

# Project 1

Anders Eriksen

## I. INTRODUCTION

## II. THEORY AND METHODS

The main point of focus in the project is the development of methods to solve ordinary differential equations (ODEs) from an object oriented way mainly through the philosophy of "write once, run repeatedly". The system we are studying for this, is a model solar system where the main assumptions are that the planets being point particles as well as the motion being coplanar. We naturally also assume no other forces at play besides gravity.

$$F_G = \frac{GM_\odot M_{\text{Earth}}}{r^2},$$

Initially, we also neglect the motion of the sun, as it is overwhelmingly massive. The origin of the suns orbit is far inside it's radius, which we have assumed to be zero. This will be gone into further detail on later, but for a 2 body system of earth and sun, we use newtons equations of motion (eom) to get:

$$\frac{d^2 \vec{r}}{dt^2} = \frac{\vec{F}_G}{M_{\text{Earth}}},$$

When modeling systems of this size, meters and kilograms are not the best units, as they are far too small. Instead, we will use astronomical units(AU), sun masses( $M_\odot$ ) and years(yr) as our measurement values. We have then,

$$1AU = 1.5 \times 10^{11} \quad 1M_\odot = 2 \times 10^{30}$$

For reference, I include the masses and orbits of relevant planets.[\[1\]](#)

Planet	Mass, kg	Distance to sun, AU
Earth	$M_{\text{Earth}} = 6 \cdot 10^{24}$	1
Jupiter	$M_{\text{Jupiter}} = 1.9 \cdot 10^{27}$	5.20
Mars	$M_{\text{Mars}} = 6.6 \cdot 10^{23}$	1.52
Venus	$M_{\text{Venus}} = 4.9 \cdot 10^{24}$	0.72
Saturn	$M_{\text{Saturn}} = 5.5 \cdot 10^{26}$	9.54
Mercury	$M_{\text{Mercury}} = 3.3 \cdot 10^{23}$	0.39
Uranus	$M_{\text{Uranus}} = 8.8 \cdot 10^{25}$	19.19
Neptun	$M_{\text{Neptun}} = 1.03 \cdot 10^{26}$	30.06

Initially, we ensure that our integration methods work, we make a special solution of the earth sun system, implementing the Forward Euler and velocity Verlet integration methods. Before that, however, we need to discretize our functions.

If we assume circular motion, then we can rewrite the force as

$$F_G = \frac{M_{\text{Earth}} v^2}{r} = \frac{GM_\odot M_{\text{Earth}}}{r^2},$$

and use this to get

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

here,  $4\pi$  has the dimensions listed. This comes from earth's period being 1yr and it's distance is 1AU.

```

N = nr. of timesteps
h = (t_end - t_start)/N // steplength
for i = 0 to i = N - 1 do
    |
    |           v_{i+1} = a_i \cdot h
    |           r_{i+1} = v_i \cdot h + r_i
    |
end

```

**Algorithm 1:** a Forward Euler integration

```

N = nr. of timesteps
h = (t_end - t_start)/N // steplength
for i = 0 to i = N - 1 do
    |
    |           a_i = a(r_i)
    |           r_{i+1} = h \cdot v_i + h^2 a_i / 2
    |           a_{i+1} = a(r_{i+1})
    |           v_{i+1} = h \cdot a_{i+1} + v_i
    |
end

```

**Algorithm 2:** a velocity Verlet integration

After ensuring that the algorithms are implemented correctly, I need to generalize the code into an object oriented (OO) implementation. This should constitute a main file which takes the various objects and binds them together. After looking through examples and some discussion amongst peers, I have elected to use the methods presented, providing my own "solver" class and otherwise incorporating code into my experiments. This is in part due to time constraint, but also because using and tweaking these objects is gonna bring a quicker working knowledge through usage and familiarity. building from scratch can come later. The main idea in the objects used, is to have one to create a type "planet", which one can then store in another object, the "solar system". There is a 3rd one, a "solver" which can find the next position of the planets after a time dt has passed using the combined gravitational force on the objects from the rest of the system and the individual positions and velocities.

As this is an exploration of the numerical methods, we need to explore the results. One way to do this, is to controll for the velocity resulting in circular orbit, both through trial and error, as well as through analytical calculation. This would be of interest comparing Verlet and Euler as well as the time needed to run each, in conjunction with the amount of FLOPS needed per step in the

loops. After exploring the 2 methods, further on, we'll use the Verlet method, as this maintains energy conservation.

Having explored the requirements for circular orbit, we now examine the case for escape velocity. Both through experimentation and analytical analysis. As an additional analysis, we will explore how the escape velocity changes as we change the power of  $r$  in our expression of force, especially when we let the power approach 3.

Having verified our equations, we can now complicate our image through adding another body to our problem. We want to add the greatest single contributor besides the sun, Jupiter. The 3-body problem is far more difficult to process than the 2-body problem. With 2 bodies, the earthen orbit is stable and unchanging. Adding in Jupiter, we need to take into account how much of an effect Jupiter has on the earth. Thus, we need the total force as the vector sum of the forces from the sun and Jupiter. Now we have pull from different directions over time, and this should affect the orbit of the planet. The force contribution of Jupiter is on the form

$$F_{\text{Earth-Jupiter}} = \frac{GM_{\text{Jupiter}}M_{\text{Earth}}}{r_{\text{Earth-Jupiter}}^2},$$

And we can ensure that the solutions remain stable, by checking that the energy of the system is conserved through the velocity verlet.

Once we know that our solutions are stable for more complicated systems, we can center our system around a center of mass, instead of a stationary sun. Thus, we now shift the positions to be in regard to the center of mass (COM) of the system. With this, the center of mass will serve as a "stationary" point, while all other bodies will revolve around it. To ensure the motion is conserved, we add a velocity to the sun so that the total momentum of the system is 0. Finally, we add in the remaining planets of the solar system. The initial positions are all taken from Mathias Vege after some discussion on Slack.[2].

Now that we have a good working system for our planets, we want to go into some detail on a phenomenon that was crucial to the success of the theory of relativity, namely the precession of Mercury's perihelion. When all classical effects are accounted for, the observed value of the perihelion precession is at around 43" per century. This arises from a small perturbation of the  $1/r^2$  dependency of the gravitational force which results in the effective precession of the ellipse. The correction to the force becomes

$$F_G = \frac{GM_{\text{Sun}}M_{\text{Mercury}}}{r^2} \left[ 1 + \frac{3l^2}{r^2c^2} \right]$$

Where  $l = |\vec{r} \times \vec{v}|$  is the magnitude of Mercury's orbital angular momentum per unit mass, with  $c$  as the speed of light in vacuum. We trace the angle of perihelion  $\theta_P$  with

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

or the ratio of Mercury's  $x$  and  $y$  position at perihelion. Using the perihelion speed of Mercury as 12.44 AU/yr with a distance to the sun of 0.3075 AU. As a control of the resolution, we can compare with the precession of the perihelion without the correction.

### III. RESULTS AND DISCUSSION

For the initial incarnation of the solution, just a simple run of either Verlet or Euler with the Earth-Sun system, we can see a change in the distance for the Euler solution, but not so for the Verlet solver. This indicates that the energy really is conserved in the integration. For 2 body problem, the question of a circular orbit is fairly simple, merely following the centripetal acceleration  $a = \frac{v^2}{r}$  which gives us a velocity of  $v = \frac{2\pi}{r}$

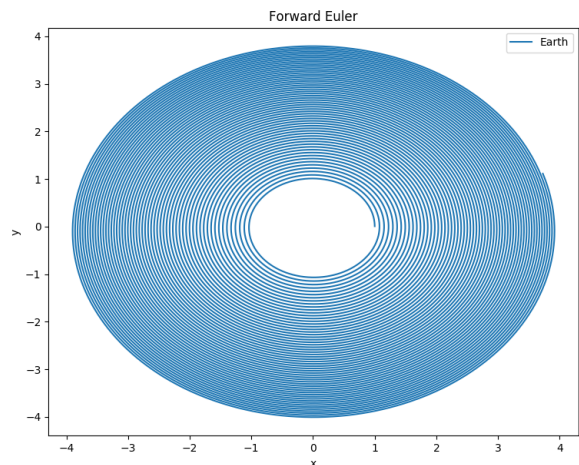


FIG. 1. Forward Euler integration of the 2 body system of a stationary sun and earth. There is a clear drift as time goes by, which emphasizes the drift in energy of the system with the Euler, since the integrations are done in a conservative potential field.

The Forward Euler integration has a clear drift over time as seen in figure 1, stemming from a failure to conserve the energy of the system. The velocity verlet, however maintains the energy, see figure 2, as well as having a numerical error proportional to steplength  $h^2$ , as opposed to Euler's  $h$ . We can also see that the resulting orbit is circular, as we should expect from the initial velocity of the simple system.

When calculating escape velocity, I used a distance  $r = 1AU$  and used the fact that the energy would need to be either 0 or greater if the planet was to escape. Therefore, I set  $E_k = E_p$  and found a velocity of  $v = \sqrt{\frac{8\pi^2}{r}}$ , the initial distance meant  $v = 2\sqrt{2}\pi$ . This is illustrated in figure 3, where we see that the motion is less and less affected by the star with distance. As this is the exact

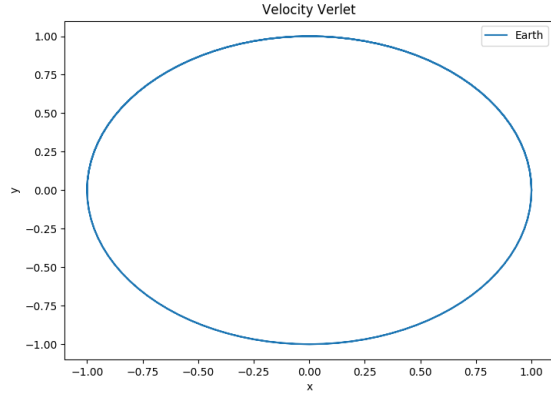


FIG. 2. Velocity Verlet integration of the 2 body system of a stationary sun and earth. This integration is run with as many steps as figure 1, which shows how much more robust the Verlet solution is compared to the Forward Euler. We can also see from the axes that despite the seeming elliptical orbit, this has more to do with improper scaling. The radius is 1 AU in all directions.

escape velocity, it will not disappear until the distance is at infinity. Currently, the velocity goes according to  $v \propto \frac{1}{\sqrt{r}}$ , but if we were to increase the density dropoff of the gravitational force according to the cube of the distance, then the escape velocity would approach  $v \propto \frac{1}{r}$ .

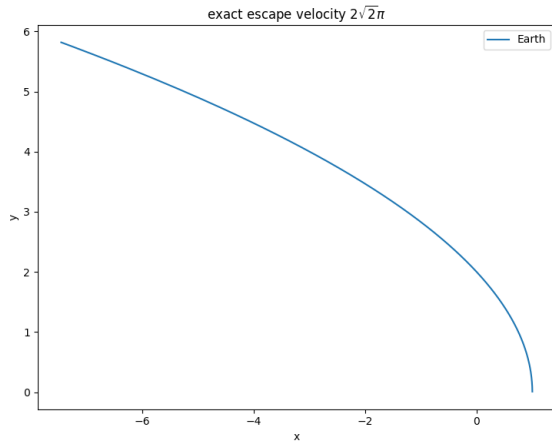


FIG. 3. this is the earths "orbit" around the sun with an escape velocity so  $E_k = E_p \rightarrow E = 0$ . For a system at a distance of 1AU, escape velocity is, as stated in the title. Measurements show that just below leads to a stable orbit with a change from  $2 \rightarrow 1.9$  in front of the square root.

At this point, we begin to leave the realm of analytical solutions explored by Newton and his contemporaries. Here, we see that the inclusion of Jupiter squishes the circular orbit of the Earth into an ellipse. This is also a challenge to pick good values for initial conditions. Therefore,

I use information on planet positions and velocities[2] as mentioned above. Varying Jupiter's mass, we can see

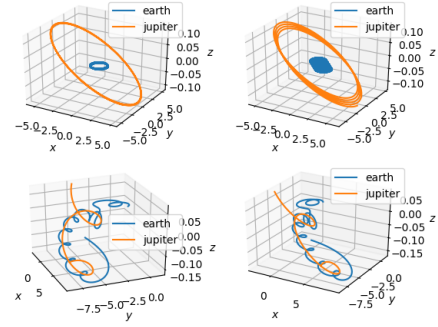


FIG. 4. The orbits of Earth and Jupiter around a stationary sun for varying masses of jupiter. We can see as we increase the mass from Jupoter's regular mass(top left) by 10(top right) and 1000(bottom row) respectively, we get increasingly rampant orbits, with the lower spiraling out completely. We can see that as Jupiter's mass approaches that of the sun, it destabilizes the system.

that it provides increasingly greater distractions from the sun with regards to the Earth untill Jupiter is as massive as the sun. At this point, the system breaks down and spirals outwards. What we can notice throughout these, is that the momentum of the system remains the same to a relative error  $\epsilon < 1e-5$ , which means that the Verlet method still holds even under these strange developements.

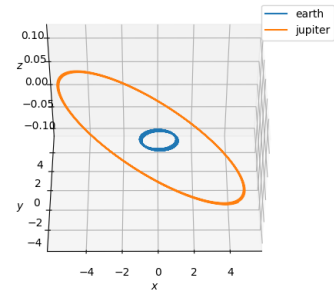


FIG. 5. An enlarged version of the first plot in figure 4. We can see, that the previously circular orbit of the earth is pushed to be elliptical. We also see some slight axial tilt in the axis. the Verlet method retains angular momentum to a great deal, and this should be close to the effect of the planet, rather than iancuracies in the numerical method.

As we have now more or less verified our numericals for various systems and number of planets, we can begin to add in both the motion of the sun as well as the remaining planets(excluding Pluto which is currently not within this deifnition) in figures 6 and 7. We can observe

a gap between the orbits that seem to almost increase exponentially with the distance from the sun. We can also see that the 5 decades of time development does not seem to be enough to complete the orbits of the furthest planets. This makes sense, as the gravitational field reduces with  $\frac{1}{r^2}$ , and so the resulting velocities that remain bound to the system have to be significantly lower, as the kinetic energy is proportional to the square of the velocities. we can also see, that the initial assumption of coplanar motion is off, but not by terribly much over all, with some tilt in either direction.

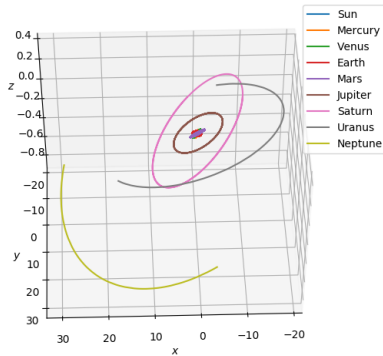


FIG. 6. The full solar system (excluding Pluto). The data is gathered over 5 decades, which is not quite enough time for the outermost planets to complete an orbit, but we see that Saturn and closer planets have seemingly closed orbits. There is a distinct distance between the outer planets Neptune, Uranus, Saturn and, to a degree, Jupiter and the inner planets beginning with Mars. This inner part has far tighter orbits in comparison.

A final consideration, is the accuracy of our potential calculations. We assumed the system followed the newtonian gravity, which seems to be fairly accurate on the whole. But there are corrections that can be made with regards to the theories of relativity. In particular, an important test of Einsteins theories was the perihelion precession of Mercury. The perihelion precession was measured to be  $43''$  per century. Newtonian gravity predicts several orders of magnitude less in this timeframe. To examine this, I ran a system of pure Sun-Mercury and set Mercury's initial values as those in the perihelion. Letting the simulation run for a century with a sampling of  $1e8 - 1e9$  points, I found a precession of  $42.8''$  for the relativistically corrected equation presented in the theory above. While the newtonian system predicts a precession of  $0.02''$ . this shows that the relativistic model is about 1000 more accurate at predicting systems of this magnitude than newton's.

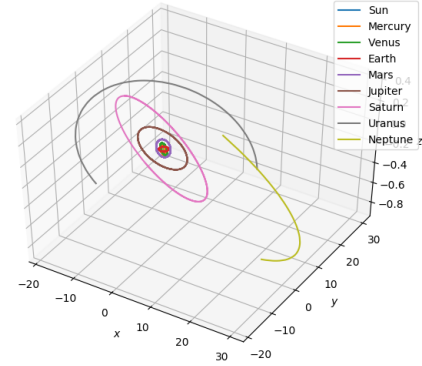


FIG. 7. A shifted angle of the same plot as figure 6, in an attempt to better showcase the inner "nucleus". We can see a progressively closer orbits as we approach the sun, almost seeming to follow an inverse proportionality to the distance from the sun.

#### IV. CONCLUSION

As we built up our program and system, we saw that the velocity Verlet method vastly outperformed the Forward Euler, maintaining a good approximation to the plante's path with around a few hundred integration points a day during our revolutions around the star and maintaining a near constant angular momentum through the years in our isolated system.

As we added complexity to our system, we also developed a program with objects that could compartmentalize the required tasks and feed the finished results into eachother, resulting in a program that could near seamlessly add planets and calculate their positions and velocities over time without more than an added line. This object orientation mainly revolved around 4 objects. A "celestial body" class, which took in the planets and assigned them positions, velocities and other physical observables which could then be accessed down the line. A "system" class which stored all the planets and could work out the forces inbetween each of them and a "Solver" class, which could solve the equations of motion for the planets in the system a step forwards in time. This way of handling things allowed for far less repetition of definitions and helped insulate mistakes.

While the systems we set up worked well, there are a few things that I did not have the time to fully automate, such as the force used, or the calls to various amounts of planets. Ideally, I envision a system in which I can give a filename and the program will implement a planet with name, position, vellocity and other attributes to create the system I wish to study. further, I would like to customize the span of the integration in both length

and timestep or number of steps. The main issue I encountered on this front, was in interpreting the input. I believe a solution here could be a textfile as input, with relevant info such as data files, integration length and similar, while the system itself has perhaps another object that can interpret input from file and put the results where they ought to go. Finally, I should have specified a folder to write the results into.

Another solution could be a python script to handle inputs and outputs, with plots as necessary. This would be a more involved project than what time allows for this time around.

## V. APPENDIX

- 
- [1] M. Hjorth-Jensen, <http://compphysics.github.io/ComputationalPhysics> (2018).
  - [2] M. Vege, Slack (2018).