Simulating the Solar System Using Ordinary Differential Equations

Håkon Fossheim

Github repository

Abstract

The aim of this project is to apply two ordinary differential equation solvers, namely the Euler method and the velocity Verlet algorithm in order to simulate the motion of the planets in the solar system. A comparison of the run times for both algorithms is made, and the Verlet algorithm proves superior. The stability of the Sun-Earth-Jupiter system is studied for different values for the mass of Jupiter and the precession of the perihelion of Mercury is simulated.

Contents

1 Introduction

This project is a demonstration of the usefulness of object orientation. Object orientation is key when dealing with many similar instances of the same equations where only a few parameters differ from each other. A two-dimensional simulation of the solar system will be modeled by solving coupled ordinary differential equations. To solve these differential equations, the Euler method and the velocity Verlet algorithm is applied to the Sun-Earth system.
First, the equations needed for simulating the Sun-Earth system will be derived. Secondly, the mentioned algorithms will be introduced. The equations used for the Sun-Earth system will then be expanded to calculate forces from multiple planets. Lastly, a simulation of the precession of Mercury's perihelion is attempted.

2 Theory

2.1 The Earth-Sun system

First consider a two-body interaction e.g. the Earth-Sun system. The force acting upon Earth from the Sun and vice versa is given by

$$F = ma = \frac{GM_\odot M_\oplus}{r^2} \tag{1}$$

Where $G = 6.67 \cdot 10^{-11}\text{m}^3\text{kg}^{-1}\text{s}^{-2}$ is the gravitational constant, $M_\odot = 2 \cdot 10^{30}\text{kg}$ is the mass of the Sun and $M_\oplus = 6 \cdot 10^{24}\text{kg}$ is the mass of the Earth. Separating this force into its x and y component, as shown in figure 1, yields

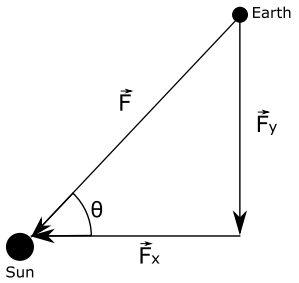

Figure 1: Caption

$$a_x = \frac{F_x}{M_\oplus} = \frac{F\cos\theta}{M_\oplus} = \frac{GM_\odot x}{r^3} \tag{2}$$

$$a_y = \frac{F_y}{M_\oplus} = \frac{F \sin\theta}{M_\oplus} = \frac{GM_\odot y}{r^3} \tag{3}$$

To replace the $GM_\odot$ term with something more manageable, first set the centripetal force equal to the gravitational force.

$$M_\odot \frac{v^2}{r} = \frac{GM_\odot M_\oplus}{r^2} \tag{4}$$

$$v^2 r = GM_\odot \tag{5}$$

Approximating the Earths orbit as circular and switching to units of Au/Year gives

$$\begin{aligned} v &= \frac{2\pi r}{P} \\ &= \frac{2\pi \text{Au}}{\text{Year}} \end{aligned}$$

Inserting this into equation 5 gives

$$GM_\odot = 4\pi^2 (1\text{Au})^3 / 1\text{yr}^2 \tag{6}$$

Inserting this into equation 2 and 3 yields the scaled equations for the acceleration

$$a_x = \frac{4\pi^2 x}{r^3} \tag{7}$$

$$a_y = \frac{4\pi^2 y}{r^3} \tag{8}$$

Where $r = \sqrt{x^2 + y^2}$.

2.2 Euler's Method

The Euler method is defined by the following recursive relations

$$x_{n+1} = x_n + v_n dt \tag{9}$$

$$v_{n+1} = v_n + a_n dt \tag{10}$$

The implementation of this algorithm in two dimensions take the following form

```
theBody.xpos += theBody.xvel*dt;
theBody.xvel += theBody.xacc*dt;

theBody.ypos += theBody.yvel*dt;
theBody.yvel += theBody.yacc*dt;
```

Integrating one step requires 18 FLOPS, but fewer floating point operations can probably be achieved.
As seen, the algorithm needs an initial conditions for the position and velocity. The acceleration is calculated using equation 7 and 8. It is worth noting that this algorithm does not conserve energy. It is therefore not ideal for calculating the orbit of planets, as they will start to spiral outwards even for quite small time-steps.
This can be easily solved by making a small adjustment to Euler's method. By first finding the velocity and then using it to calculate the position, energy will be conserved. This algorithm is called the Euler-Cromer algorithm and is given by

$$v_{n+1} = v_n + a_n dt \tag{11}$$

$$x_{n+1} = x_n + v_{n+1} dt \tag{12}$$

2.3 The Velocity Verlet Algorithm

To derive the velocity Verlet algorithm, start by Taylor expanding x_{i+1} and v_{i+1}

$$x_{i+1} = x_i + hx_i' + \frac{h^2}{2}x_i'' + h^3 \tag{13}$$

$$v_{i+1} = v_i + hv_i' + \frac{h^2}{2}v_i'' \tag{14}$$

Where $x_i'' = v_i' = a_i$ is given by equation 7 and 8. Thus, the only unknown is v_i''. To find it, let's Taylor expand v_{i+1}

$$v_{i+1}' = v_i' + hv_i'' + \mathcal{O}(h^3) \tag{15}$$

$$v_i'' = \frac{v_{i+1}' - v_i'}{h} = \frac{a_{i+1} - a_i}{h} \tag{16}$$

Inserting this into equation 14 gives

$$v_{i+1} = v_i + ha_i + \frac{h}{2}(a_{i+1} - a_i) \tag{17}$$

$$v_{i+1} = v_i + \frac{h}{2}a_i + \frac{h}{2}a_{i+1} \tag{18}$$

The algorithm is then as follows

$$x_{i+1} = x_i + hv_i + \frac{h^2}{2}a_i \tag{19}$$

$$v_{i+1} = v_i + \frac{h}{2}a_i + \frac{h}{2}a_{i+1} \tag{20}$$

Here a_{i+1} is unknown. The algorithm therefore first calculates x_{i+1}, then finds $a_{i+1}(x_{i+1}, t_{i+1}) = \frac{F}{m}(x_{i+1}, t_{i+1})$ and finally calculates v_{i+1}.
The implementation of this algorithm uses 34 FLOPS, but this is most likely excessive and can probably be reduced by better coding.
The velocity Verlet algorithm is a symplectic integrator. The most important property of the velocity Verlet algorithm of concern in this project is that it conserves energy and angular momentum [1].

2.4 Multi Body Problem

Adding a third body, e.g. Jupiter requires a few alterations to the equations of motion. Let's look at the x-component of the force on Earth from Jupiter.

$$F_x^{EJ} = -\frac{GM_\odot x}{r_{EJ}^3}(x_E - x_J) \tag{21}$$

Here $r_{EJ} = \sqrt{(x_E - x_J)^2 + (y_E - y_J)^2}$
The acceleration of Earth due to the force from the Sun and Jupiter is then given by

$$a_x = \frac{dv_x^E}{dt} = -\frac{GM_\odot x}{r^3} - \frac{GM_J(x_E - x_J)}{r_{EJ}^3} \tag{22}$$

To scale this equation, multiply both sides of equation 6 by $\frac{M_J}{M_\odot}$ to give $GM_J = 4\pi^2 \frac{M_J}{M_\odot} \text{AU}^3/\text{yr}^2$. Inserting this into equation 22 and using units of AU and yr gives the following scaled, coupled ODEs

$$\frac{dv_x^E}{dt} = -\frac{4\pi^2 x_E}{r^3} - \frac{4\pi^2 \left(\frac{M_J}{M_\odot}\right)(x_E - x_J)}{r_{EJ}^3} \tag{23}$$

$$\frac{dx^E}{dt} = v_x^E \tag{24}$$

The equations for y-component of the acceleration and velocity are similar. It is worth noting that once the force from e.g. Jupiter on Earth is calculated, the force on Jupiter from Earth is also know, since Newtons third law states that these have to be equal to each other.

Any number of bodies can be added through the same procedure as above, but the values of mass and position must be altered. This is where object orientation comes in handy. Instead of having say 8 separate versions of equation 23 for each of the planets, a loop can be created which iterates over all the planets and extracts the relevant values on each iteration. The class "CelestialBody" stores all the relevant information about a planet and all the instantiations of this class are stored in the class "SolarSystem".

2.5 Escape Velocity

Escape velocity is achieved when the kinetic energy of an object is equal to that of its gravitational potential energy. The object is then free to escape its gravitational well. Let's find the escape velocity for Earth in the Earth-Sun system

$$|E_k| = |E_p| \tag{25}$$

$$\frac{1}{2}M_E v_{esc}^2 = \frac{GM_\odot M_E}{r} \tag{26}$$

$$v_{esc}^2 = \frac{2GM_\odot}{r} \tag{27}$$

Using $GM_\odot = 4\pi^2 \mathrm{AU}^3/\mathrm{yr}^2$ and switching to units of AU and yr gives

$$v_{esc} = \sqrt{8}\pi \mathrm{AU/yr} \tag{28}$$

Or in units of km/s:

$$v_{esc} = 42.15\,\mathrm{km/\,sec} \tag{29}$$

2.6 Precession of Mercury's Perihelion

Albert Einstein proposed three experimental tests of his theory of general relativity, one of them being the precession of Mercury's perihelion. Mercury moves at high velocity in the Sun's gravitational field. Only comets and asteroids approach the sun closer at perihelion. It therefore offers unique opportunities for testing the theory of general relativity [2]. In a Newtonian two-body system, the bodies will orbit indefinitely in perfect ellipses around the centre of mass. General relativity on the other hand, predicts that the closest point in Mercury's orbit to the centre of mass will precess at a rate of 43" per century.

3 Implementation

First an instantiation of the class SolarSystem is created. Then the desired bodies are added to the vector "the_bodies", where each object of the vector is an instantiation of the class CelestialBody. The constructor of CelestialBody takes 7 arguments: the mass of the body, the initial x and y positions, the initial x and y velocities, the value of β in equation 31 and finally a bool determining whether the constructor should give the acceleration a relativistic correction or not.
The class SolarSystem contains most of the relevant functions for this project, including a function to integrate and write all the objects of the solar system to file, a function for comparing the run time between Euler's method and the velocity Verlet algorithm and a function for calculating the precession of Mercury.

4 Results

The following plots do not include the Sun. This is due to relentless error messages when attempting to add it to the system. The equations used to simulate the orbits of the planets do however take the Sun into account, and it is kept fixed as the center of mass throughout the project.
Figure 2 shows the Earth-Sun system. As seen, Earth's orbit is spiraling outwards. This is due to the fact that Euler's method does not conserve energy, as discussed in the theory section.

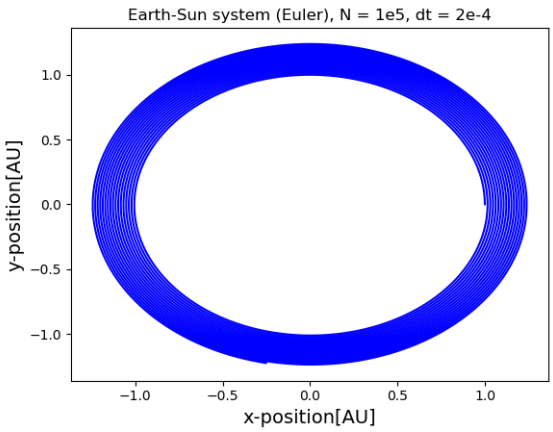

Figure 2: Caption

The velocity Verlet algorithm is vastly superior to Euler's method in calculating the orbits of planets, as can be seen in figure 3. Unlike Euler's

method, the velocity Verlet algorithm produces an orbit which doesn't spiral, but stays put.

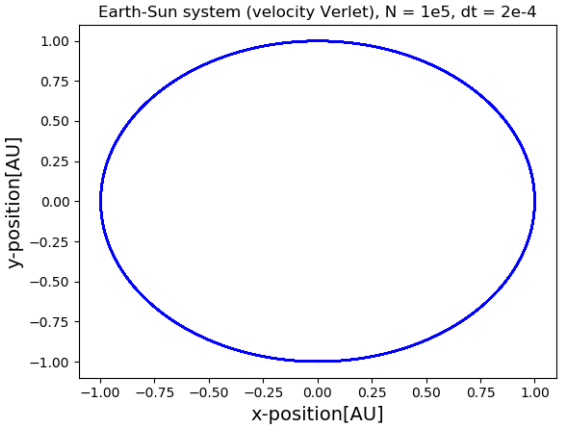

Figure 3: Caption

The run times for Euler's method and the velocity Verlet algorithm are shown in table 1. For lower N values, the Verlet algorithm is a little bit faster, but gets slower than Euler's method for large N. The run time for the algorithms depend on what the machine is doing at the time, and this might be the origin of the peculiar result seen in table 1.

Table 1: Run times for the Euler method vs. the velocity Verlet algorithm.

N	Euler(s)	Verlet(s)
$1 \cdot 10^1$	$8.82 \cdot 10^{-4}$	$7.29 \cdot 10^{-4}$
$1 \cdot 10^2$	$6.88 \cdot 10^{-4}$	$4.97 \cdot 10^{-4}$
$1 \cdot 10^3$	$5.92 \cdot 10^{-4}$	$1.01 \cdot 10^{-3}$
$1 \cdot 10^4$	$3.02 \cdot 10^{-3}$	$3.11 \cdot 10^{-3}$
$1 \cdot 10^5$	$2.33 \cdot 10^{-2}$	$2.49 \cdot 10^{-2}$

Figure 4 shows the orbit of Earth around the sun for different values of initial velocity. When the initial velocity is increased from 29.79 km/ sec to 35 km/ sec, the orbit gets elongated, but Earth does not have enough kinetic energy to escape from the gravitational well of the Sun. Through trial and error, the escape velocity was found to be about 42 km/ sec. This is in accordance with the theoretical value, which is 42.15 km/ sec.

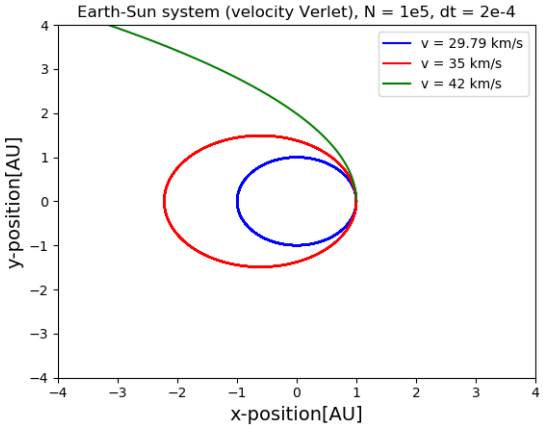

Figure 4: Caption

Next, let's change the gravitational force by replacing

$$F_G = \frac{GM_\odot M_{Earth}}{r^2} \tag{30}$$

with

$$F_G = \frac{GM_\odot M_{Earth}}{r^\beta} \tag{31}$$

Figure 5 shows three different values of β. As β is increased, the gravitational force gets weaker. The Sun starts to lose it's "grip" on Earth when β approaches 3. For β = 3.5, the gravitational force is too weak to keep Earth in it's orbit.

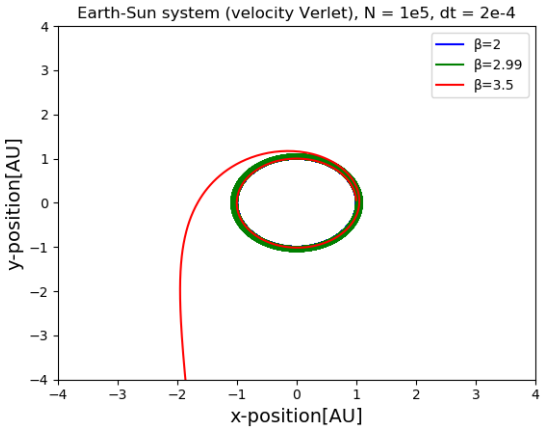

Figure 5: Caption

Let's now study what happens when a third body is added, namely Jupiter.

Figure 6 shows the Earth-Jupiter-Sun system, with Jupiter having a mass of $1.898 \cdot 10^{27} kg = 1 M_J$, which is about 1000 times less than the mass of the sun. As seen, the orbit of Earth is not visibly affected.

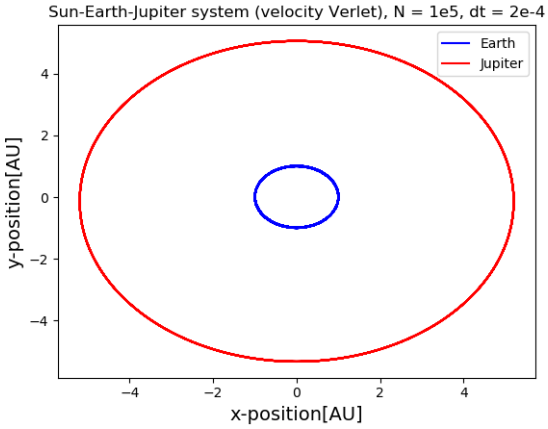

Figure 6: Caption

Increasing the mass of Jupiter by a factor of 10 changes the orbit of Earth, as seen in figure 7.

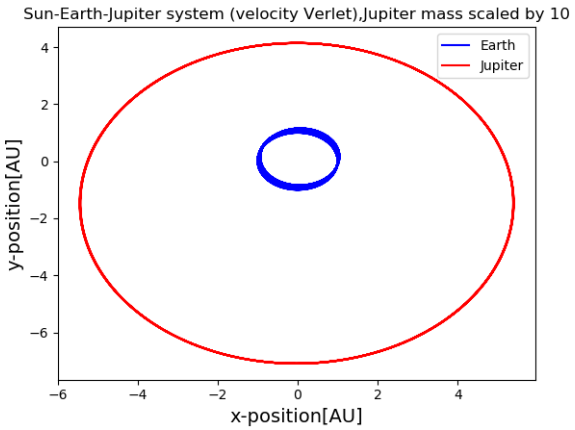

Figure 7: Caption

Figure 8 shows that increasing the mass of Jupiter by a factor of 1000 makes the system unstable.

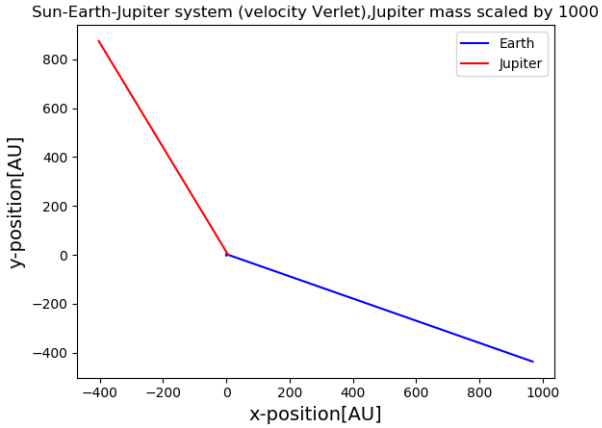

Figure 8: Caption

Figure 9 shows a simulation of all the planets in the Solar system.

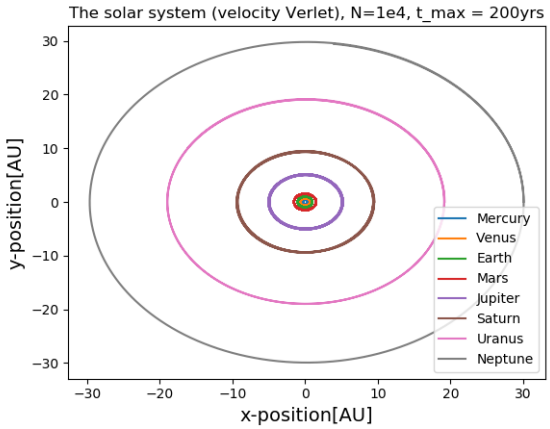

Figure 9: Caption

Next, a general relativistic correction is added to the Newtonian force, as discussed in the theory section. The angle of the perihelion is extracted on each orbit around the sun, and plotted in figure 10. The origin of the zigzaggy behavior is not known. The average perihelion angle does however increase at a rate of 175.8”/100yrs. This is much more than the theoretical value of 43”/100yrs.

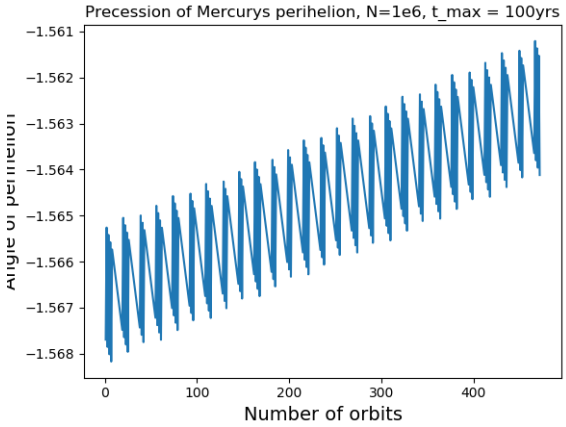

Figure 10: Caption

5 Discussion

Even though the velocity Verlet algorithm uses more FLOPS than Euler's method, it does not consistently use more CPU time. As seen in table 1, for low values of integration points, the Verlet algorithm had a slightly lower run time. This is likely due to errors. Since the run time is so short, factors like how much workload the computer is under at the time will make a larger contribution to the CPU time.
A simulation of the precession of Mercury's perihelion was attempted. The results gave a precession of 175.8″/100yrs, which is not in agreement with the theoretical value of 43″/100years. This erroneous result might be due to not having a sufficient amount of integration points.

6 Conclusion

Through object orientation, the solar system has been simulated. This would otherwise be a strenuous task if one was to stick with procedural programming.
Two integrators have been implemented, namely Euler's method and the velocity Verlet algorithm. The Verler algorithm proved superior when calculating the orbits of planets, since it preserves energy and angular momentum. The run times for the two algorithms were quite similar even though Euler's method required considerably fewer FLOPS.
The escape velocity of Earth was found through trial and error to be about 42 km/ sec, which is in agreement with the theoretical value of 42.15 km/ sec.
Lastly, the precession of Mercury's perihelion was simulated by adding a

relativistic correction to gravitational force. The simulation gave a precession of 175.8″/100yrs compared to the actual value of 43″/100years.

References

[1] Hiroshi Kinoshita, Haruo Yoshida, and Hiroshi Nakai. "Symplectic integrators and their application to dynamical astronomy". In: *Celestial Mechanics and Dynamical Astronomy* 50.1 (Mar. 1990), pp. 59–71. ISSN: 1572-9478. DOI: 10.1007/BF00048986. URL: https://doi.org/10.1007/BF00048986.

[2] G V Kraniotis and S B Whitehouse. "Compact calculation of the perihelion precession of Mercury in general relativity, the cosmological constant and Jacobi's inversion problem". In: *Classical and Quantum Gravity* 20.22 (2003), p. 4817. URL: http://stacks.iop.org/0264-9381/20/i=22/a=007.