

Computational Physics HW3

Ethan Rooney

2020-02-12

1 Introduction

This weeks homework centers around the implementation of the Rutta-Kunga 2 method for numeric integrators.

I will be implementing this algorithm on a variety of systems, initially I will be modeling an idealized version of Earths orbit, assuming that the planet is in a perfectly circular orbit and then experimenting with more and more elliptical orbits to see how the RK2 method begins to break when the linearity of a system varies drastically over a fixed time step. As seen in the highly eccentric orbit covered in problem 4.

The core of the RK2 method is to find a better way to approximate an integral when compared to Euler's method. For RK2 once the differential equation is known, the slope at the initial time t_0 and initial position x_0 is computed, a half step is taken along that line, the approximate value of the slope at the midpoint is then calculated, and then a full step is taken along the newly computed slope of the equation from (t_0, x_0) to (t_i, x_i) . For well behaved differential equations, this method provides a more accurate method of integration than Euler's method.

$$k_1 = f(y'(t_0), t_0) \quad (1)$$

$$y_1(t_0 + \frac{h}{2}) = y'(t_0) + k_1 \frac{h}{2} \quad (2)$$

$$k_2 = f(y_1(t_0 + \frac{h}{2}), t_0 + \frac{h}{2}) \quad (3)$$

$$y'(t_0 + h) = y'(t_0) + k_2 h \quad (4)$$

2 Results

2.1 Question 1

Given Equation 5: if we let $r = 1\text{AU}$ and $\ddot{r} = a_c$ we can then use Equation 6 to find $GM_\odot = v^2 r$. We know that v is the tangential velocity of the earth, which must travel $1\text{AU} \times 2\pi$ rad per 1 Year we find $GM_\odot = 4\pi^2 \text{AU}^3 \text{Year}^{-2}$.

$$\ddot{r} = \frac{GM_\odot}{r^2} \hat{r} \quad (5)$$

$$a_c = \frac{v^2}{r} \quad (6)$$

2.2 Question 2

As seen in Figure 1 we see the orbit of Earth as we would expect for the initial conditions given.

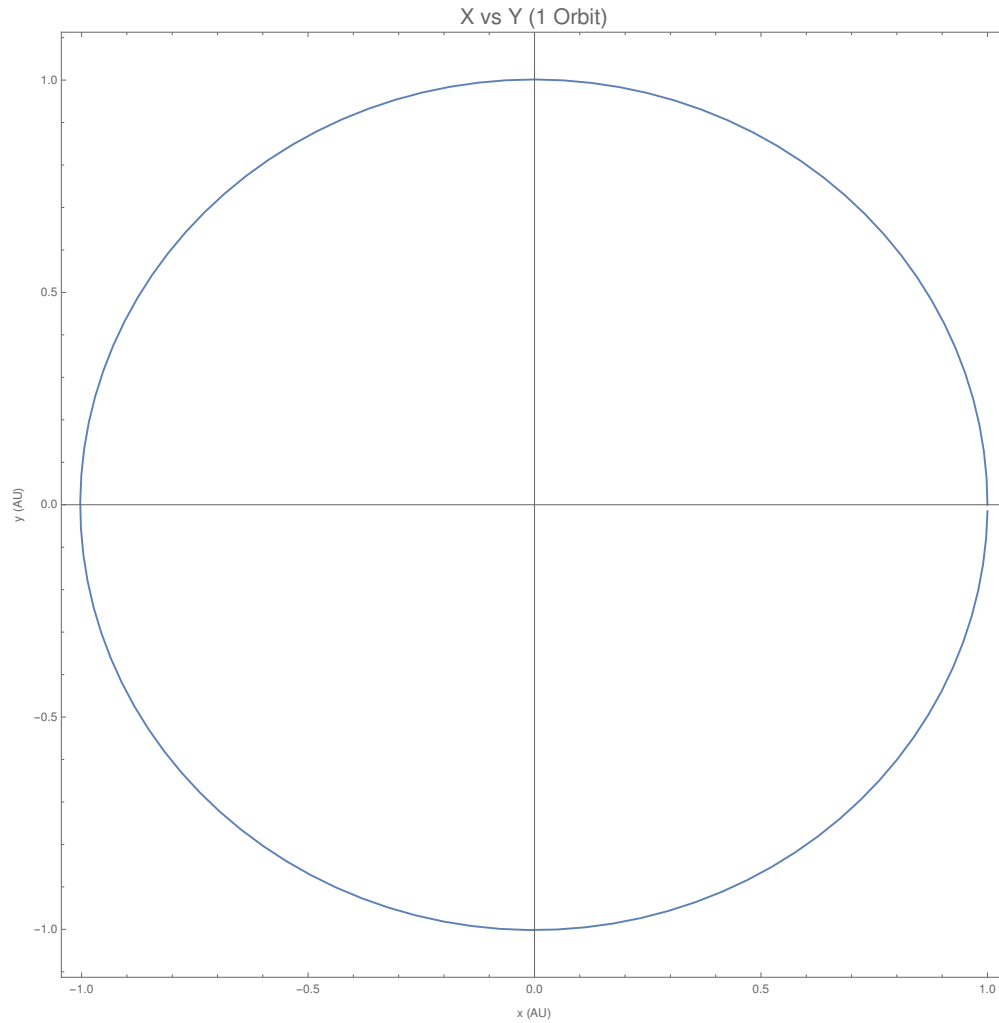


Figure 1: Plot of the orbit of Earth as simulated with RK2

While in Figure 1 we see how error in the conservation of energy scales with step size. The slope of the Log-Log plot is 3 (See Equation 7). This means that for a 10 fold increase in step size, the error of the integrator is reduced by 1000. This is consistent with the expected scaling of RK2.

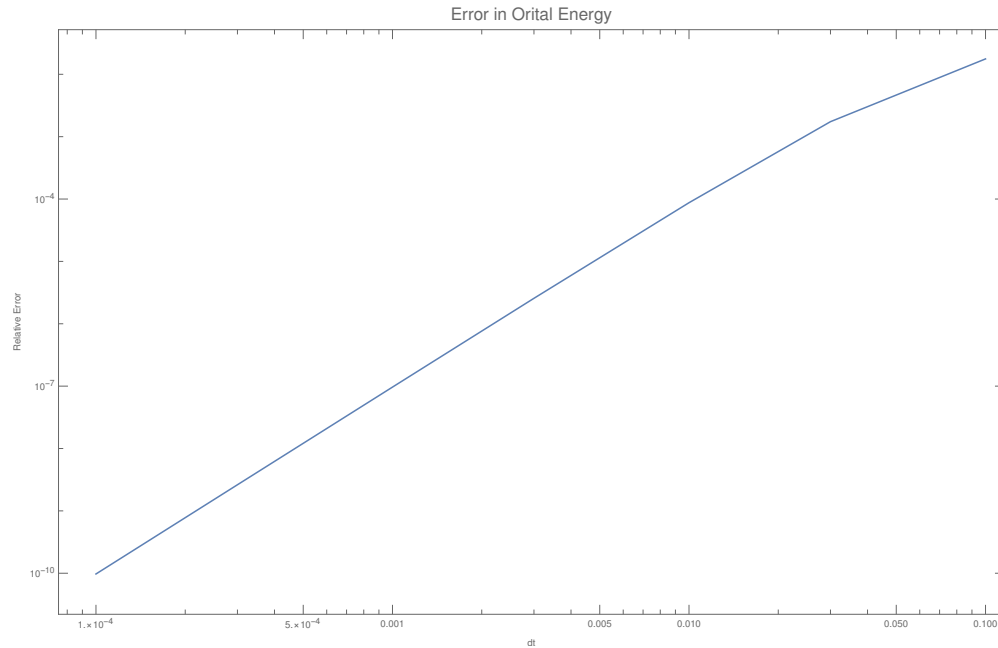


Figure 2: Error in the Energy Simulated by RK2 compared to Energy Conservation

$$Slope = \frac{\log(y_1) - \log(y_2)}{\log(x_1) - \log(x_2)} = \frac{\log(0.01) - \log(0.001)}{\log(8.73084 \times 10^{-5}) - \log(9.65174 \times 10^{-8})} \simeq 3 \quad (7)$$

Based on the Above graph, the time step required for an accuracy of 1:1000 is $dt \simeq 0.0245$.

2.3 Question 3

Seen in Figure 3 is an orbiting body with a Ratio of aphelion:perihelion of 1.857.

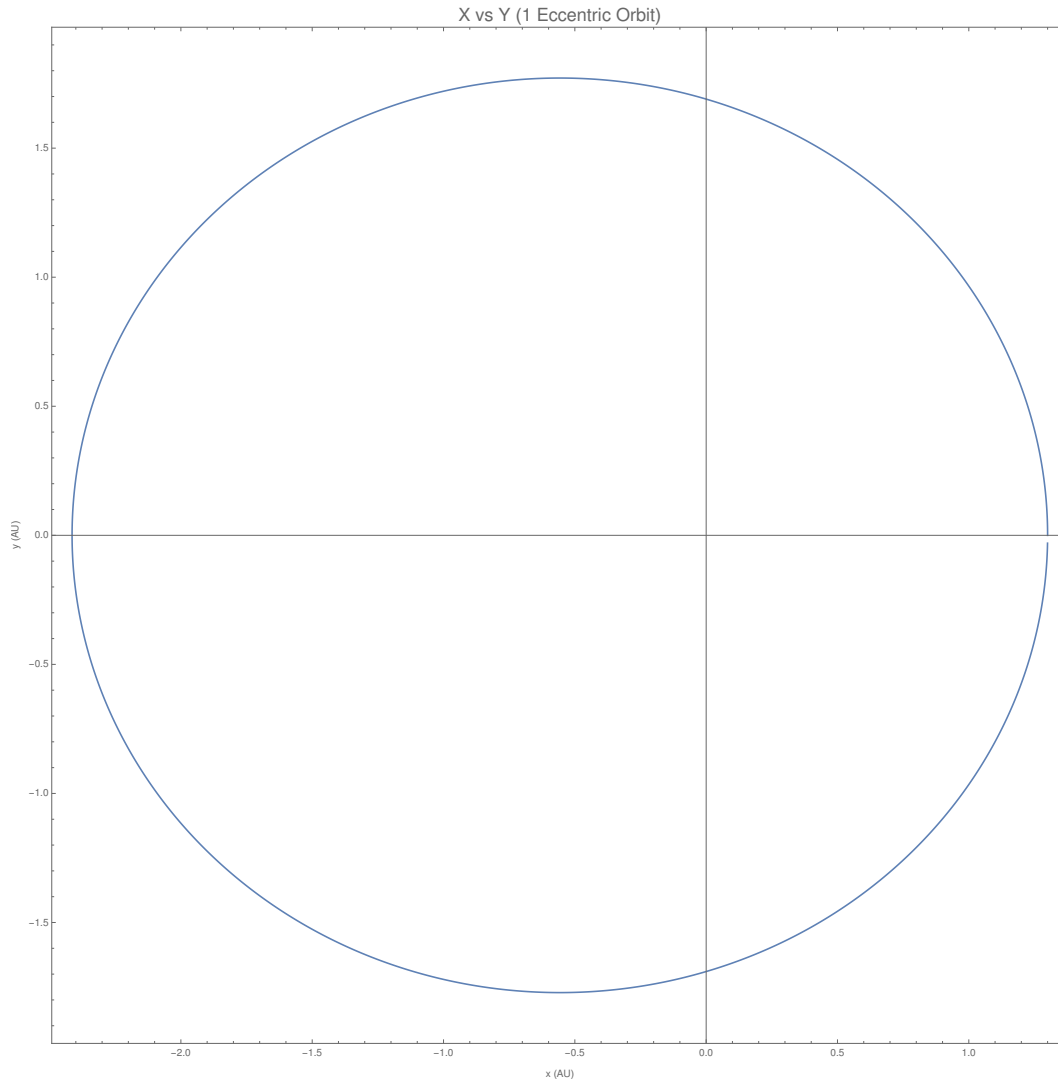


Figure 3: Eccentric Orbit of a Planet with Perihelion of 1.3 AU, with an initial velocity of 2π AU/Year.

In Figure 4 We see the energy of the planet as it makes one complete Revolution. The Kinetic energy of the Planet is at a max when the Potential Energy is at a minimum, which occurs at Perihelion. Furthermore the Total energy of the system is constant and negative, indicative of a bound system. This is consistent with my expectations.

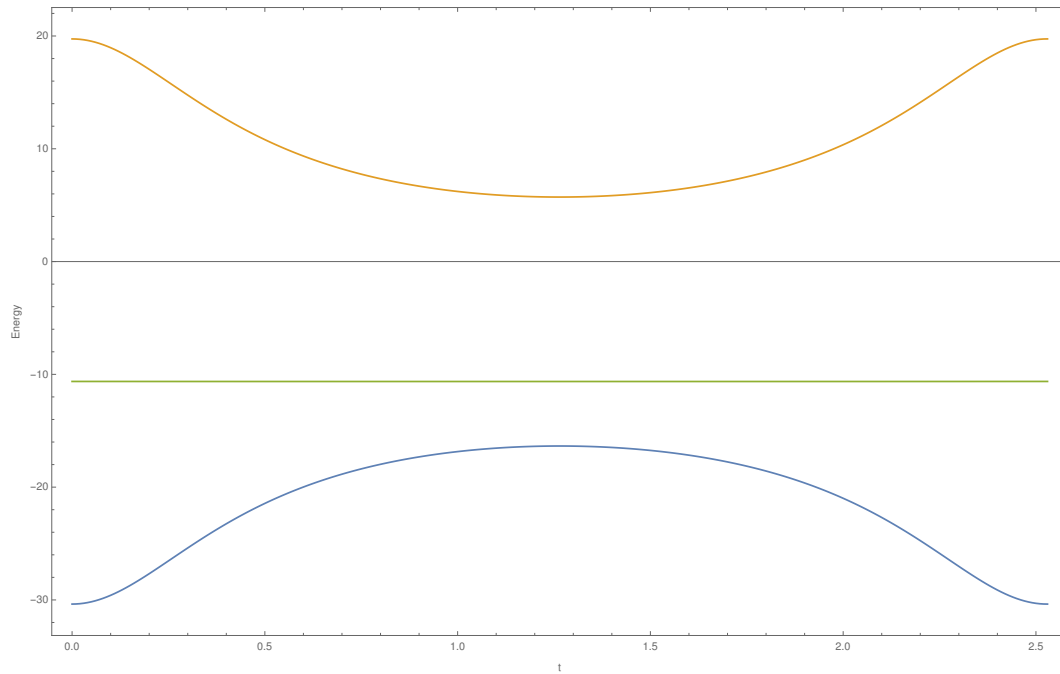


Figure 4: The Energy of a slightly eccentric orbiting body. The orange line is the Kinetic Energy, the blue is Potential Energy, while Green is the total energy.

2.4 Question 4

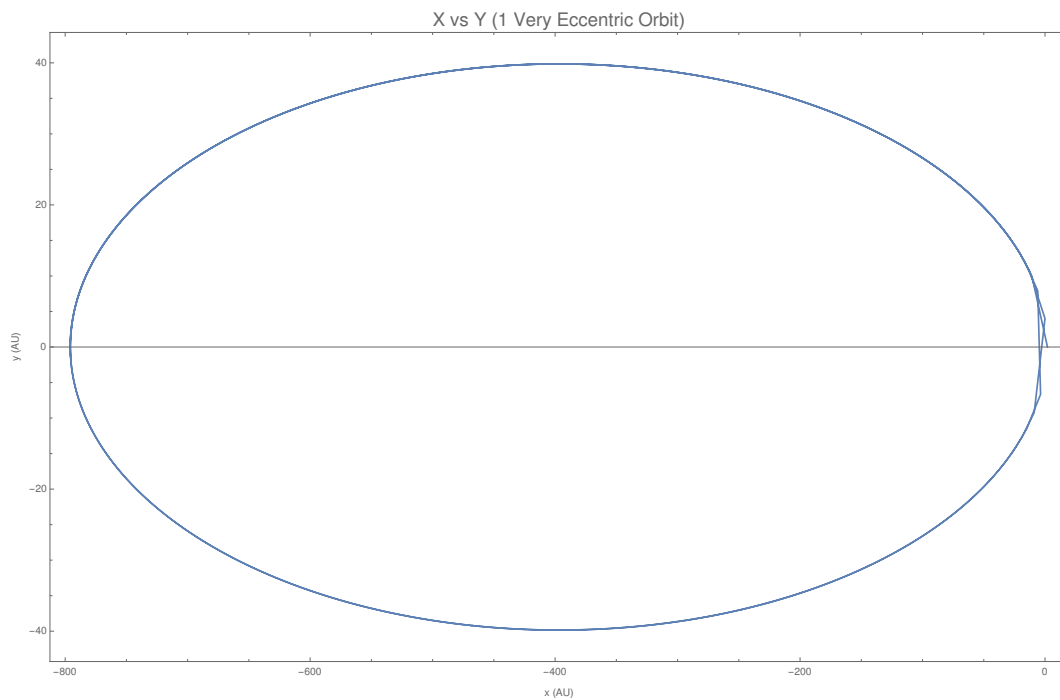


Figure 5: Orbit of an eccentric body over 20000 years with a time step of 0.001. Orbit is ~ 7970 years long.

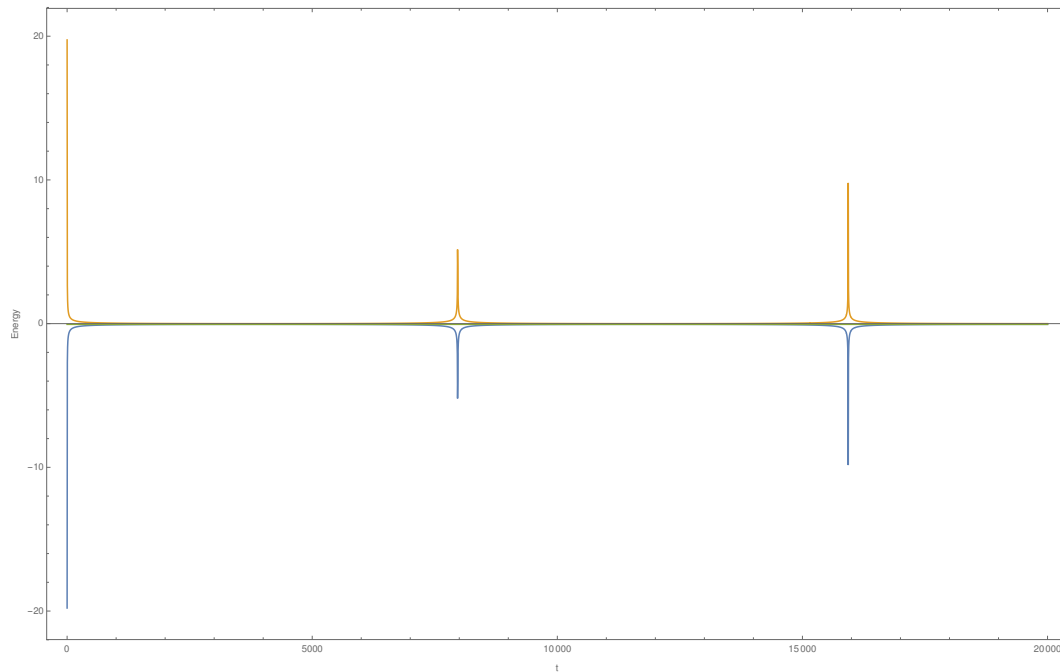


Figure 6: The Energy of a very eccentric orbiting body. Orange represents the Kinetic Energy, Blue the Potential Energy and Green the Total Energy

In Figure 6 see a plot of the energy transformation for a highly elliptical orbit with a period of 7970 years the orbit behaves much the way I expect it to. The small artifacts in Figure 5 near the perigee are due to the 5 year time steps used for plotting.

2.5 Question 5

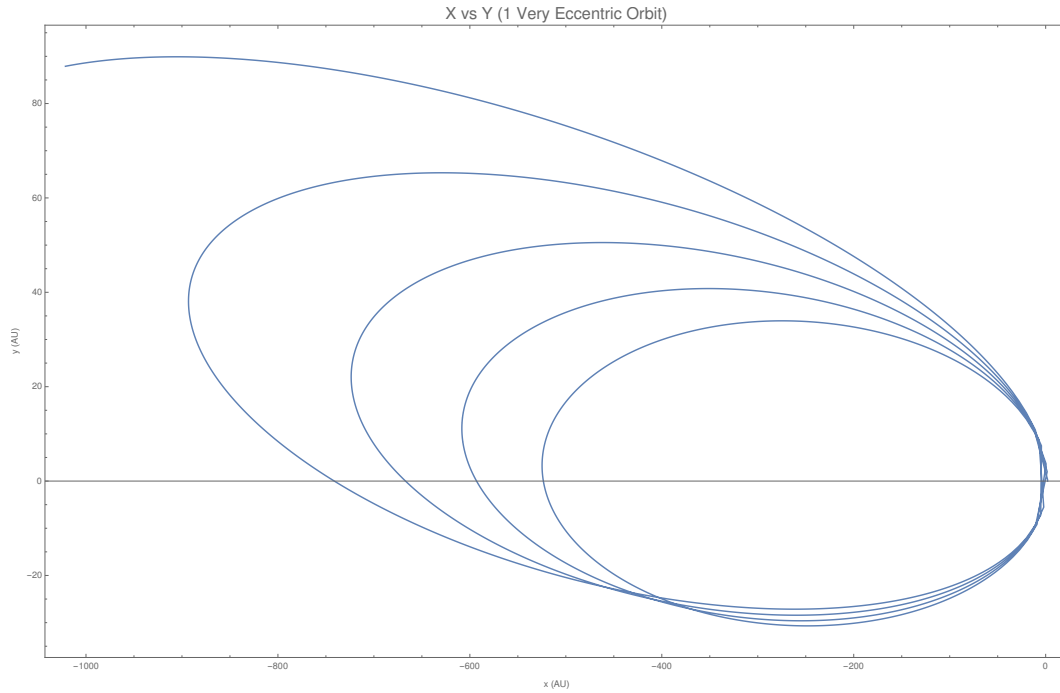


Figure 7: Something isn't right with our simulation. The planet shouldn't be leaving, yet it seems to get slingshot out of the system.

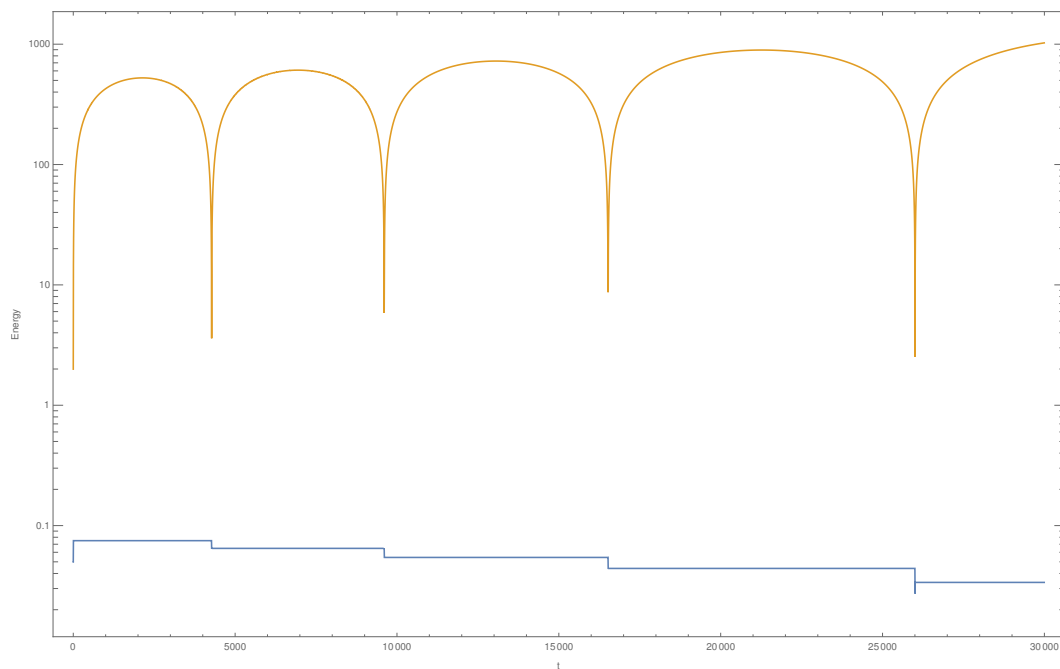


Figure 8: In the graph above we see radius of the orbit and the absolute value of the total energy. Every time the radius gets smaller energy seems to leak into the simulation.

For Highly eccentric orbits, a loss of precision of the simulation occurs every time the planet passes

through the perigee. For highly elliptical orbits this happens because during the passing of perigee there is a rapid change in both the direction and velocity of the orbiting body, the RK2 algorithm is insufficient to keep up with these rapid changes and would need to use a shorter timescale to preserve accuracy. However, long orbits means that a lot of extra computation is required for this solution. Where much of the time the larger time step would be sufficient.

Bonus

No, these orbits are no longer elliptical. Once we move to a universe where gravity scales with $\frac{1}{r^3}$ energy is not conserved in orbital mechanics and any small deviation from a perfectly circular orbit will lead to the planet being ejected from orbit, as we can see in Figure 9. The initial condition for this system was a radius 1 part in 10000 larger than perfectly circular. Over a 20 year period the planet is flung further and further into space.

If you look at Figure 10 we can see that the total energy of the system is not conserved.

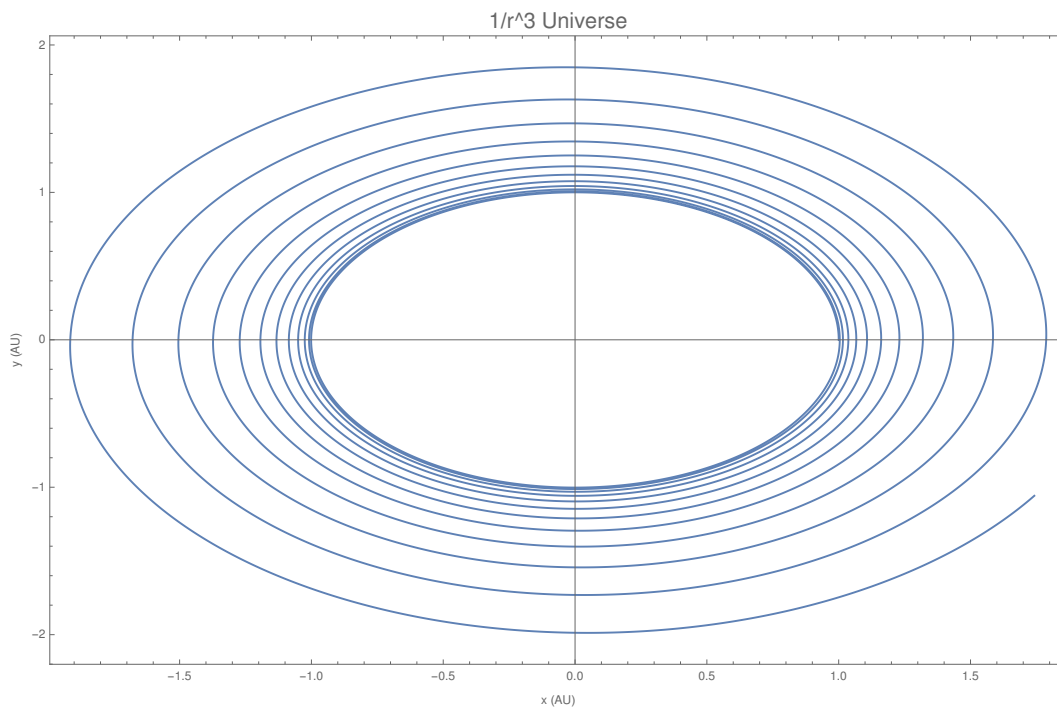


Figure 9: The tiniest nudge can fling a planet out of orbit in this system.

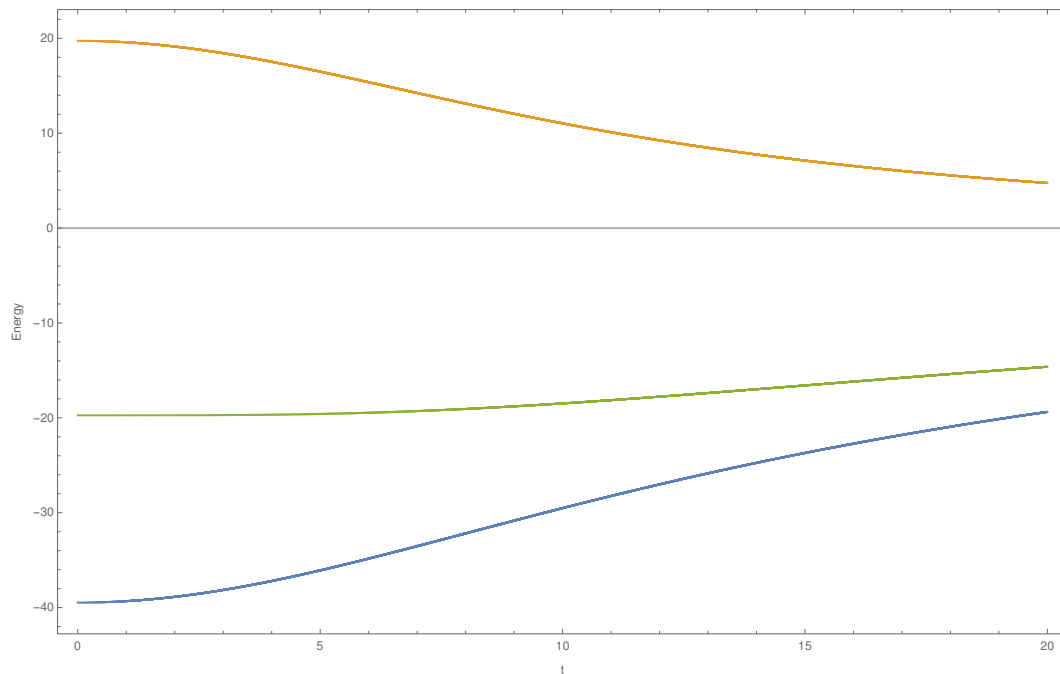


Figure 10: Energy is no longer conserved in this system. Energy bleeds into the system and the planet is flung out into space. Green is the Total Energy of the system, Orange is the Kinetic Energy, and Blue is the Potential Energy.

3 Conclusions

Rutta-Kunga method provides a much quicker method of approximating solutions to differential equations than Euler's Method. Also there is a rather nice package for vim called 'xuhdev/vim-latex-live-preview' it really simplifies the process of writing \LaTeX . Additionally I managed to find the problem with Mathematica crashing on file saves. If more than one Kernel is open then they can conflict in file browser `SetDirectory`. When that happens Mathematica freezes and all progress can be lost.