Hw 3

Stefany Fabian Dubón*
Bryn Mawr College
(Dated: February 16, 2023)

[Solve exercises 3.6, 3.7]

1. INTRODUCTION

For this homework assignment, an introduction to Deterministic chaos and the Feigenbaum plot was given which is determined by the equation:

$$x' = rx(1 - x) \quad (1)$$

This is an iterative map, and the way it works is that for a given value of the constant r, an input value of x is fed into the right -hand side of the equation, which then gives a value of x'. This process is then repeated multiple times and gives one of three possible scenarios. 1) The value settles down to a fixed number and stays there. 2) It settles into a periodic pattern 3) It generates what looks like a random sequence of numbers , that although might not look like it, but are very predictable.

The Mandelbrot set was also introduced, which is a mathematical object that contains structure within structure, and involves the repeated iteration of the following equation:

$$z' = z^2 + c \quad (2)$$

Here, we take an initial value of z and feed it into the equation to get a new value of z'. A point is in the Mandelbrot is in the set only if the iteration never passes z =2.

2. EXERCISE 3.6

Used Equation 1 to create a program that calculates and displays the behavior of the logistic map. I first define the given information such as x=0.05 and the number of iterations (1000). I then had to define the logistic map function that takes r, x, and the number iterations, which computes the behavior of the logistic map (using the Equation 1) for 1000 iterations and returns a list of the values for x. Since I was given a range for r, I then initialize the range from 1 to 4 in steps of 0.01. The next step is to iterate over each value of r and compute the logistic map for r, using the previously set arguments. Finally, I set the horizontal (r) and vertical (x) axis, and plot the logistic map as a scatter plot.

Answer to the questions

a) For a given value of r, a fixed point happens when when the value settles down and stays there. For example if I put x=0 on the right hand side, I will get $x' = 0$, which looks like just a point on the Feigenbaum plot. A limit circle, settles down into a periodic pattern, where it repeats a set of values in sequence

b) Based on my plot the edge of chaos happens at around 3.65

3. EXERCISE 3.7

Equation 2 used to get the Mandelbrot set

Equation used to find all values of c:

$$c = x + iy \quad (3)$$

To create a program for the Mandelbrot set, I first set the N x N grid spanning the given region where $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$, then I set the number of iterations to be performed. Next, I had to create a a while statement for that it would loop over all the points in the grid, and for each point, the Mandelbrot iteration could be performed. To do this I sent z = 0 as the initial starting value and repeatedly apply the formula $z' = z^2 + c$ until the value of z passes 2 or the maximum number of iterations is reached. Then, I assigned the color corresponding to the grid point.This means that if —z— becomes greater than 2, then the point c is not in the Mandelbrot set and the corresponding grid point will be white; however, if the maximum number of iterations is reached and —z— is still less than 2, then the grid point is black. Finally, I made a density plot in which grid points inside the Mandelbrot set are colored black and those outside are colored white.

---

*Electronic address: sfabiandub@brynmawr.edu; URL: Optionalhomepage

4. RESULTS

For exercise 3.6, I got the following logistic map:

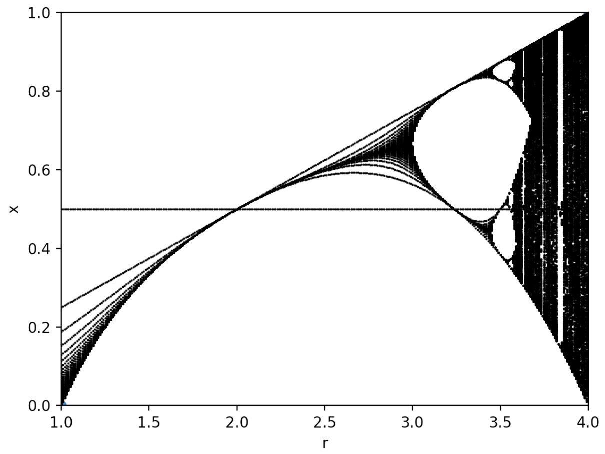

FIG. 1: Feigenbaum plot

For exercise 3.7, I got the following plot.

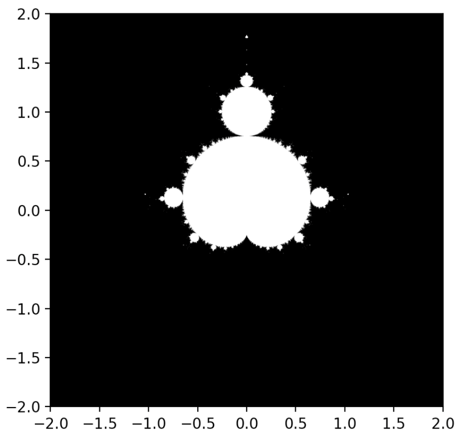

FIG. 2: Image of the Mandelbrot set

5. CONCLUSION

Survey Questions The homework took me some time, mainly because I had some trouble with the creation of the grid in exercise 3.6 and because vscode kept telling me that there wasn't a matplotlib module. I wasn't ab;le to produce any of the plots in vscode because it stills shows an error but when I tried to do it straight from the terminal it did work. I learned on how to put graphs and figures in LaTeX, which is a good skill to have and about iterations. The most interesting problem to me was 3.7 because I create a pretty cool plot. I think that the homework is a better length now.

Pseudo Code for 3.6

```
x = 1/2
num_iter = 1000

   # Behaviour of the logistic map
def log_map(r, x, num_iter)
 x_list [x]
  for i in range (num_iter)
      x = r x (1-x)
     x_list.append (x)

 return (x_list) <- values of x as

  # Use the range of r values giv
r-val = np.arrange (1, 4, 0.01)

  # Now iterate over each value
For r in r_values :
   r_values = logistic_map (r, x, ,
plt.scatter ()

 # set horizontal & vertical axis
 x = plt.xlim (1, 4)
 y = plt.ylim (0, 1)
plt.xlabel ('r')
plt.ylabel ('x')
plt.scatter (x, y)
plt.show ()
```

Pseudo Code for 3.7

```
# Define the resolution of the image
 N = 100 - to start

 # Number of iterations
num_iter = 100

# Create the grid of complex numbers
  x-values = (-2, 2, N)   (given range)
  y-values = (-2, 2, N)
  grid = array (x + y · 1j)

# Initialize z, i for a given complex val
 for c in grid:
   z = 0
   i = 0

 # create a while statement to implement
while < number of iterations and to the
 z = z^2 + c <- formula
 i += 1
 # assing color
 if abs(z) >= 2
     assign color white
 else :
     assign color blk
plt.inshow ( ) plot the Mandelbrot se
plt.show
```