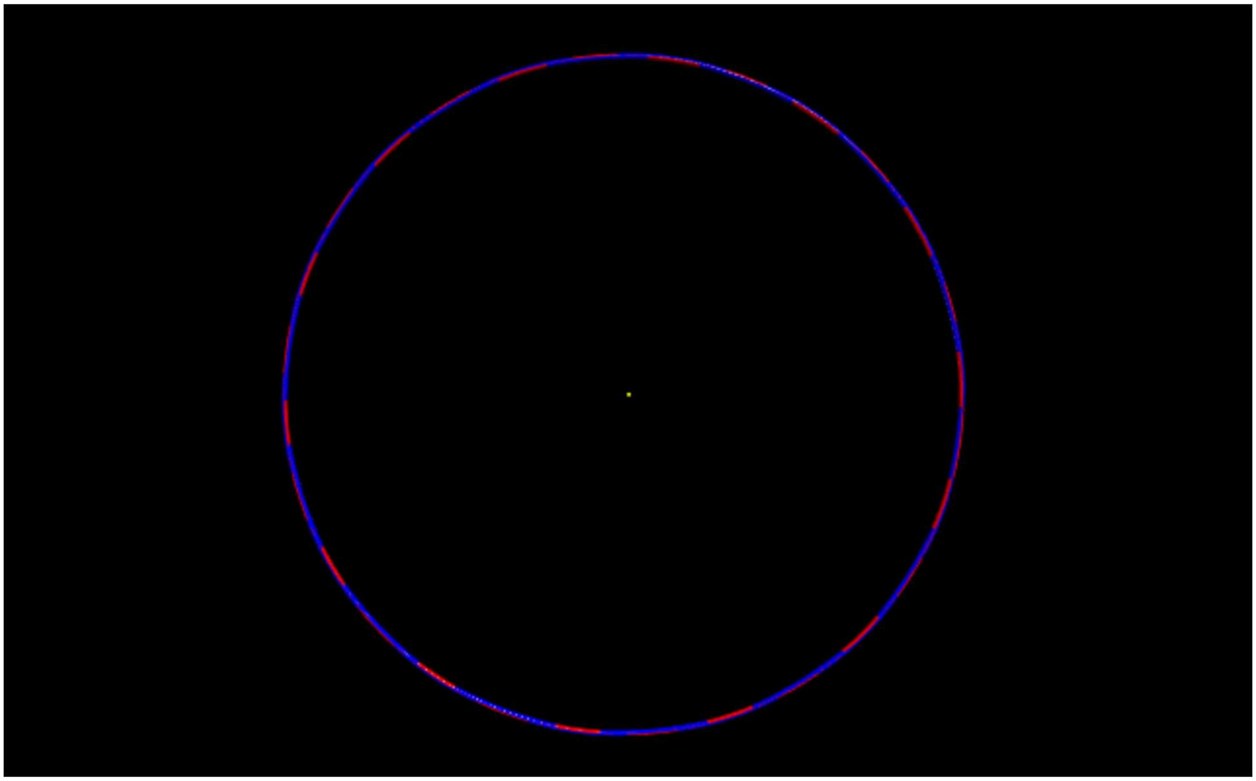
Hunter Wiese

August 14, 2017

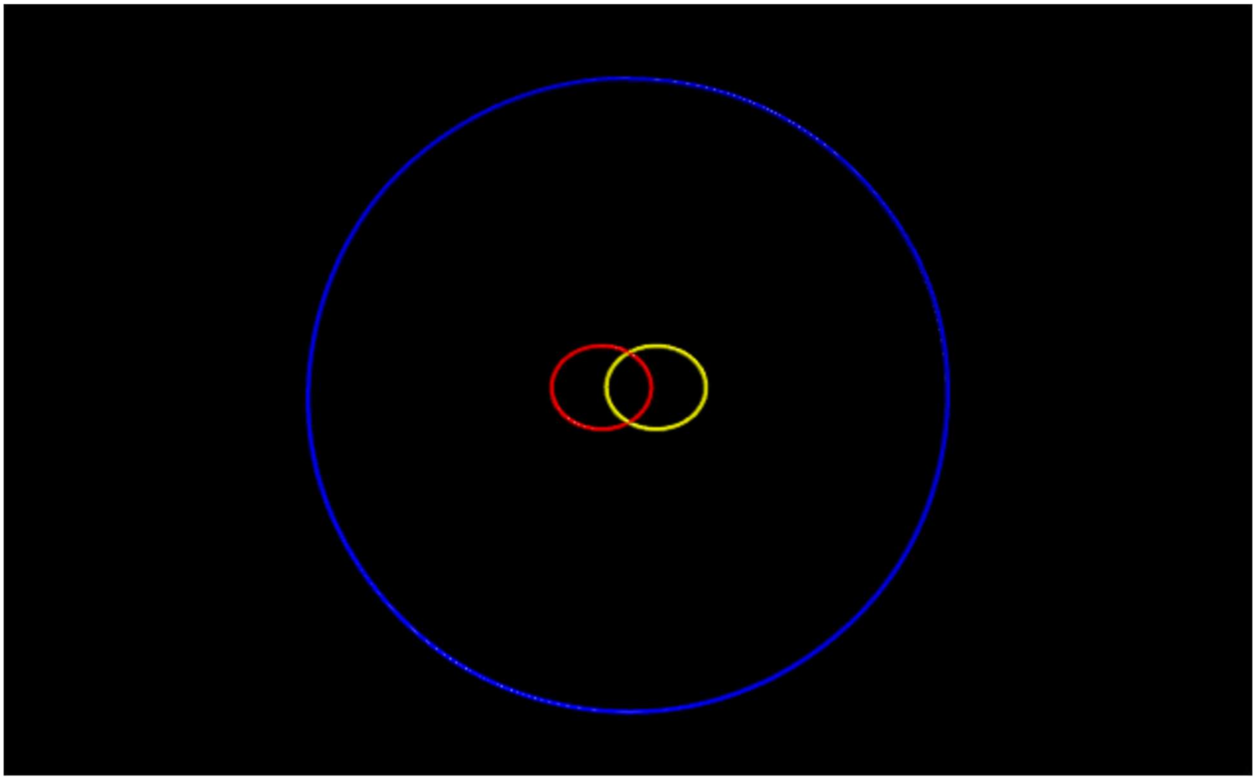
Three Body Simulation Report

I chose to do a three-body problem for my final project to further increase my understanding of the RK4 method of solving ODE's, and because I figured the final program could be interesting to experiment with. The program utilizes three second order differential equations, which are integrated with respect to the x, y, and z dimensions to produce the displacement of each object. This meant that the program required eighteen initial conditions to run, which made bug fixing and testing very tedious. I read online that there are more efficient ways to represent the three-body problem using fewer equations, but I couldn't figure out how to implement them using the RK4 method. In the future, this could be a good way to improve the program. Once I got the program working, however, I saw some interesting results.

The first simulation I created was one of the Sun, Earth, and Moon. I kept the Sun fixed, and took the initial velocities and positions of the Earth and the Moon from google, to see if my simulation would create stable orbits based on these conditions, and I was happy to see that it did. As a second test of the simulation I found the period of the Earth, which was about 373 days. Considering the precision limitations of the program, and the specificity of the data I used for the initial conditions, this is a pretty good result. The major issue with this simulation, however, is that it is difficult to tell whether the Moon is orbiting the Earth correctly. It does follow the same general path as the Earth, but in both matplotlib and vpython the scale of the problem is too large to allow me to view the difference between the paths the Earth and Moon take. Another way to improve this program in the future would be to find a way to see whether the Moon is orbiting the Earth as expected.



The second simulation I created was a binary star system with a single planet orbiting both stars. For this simulation, I attempted to come up with the initial conditions myself that would produce stable orbits for two solar mass stars and one Earth mass planet. I began by setting the stars two AU apart, and the planet four AU from the center of mass of the two stars, on a line perpendicular to one drawn between the stars. I found that at these locations, the stars formed a stable orbit if I gave them initial velocities of 10,000m/s in opposite directions. I was surprised to find that the stable orbit produced was not elliptical, but was instead two interlocking ellipses, like a Venn Diagram. The two stars orbited the center of mass of the system with periods of about 390 days. To get the initial velocity of the planet I assumed that the system would work similarly to a system with one star that was double a solar mass. Using GMm/r^2 = mv^2/r, and reducing this equation to GM/r = v^2, I found that for a system with mass 2M and radius 4r, the velocity should be (v/sqrt(2)), so G2M/4r = v^2/2 → GM/r = v^2. So I set the planet's initial velocity to be Earth's velocity divided by sqrt(2) and found that this did indeed produce a stable orbit with a period of about five years 348 days.



This project taught me a lot about solving ODE's in python, and more than anything about bug fixing, so I believe it was worth doing. I was also able to get some interesting results from the two simulations I ran, with the period of the Earth being only eight days off from what it should be. I was surprised by how hard this program was for my computer to run, when it runs other, seemingly more complicated, programs very quickly. It shows how important the efficiency of a program is. While I'm sure I could easily find a more efficient, and better animated, n-body simulation online, finishing this project was still very satisfying after struggling to program it for so long. Making predictions based on physics and astronomy and testing them with my own code was a very cool experience, and even provided some unexpected results, like the orbits of the binary stars.