Report

of

Exercise sheet 1:

Orbit / Satellite motion

for

269002 VO Computational Concepts

in Physics I (2021W)

Student name:	Clemens Wager, BSc
Matriculation nr:	01635477
Degree program:	Master of Computational Science
Supervisor:	Univ.-Prof. Dr. techn. Cesare Franchini

Vienna, 10.01.2022

Introduction

The two-body problem belongs to classical mechanics and finds application in astrophysics. It can be used to predict the motion of two massive objects, whose gravitational forces interact.

A simulation of this behavior was studied in this computer experiment. Specifically, the trajectory of a satellite orbiting the sun. Moreover, the simulation confirms Kepler's third law of planetary motion.

Method

Theory

The main goal was to simulate the satellite's motion for one complete orbit in two dimensions. Therefore, we must compute its positions at different timesteps. The orbits center is the sun which has a mass of $M = 1.99e30\,kg$ and for reasons of simplicity the satellite has a mass of $m = 1\,kg$. The universal gravitational constant $G = 6.67e-11\,m^3kg^{-1}s^{-2}$. The satellite has an initial position r and an initial velocity v. Both variables have an x and y component as the simulation takes place in 2D space.

We can calculate the gravitational force that pulls the satellite into the center:

$$F_G = \frac{GmM}{|\mathbf{r}|^3}\mathbf{r}$$

From Newton's second law of motion we have:

$$\begin{aligned} \frac{dx^2}{dt^2} &= \frac{F_{G,x}}{m} \\ \frac{dy^2}{dt^2} &= \frac{F_{G,y}}{m} \end{aligned}$$

Thus we have can define the x and y components of FG as

$$\begin{aligned} F_{G,x} &= -\frac{GmM}{r^2}\cos\theta = -\frac{GmM}{r^3}x \\ F_{G,y} &= -\frac{GmM}{r^2}\sin\theta = -\frac{GmM}{r^3}y \end{aligned}$$

We plug in the definition of FG to get

$$\begin{aligned} \frac{dx^2}{dt^2} &= -\frac{GMx}{r^3} \\ \frac{dy^2}{dt^2} &= -\frac{GMy}{r^3} \end{aligned}$$

and split the two second order differential equations (DE) into two first order DEs like so:

$$
\begin{aligned}
\frac{dv_x}{dt} &= -\frac{GMx}{r^3} \\
\frac{dx}{dt} &= v_x \\
\frac{dv_y}{dt} &= -\frac{GMy}{r^3} \\
\frac{dx}{dt} &= v_y
\end{aligned}
.
$$

In our solar system GM is a constant that can be defined as $GM = 4\pi^2\, AU^3$. When we plug in this constant we get

$$
\begin{aligned}
\frac{dv_x}{dt} &= -\frac{4\pi^2 x}{r^3} \\
\frac{dx}{dt} &= v_x \\
\frac{dv_y}{dt} &= -\frac{4\pi^2 y}{r^3} \\
\frac{dx}{dt} &= v_y
\end{aligned}
.
$$

Now we have all the equations we need to calculate the acceleration and the velocity at time t. Thus, we can compute future positions that lie on the satellite's trajectory through a series of discretization steps. Position and velocity will be propagated with the help of the *Euler-Cromer method.* The accuracy of this process can be increased by decreasing the timestep Δt.

Algorithm

To propagate the satellite's trajectory the Euler-Cromer Method was used. This algorithm is well-suited for systems where energy is conserved. Below is a Pseudocode of the algorithm and the python implementation from the program code.

Algorithm 1 Euler-Cromer Method

```
1: $v_{x,i+1} = v_{x,i} - \frac{4\pi^2 x_i}{r_i^3}\Delta t$
2: $x_{i+1} = x_i + v_{x,i+1}\Delta t$
3: $v_{y,i+1} = v_{y,i} - \frac{4\pi^2 y_i}{r_i^3}\Delta t$
4: $y_{i+1} = y_i + v_{y,i+1}\Delta t$
```

Algorithm 1 is the Euler-Cromer Method written in pseudocode. v is the velocity and x_i and y_i are the x and y coordinates of the position.

```
def euler_cromer(GM, dt, time, r, v, normR):
    """
    Implement Euler-Cromer method in function
    Input:
      GM: precalculated G*M
      dt: timestep
      time: current time
      r: current position
      v: current velocity
      normR: distance to sun
    Return:
      time: updated time
      r: updated position
      v: updated velocity
    """
    # update acceleration
    accel_x = -GM * r[0] / (normR ** 3)    # accel x
    accel_y = -GM * r[1] / (normR ** 3)    # accel y

    # update velocity
    v[0] = v[0] + dt * accel_x      # update v_x
    v[1] = v[1] + dt * accel_y      # update v_y

    # update position
    # The Euler-Cromer step (using updated velocity)
    r[0] = r[0] + dt * v[0]         # update r_x
    # The Euler-Cromer step (using updated velocity)
    r[1] = r[1] + dt * v[1]         # update r_y

    time = time + dt                # update time
    return time, r, v   # return updated values
```

Equations solved

To make all the measurements the following equations were solved:

Semi-major axis a, where r_{max} is the position that is the furthest away from the sun and r_{min} the position closest to the sun:

$$a = \frac{r_{\max} + r_{\min}}{2}$$

Semi-minor axis b:

$$b = \sqrt{r_{\max} r_{\min}}$$

Eccentricity e:

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

Perihelion distance (closest point to the sun) with the semimajor axis a and eccentricity e:

$$p = a(1 - e)$$

Total energy in the system E (Total energy = Kinetic energy – Potential energy):

$$E = \frac{1}{2}mv^2 - \frac{GmM}{r}$$

Implementation

The simulation was implemented in Python 3.7 and uses some libraries for scientific computing: math, numpy and matplotlib.pyplot. The implementation incorporates some parts of the source code given in the lecture. Before executing the program, the user can adjust the simulation parameters and turn functions on and off to run different experiments. For example, if the bool-variable `confirmKeplersThirdLaw` is set to `True`, the program specifically simulates and plots several orbits and then plots a graph to illustrate the ratio described by Kepler's third law.

Results and Discussion

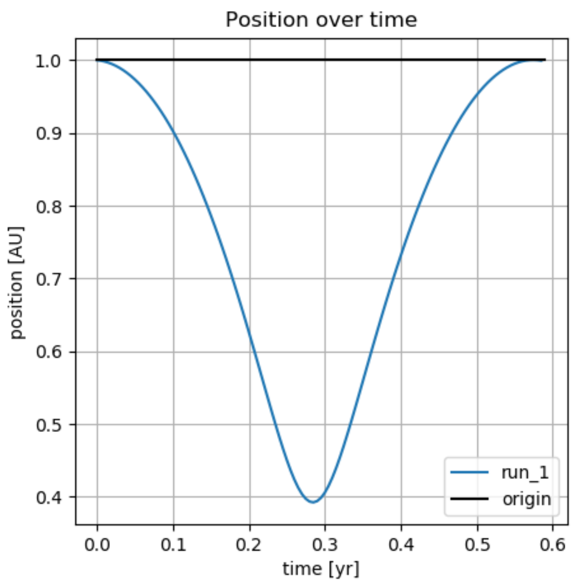
In order to validate the accuracy of the simulation the program was executed with the given initial parameters: position $r_{x,y} = (1,0)$, the initial velocity $v_{x,y} = (0, \frac{3\pi}{2})$ and the timestep $dt = 0.005\, yr$.

Table 1 shows that there were small deviations between the resulting measurements and the expected results. The differences probably come from round-off errors and can be decreased with smaller timesteps.

Expected result	Actual result
$T \approx 0.58\, yr$	$T \approx 0.59\, yr$
$e = 0.436$	$e = 0.4365$

Table 1: Expected result and actual result to examine simulation accuracy.

The satellite's motion was plotted to visualize the trajectory. In the left plot of Figure 1 we see that the period is about 0.59 years on the x-axis. Further, the perihelion distance $p = 0.392\, AU$ is represented by the global minimum. From the right plot it becomes obvious that the orbits eccentricity is far from 1, which would be a perfect circle. Moreover, the semimajor axis measures 0.696 AU and the semiminor axis 0.627 AU.



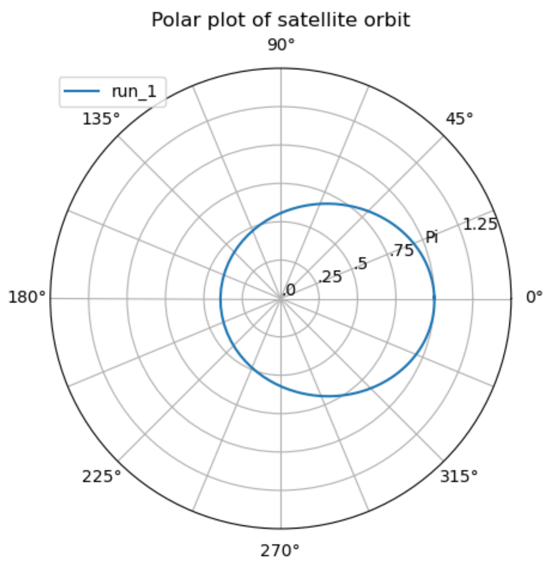


Figure 1 illustrates the orbit's trajectory as blue "run_1" line. On the left side, there is the satellite's position over time with time in years on the x-axis and distance from the sun in AU. The distance from the sun was calculated as $||r0 + r1||_2$, where $r1$ is the subsequent position of $r0$. The black "origin" line marks the initial position. On the right side, there is a polar plot visualizing the satellite's angular displacement over the course of one orbit.

The second aim of this computer experiment was to confirm Kepler's third law (KTL) of planetary motion with the simulation program. KTL states that

$$\frac{a^3}{T^2} = \frac{G(M+m)}{4\pi^2} \approx \frac{GM}{4\pi^2} \approx 7.496 \cdot 10^{-6} \left(\frac{\mathrm{AU}^3}{\mathrm{days}^2}\right) \text{ is constant},$$

where a is the semimajor axis and T is the period.

The orbits plotted for of this simulation experiment can be seen in the appendix figure 3. In order to show that the program follows KTL the data was visualized on a loglog plot in green against a black line that follows KTL exactly. Obviously, the gradient of the graphs differs which is probably caused by round-off errors. Interestingly, in the lower section (the first 6 datapoints) the green line is runs almost flat. However, the black line also has a knee in the lowest visible datapoint. Overall, the simulation follows along KTL but for reasons unknown the first few measurements do not match the ideal line.

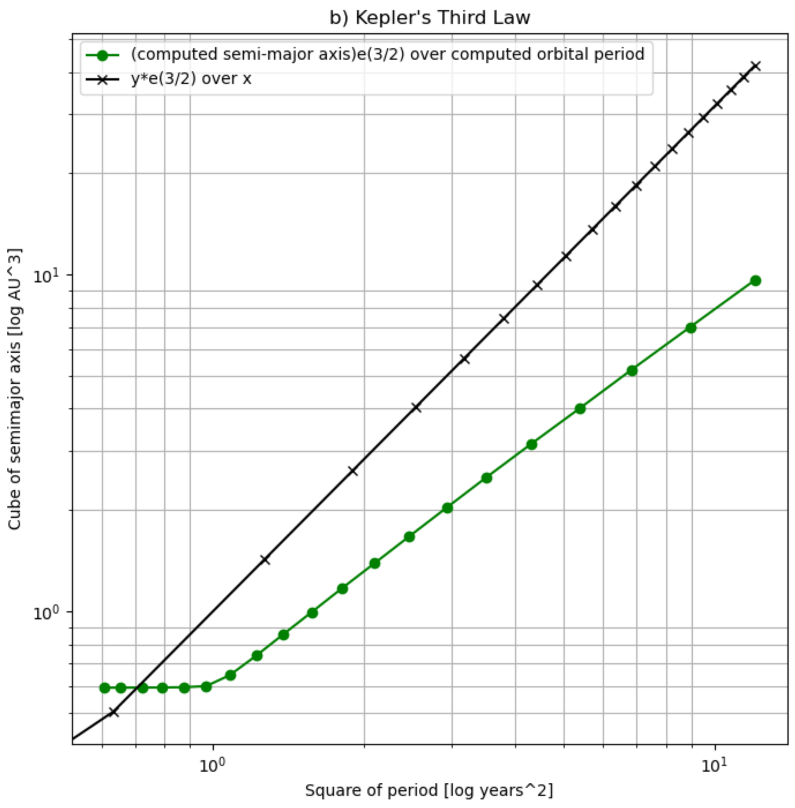


Figure 2 is a loglog plot of two graphs. The x-axis is the square of the period, and the y-axis is the cube of the semimajor axis (both axes are log scaled). The black line represents the exact ratio of period and semimajor axis described in Kepler's third law. The green line is made of the corresponding data measured in 20 simulations.

Conclusion

In this computer experiment the trajectory of a satellite orbiting the sun was studied in a closed system. Therefore, a simulation incorporating the Euler-Cromer method was implemented. To study the system the program took certain measurements and graphically illustrated the results. Further the measured values were compared to their analytical solutions. Moreover, the simulation could also confirm Kepler's third law of planetary motion to some degree.

Appendix

Size	Value
period T	0.59 yr
total energy E	-28.3751 M AU^2 / yr^2
semimajor axis a	0.6964 AU
semiminor axis b	0.6265 AU
eccentricity e	0.4365
perihelion distance p	0.3924 AU
maximum r	1.0003 AU
minimum r	0.3924 AU

Table 2: All measurements from simulation with validation parameters.

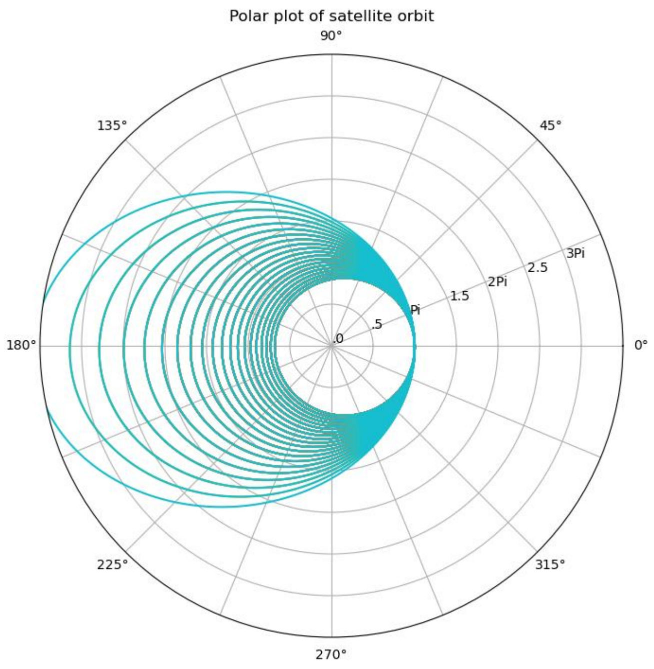


Figure 3 shows the 20 orbits that were simulated to gather data for the task of confirming Kepler's Third Law. Their measurements (period and semimajor axis) were used to plot the graph on Kepler's third law.