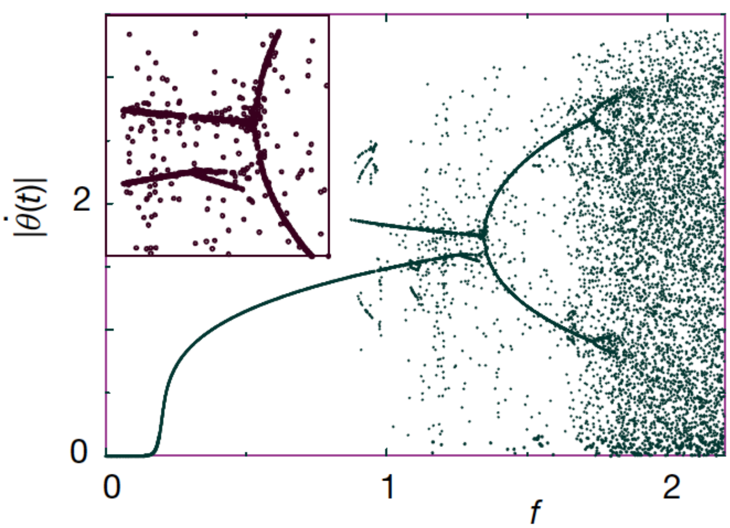
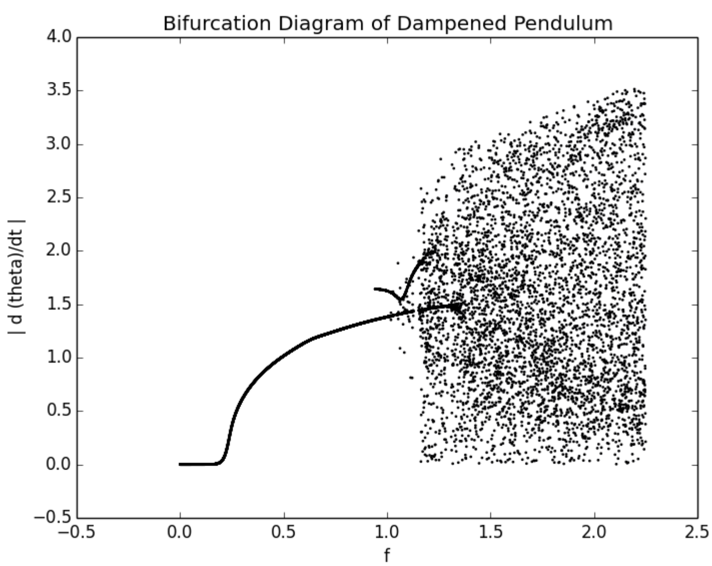
Computational physics
Assignment 11
Roy Rinberg

15.3
Create the bifurcation diagram from the book:



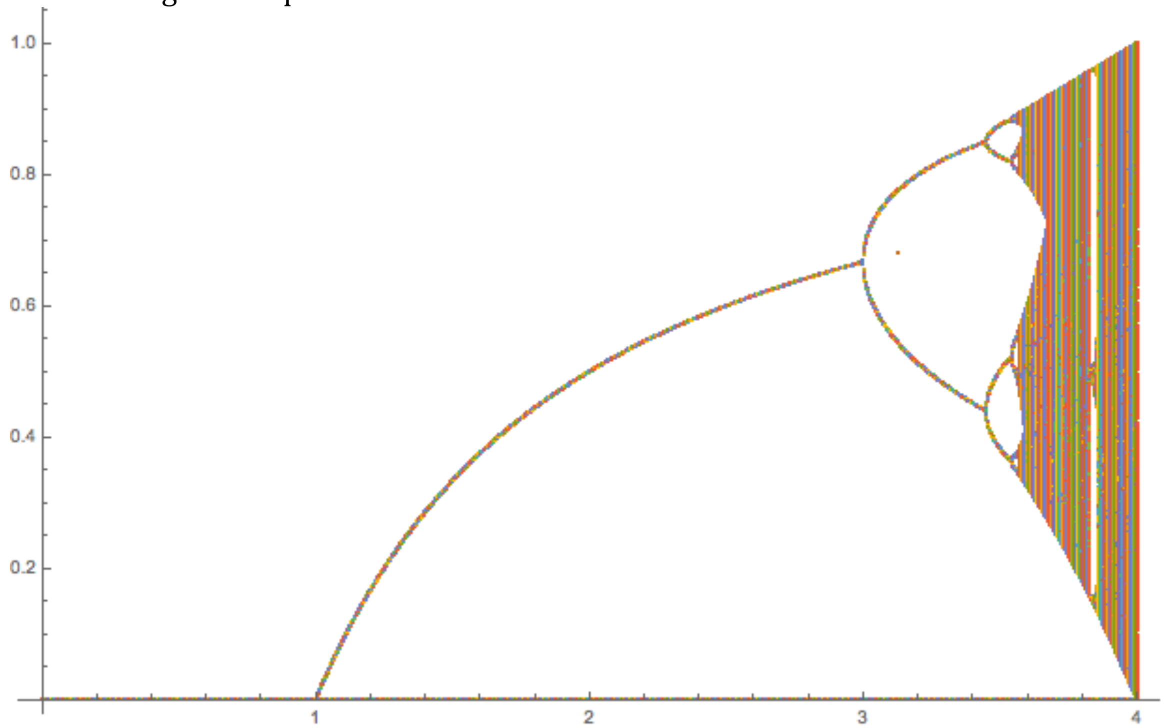
And so I did this:



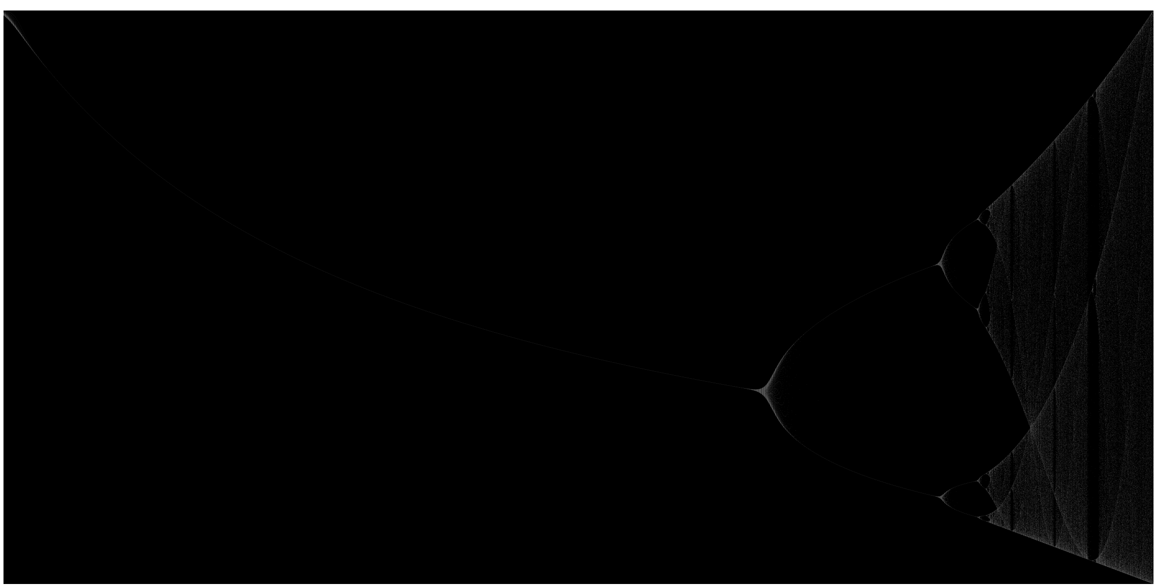
By iterating through RK4 a certain (specified) number of times in time, and then plotting the last value of d Theta /dt.

It doesn't look perfect, but it looks relatively close to what we expected – the chaotic behavior begins occurring earlier in f than in the book, and also there are fewer meaningful bifurcations.

Here is a logistic map that I made in mathematica:

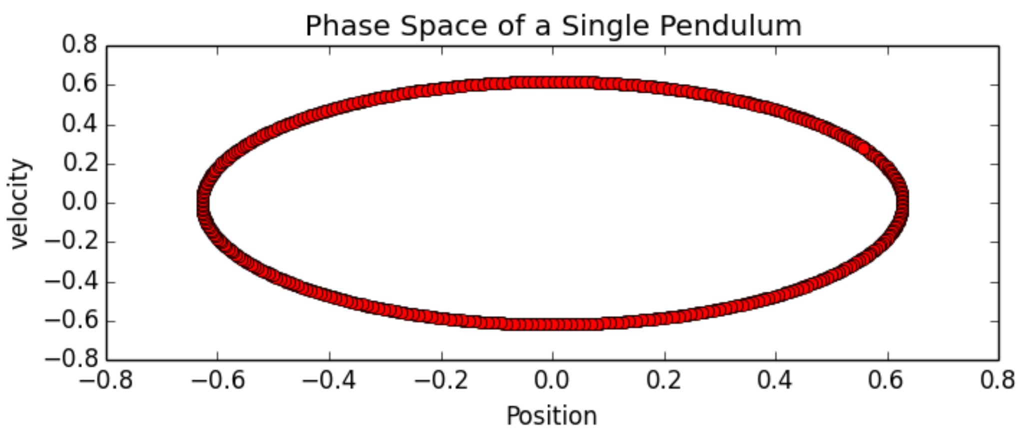


Here is a logistic map that I made in python, largely from scrapping code from the internet (it looks a lot nicer than in mathematica), but also is white on black– don't print this – delete this picture before printing, it will be a waste of ink.



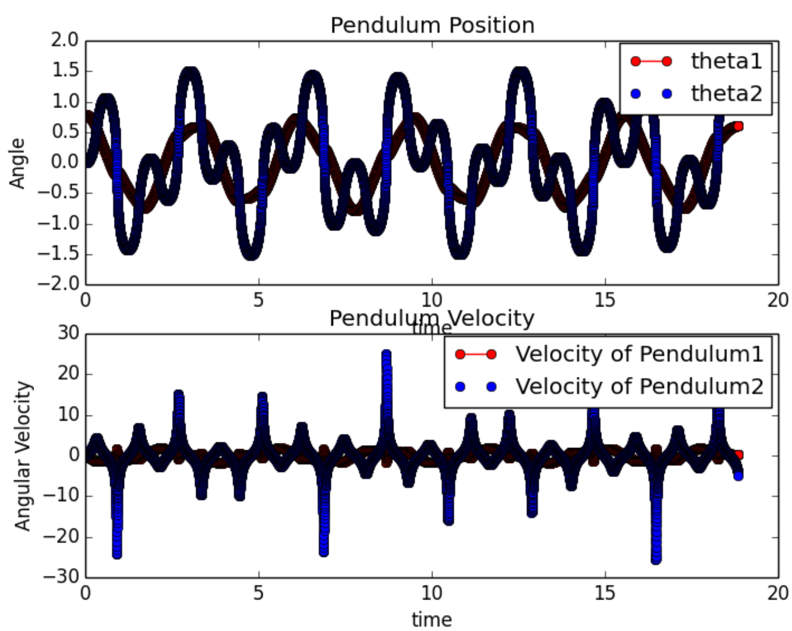
This week I worked mostly on creating a double pendulum simulation entirely from scratch – entirely using Lagrangian equations of motion, and applying rk4 technique we developed previously

To prove that the methodology works, here is the same algorithm applied on a single pendulum:

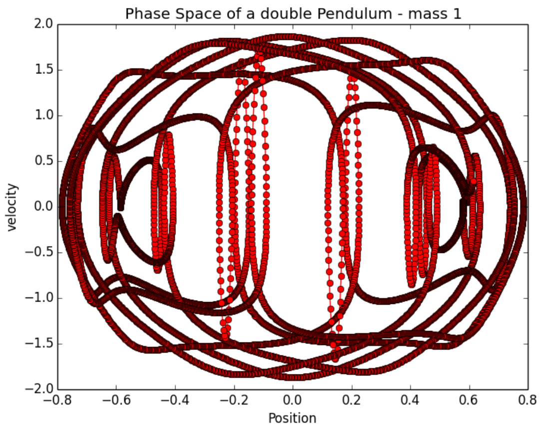


The following images are for an evolution in time for initial conditions : m1 = 2; m2 =1; L2 = 2; L1 =1; angular velocity 1 = 0; angular velocity 2 = 0; angle2 = 0; and angle1 = Pi/4

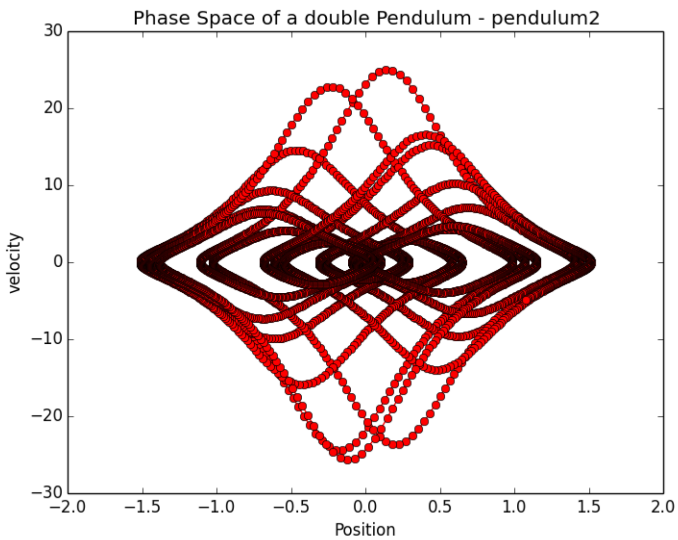
Here is a plot of the angles of both individual pendulums as a function of time, as well as the angular velocities of both pendulums as a function of time.
The Position of Mass 1 is largely sinusoidal, and mass 2 looks more like a beat frequency.



Here is a plot of the phase space of mass 1 of the double pendulum

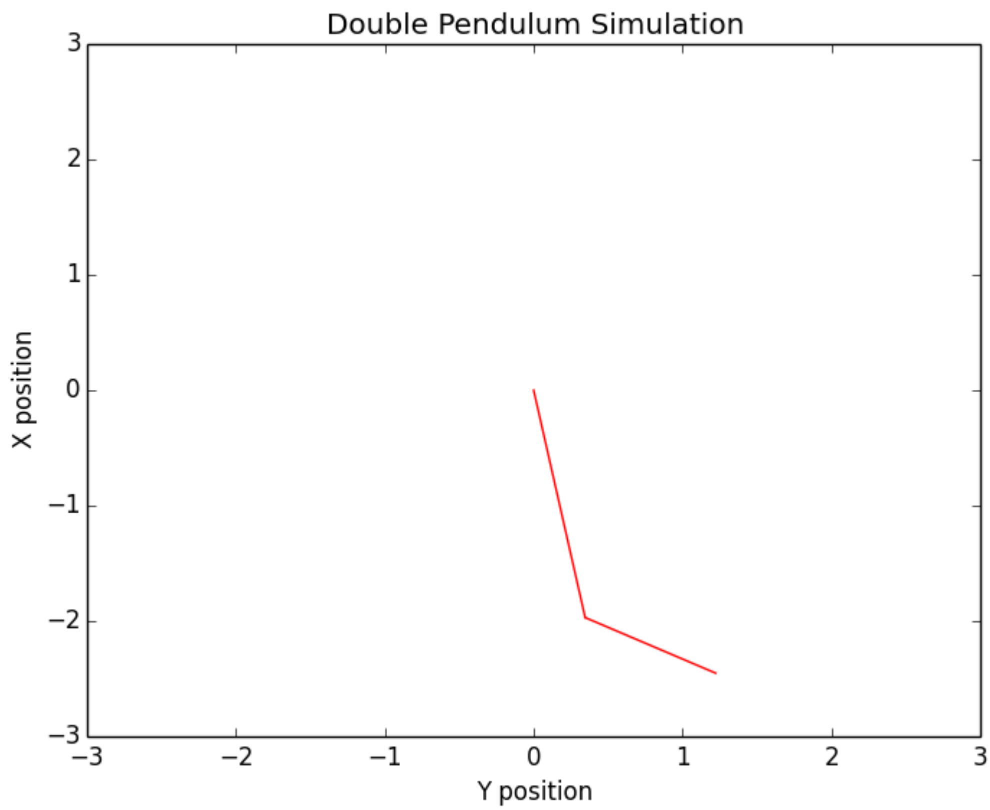


Here is a plot of the phase space of mass 1 of the double pendulum



Here is a snapshot of a simulation of a double pendulum that I created:

.


Double Pendulum Simulation
X position
Y position
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3