My research focuses on describing the patterns and attributes of the idealized physical model of the “double pendulum”. This system is somewhat special because its motion is chaotic - meaning that small differences in the initial conditions lead to drastic changes in future states of the system. This is in sharp contrast to the single pendulum, which has a periodic motion. Of note is that the double pendulum is only truly chaotic without any air resistance or friction whatsoever, for with resistance the pendulum would come to rest as time progresses further. I created a program that predicts the position of both point masses and any time using the 4th-order Runge Kutta method in Octave (free MatLab alternative) and then C++. The program has adjustable initial conditions, adjustable masses and rod lengths, and a choice of step-size to balance accuracy vs. efficiency. I was able to plot the double pendulum’s trajectory using Octave built-in plotting functions, but the primary focus of my research was modelling a certain condition of this system: whether or not either of the pendulums ever “flip”, and when this happens. When either of the pendulums’ angles reaches more than 2\*pi greater than their initial position, they have inverted. Whether or not this occurs, and how long it takes for it to occur, depends on what position the pendulum is dropped (initial condition). For this project I normalized the weights to be 1 kilogram each, the rod lengths to be 1 meter each, and gravity to be 1 m/sec^2, for simplicity. There are two components to the initial condition: the angle of the first pendulum, “theta”, and the angle of the second, “phi”. Both of them range from 0 to 2\*pi radians. With these two variables, we have a 2\*pi by 2\*pi grid representing all the possible position the pendulums can drop from. I ran the runge-kutta solver and timed how long it took for the inversion to occur for each pixel in the grid (adjustable mesh size), and color-coded it, red being the longest and blue being the shortest time. I have been trying to understand the properties of the image, and whether or not it is a fractal, by zooming in on various regions and re-generating it there. This is the image generated (covering the top quarter of the domain):

