Homework 3: Ordinary Differential Equations, Part 1

Caden Gobat

February 17, 2021

1 Introduction

Last week we explored some simple integration techniques for evaluating ODEs numerically. This week we will use a second-order Runge-Kutta method to solve the differential equation of motion due to gravity. Newton tells us that the magnitude of the gravitational force between the Sun and Earth is $F_G = \frac{GM_\odot M_\oplus}{R^2}$. We also know from his second law that the Earth will move around the Sun with an accerlation described by $\vec{F} = M_\oplus \mathbf{a} = M_\oplus \ddot{\mathbf{x}}$. Thus, solving for the acceleration gives $\ddot{\mathbf{x}} = \frac{GM_\odot}{R^2}$, which can also be written as

$$\ddot{\mathbf{r}} = \frac{GM_\odot}{r^2}(-\hat{\mathbf{r}}) \tag{1}$$

where $\mathbf{r} = \langle x, y\rangle$ and $r = \sqrt{x^2+y^2}$.

The second-order Runge-Kutta formula (RK2) is as follows:

$$\begin{aligned}
\frac{d\phi}{dt} &= f(\phi, t)\\
k_1 &= \Delta t \cdot f(\phi_n, t_n)\\
k_2 &= \Delta t \cdot f\left(\phi_n + \frac{k_1}{2}, t_n + \frac{\Delta t}{2}\right)\\
\phi_{n+1} &= \phi_n + k_2
\end{aligned}$$

where in this case ϕ is a 4-element vector made up of $\langle x, y, v_x, v_y\rangle$ and f is the derivatives function that uses Eq. 1 to calculate velocity based on acceleration and position based on velocity (*this* equation is time-independent, but it is possible that the derivatives function could depend on the actual time t as well). In this report I showcase some of the results that an implementation of this algorithm in `C` is capable of producing.

2 Results

Question 1

- $GM_\odot$ in "Earth orbit" units (AU and years) has a numerical value of $(2\pi)^2 = 4\pi^2 \approx 39.4784$
- Assuming an ideally circular orbit, Earth must travel a distance of $2\pi(1\text{ AU})$ in a year. Thus its tangential velocity is simply $2\pi\ \frac{\text{AU}}{\text{year}}$.

Question 2

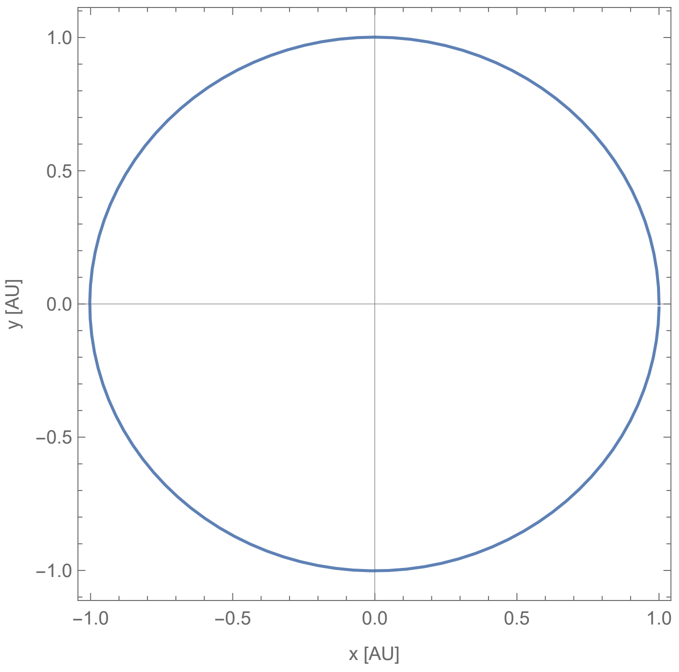

Figure 1: Using the initial conditions discussed in **Question 1**, we can simulate an orbit for the Earth that is approximately circular over the course of a year, as anticipated.

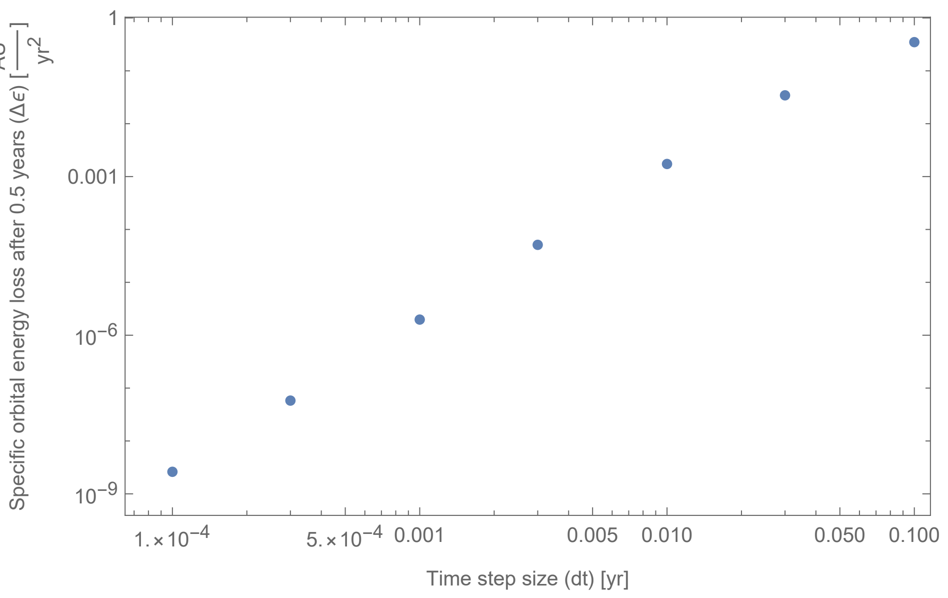

Figure 2: Specific orbital energy per unit mass lost after half a year of simulation using various step sizes (dt). Larger steps lead to greater non-conservation of energy. Log slope is approximately 2.781, meaning this algorithm converges even faster than $\mathcal{O}(n^2)$ (since $n \propto 1/dt$), which is what is expected.

In order to attain an error of $\lesssim 0.001$, we need a dt of just under 0.01 years.

Question 3

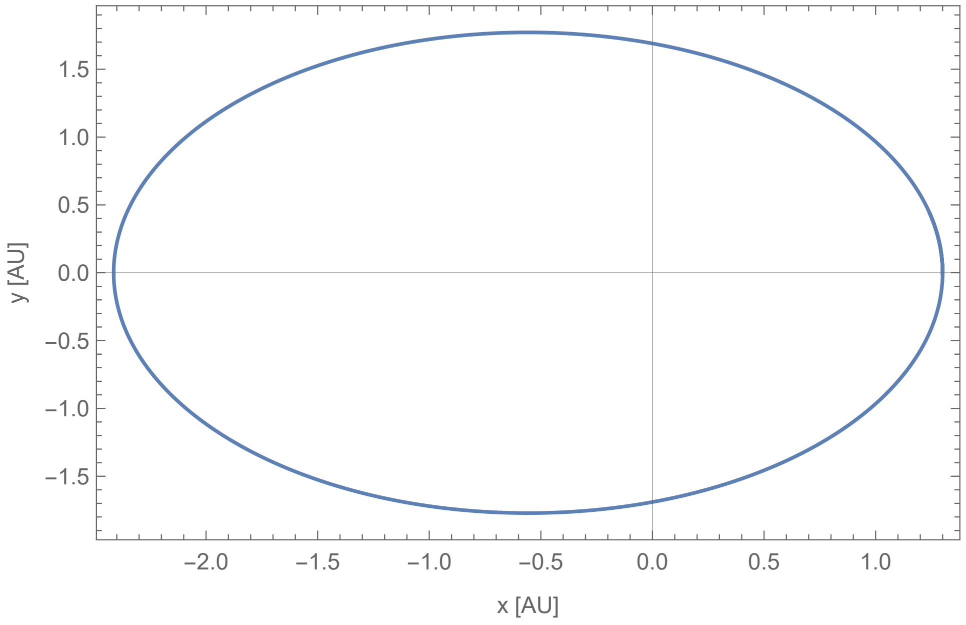

Figure 3: Trajectory of an eccentric orbit using $(x_0, y_0) = (1.3, 0)$; $(v_{x,0}, v_{y,0} = (0, 2\pi)$; and $dt = 0.001$ yr.

This orbiting body starts at perihelion, or $R_{\min}$, of 1.3 AU. It reaches a maximum distance (aphelion) of 2.414 AU, meaning the ratio is $\frac{r_a}{r_p} = 1.857$.

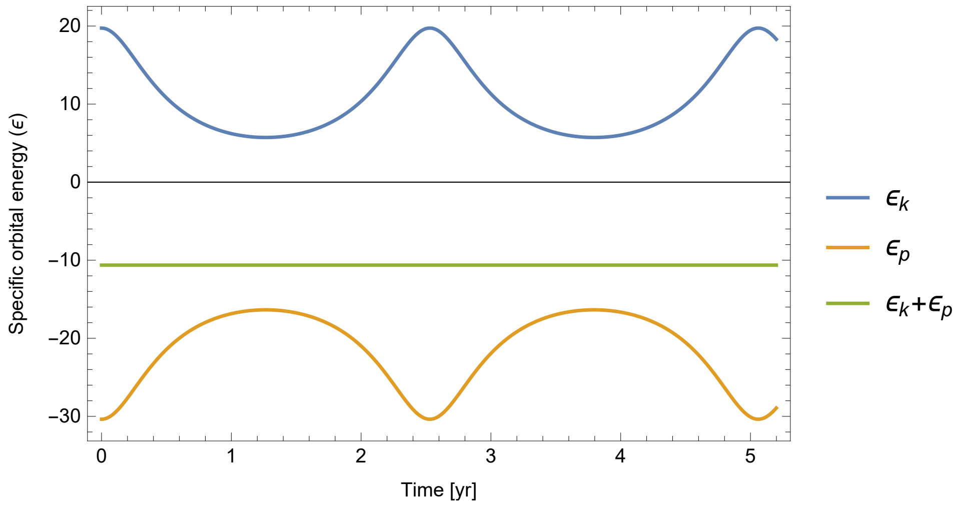

Figure 4: Orbital energies per unit mass, plotted over two complete periods. As expected, the total energy remains nearly constant, with potential and kinetic being converted back and forth into one another throughout the orbit. Furthermore, it also makes sense that kinetic energy is at a maximum at the closest approach to the Sun (when velocity is greatest), and potential is greatest when it is furthest away.

Question 4

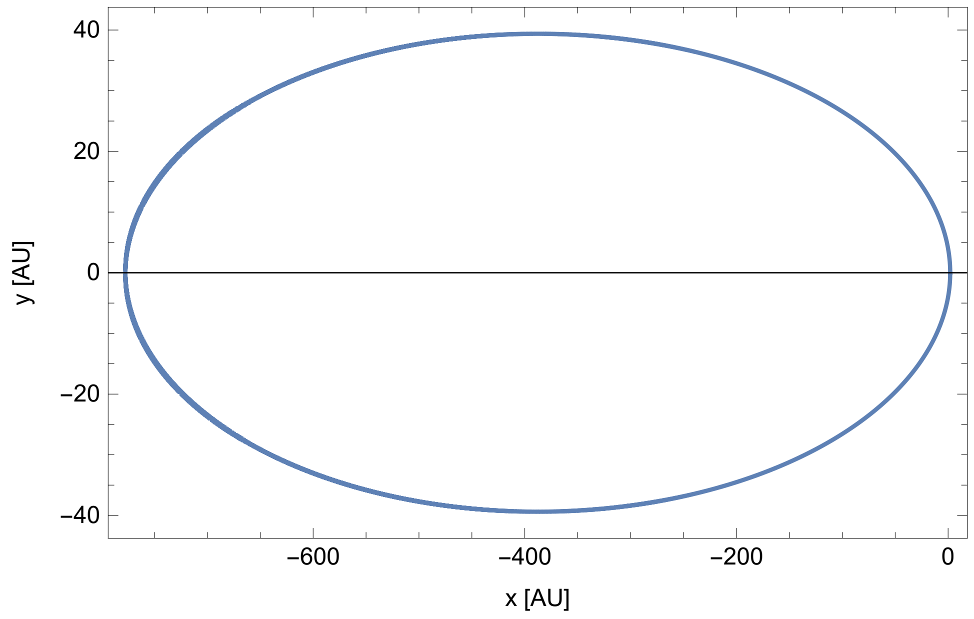

Figure 5: Simulated trajectory of one complete orbit of a comet with $(x_0, y_0) = (1.995, 0)$ and $v_0 = 2\pi$ AU/yr.

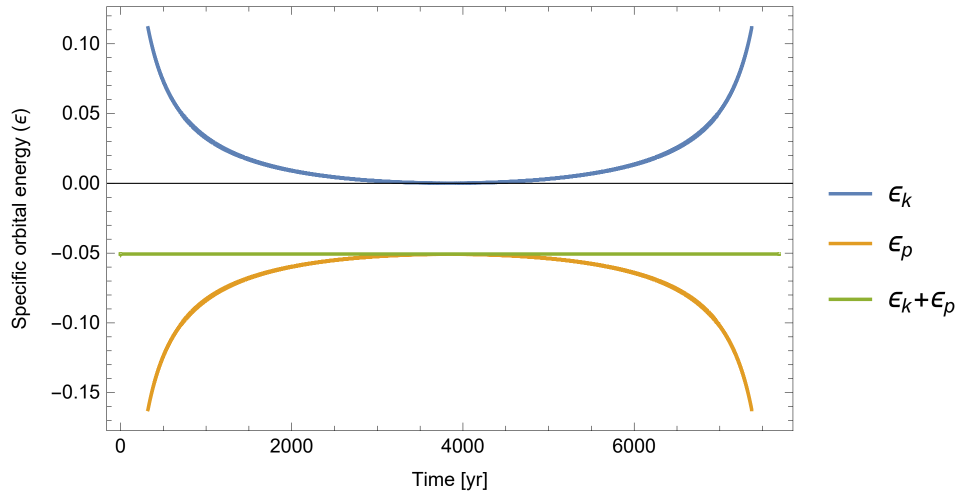

Figure 6: Orbital energy per unit mass for the comet whose trajectory is shown in Fig. 5. The timescale of the period is more apparent here, and the behavior whereby the total ϵ remains constant over time holds true.

This setup yields a period of approximately $\boxed{\text{7694 years}}$ for this first revolution.

Question 5

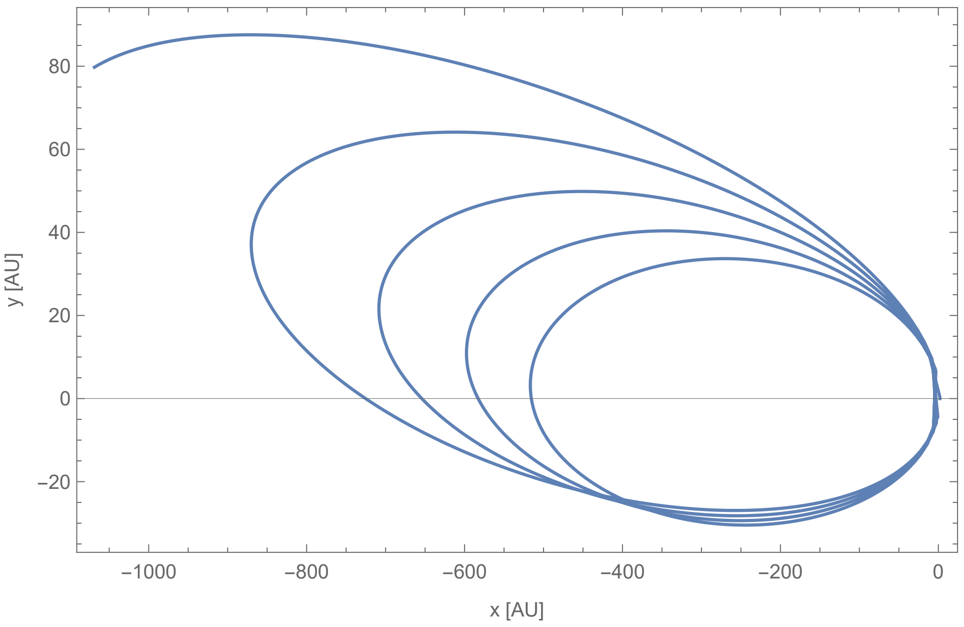

Figure 7: Multiple orbits over a time domain of 30,000 years using the same initial conditions as in **Question 4**. Clearly, there is some error introduced over the course of several revolutions, as the simulation does not retrace the same path every time.

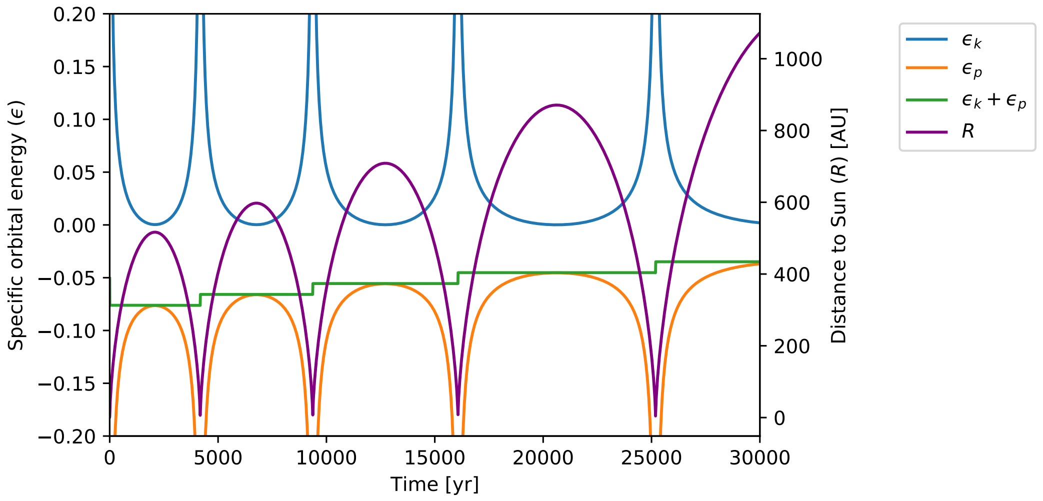

Figure 8: Specific orbital energy as plotted previously, but this time overlaid with distance from the Sun as a function of time as well. Here the total ϵ does not remain constant, as there is a bump up each time the comet reaches perihelion (the minimum distance to the Sun). This is because of the fact that in an orbit this eccentric, the velocity at perihelion is relatively much greater, so the step size (however small) is less capable of capturing all of the motion during the close approach as during the rest of the orbit, so the high velocity "overshoots." This issue with the velocity causes an erroneous increase in kinetic energy at perihelion, which in turn increases the total energy and results in an increase in the size of the orbit with each consecutive period.

3 Conclusions

This assignment was a great introduction to higher-order numerical integration techniques, and involved a physically intriguing problem at the same time. I have done Runge-Kutta integration in Python in the past to simulate orbits, so implementing similar things in `C` was a fun exercise.

I did not have many significant challenges other than an initial misunderstanding of what **Question 2** was asking in regards to the non-conservation of ϵ, but I was able to figure it all out.

I look forward to continuing to explore numerical methods.