Homework 1

Shaun Fedrick
Computational Physics Haverford College
(Dated: February 11, 2020)

1. INTRODUCTION

This homework concerned the basics of python plotting use matplotlib.
The process was essentially:

- import matplotlib
- Create a figure: $figure = matplotlib.pyplot.figure()$
- Create a subplot: $ax = figure.add_subplot(111)$
  - with create an a subplot you can create multiple subplots to the same figure to get multiple graphs onto one figure.Subplot(int X,int Y,int n) denotes how many graphics can be placed in a given row column section.For example, if you were to use subplot(232) that would mean that there are 2 available rows for each of the 3 available columns and you would want to place the given plot in the second box a_{12} , in the 2 by 3 matrix.
- Create a title: figure.suptitle("Title")
  - This is the global title for all subplots you can also make individual titles for each subplot
- Label X and Y axis: $ax.set\_xlabel(r"Xlabel")$ & $ax.set\_ylabel(r"Ylabel")$
- Plot to a designated $subplot = ax.scatter(x, y, s = 10, c =' colorcode', marker = "o", label =' DataLabel')$ or $subplot = ax.plot(x, y, s = 10, c =' colorcode', marker = "o", label =' DataLabel')$
- Make a legend for designated subplot: $ax.legend(loc =' upperleft')$

2. 3.1

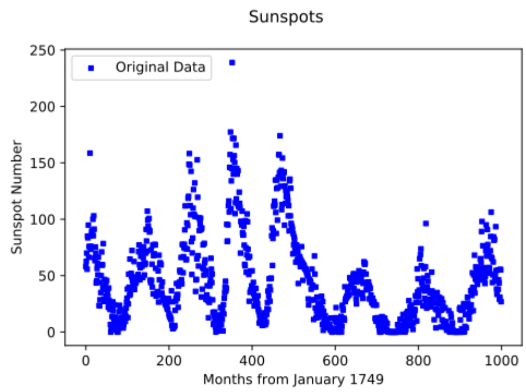

FIG. 1:

Problem 3.1 involved plotting data from from a .txt file containing sunspot data. A running average was then computed from the data by taking the previous 5 points and the next 5 points and averaging them. At the end points I just repeated the original point at x_i for the points that couldn't be accessed at the beginning or end of the list i.e points $x_{i-1}, x_{i-2}, x_{i-3}, etc$ and $x_{i+1}, x_{i+2}, x_{i+3}, etc$.

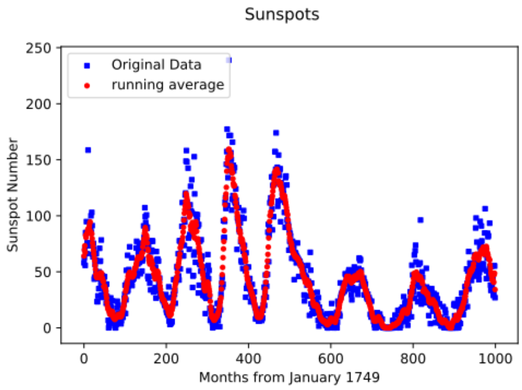

FIG. 2:

3. 3.2

Problem 3.2 involved plotting data using polar coordinates. I accomplished this goal by making a function that converted r and theta into x and y using $x = r * Cos(\theta)$ $y = r * sin(\theta)$. I then used this to make a Deltoid and a Butterfly plot.

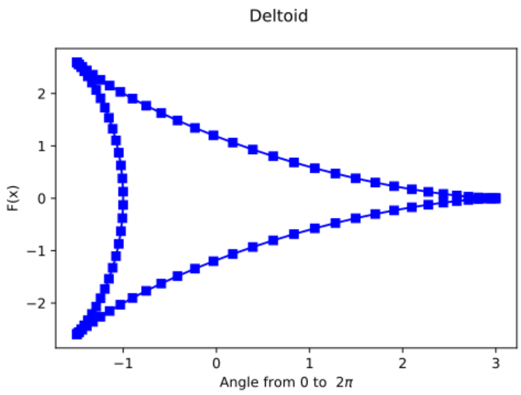

FIG. 3: Deltoid plot using Cartesian plot

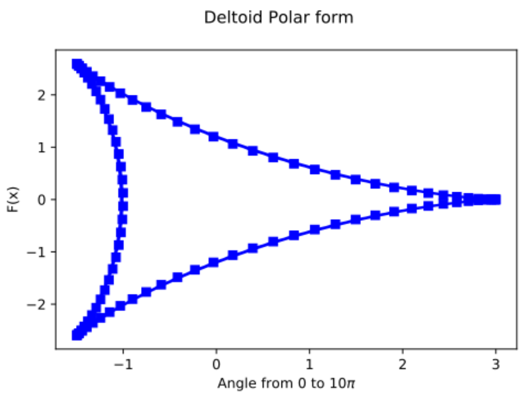

FIG. 4: Deltoid plot using Polar plot

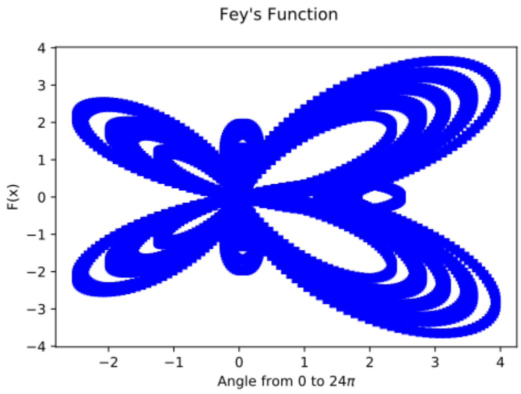

FIG. 5: Butterfly plot using Polar plot

4. 3.6

Data: Feigenbaum Plot
Result: Wrote the Feigenbaum plot recursively
This was done by setting the base case to when 1000 iterations are completed
if *The base case was reached* **then**
return $rx(1-x)$
else
send it back through the function with the new value of $x = rx(1-x)$
end

Algorithm 1: How to Plot the Feigenbaum function recursively

I then used this function to append values to a list which I plotted against a linear space of r values from 1 to 4 with a step size of 0.01 I hypothesized that a steady point would look like a series of consistent points. An oscillating point would like sinusoidal and chaos would have no coherent structure. This is past the deadline, but I have to revise this. After watching a Veritasium video on chaos I realized my plot was not completely correct. Even though I followed the instructions to run my fig tree function a 1000 times each iteration I did not get the branches shown in the studies Veritasium mentioned. As a result, I went back to my code and realized that my plot was not correct because to capture the oscillations of points you need to use a number of iterations that are spaced in mod 4 space. This meant that I should've plotted 1000,1001,1002, and 1003 on the same plot to show all the branches of the fig tree. And I now think that the sinusoidal points look like branches in the tree instead of the vaguely oscillatory motion I noticed in my other plot, which could be attributed to the function settling.

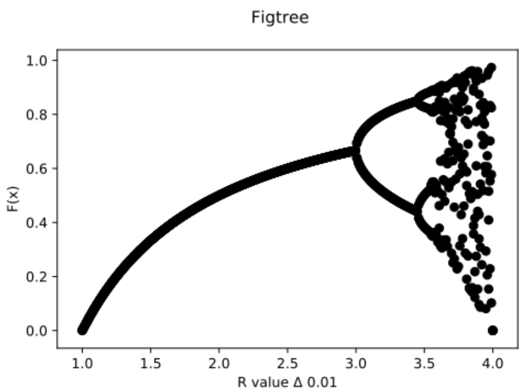

FIG. 6: Fig Tree plot

Based on my plot I would say the edge of chaos occurs at $r = 3.7$

5. CONCLUSIONS

Overall I enjoyed this homework far more than the last because of the ample amount of choice given to choose interesting problems. I found the skills I learned in this problem set to be very useful for my future stem career, and the timing needed for the homework to be ample. I thoroughly enjoyed this problem set.