# Duffing Strobe Application
# PHY 230

Avi Vajpeyi

April 13, 2016

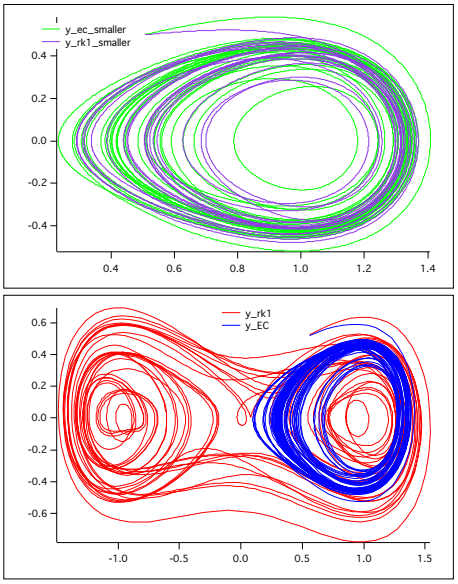
## 1 Complications with Application

We initially had a lot of difficulty making the application work as the animate function was not working correctly. It would initially draw on the screen, but then would not update the drawing. After placing 'NSLogs' throught the program, we noticed that the animation was calling draw rect once, and would even the calculate function repeatedly, but was not sending drawing messages for some reason. We could see the values that we were trying to draw changing but were not able to get the draw rect to reflect those changes.
We fixed this by copying the code into an application provided by Dr. Lindner and we were then able to run it. Sadly, we were not able to figure out what was initially wrong.
Later we had issues making the duffing sections appear for anything other than rk1. For the other algorithms, we were getting a point particle moving around. We decided to print out and plot the data for the position to see what was going on, and can see the plots in figure 1.
With these graphs, we figured out that it was actually an initialization error that was causing the sections to appear as they were. We changed the amplitude from 0.1 to 0.19, and this fixed the issue.
We also realized that it would be wise to instead of allowing the user to explicitly change the time step, for the user to chose how many sections of a period one time step should be cut into. This helped make the duffing sections appear better.

## 2 Analyzing different algorithms

After getting the application to work we started studying the various integration algorithms by altering the time step. First, I set the algorithm to RK4, and decreased the time step. The algorithm failed at around 8 periods per dt. I then switched to RK3 which failed at 14 periods per dt. RK2 failed at 19 periods per dt, and EC failed at 11! Coming in last place, was RK1, which failed at around 107 periods per dt.

Figure 1: Graphs of Euler-Cromer and RK1 Duffing Sections. We expect that the EC should be better. Comparing the first graph, with a lower time step, and the second graph, with a bigger time step, we can see that RK1 changes more. As the second graph shows what the application was initially showing, we know that our input variables are incorrect.




It was interesting to note how robust EC is, however, I also noted that it became inaccurate at lower number of periods per dt.

Figure 2: A screen shot of the Application