PHY407: Lab 1

Sky Kapoor and Chris Compierchio

September 16th, 2022

Work Distribution: We worked on this assignment together in-person, so it was split quite evenly. We each wrote one pseudocode. The base code for question 1 was written together, and the structure of this code was used again in question 2. We alternated between parts for the rest of the questions, and then switched and checked each other's work at the end.

Question 1

a) Rearranging equations, nothing to submit

b) Pseudocode and explanatory notes:

1. Define our constants: G as $39.5AU^3M_S^{-1}yr^{-2}$, M_S as 1 solar mass.
2. Pick a value for Δt, as well as a time period to integrate over.
3. Set values for initial conditions: $v_{x0}, v_{y0}, x_0, y_0, t_0$.
4. Write down position and velocity equations using rearranged equations from part a.
5. Define r as $\sqrt{x^2+y^2}$.
6. Using the Euler-Cromer method, update the velocity.
7. Use this updated velocity to update the position.
8. Repeat steps 5 and 6 until the chosen time period is integrated over.
9. Plot velocity vs. time and x vs. y.

Explanatory notes:
The code stores the initial conditions and constants in variables as outlined in the pseudocode. Then, it creates empty arrays for each value. The Euler-Cromer method uses the value x_{i+1} to calculate the value of v_{i+1} instead of just x_i as seen in the Euler method. We use this value when writing down the position and velocity equations in our code. Then, we update the positions with new velocities in order to continue the loop. The arrays for the values of x and y, as well as vx and vy, are completely filled when the loop is completed.

c) Python code and plots

All Python code is included in the Quercus submission as .py files.

Notice the velocity vs time plot (figure 1) is perfectly periodic, and the X vs Y plot (figure 2) is elliptical with no perturbations. This means that angular momentum is conserved throughout the orbit of Mercury about the Sun.

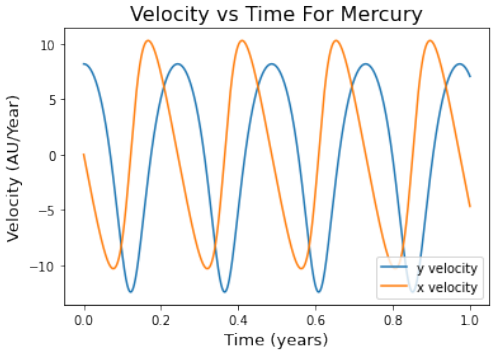


Figure 1: Velocity vs. Time plot for Mercury

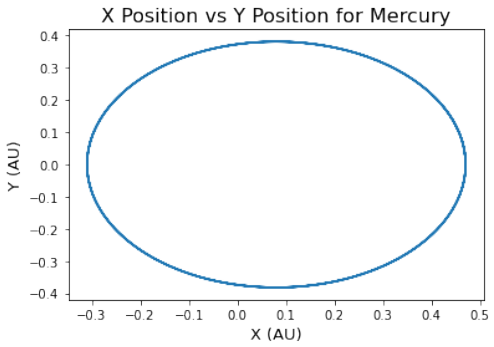


Figure 2: Orbital precession for Mercury

d) Altered gravitational force

Notice the velocity vs time plot (figure 3) is not perfectly periodic, and the X vs Y plot (figure 4) is elliptical but with many perturbations. This means that angular momentum is NOT conserved throughout the orbit of Mercury about the Sun, and that the aphelion moves in time.

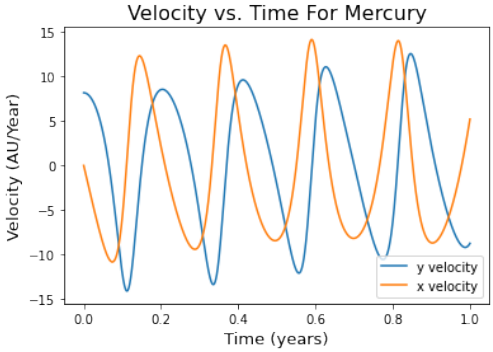


Figure 3: v vs. t, altered gravitational force

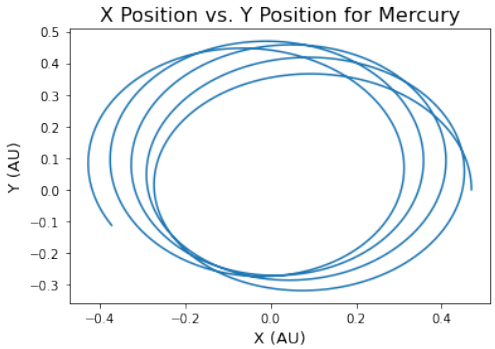


Figure 4: Orbital precession, altered grav. force

Question 2

a) Pseudocode and explanatory notes:

1. Define constants as in Question 1, as well as: M_J as $10^{-3}M_S$ and a_J as $5.2AU$.
2. Pick a value for Δt, as well as a time period to integrate over.
3. Set values for initial conditions for Jupiter: x_J, y_J, v_{xJ}, v_{yJ}
4. Set values for initial conditions for Earth: x_E, y_E, v_{xE}, v_{yE}
5. Write down position and velocity equations for Jupiter around the Sun.
6. Write down position and velocity equations for Earth around the Sun.
7. Calculate distance between Jupiter and Earth as a function of time.
8. Determine the gravitational force on Earth from Jupiter and the Sun.
9. Plot Jupiter and Earth's orbits.

Explanatory notes:
This code works very similarly to the code from Question 1, and much of the same structure is

used. The difference is that arrays are made for the x, y, vx, and vy values of both Jupiter and the Earth. This allows us to plot two distinct orbits and compare them with one another.

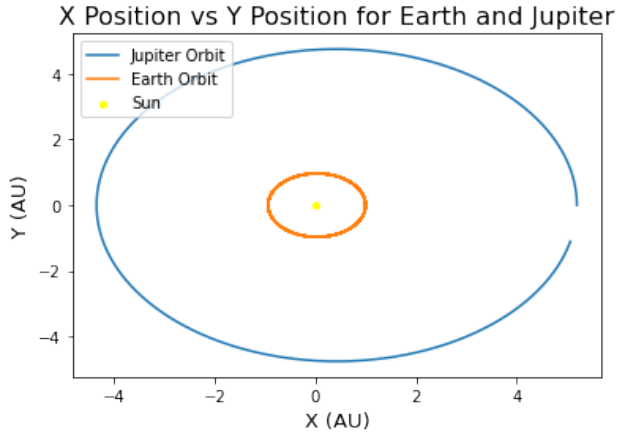


Figure 5: Orbits of Jupiter and Earth for a 10-Earth-Year period

Notice the incomplete orbit for Jupiter – this is because Jupiter's orbital period is equal to 12 Earth years, and the program was run for 10 Earth years.

b) Plots and long-term behaviour

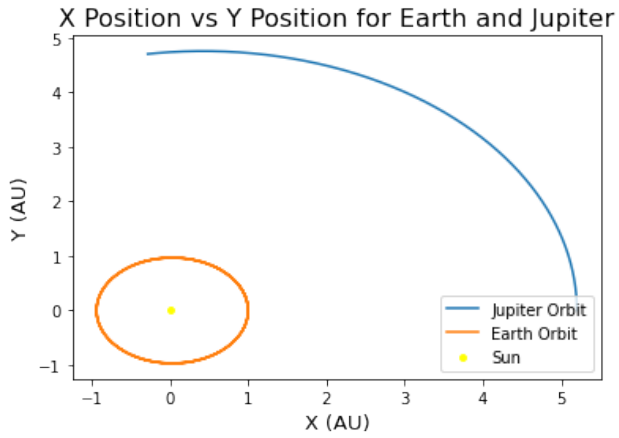


Figure 6: Long-term behaviour for Earth as influenced by 1000x Jupiter's mass

As seen in the figure above, Jupiter's orbit shows that it gets larger over time – its long-term behaviour shows an increase in orbital radius.

c) Asteroid

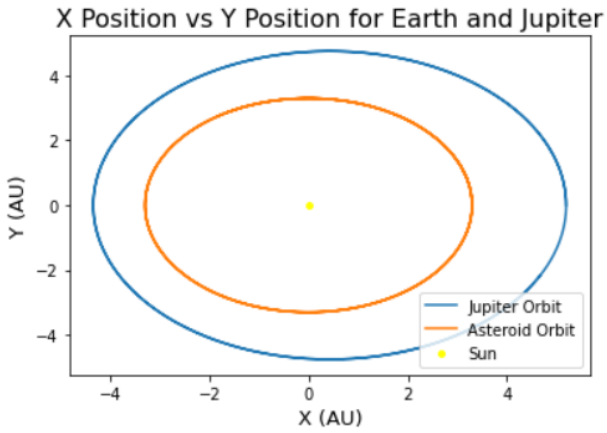


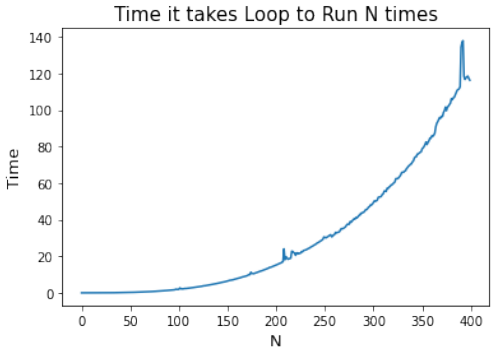
Figure 7: Asteroid orbital behaviour with Jupiter and Sun

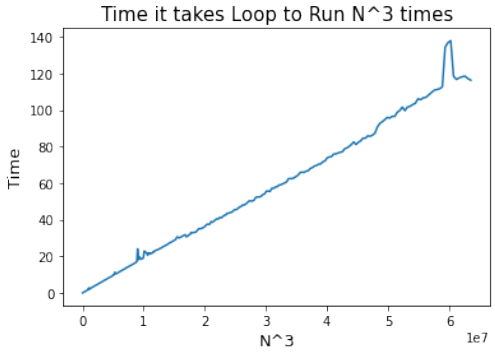
Despite the instructions, we failed to see any significant perturbations in the asteroid's orbit. Attempts to rectify this error were fruitless.

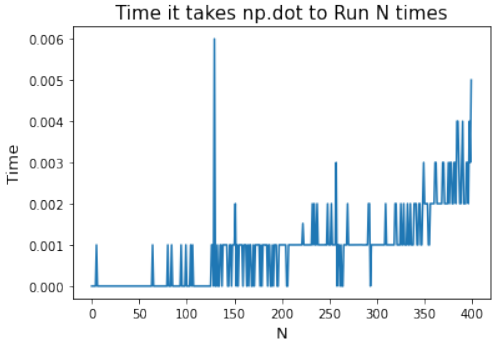
Question 3

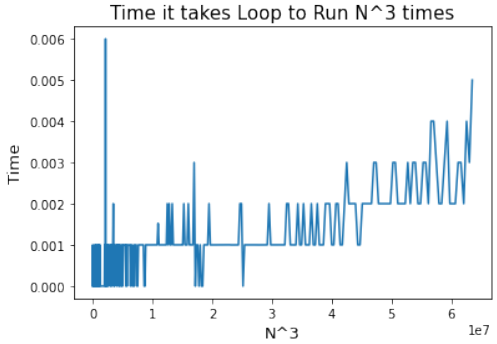
Timing matrix multiplication

Using the code fragments from the textbook, Python takes a significant period of time to carry out the desired operation when the value of N gets high. Using numpy.dot, we can see that it takes far less time for Python to carry out the same operation. The time fluctuates a lot more using numpy.dot, but this is seemingly random. As predicted by the textbook, we see a nonlinear result for N and a linear result for N^3.









A comparison between the two methods at a value of N = 100 (for runtime purposes) is shown in the plots below.