Homework 4: Ordinary Differential Equations, Part II

Caden Gobat

February 24, 2021

1 Introduction

Last week we introduced Runge-Kutta methods for calculating numerical solutions to ODEs. We can apply this algorithmic strategy to integrate the forces between an arbitrary (N) number of bodies in an interacting system. This allows us to numerically compute the positions of objects in space based on our knowledge of how their relative positions affect their accelerations and thus velocities.

We recall the steps of the RK2 (second-order Runge-Kutta) method: for a given state vector ψ_n and derivatives function $\mathbf{F}(\psi, t) = \dot{\psi}$,

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{F}(\psi_n, t_n) = \mathbf{F}(\psi_n) \\ \mathbf{k}_2 &= \mathbf{F}(\psi_n + \mathbf{k}_1 \delta t/2, t_n + \delta t/2) = \mathbf{F}(\psi_n + \mathbf{k}_1 \delta t/2) \\ \psi_{n+1} &= \psi_n + \delta t \mathbf{k}_2 \end{aligned} \tag{1}$$

where δt is the chosen finite time step size. For this application, the derivatives do not depend on the time t itself, so $\mathbf{F}$ is a function only of ψ.

In order to determine the initial conditions that will give the systems physically realistic parameters, we can consider Newton's law of gravitation and the formula for centripetal acceleration: for two bodies who interact only through gravity,

$$a = \frac{v^2}{r} = \frac{Gm_1m_2}{\mu r^2} \implies v = \sqrt{\frac{Gm_1m_2}{\mu r}} = \sqrt{\frac{G(m_1+m_2)}{r}} \tag{2}$$

Because this can be scaled to incorporate as many bodies as desired, knowing this allows us to pick any starting positions, calculate the relative rs between each body, and determine the velocities that will make their orbits work out.

We can implement this algorithm in `C` (along with a formula for Newtonian gravity, which will act as our $\mathbf{F}$) in order to gain insight into how these kinds of systems behave.

2 Results

Question 1

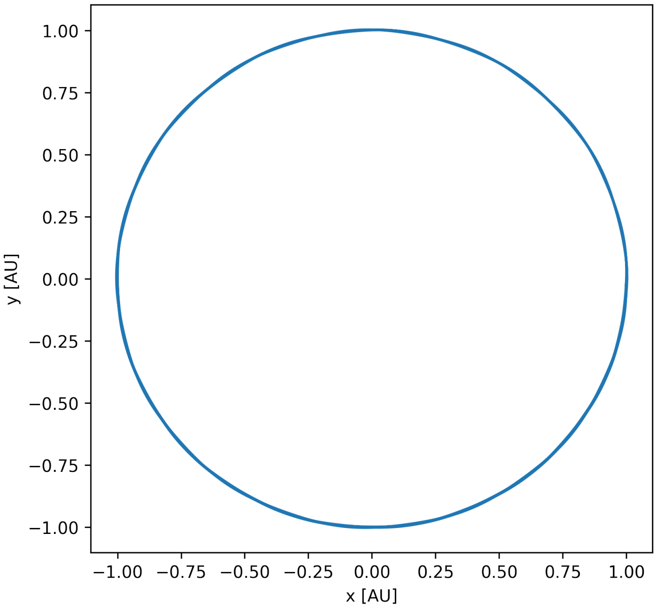

Figure 1: Orbital trajectory over 3 years of the third (smallest) body in a 3-body system where $M_1 = M_\odot$, $M_2 = 3 \times 10^{-6} M_\odot$, $M_3 = 3.7 \times 10^{-8} M_\odot$, $r_{12} = 1$ AU, and $r_{23} = 0.0025$ AU. The calculated period of the medium-mass body (the "planet") is between 1.0013 and 1.0014 years.

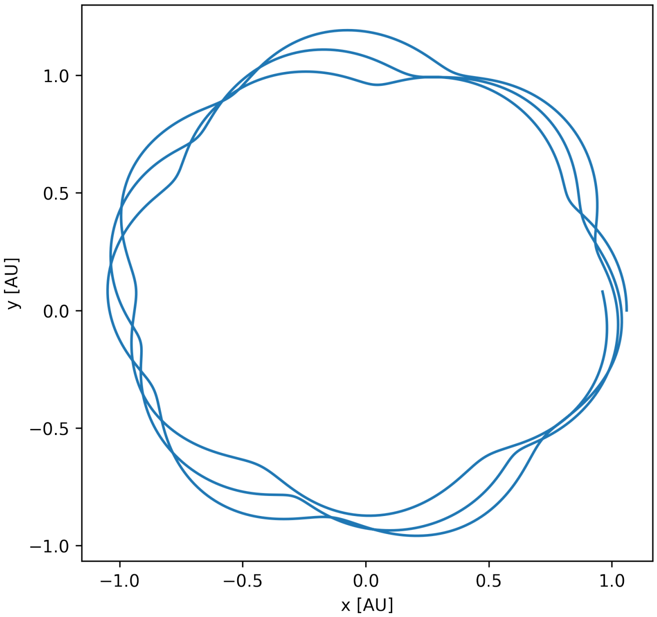

Figure 2: Orbital trajectory over 3 years of the third (smallest) body in a 3-body system where $M_1 = M_\odot$, $M_2 = 10^{-2} M_\odot$, $M_3 = 10^{-4} M_\odot$, $r_{12} = 1$ AU, and $r_{23} = 0.0025$ AU. The calculated period of the medium-mass body (the "planet") is between 1.0188 and 1.0189 years.

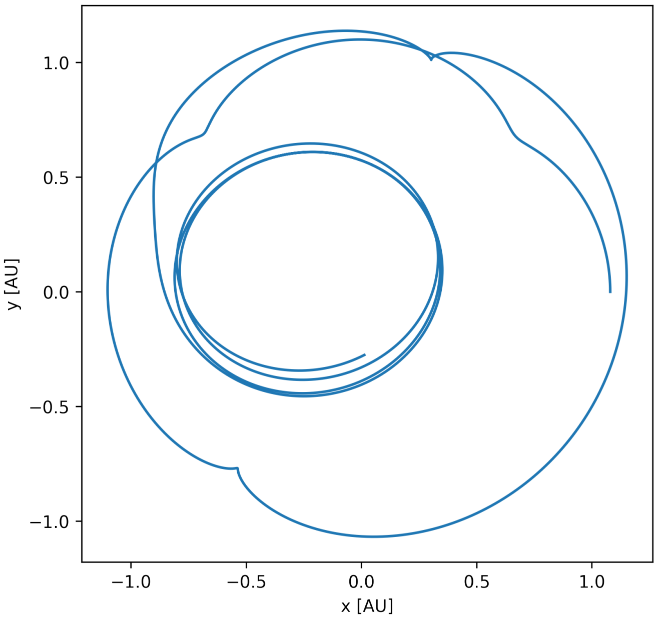

Figure 3: Orbital trajectory over 3 years of the third (smallest) body in a 3-body system where $M_1 = M_\odot$, $M_2 = 10^{-2} M_\odot$, $M_3 = 10^{-4} M_\odot$, $r_{12} = 1$ AU, and $r_{23} = 0.08$ AU. The calculated period of the medium-mass body (the "planet") is between 1.0177 and 1.0178 years.

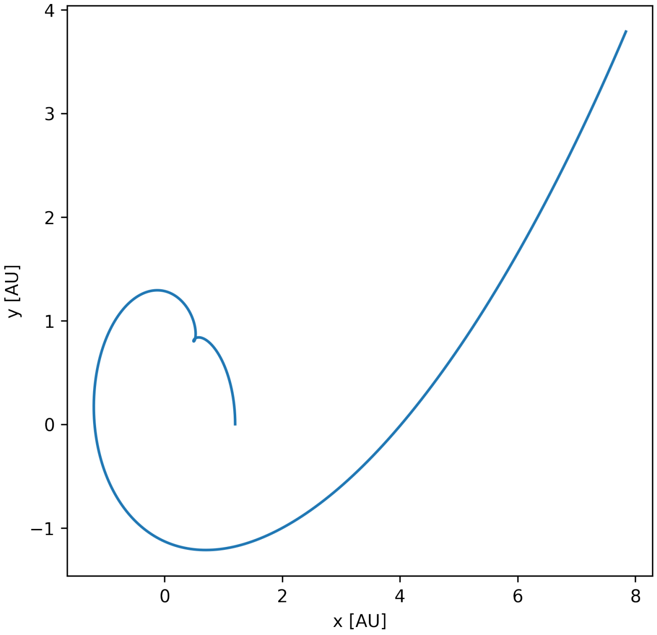

Figure 4: Orbital trajectory over 3 years of the third (smallest) body in a 3-body system where $M_1 = M_\odot$, $M_2 = 10^{-1} M_\odot$, $M_3 = 10^{-4} M_\odot$, $r_{12} = 1$ AU, and $r_{23} = 0.2$ AU. The calculated period of the medium-mass body (the "planet") is between 1.0192 and 1.0193 years.

The period for the planet in each case is ~ 1 year because the initial velocities are specified such that a circular orbit will complete 1 revolution in 1 year. Changing dt has little effect on these results, as the orbits for the planet are still mostly circular or elliptical, and so there is not as much accuracy to be lost in small, nuanced movements as is the case for the third, smallest body.

Question 2

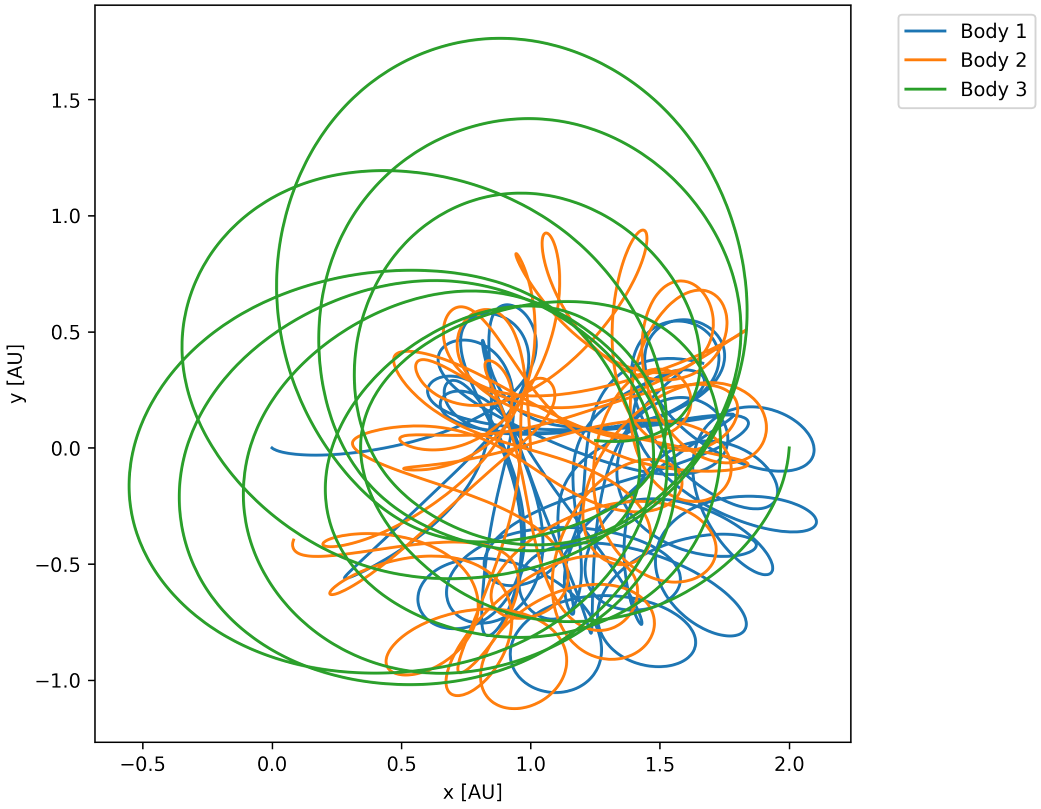

Figure 5: Trajectories of three bodies over 8 years when $M_1 = M_2 = M_3 = M_\odot$ with initial conditions satisfying $x_1 = (0, 0)$ AU, $v_1 = (1, 1)$ AU/yr, $x_2 = (1, 0)$ AU, $v_2 = (0, 6)$ AU/yr, $x_3 = (2, 0)$ AU, $v_3 = (0, 6)$ AU/yr at time $t = 0$. Positions have been adjusted so as to be recentered on the center of mass at each time step in order to visualize the inherent motion of the system without overall drift.

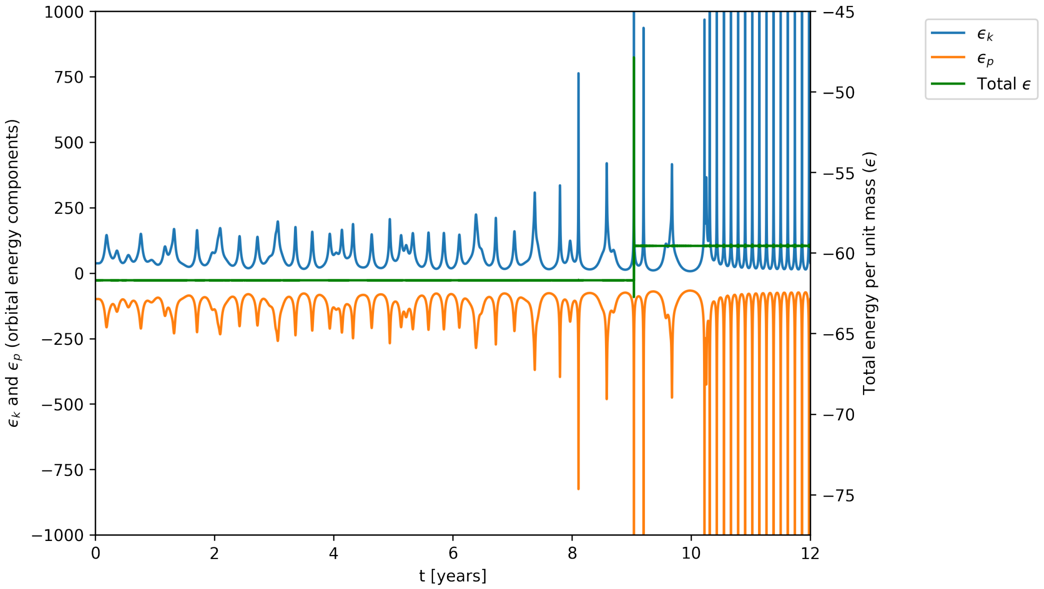

Figure 6: Specific orbital energy (total energy per unit mass) breakdown of the 3-body system over 12 years. Total energy ($\epsilon_k + \epsilon_p$) remains constant, as expected, for the first ~ 9 years of the simulation, after which an obvious discontinuity occurs. Note the separate scaling for the individual energy components (left vertical axis) and the total energy (right vertical axis).

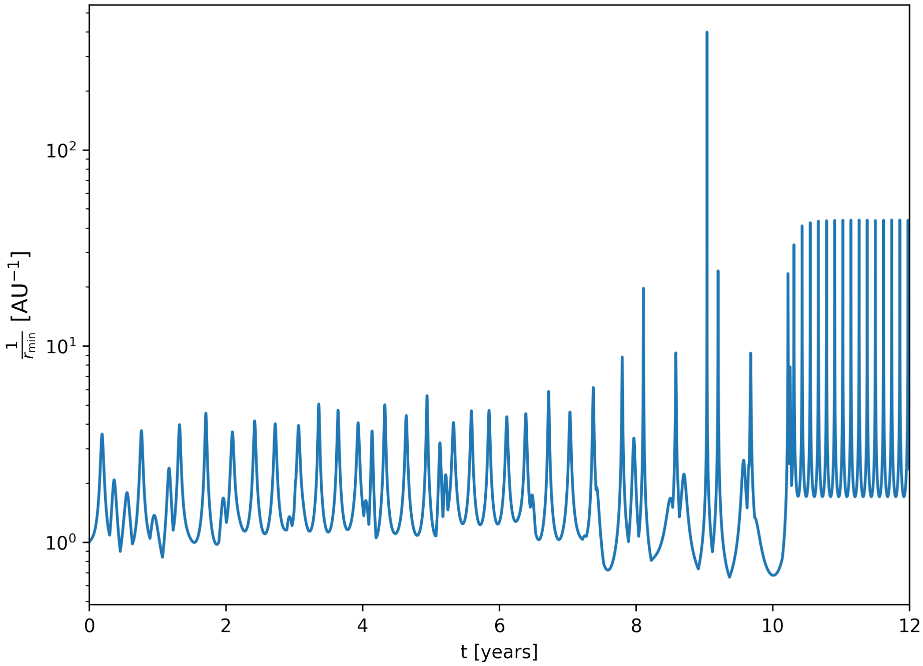

Figure 7: Reciprocal of the separation between the two closest bodies at any given time over a 12-year simulation. This quantity $\frac{1}{r_{\min}}$ is proportional to $a_{\max}$—a result that follows from Eq. 2. Note the choice of logarithmic scaling on the vertical axis to better show features at all scales. This plot provides some insight into the discontinuity in energy seen in Fig. 6. At right around $t \simeq 9$ years, we observe a large spike in the reciprocal of the smallest separation. This corresponds to pronounced spike in the acceleration of the system as well, which means that the finite time step dt is less able to "keep up," thus introducing error.

This error is somewhat difficult to address, as it results from the fact that the objects pass within $\sim \frac{1}{400}$ AU of one another. Given that this distance is approximately half the radius of the Sun itself ($R_\odot \approx \frac{1}{215}$ AU), if we interpret each of these three bodies as a Sun-like object, it would mean two of them actually come into contact with one another at this point. There is no way to reliably predict what would happen in this case using this code which treats all bodies as point masses, and the acceleration is so great than an *extremely* fine time resolution around this time would be needed in order to accurately accommodate the magnitude of the quantities involved.

Question 3

The computational complexity of performing an N-body simulation like this is $\mathcal{O}(n^2)$. For every new body introduced, one must run all of the calculations for each of the other bodies' relations to it. For example, a system of 2 bodies requires only one relationship to be calculated (plus the "self"-computation, in some sense). Doubling N to a 4-body system now requires computing a total of 8 interactions. Thus we observe that a doubling of N results in a quadrupling ($\times N^2$) of the required computations.

3 Conclusions

As compared to previous assignments, I had a few more technical difficulties this time around. At first my `C` code was not functioning properly at all, and I had to scale back on some of the generalizability that I had tried to implement during the debugging process.

Additionally, the requested timestep of $dt = 10^{-6}$ years is extremely computationally demanding, especially when the time domain spans 8 or 12 years. I found myself sitting idle for long periods of time simply waiting for my code to finish execution, before even knowing if the results were good or not. Additionally, the simulation results and output are at least 1–2 GB in size for some of the time domains required in this assignment. I compared the shapes of the orbits using $dt = 10^{-5}$ years and $dt = 10^{-6}$ years and found only minimal differences, so I hesitate to say that the marginally increased accuracy afforded by the smaller time step is worth the increased computational time.