + 3 \frac{|\mathbf{r}^j \times \mathbf{v}^j|^2}{|\mathbf{r}^j|^2 c^2} \right] \sum_{k \neq j}^J M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (6)$$

In conclusion, in order to simulate the movement of Newtonian/relativistic gravitational objects, the following set of simultaneous second order differential equations must be solved:

Newtonian:

$$\frac{d^2}{dt^2} \mathbf{r}^j = G \sum_{k \neq j}^J M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (7a)$$

Relativistic:

$$\frac{d^2}{dt^2} \mathbf{r}^j = G \left[1 + 3 \frac{|\mathbf{r}^j \times \mathbf{v}^j|^2}{|\mathbf{r}^j|^2 c^2} \right] \sum_{k \neq j}^J M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^3} \quad (7b)$$

for all $j \in \{1, \dots, J\}$, provided an appropriate set of initial conditions $\{t_0, \mathbf{r}^j(t_0), \mathbf{v}^j(t_0)\}_j$ for each object.

2. Experimental gravity

Seeing that the $1/r^2$ dependence of the gravitational acceleration is particular to Newtonian gravity, it is tempting to analyse a generalised dependence $1/r^\beta$. Clearly for β values slightly greater than 2, the gravitational pull from the Sun on the planets would just fail to stabilise the planets' kinetic energy, leaving the planets slowly drift away. For β values slightly less than 2, the orbits are likely to remain stable despite loosing their original approximately-elliptic shape. Introducing such a generalised dependence is easy, instead of the $1/|\delta \mathbf{r}^{kj}|^3$ dependence in (3), the gravitational acceleration would instead have a $1/|\delta \mathbf{r}^{kj}|^{\beta+1}$ dependence:

$$\mathbf{g}^{kj}(\beta) = G M_k \frac{\delta \mathbf{r}^{kj}}{|\delta \mathbf{r}^{kj}|^{\beta+1}} \quad (8)$$

The equivalent net " β -Newtonian" gravitational acceleration is then found as before: $\mathbf{g}^j(\beta) = \sum_{k \neq j}^J \mathbf{g}^{kj}(\beta)$.

3. Units

Knowing that the simulations will take place on the scale of the solar system, it is computationally expensive to use SI units. The simulations will therefore be run using the so-called astronomical, or au, system of units. This system of units is inspired by Earth's orbit about the Sun and thus defines the unit of time and space according to the period and average radius of the orbit. The unit of

length is the astronomical unit au, which today is defined exactly as $1 \text{ au} = 149\,597\,870\,700 \text{ m}$. The unit of time is the Julian year, i.e. $365.25 \text{ days} = 31\,557\,600 \text{ s}$.

The astronomical units is not a strictly closed system in that several units of mass are used, this report will use the most computationally friendly one: the solar mass $1 M_\odot = 1.988500 \cdot 10^{30} \text{ kg}$.

From the unit definitions of length and time, the speed of light in a vacuum c can be expressed in astronomical units as

$$c = 299792458 \text{ m/s} \cong 63241.08 \text{ au/yr} \quad (9)$$

Using the unit definitions for length, time and mass, the universal gravitational constant can be expressed in astronomical units as

$$G = 6.67408 \cdot 10^{-11} \text{ m}^3 \text{ s}^{-2} \text{ kg}^{-1} \quad (10a)$$

$$\cong 39.4774 \text{ au}^3 \text{ yr}^{-2} M_\odot^{-1} \quad (10b)$$

It is not a coincidence that the value for G in au units is approximately equal to $4\pi^2$, Kepler's 3rd Law of two-body orbits states that the semi-major axis a of an elliptical orbit is proportional to the square of the orbital period T :

$$\frac{a^3}{T^2} = \frac{G(M_1 + M_2)}{4\pi^2} \quad (11)$$

where M_1 and M_2 are the masses of the bodies involved. For the Earth-Sun system, $a^3/T^2 = 1 \text{ au}^3/\text{yr}^2$ and $M_\odot + M_{\text{Earth}} \approx M_\odot$ so that $G = 4\pi^2 \text{ au}^3 \text{ yr}^{-2} M_\odot^{-1}$.

4. Circular orbit

Assuming a single gravitational body with mass m is orbiting a massive object with mass M whose motion is negligible, the gravitational acceleration would equal the centripetal acceleration of the body:

$$m \frac{v_{\text{circ}}^2}{R} = \frac{GmM}{R^2}$$

where v_{circ} is the orbital speed of orbiting body. When rearranged, this yields the orbital speed as a function of the orbit radius and mass of the massive object:

$$v_{\text{circ}} = \sqrt{\frac{GM}{R}} \quad (12)$$

5. Planetary escape velocity

In the unlikely event that a planet and the Sun is isolated, and the planet is given escape velocity in the direction radially outwards from the Sun, conservation of energy would imply that the planet's initial kinetic energy would equal the total work done by the Sun on the

planet. The work done needed for the planet to escape the Sun (meaning travel infinitely far away without external forces acting on Earth during the journey) starting from a distance R is given by

$$\begin{aligned} W_{\text{esc}} &= - \int_{C_R} d\mathbf{r} \frac{-GM_p M_\odot}{|\mathbf{r}|^3} \mathbf{r} = GM_p M_\odot \int_R^\infty \frac{dr}{r^2} \\ &= -GM_p M_\odot \frac{1}{r} \Big|_R^\infty = -GM_p M_\odot \left(0 - \frac{1}{R}\right) \\ W_{\text{esc}} &= \frac{GM_p M_\odot}{R} \end{aligned} \quad (13)$$

where M_p is the planet's mass and C_R is a straight curve starting a distance R from the origin and directed radially outwards from the Sun. The initial kinetic energy of the planet is $K_{\text{esc}} = \frac{1}{2} M_p v_{\text{esc}}^2$, hence the escape velocity of the planet is given by

$$W_{\text{esc}} = K_{\text{esc}} \implies v_{\text{esc}} = \sqrt{2 \frac{GM_\odot}{R}} \quad (14)$$

6. The perihelion precession of the planets

The French mathematician Urbain Le Verrier discovered in the mid-19th century that the perihelion (the point on a planetary orbit that is closest to the Sun) of Mercury, in direct violation with contemporary Newtonian celestial mechanics, precessed about Solar System's planar axis. The angular velocity of the precession, which was not fully explained until the advent of general relativity, measures a tiny 43" per century. Using the relativistic correction in (6), the precession should be reproducible. The distance from Mercury to the Sun and the orbital speed of Mercury at perihelion is 0.3075 AU and 12.44 AU/yr respectively.

B. Numerical integration - Initial Value Problems

The following theory is based on Lectures in FYS-3150. [1]

The problem at hand (equation (7)) is a so-called differential equation initial value problem. An analytic solution does in general not exist, meaning numerical methods are necessary. This report will introduce two integration methods (or schemes), the Forward-Euler method and the Velocity-Verlet method.

1. Preliminary

In order to maintain some level of generality, both methods will be discussed in the absence of knowledge of the gravitational acceleration. Newton's second law of motion is second order ordinary differential equation, which in the most general case may be written as

$$\mathbf{f}''(t) = \frac{d^2 \mathbf{f}}{dt^2}(t, \mathbf{f}(t), \mathbf{f}'(t)) \quad (15)$$

where $\mathbf{f}(t)$ is the solution in question and $d^2 \mathbf{f}/dt^2$ is a known function. To avoid having to write out $\mathbf{f}(t)$, its derivatives, etc., the following will introduce a simplified notation.

Let t_i denote the $(i+1)^{\text{th}}$ element in the linearly spaced set of t values:

$$\{t_i = t_0 + ih : i \in \mathbb{Z}^+, t_0 \in \mathbb{R}, h \in \mathbb{R}^+\} \quad (16)$$

Furthermore, define $\mathbf{f}_i = \mathbf{f}(t_i)$ and let prime-notation, e.g. \mathbf{f}' , denote taking the derivative with respect to t .

Using this notation, the first- and second-order Taylor expansion of \mathbf{f} about t_i , evaluated at t_{i+1} , is simply

$$\mathbf{f}_{i+1} = \mathbf{f}_i + h\mathbf{f}'_i + \mathcal{O}(h^2) \quad (17a)$$

$$\mathbf{f}_{i+1} = \mathbf{f}_i + h\mathbf{f}'_i + \frac{h^2}{2}\mathbf{f}''_i + \mathcal{O}(h^3) \quad (17b)$$

In case \mathbf{f} represents the position vector \mathbf{r} , the derivatives \mathbf{r}' and \mathbf{r}'' are simply the velocity and acceleration vectors \mathbf{r} and \mathbf{a} . In case \mathbf{f} represents the velocity vector \mathbf{v} , the derivatives \mathbf{v}' and \mathbf{v}'' represent the acceleration and the derivative of the acceleration vectors \mathbf{a} and \mathbf{a}' .

2. Forward-Euler

The Forward-Euler integration scheme is a low-order, somewhat inaccurate, integration method originating from some of the earliest and most tedious attempts of numerical calculus. The method has been included for exactly this reason: namely to highlight the difference in accuracy between different integration methods.

The Forward-Euler scheme is easy to outline, it follows exactly from the first-order Taylor expansions of \mathbf{r} and \mathbf{v} :

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h\mathbf{v}_i + \mathcal{O}(h^2) \quad (18a)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + h\mathbf{a}_i + \mathcal{O}(h^2) \quad (18b)$$

Note that the order-terms are ignored in the integration. Each step in a Forward-Euler integration carries an error of order h^2 , thus with N steps the global error made in an integration is equal to $N\mathcal{O}(h^2)$. It is customary to let t run from $t = 0$ to some $t = t_N = Nh$ so that the global error may be written as $t_N\mathcal{O}(h)$.

3. Velocity-Verlet

The Velocity-Verlet method is a second-order integration scheme that, as the name implies, focuses on the velocity. The method is energy-conservative in that if \mathbf{r} and \mathbf{v} represents the position and velocity of an object in an inertial frame of reference, its total mechanical energy would be conserved in time.

The Velocity-Verlet method follows from the second-order Taylor expansions of \mathbf{r} and \mathbf{v} :

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h\mathbf{v}_i + \frac{h^2}{2}\mathbf{a}_i + \mathcal{O}(h^3) \quad (19a)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + h\mathbf{a}_i + \frac{h^2}{2}\mathbf{a}'_i + \mathcal{O}(h^3) \quad (19b)$$

The term \mathbf{a}'_i is now approximated using a first-order Taylor expansion of \mathbf{a}_i :

$$\mathbf{a}_{i+1} = \mathbf{a}_i + h\mathbf{a}'_i + \mathcal{O}(h^2)$$

such that

$$\mathbf{a}'_i = \frac{\mathbf{a}_{i+1} - \mathbf{a}_i}{h} + \mathcal{O}(h)$$

Inserting into the above equations thus yields

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h\mathbf{v}_i + \frac{h^2}{2}\mathbf{a}_i + \mathcal{O}(h^3) \quad (20a)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \frac{h}{2}[\mathbf{a}_i + \mathbf{a}_{i+1}] + \mathcal{O}(h^3) \quad (20b)$$

Unlike the Forward-Euler method, the Velocity-Verlet method is invalid for accelerations dependent on velocity: note in particular that \mathbf{v}_{i+1} is dependent on \mathbf{a}_{i+1} . In case the acceleration is dependent on the position (such as in the case of gravitational acceleration), the position vector \mathbf{r}_{i+1} must be evaluated before \mathbf{v}_{i+1} . That is, the Velocity-Verlet method is order-specific with respect to evaluating \mathbf{r}_{i+1} and \mathbf{v}_{i+1} for accelerations dependent on position. The global error made in a Velocity-Verlet integration is $N\mathcal{O}(h^3) = t_N\mathcal{O}(h^2)$.

III. METHOD

A. Algorithms

Having introduced the integration methods in the theory section, this section will take a step further and discuss the actual algorithms used in the simulations. That is, implementing the Forward-Euler and Velocity-Verlet methods as an algorithm for integrating the gravitational N -body problem.

In the following sections, exactly *how* to compute the gravitational acceleration $\mathbf{g}_i^j = \mathbf{g}^j(t_i)$ is momentarily ignored as it is treated later.

1. Static/Stationary objects

It is not uncommon that a particular object in a simulation barely moves, e.g., the Sun in the Sun-Earth system. More often than not, the movement of such an object has no impact on the results, meaning no important details are lost by ignoring it. Therefore, this project will divide gravitational objects into dynamic bodies and

static (or stationary) bodies. In order to differentiate between dynamic and static objects, the complete set of object indices $\{1, \dots, J\}$ is divided into two subsets J_{dyn} and J_{stat} , containing the indices corresponding to dynamic and static indices respectively.

The introduction of static bodies may significantly reduce the number of computations per simulation. This would not only improve the programs' efficiency with respect to program run-time, but it may also reduce the risk of accumulating floating point errors.

2. Forward-Euler integration

The first method is the Forward-Euler integration method, which is in fact particularly easy to implement: the next position and velocity vectors \mathbf{r}_{i+1} and \mathbf{v}_{i+1} can be computed directly as stated in equations (18). Hence, for a complete Forward-Euler integration with N steps of time-step h , the algorithm may be written as:

Algorithm 1 Forward-Euler integration

```

1: Set initial condition  $t_0$ .
2: for  $j \in J_{\text{dyn}}$ :
3:   Set initial conditions  $\mathbf{r}_0$  and  $\mathbf{v}_0$  for object  $j$ .
4: end for
5: for  $i \in \{0, \dots, N\}$ :
6:   for  $j \in J_{\text{dyn}}$ :
7:     Compute  $\mathbf{g}_i^j$ .
8:     Compute  $\mathbf{r}_{i+1}^j = \mathbf{r}_i^j + h\mathbf{v}_i^j$ .
9:     Compute  $\mathbf{v}_{i+1}^j = \mathbf{v}_i^j + h\mathbf{a}_i^j$ .
10:   end for
11: end for
```

Note that in the actual implementation of Algorithm 1, the gravitational acceleration \mathbf{g}_i^j is computed for each $j \in J_{\text{dyn}}$ before updating the position and velocities: \mathbf{r}_{i+1}^j and \mathbf{v}_{i+1}^j . This is simply a result of the specific program structure and has no effect on the results. This comment is included for clarity.

3. Velocity-Verlet integration

The second method is the Velocity-Verlet integration method. The basic principle behind Algorithm 1 carries over to the Verlet algorithm, however, the inclusion of \mathbf{g}_{i+1}^j in the calculation of \mathbf{v}_{i+1}^j poses an additional challenge. The gravitational acceleration is dependent on position, thus \mathbf{r}_{i+1}^j must be computed before \mathbf{v}_{i+1}^j . However, it would be rash to compute both \mathbf{g}_i^j and \mathbf{g}_{i+1}^j at each step, seeing that \mathbf{g}_{i+1}^j eventually becomes \mathbf{g}_i^j in the next step. For a complete Velocity-Verlet integration with N steps of time-step h , the algorithm can be written as:

Algorithm 2 Velocity-Verlet integration

```

1: Set initial condition  $t_0$ .
2: for  $j \in J_{\text{dyn}}$ :
3:   Set initial conditions  $\mathbf{r}_0$  and  $\mathbf{v}_0$  for object  $j$ .
4:   Compute  $\mathbf{g}_0^j$ .
5: end for
6: for  $i \in \{0, \dots, N\}$ :
7:   for  $j \in J_{\text{dyn}}$ :
8:     Compute  $\mathbf{g}_{i+1}^j$ .
9:     Compute  $\mathbf{r}_{i+1}^j = \mathbf{r}_i^j + h\mathbf{v}_i^j + \frac{1}{2}h^2\mathbf{g}_i^j$ .
10:    Compute  $\mathbf{v}_{i+1}^j = \mathbf{v}_i^j + \frac{1}{2}h(\mathbf{g}_i^j + \mathbf{g}_{i+1}^j)$ .
11:    Set  $\mathbf{g}_i^j = \mathbf{g}_{i+1}^j$ .
12:   end for
13: end for

```

Again, note that the actual implementation of Algorithm 2 is slightly different due to program structure: Here, the position is first updated using \mathbf{g}_i^j , next \mathbf{g}_{i+1}^j is computed such that finally \mathbf{v}_{i+1}^j may be computed. This comment is included for clarity.

4. Evaluating the gravitational acceleration

Regardless of the chosen integration method, the gravitational acceleration on each dynamic object from every other object needs to be computed at every time step. Consider first the simplest case: standard Newtonian gravity (equation (4)). The following algorithm loops over each object and computes its contribution to the net gravitational acceleration on object j :

Algorithm 3 Computing the Newtonian \mathbf{g}_i^j .

```

1: Initiate  $\mathbf{g}^j = \mathbf{0}$ .
2: for object  $k \in \{1, \dots, N\}$ :
3:   if  $k \neq j$ :
4:      $\delta\mathbf{r}_i^{kj} = \mathbf{r}_i^k - \mathbf{r}_i^j$ .
5:      $\mathbf{g}_i^j += M_k \delta\mathbf{r}_i^{kj} / |\delta\mathbf{r}_i^{kj}|^3$ .
6:   end if
7:    $\mathbf{g}_i^j *= G$ 
8: end for

```

In order to extend Algorithm 5 to the relativistic case, only a simple adjustment is necessary:

Adjustment Relativistic correction to \mathbf{g}_i^j .

$$\mathbf{g}_i^j *= G \left[1 + 3|\mathbf{r}_i^j \times \mathbf{v}_i^j|^2 / (|\mathbf{r}_i^j|^2 c^2) \right]$$

Similarly, extending Algorithm 5 to the β -Newtonian case requires only the following adjustment:

Adjustment β -Newtonian correction to \mathbf{g}_i^j .

$$\mathbf{g}_i^j += M_k \delta\mathbf{r}_i^{kj} / |\delta\mathbf{r}_i^{kj}|^{\beta+1}.$$

B. Simulations

The following section presents the various scenarios simulated in this project.

1. The Earth-Sun system

The first system to be simulated is the simple two-body Earth-Sun system. Initially, Earth will be placed in a circular orbit in order to assess the stability of the two integration algorithms as a function of different time step lengths h . The Sun is categorised as a static body located at the origin. To simplify, Earth is placed at $\mathbf{r}_0 = (1, 0, 0)$ au with velocity $\mathbf{v}_0^{\text{Earth}} = (0, v_{\text{circ}}^{\text{Earth}}, 0)$, where equation (12) with $M = M_{\odot} = 1$ and $R = 1$ au yields an orbital speed of $v_{\text{circ}}^{\text{Earth}} = 2\pi$ au/yr.

As an ideal orbit would have $|\mathbf{r}_i| = 1$ for all i , the standard deviation in $|\mathbf{r}|$ will be used as a measure for the stability of the results. Additionally, knowing that the Velocity-Verlet algorithm is energy-conservative, stability will also be evaluated using the standard deviation in the mechanical energy E and angular momentum \mathbf{L} :

$$E = K + U = \frac{1}{2}M_E|\mathbf{v}|^2 - \frac{GM_E M_{\odot}}{|\mathbf{r}|} \quad (21a)$$

$$|\mathbf{L}| = |\mathbf{r} \times M_E \mathbf{v}| \quad (21b)$$

Note however that the only action of M_E in the above equations is to scale the values. Hence, the underlying conserved quantities are actually

$$\frac{E}{M_E} = \frac{1}{2}|\mathbf{v}|^2 - \frac{GM_{\odot}}{|\mathbf{r}|} \quad (22a)$$

$$l = |\mathbf{r} \times \mathbf{v}| \quad (22b)$$

The Velocity-Verlet algorithm is bound to produce better results than the Forward-Euler algorithm. In all remaining the simulations, only the Velocity-Verlet algorithm will be used.

Having checked the stability of the algorithms, the escape velocity of Earth will be studied. Again the Sun is static at the origin and the origin is placed at $\mathbf{r}_0 = (1, 0, 0)$ au, now however with initial velocity $\mathbf{v}_0^{\text{Earth}} = (v_{\text{esc}}^{\text{Earth}}, 0, 0)$. Using astronomical units and $R = 1$ au, equation (14) yields $v_{\text{esc}}^{\text{Earth}} = 2\sqrt{2}\pi$ au/yr. Although this is perfect for the ideal system, numerical errors may affect the simulated escape velocity to a small degree. In order to test the accuracy of the simulated escape velocity, the simulation will be run using a small dense set of normally distributed velocities, centered about the ideal escape velocity.

Furthermore, it is of academic interest to study the same problem with the standard Newtonian gravity replaced by the general β -Newtonian gravity. Earth will be given the best simulation escape velocity from the $\beta = 2$ case and the simulation will be run for several β values in the range [2, 3].

2. The Earth-Jupiter-Sun system

Jupiter will now be introduced to the Earth-Sun system such that the complete system is a so-called three-body system. The idea is to study how Earth's two-body circular orbit from the first simulation is affected by the presence of Jupiter with special emphasis on the stability of the orbit. The stability will be measured using the same measures as introduced above.

As opposed to the two-body system (ignoring parabolic trajectories), the three-body system is not inherently stable. There are many ways to initiate the Earth-Jupiter-Sun system, but in order to study the specific effect of Jupiter's perturbation of Earth's orbit, some restrictions need to be introduced: First of all, Jupiter will be placed in a circular orbit such that the potential instability of the Jupiter-Sun system does not affect Earth's orbit. Using equation (12) with $M = M_\odot = 1$ and $R = 5.2$ au (Jupiter's semi-major axis) yields an orbital speed of $v_{\text{circ}}^{\text{Jupiter}} \cong 2.755$ au/yr. Next, assuming a two-dimensional system, the direction from the Sun to Earth $\hat{\mathbf{r}}^{\text{Earth}}$ and the Sun to Jupiter $\hat{\mathbf{r}}^{\text{Jupiter}}$ will necessarily become equal at least once every year. Hence, the initial angular direction of Jupiter is arbitrary. Therefore Jupiter is placed at $\mathbf{r}_0^{\text{Jupiter}} = (5.2, 0, 0)$ au such that Jupiter's gravitational pull is the strongest at the start of the simulation.

Having explored Jupiter's standard perturbation of Earth's orbit, the simulation is now rerun using a larger mass for Jupiter. In particular, using masses $10M_{\text{Jupiter}}$ and $1000M_{\text{Jupiter}}$.

3. The Solar System: The N-body problem

The simple system is now expanded to include every planet in the solar system. Moreover, the Sun will now be classified as a dynamic object to mimic the real solar system as much as possible. The simulation will be run in three-dimensions using initial conditions from the following NASA website: <https://ssd.jpl.nasa.gov/horizons.cgi#top>. [3] The coordinates are extracted using these settings:

```
Ephemeris Type:
VECTORS
Target Body in
{Sun [Sol], Mercury, Earth [Geocenter], Mars,
Jupiter, Saturn, Uranus, Neptune}
Coordinate Origin:
Solar System Barycenter (SSB)
Time Span:
Start=2018-10-22, Stop=2018-10-23, Step=1 d
Table Settings:
default
Display/Output:
default
```

A document named `NASA_data.dat` containing all the data can be found on the GitHub repository.

4. The relativistic Mercury-Sun system

The final simulation will study the two-body Mercury-Sun system using the relativistic correction to the gravitational acceleration. The main and only purpose of this simulation is to reproduce the observed perihelion precession of Mercury, thereby arguing for the validity of general relativity.

The simulation will take place in the x - y plane with the x -axis defined along the initial perihelion of Mercury. Again, the Sun will be treated as a static object, while Mercury is given initial conditions $\mathbf{r}_0^{\text{Mercury}} = (0.3075, 0, 0)$ au and $\mathbf{v}_0^{\text{Mercury}} = (0, 12.44, 0)$ au/yr. The simulation will run a whole century with the intention of reproducing a perihelion angle shift of $\Delta\theta_p = 43''$.

In order to measure the shift, the angle of perihelion θ_p is determined using polar coordinate transformations:

$$\theta_p = \arctan\left(\frac{y_p}{x_p}\right) \quad (23)$$

where $\mathbf{r}_p = (x_p, y_p, 0)$ is the position vector of the perihelion of Mercury.

IV. RESULTS

A. The Earth-Sun System: Circular Orbit

Figure 1 shows Earth's circular orbit when simulated with a stationary Sun, ideal initial conditions (for circular orbits) and integrated using the Verlet-Velocity algorithm with $N = 1000$ integration steps over the course of 1 yr. The simulation, i.e. the orbit, is clearly completely stable.

The stability of the Forward-Euler and Velocity-Verlet algorithms is measured in figures 2, 3 and 4 as functions of the integration step length h . Not surprisingly, the Velocity-Verlet algorithm improves on the Forward-Euler algorithm in more or less every way. In particular, the gradient of the two trends in figure 2 highlights the global error made in each algorithm. Whereas the Velocity-Verlet algorithm scales as $\mathcal{O}(h^2)$, corresponding to a gradient of about 2, the Forward-Euler algorithm scales as $\mathcal{O}(h)$ and has therefore a gradient of about 1. As expected, improving the step length of the simulation leads to improved energy-conservation. For the case of Forward-Euler, the improvement does not seem to follow any particular pattern. For the case of Velocity-Verlet, the gradient is unmistakably constant. The same relationships can be found in figure 4, although the Velocity-Verlet algorithm performs surprisingly well for with respect to $\text{std}(l)$.

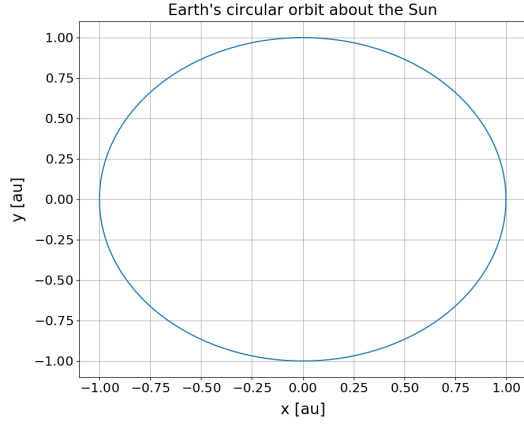


Figure 1: A plot of the two-dimensional simulation of Earth's orbit using initial conditions $x = 1$ au, $y = 0$ and an initial velocity equal to the ideal velocity for circular orbits.

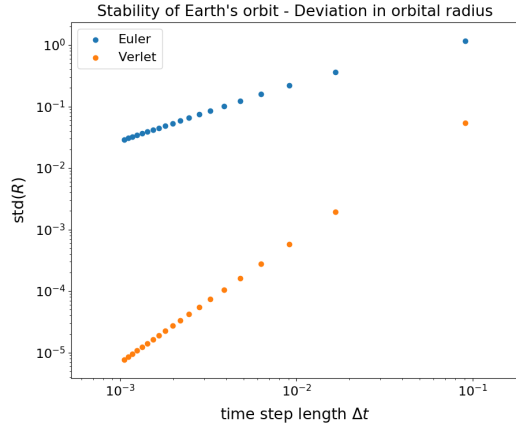


Figure 2: The stability of Earth's orbit in the two-body Earth-Sun system, measured using the standard deviation in the orbit's radius $|\mathbf{r}_i|$ with respect to the integration step length h .

B. The Earth-Sun System: Escape Velocity

Figure 5 shows the effect of increasing the β -gravity parameter in (8) from 2 to 3. As Earth's initial velocity is kept constant, the gravitational attraction from the Sun for $\beta > 2$ is unable to trap Earth in its potential well. The simulations integrate the system using the Velocity-Verlet algorithm for 200 years with $N = 100\,000$ integration steps.

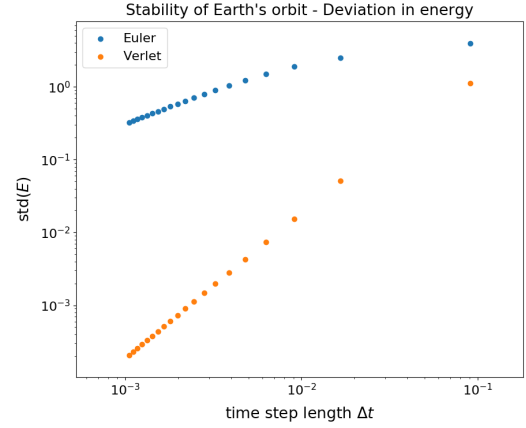


Figure 3: The stability of Earth's orbit in the two-body Earth-Sun system, measured using the standard deviation in the orbit's mechanical energy E/M_E (equation (21a)) with respect to the integration step length h .

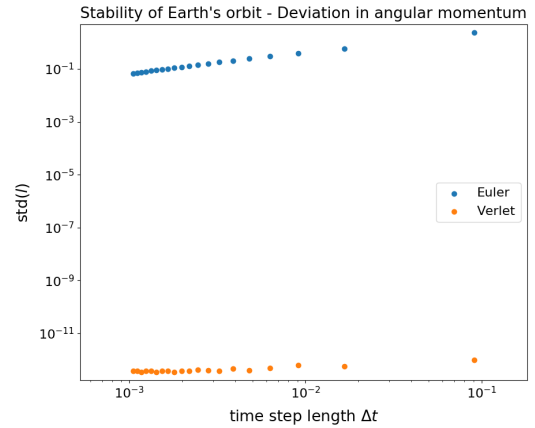


Figure 4: The stability of Earth's orbit in the two-body Earth-Sun system, measured using the standard deviation in the orbit's angular momentum l (equation (21b)) with respect to the integration step length h .

C. The Earth-Jupiter-Sun System

This three-body system was integrated 4 times using the Velocity-Verlet algorithm over the course of 2 years with $N = 200\,000$ integration steps. The first integration did not include Jupiter, while the 3 latter integrations included Jupiter with a mass of M_J , $10M_J$ and $1000M_J$ respectively. The results are shown in figure 6. It is clear that Jupiter's standard perturbation of Earth's orbit is negligible. For the greater masses the effect does play a role as the radius begins to oscillate about a constant distance of 1. For the $1000M_J$ case the motion is increasingly unstable as time continues, meaning the oscillation could increase to a point of significance in the long run.

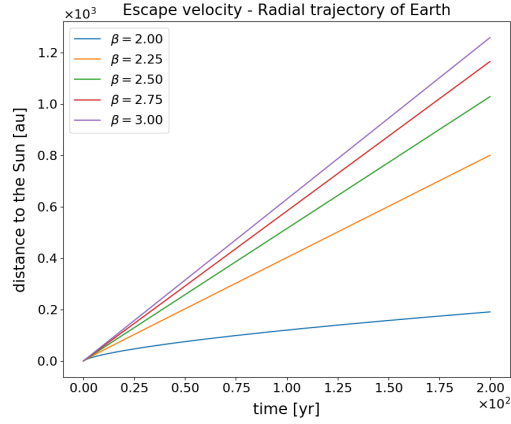


Figure 5: The effect of increasing the β -gravity parameter on the radial trajectory of Earth during a simulation of gravitational escape.

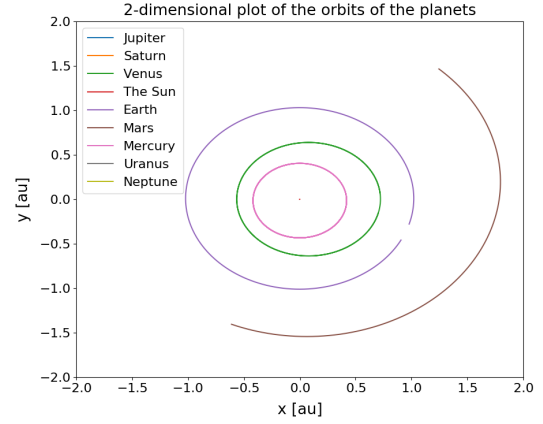


Figure 7: A two-dimensional plot of a three-dimensional simulation of the Solar System's planetary orbits. The initial conditions were randomised with inbedded perturbed orbits.

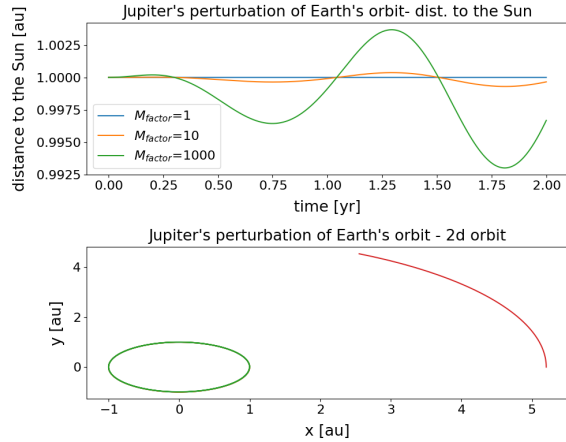


Figure 6: Jupiter's perturbation of Earth's orbit about the Sun. The red orbit in the two-dimensional plot is Jupiter's orbit. All three Earth orbits are plotted in the two-dimensional plot.

D. The Solar System

After many attempts, I declared myself unable to use the NASA data. I do not know if this is because I am reading the data wrong, the units wrong, or something entirely different. So I decided to use random generated initial conditions instead. I choose 8 uniformly distributed angles $\theta \in [0, 2\pi]$ and placed the planets in circular orbits according to (12) using the "Distance to sun in AU" data from the project pdf. Then I gave the orbits a small disturbance in order to emulate perturbed orbits.

The system was run for 1 year using the Velocity-Verlet algorithm with $N = 500\,000$ integration steps, the results are shown below in figures 7 and 8

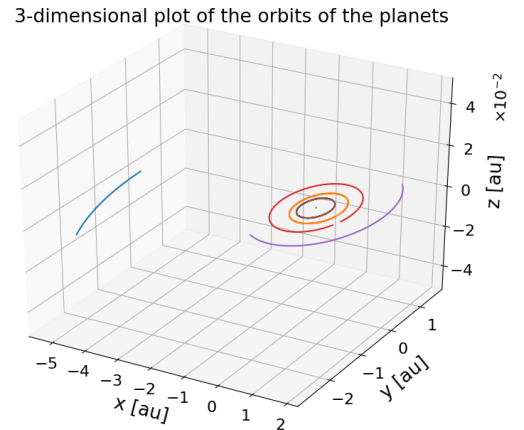


Figure 8: A plot of a simulation of the Solar System's planetary orbits. The initial conditions were randomised with inbedded perturbed orbits.

E. The Perihelion Precession of Mercury

Integrating the Mercury-Sun system using the Velocity-Verlet algorithm for 50 years with $N = 5\,000\,000$ integration steps yielded a total perihelion precession of $26.66''$, which implies a shift of $53.31''$ over the course of 100 years. Sadly this is a little to much compared to the expected shift $46''$: an error of about 15.9%. This is likely a result of an accumulative numerical error, which has increased to the point where its effects is of a significant order.

V. DISCUSSION

Despite the success of the Velocity-Verlet algorithm in this report, the perihelion precession was reproduced with an error of 15.9%. One way to tackle this would be to increase the number of integration points, however this solution fails once machine precision is reached. Another way would be to exchange the algorithm with a higher-order integration scheme, albeit it would also carry with it a greater number of computations per simulation.

VI. CONCLUSION

Overall, Project 3 has been a descent success. The Velocity-Verlet algorithm was shown, as expected, to be overly favoured over the simpler Forward-Euler algorithm. The Earth-Sun system (both the circular orbit scenario and the β -Newtonian escape-velocity scenario) behaved as predicted in the theory section. The perturbations of Earth's orbit due to the gravitational attraction from Jupiter were shown to have miniscule effects. On the other hand, the "10 M_J Jupiter" instigated an oscillation in the orbital radius, while the "1000 M_J Jupiter" drove the orbit unstable. The perihelion precession of Mercury was reproduced with an error of 15.9%.

-
- [1] Morten Hjorth-Jensen. Computational physics lectures: Ordinary differential equations, 2017. <http://compphysics.github.io/ComputationalPhysics/doc/pub/ode/pdf/ode-print.pdf>.
 - [2] Morten Hjorth-Jensen. Fys3150 computational physics - project 3, 2018. <http://compphysics.github.io/ComputationalPhysics/doc/Projects/2018/Project3/pdf/Project3.pdf>.
 - [3] NASA. Horizons web-interface, 2018. <https://ssd.jpl.nasa.gov/horizons.cgi#top>.