Project 3 in FYS3150

Bendik Steinsvåg Dalen, Ulrik Seip

November 9, 2018

https://github.com/UlrikSeip/Projects/tree/master/prosjekt3

1 ABSTRACT

In this project we simulate the orbits of all the 8 planets in the solar system, and Pluto. Comparing the Forward Euler and the Velocity Verlet methods we find the Velocity Verlet method to be preferable due to its conservation of energy. We then test the Velocity Verlet method against the analytically derived escape velocity and perihelion of Mercury.

2 INTRODUCTION

Our solar system is littered with asteroids, planets and moons. This plethora objects floating around in space makes for a perfect exercise in solving multi body differential equations in 3 dimensions.

When simulating orbits for several celestial bodies with high accuracy, the computation can be expensive, and so it is paramount to strike a balance between efficiency and accuracy. To explore this balance, we will run simulations using the Velocity-Verlet integration method, and comparing with the Forward-Euler method. Having found the optimal way to simulate the orbits, we move on to test the effect of the gravitational pull between planets, and complete a full model for all planets of the solar system, and Pluto.

We also wanted to look at the perihelion precession of Mercury to test the stability of our code, but weren't able to due to time constraints.

3 METHOD

3.a Newton's law of gravitation

One of the most common representations of Newton's law of gravitation on Earth is

$$\boldsymbol{F}_G = \frac{M_{\text{Earth}} v^2}{r}\hat{\boldsymbol{r}} = \frac{GM_\odot M_{\text{Earth}}}{r^2}\hat{\boldsymbol{r}}, \tag{1}$$

where $G = 6.67 \cdot 10^{-}11\frac{m^3}{kgs^2}$ is the gravitational constant, m_1 and m_2 are the masses of the bodies exerting a force upon each other, F_G is said force, and r is the distance between the bodies. Using Kepler's laws this can be further simplified to

$$\boldsymbol{F}_G = \frac{M_\odot M_{\text{Earth}} 4\pi^2}{r^2}\frac{\text{AU}^3}{\text{yr}^2}\hat{\boldsymbol{r}}. \tag{2}$$

We can then rewrite this for a point mass and acceleration as

$$\boldsymbol{a} = \frac{M_\odot 4\pi^2}{r^2}\frac{\text{AU}^3}{\text{yr}^2}\hat{\boldsymbol{r}}. \tag{3}$$

3.b The Forward Euler method

To use equation 3 for the Forward Euler method we need an expression for $\Delta\boldsymbol{v}$. We therefore introduce a time step dt. We also define $\hat{\boldsymbol{r}} = \cos(\theta)\hat{\boldsymbol{i}} + \sin(\theta)\hat{\boldsymbol{j}} + \cos(\phi)\hat{\boldsymbol{k}}$. This gives us

$$\frac{d\boldsymbol{v}}{dt} = \frac{v_{i+1} - v_i}{dt} = -4\pi^2\frac{M_\odot}{r^2}\hat{\boldsymbol{r}}$$

$$\boldsymbol{v}_{i+1} = -4\pi^2\frac{M_\odot}{r^2}dt\hat{\boldsymbol{r}} + \boldsymbol{v}_i = -\boldsymbol{a}_i dt + \boldsymbol{v}_i. \tag{4}$$

We can do something similar to find $\Delta\boldsymbol{x}$:

$$\frac{d\boldsymbol{x}}{dt} = \frac{x_{i+1} - x_i}{dt} = v_{i+1}\hat{\boldsymbol{r}} = \boldsymbol{v}_{i+1}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{v}_{i+1}dt + \boldsymbol{x}_i = \boldsymbol{v}_{i+1}dt + \boldsymbol{x}_i. \tag{5}$$

3.c The Velocity Verlet method

From [1] we know that the Verlet formula for a specific x_i is

$$x_{i+1} = 2x_i - x_{i-1} + h^2 x_i^{(2)} + O(h^4), \tag{6}$$

where h is the timestep, $x^{(2)}$ is function 3, and O is the truncation error. We also know that the velocity is

$$x_i^{(1)} = \frac{x_{i+1} - x_{i-1}}{2h} + O(h^2). \tag{7}$$

Unfortunately function 6 is a bit difficult to use as we only know the initial position, and thus can't find x_1 or x_2, and so forth. To help with this we can rewrite them into

$$x_{i+1} = x_i + h x_i^{(1)} + \frac{h^2}{2} x_i^{(2)} \tag{8}$$

and

$$x_i^{(1)} = x_{i-1}^{(1)} + \frac{h}{2}\left(x_i^{(2)} + x_{x-i}^{(2)}\right), \tag{9}$$

see section 5.a for more details.

3.d The code for the implementations

Both the Velocity Verlet and the Forward Euler algorithms were implemented using a combination of python and Julia. Python happens to be a convenient language for making an object oriented implementation of a solar system simulation, while Julia is in most cases more than 10 times faster than Python, and even rivals c++, when it comes to computation speeds.

3.e Testing the algorithms

To test our algorithm we can check if the escape velocity of our planets is correct. We find it analytically by using conservation of energy, and the fact that the kinetic energy, and the potential energy of our planets are defined with opposite signs. We therefore start with the equation:

$$E_k + E_p = \tfrac{1}{2} M_E v_{esc}^2 - G\frac{M_\odot M_E}{r} = 0$$

If we then isolate v_{esc} on the left side of the equation, we end up with the following equation:

$$V_{esc} = \sqrt{\frac{2GM_\odot}{r}} \tag{10}$$

To test our implementation against the analytical solution we simply run a simulation for an imaginary planet with the initial analytical velocity, and see if it has returned after several thousand years. We can also try simulating with a slightly lower initial velocity, and se if the planet does indeed stay in orbit with anything less than the analytical initial velocity.

Furthermore the algorithms can be tested against each other for both accuracy and consistency.

3.f The three-body problem

We now have a good basis to extend our algorithm to study the three-body problem, by adding Jupiter to the equation. The force between Earth and Jupiter is

$$\boldsymbol{F}_{\text{Earth-Jupiter}} = \frac{GM_{\text{Jupiter}} M_{\text{Earth}}}{r^2_{\text{Earth-Jupiter}}}\hat{\boldsymbol{r}}. \tag{11}$$

For simplicity's sake we will keep the Sun fixed in the centre of mass, or origo, for now. For each timestep we then calculate the force on Earth and Jupiter from the Sun, and the the force between Earth and Jupiter, and add them up. We then get an position array for both Earth and Jupiter.

To test the stability of our Verlet solver we also studied what effect increasing the magnitude of Jupiter by 10 and 1000 would have on the system.

3.g Final model for all planets of the solar system

We now almost have a working model of the solar system. Firstly we calculated the three-body problem of the Earth, Sun and Jupiter again, but now treating the Sun as an object, instead of fixing it in the centre of mass. We then get the path of the objects around the centre of mass. The initial values of the Sun was found by setting the position in origo, and making sure the momentum of the Sun was the negative of the total momentum of the other planets.

Finally we simply added the remaining planets, and Pluto, to the calculation as extra bodies. The acceleration on each body was found as above, by calculating the force on each object from the other objects.

4 RESULTS

4.a Testing the algorithms

As can be seen in figure 1 the planet did indeed escape with our analytical initial velocity, and figure 2 shows that a slight reduction to said initial velocity places the planet back in orbit.

When comparing the Velocity Verlet and Forward Euler algorithms we observe an increase in the orbits radius for both cases both in 2 and 3 dimensions. In figure 3, 4, 5, 6 we observe a higher increase for the Forward Euler Method

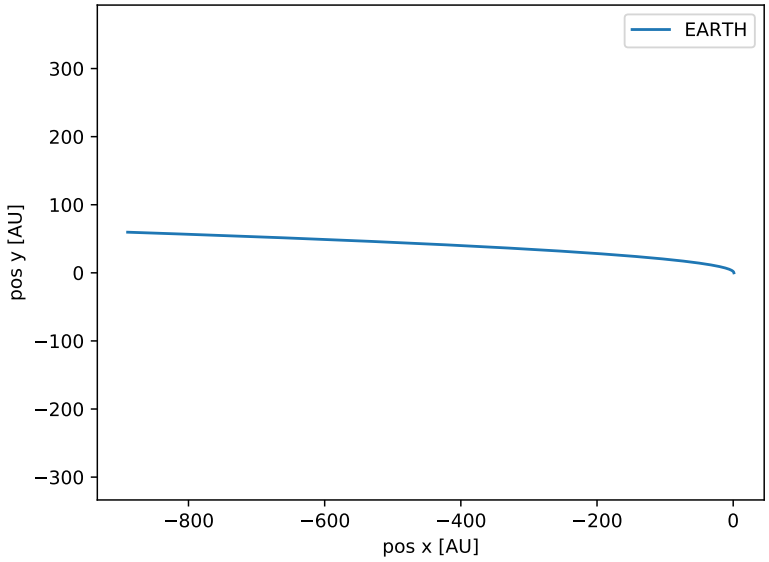

Figure 1: A plot of the simulated minimal escape velocity over 2000 years

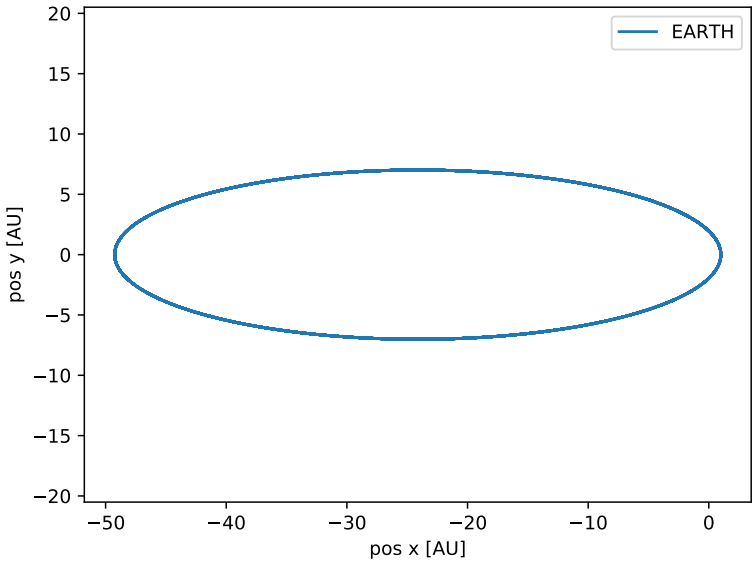

Figure 2: Initial velocity at .99 if escape velocity over 2000 years

4.b The three-body problem

The plot of the path of Jupiter and Earth can be seen in figure 7, with an elapsed time of $t = 30$yr and a timestep of 10^{-5}yr. The path when Jupiter has 10 and 1000 times more mass, with the same time interval, can be seen in figure 8 and 9 respectively.

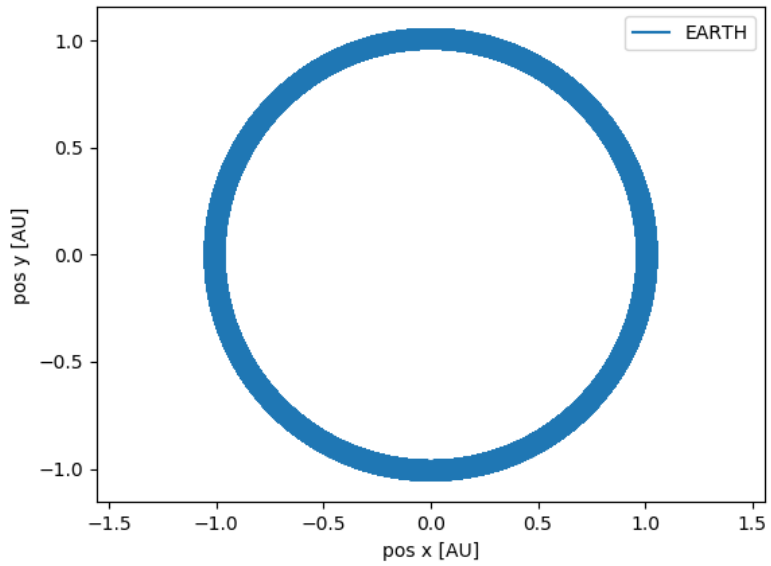

Figure 3: A 10000 year simulation with the forward euler method in 2d, $dt = \frac{1}{100} years$

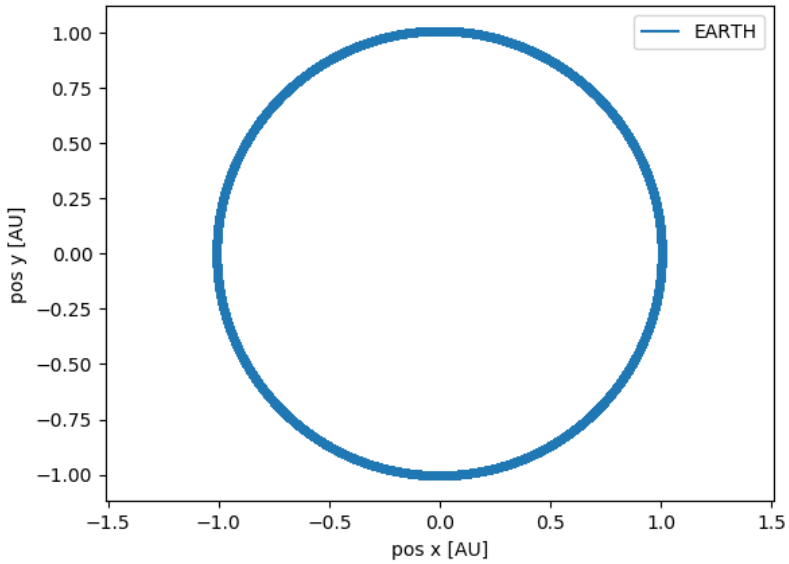

Figure 4: A 10000 year simulation with the velocity verlet method in 2d, $dt = \frac{1}{100} years$

4.c Final model for all planets of the solar system

The plot of the path of Jupiter, Earth and the Sun can be seen in figure 10, with an elapsed time of $t = 20$yr and a timestep of 10^{-5}yr. The plot of the path of all the planets can be seen in 11, with an elapsed time of $t = 300$yr and a timestep of 10^{-3}yr. The path of the planets closer to the centre can be seen in 12.

4.d Testing the code

As we can see quite clearly from figure 1 that the planet leaves its orbit, and we can see from figure 2 that there is no room for reducing the initial velocity. This means that our code works as it should.

4.e Velocity Verlet vs Forward Euler

When it comes to computation times the difference between the two methods is minimal, and a rough flop count confirms that they should be approximately equal. When it comes to accuracy there is a big difference however. Looking at the different runs in figure 5 and figure 6, we can see that as time passes the orbit that should be stable widens quite a lot more for the Forward Euler simulation, than for the Velocity Verlet simulation.

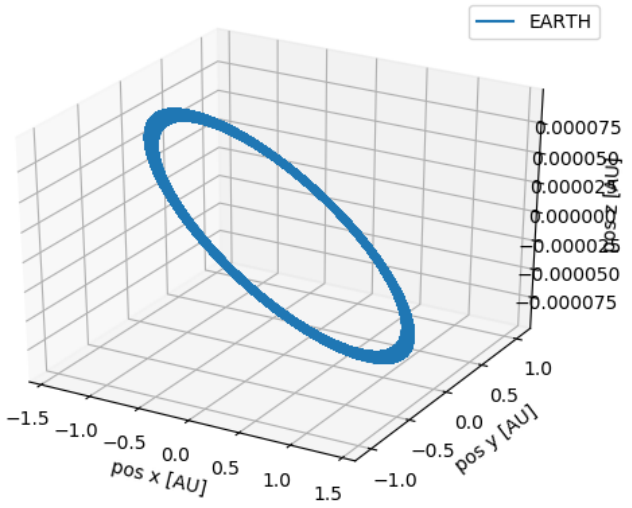

Figure 5: A 10000 year simulation with the forward euler method in 3d, $dt = \frac{1}{100} years$

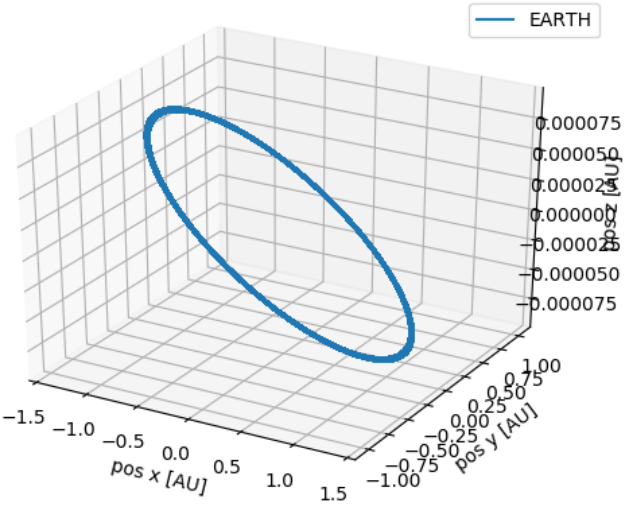

Figure 6: A 10000 year simulation with the velocity verlet method in 3d, $dt = \frac{1}{100} years$

4.f The three-body problem

We see that the normal path of Earth and Jupiter are stable, suggesting that the Verlet algorithm we have created is a good approximation. When Jupiter's mass is 10 times stronger there is no noticeable difference in the path. When it is 1000 times stronger, Earth spins around the Sun and Jupiter a couple of times, before shooting out into space, while Jupiter has a stable orbit around the sun. This suggest that the Verlet method is good enough to simulate a stable path when Jupiter's mass is 10 times stronger, but not when it is 1000 times stronger. This makes sense as when $1000 M_{\text{Jupiter}}$ is about the same as $M_{\odot}$, which would make it pull on Earth as strongly as the Sun. If we hadn't fixed the Sun in the centre of mass, then it and Jupiter would either have an unstable orbit, or we would get something close to a Binary star system.

4.g Final model for all planets of the solar system

In conclusion we see that our algorithm produces nice, stable orbits for all the planets. The only planet which seem to not be completely stable is the planet closest to the sun, Mercury, but this could just be that its path moves around more due to its closeness to the centre of mass. We can also see that we get a moon like effect between the Earth and Jupiter by increasing Jupiter's mass significantly. This suggests that our algorithm should handle both moons and random asteroids as well as planets. All in all it seems like the Verlet method produced a good approximation of our solar system.

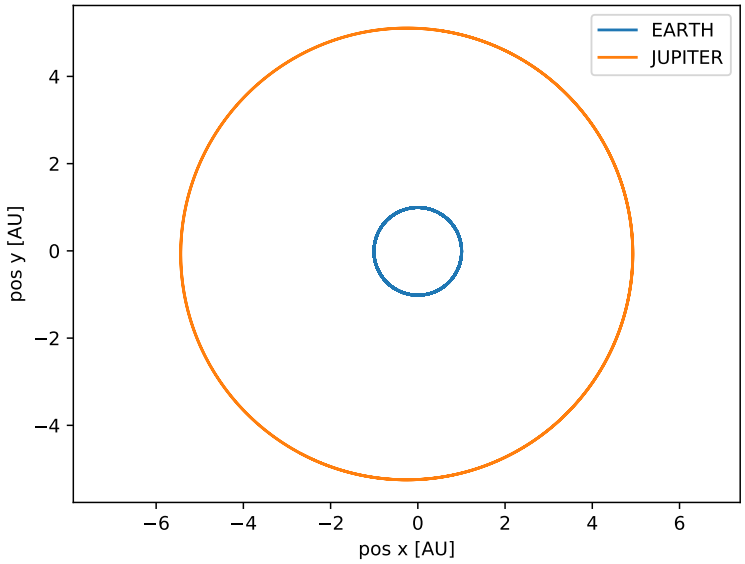

Figure 7: A plot of Jupiter and Earth's path around the sun

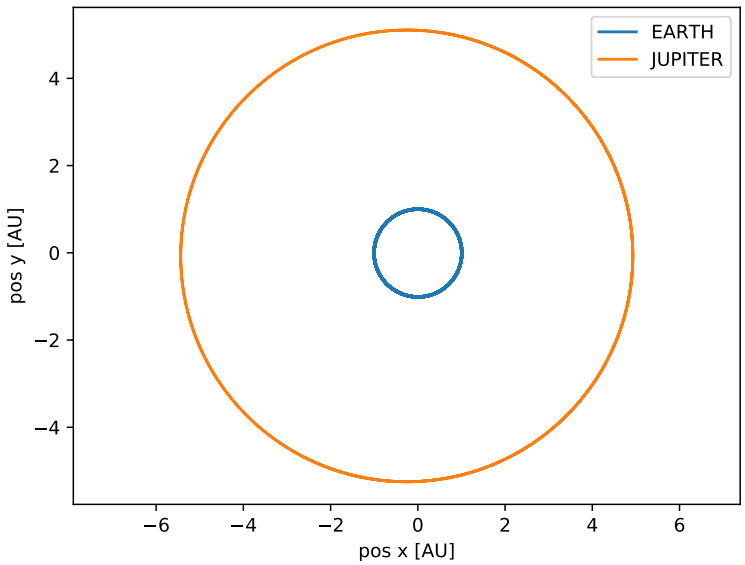

Figure 8: A plot of Jupiter and Earth's path around the sun, when Jupiter has 10 times the mass

5 APPENDICES

5.a The Velocity Verlet method math

Firstly function 8:

$$x_i^{(1)} = \frac{x_{i+1} - x_{i-1}}{2h} \Rightarrow 2hx_i^{(1)} = x_{i+1} - x_{i-1} \tag{12}$$

$$x_{i-1} = x_{i+1} - 2hx_i^{(1)} \tag{13}$$

$$x_{i+1} = 2x_i - x_{i-1} + h^2 x_i^{(2)} = 2x_i - \left(x_{i+1} - 2hx_i^{(1)}\right) + h^2 x_i^{(2)} \tag{14}$$

$$2x_{i+1} = 2x_i + 2hx_i^{(1)} + h^2 x_i^{(2)} \tag{15}$$

$$x_{i+1} = x_i + hx_i^{(1)} + \frac{h^2}{2} x_i^{(2)} \tag{16}$$

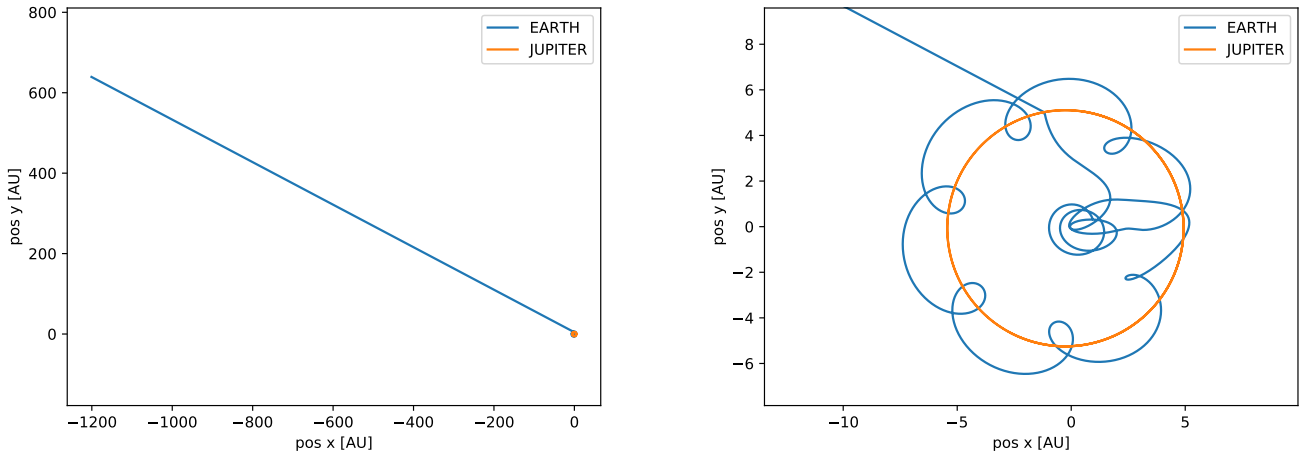

Figure 9: A plot of Jupiter and Earths path around the sun, when Jupiter has 10000 times the mass

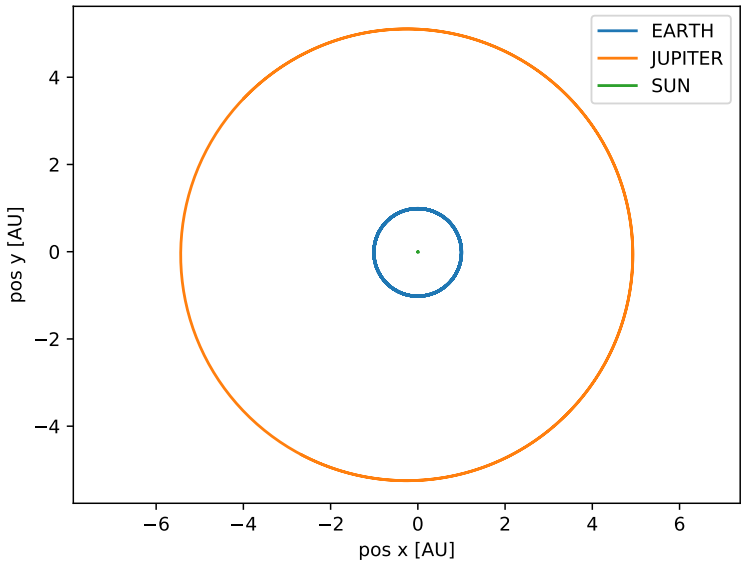

Figure 10: A plot of Jupiter, Earth and the Sun's path around the centre of mass. You can't see it here, but the Sun moves in a small circle in the centre

Then function 9:

$$x_{i+1} = x_i + hx_i^{(1)} + \frac{h^2}{2}x_i^{(2)} \Rightarrow x_i = x_{i-1} + hx_{i-1}^{(1)} + \frac{h^2}{2}x_{i-1}^{(2)} \tag{17}$$

$$x_{i+1} = x_{i-1} + hx_{i-1}^{(1)} + \frac{h^2}{2}x_{i-1}^{(2)} + hx_i^{(1)} + \frac{h^2}{2}x_i^{(2)} \tag{18}$$

$$x_i^{(1)} = \frac{x_{i-1}^{(1)}}{2} + \frac{h}{4}x_{i-1}^{(2)} + \frac{x_i^{(1)}}{2} + \frac{h}{4}x_i^{(2)} = x_{i-1}^{(1)} + \frac{h}{2}\left(x_i^{(2)} + x_{i-1}^{(2)}\right) \tag{19}$$

6 REFERENCES

References

[1] Computational Physics, Lecture Notes Fall 2015, Morten Hjort-Jensen p.215-220

[2] https://ssd.jpl.nasa.gov/?horizons#top

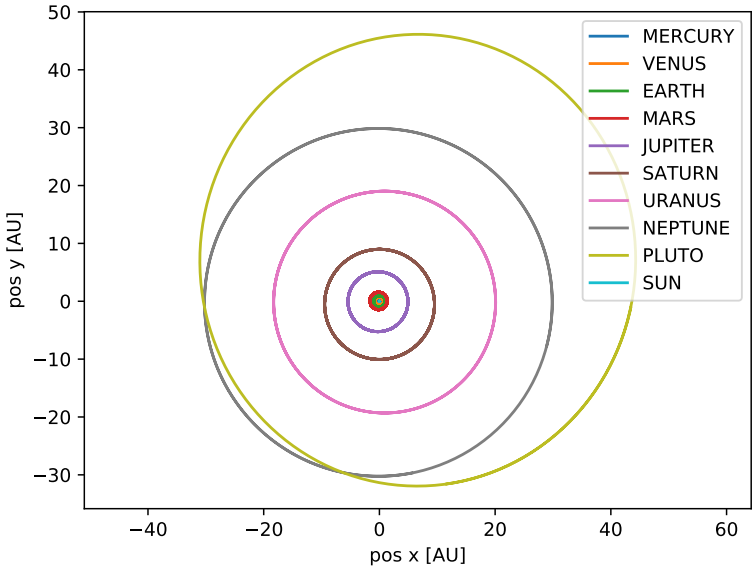

Figure 11: A plot of the Sun, all the planets and Plutos path around the centre of mass

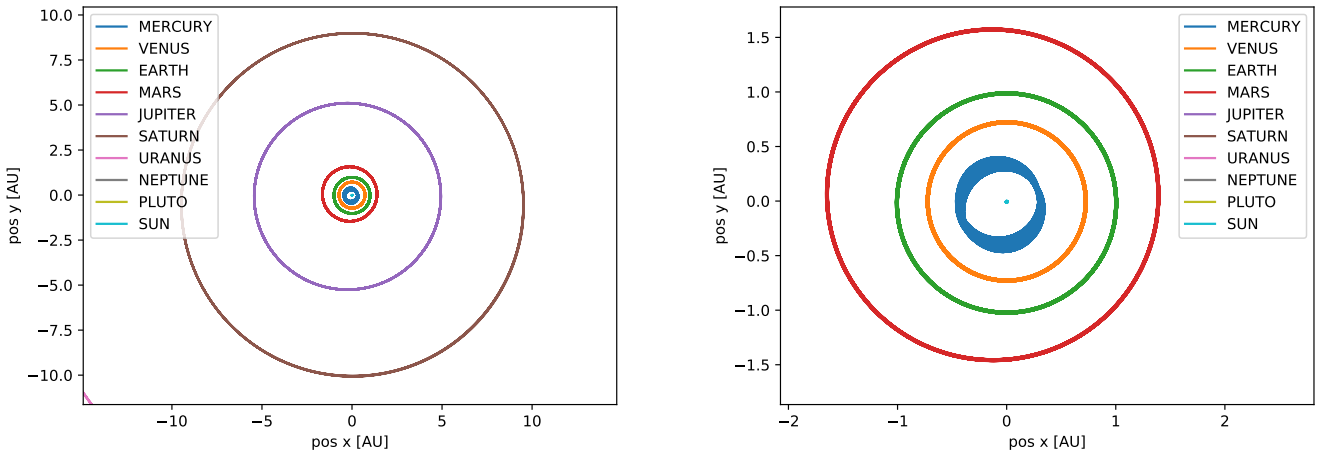

Figure 12: A plot of the Sun and the planet's path around the centre of mass when we zoom in on the planets closer to the Sun