FYS3150 Computational Physics - Project 3

Nicholas Karlsen

This is an abstract

INTRODUCTION

Lastly, the source code for any code discussed in this report can be found on my Github at: https://github.com/nicholaskarlsen/FYS3150

THEORY, ALGORITHMS AND METHODS

Newton's law of universal gravitation

Between every body, there is a force of attraction inversly proportional to the square of the separation, or more precisely, the force acting on some body with mass m due to a mass m' is

$$\mathbf{F} = G\frac{mm'}{|\mathbf{r}-\mathbf{r}'|^2}\hat{\mathbf{u}}_{\mathbf{r}-\mathbf{r}'}, \quad \hat{\mathbf{u}}_{\mathbf{r}-\mathbf{r}'} = \frac{\mathbf{r}-\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|} \tag{1}$$

Where G is the gravitational constant and $\mathbf{r}$ denotes the vector from the body of mass m to mass m'

If we choose to work in the cartesian coordinate system centered on the body with mass m', then $\mathbf{r} = -(x, y, z)$ and $|\mathbf{r}| = \sqrt{x^2+y^2+z^2} = r$.

By Newtons second law, the acceleration on body 1 due to the gravitational pull of body 2 can then be written as

$$\mathbf{a} = \frac{1}{m}\mathbf{F} = G\frac{m'}{r^2}\frac{\mathbf{r}}{r} = -G\frac{m'}{r^2}\frac{(x,y,z)}{r} \tag{2}$$

Written out component-wise in terms of the positions, we get the set of coupled differential equations

$$\frac{\partial x^2}{\partial t^2} = -G\frac{m'}{r^2}\frac{x}{r}, \quad \frac{\partial y^2}{\partial t^2} = -G\frac{m'}{r^2}\frac{y}{r}, \quad \frac{\partial z^2}{\partial t^2} = -G\frac{m'}{r^2}\frac{z}{r} \tag{3}$$

Solving ODE numerically

Forward Euler

Consider a function $y(t)$, which derivative, $y'(t,y)$ has a known form.

Velocity-Verlet Algorithm

```
for  $i = 0, \dots, N-1$
    $\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_i\Delta t + \frac{1}{2}\mathbf{a}_i(\Delta t)^2$
    $\mathbf{v}_{i+1} = \mathbf{v}_i + \frac{1}{2}(\mathbf{a}_{i+1} + \mathbf{a}_i)\Delta t$
```

Forward Euler Algorithm

```
for  $i = 0, \dots, N-1$
    $\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i\Delta t$
    $\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_i\Delta t$
```

Euler-Cromer Algorithm

```
for  $i = 0, \dots, N-1$
    $\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i\Delta t$
    $\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+1}\Delta t$
```

RESULTS AND DISCUSSIONS

Object orientation

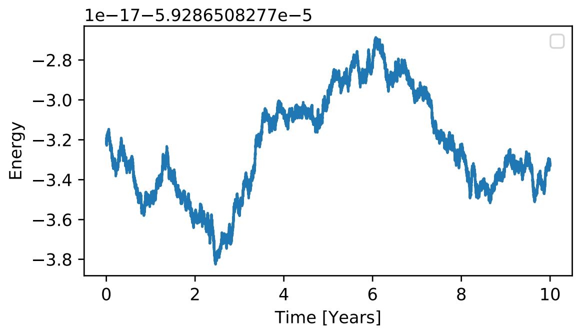
When designing the class solarsystem, i wanted to strike a balance between utilizing the benefits, and expandability you get from object orientation whilst also not abstracting the data too much. As such, the class simply manipulates arrays of a particular format in a particular way. Not abstracting the operation by creating objects for each planet (or something else of that nature), which seems to be a common pitfall of object orientation as i percieve it.

A particular benefit in the way i wrote my code, is that there is no difference if the input data is in 2 Dimensions or 3. The code will work just the same either way by making full use of the way Numpy arrays work.

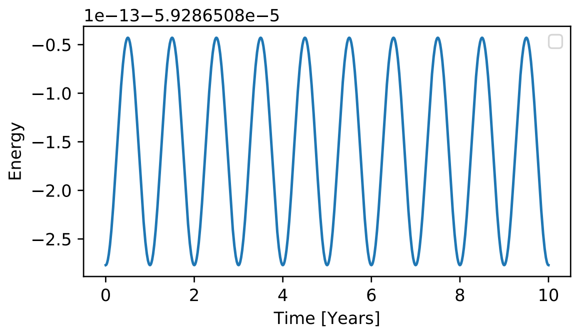
CONCLUSIONS

In trying to generalize my code through the use of object orientation, i had to strike a balance between generality and complexity

[1] M. Hjorth-Jensen, Computational Physics - Lecture Notes 2015, (2015).

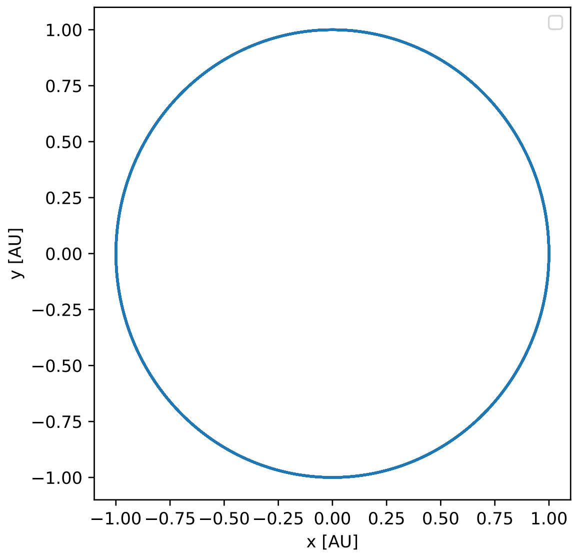


(a) Velocity-Verlet



(b) Euler-Cromer

FIG. 1: The fluctuation of the total energy of the Earth in the Earth-Sun system for solutions using the Velocity-Verlet algorithm (a) and the Euler-Cromer algorithm (b)



Fetching data from Horizons using horizons.py

In order to streamline the process of fetching data from the Horizons system i created a small script, horizons.py that utilizes the Astroquery python package and returns only the select data that i am interested in. The function, fetch_data takes input as a dictionary, rather than a list of id numbers. Whilst this may not be as extensible or practical, it makes the code easier to read and understand, by making the instant connection between the planet name and id. For the purposes of this project, i find that much more valuable since i am only dealing with a limited amount of planets anyway.