# Math 5330, Mathematical Modeling: Bifurcation diagrams for iterated maps

# *In class, 11/6/2019*

## Bifurcation diagram for logistic map

6.

Write a program to generate a bifurcation diagram for the logistic map (Equation (3) in May’s paper), as shown in:

<https://en.wikipedia.org/wiki/Bifurcation_diagram> (see first figure in the “Logistic map” section)

The code is constructed as follows (this is just a “pseudocode”—you will need to write the actual program using Python syntax).

Initialize: a\_min, a\_max, n\_a, min\_iter, n\_iter

Initialize x\_vec and y\_vec as all zeros (length is n\_iter – min\_iter)

Initialize a\_vec as n\_a values between a\_min and a\_max (use linspace)

Loop over values of a\_vec (use index ii)

set y = 0.5

Iterate the function F(y,a\_vec[ii]) min\_iter times (you will be updating the value of y, while leaving a\_vec[ii] fixed)

Loop over index jj with n\_iter – min\_iter iterations, and store the next n\_iter – min\_iter iterations of the function in y\_vec as follows:

Compute y=F(y,a\_vec[ii]) to obtain the next value of y

x\_vec[jj] = (ii’th value of a\_vec)

y\_vec[jj] = y

Use scatter to plot x\_vec (on the *x* axis) versus y\_vec (on the *y* axis). Use a marker size of 0.01, and color blue.

To test your code, use the following values:

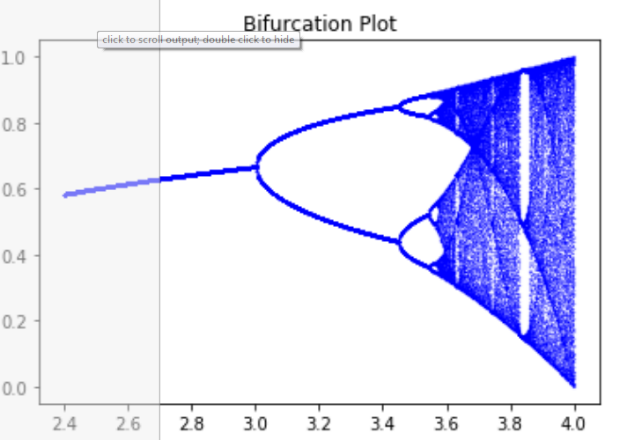
a\_min= 2.4

a\_max = 4

n\_a = 2000 (when testing, use n\_a = 200)

min\_iter = 100

n\_iter = 200

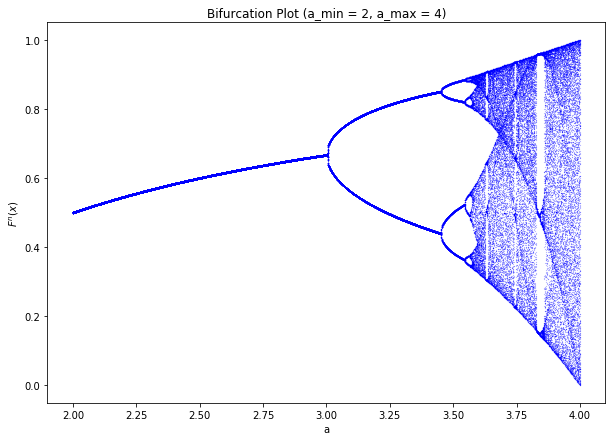


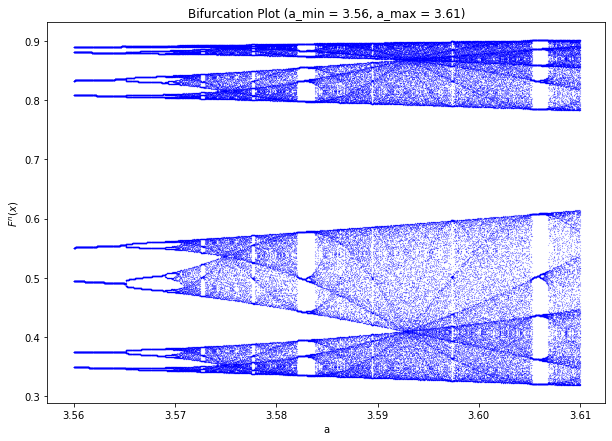
## To hand in:

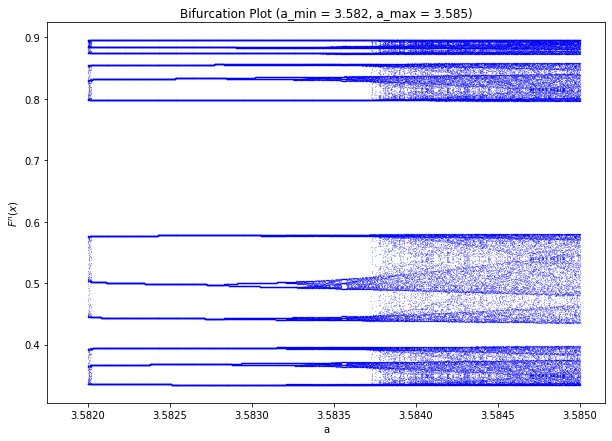
* + Your code (as .py file)
  + In your Word file: (a) Plot for a\_min = 2, a\_max = 4; (b) Plot for a\_min = 3.56, a\_max = 3.61; (c) Plot for a\_min: 3.582, a\_max: 3.585. (As before, save your plots using the snipping tool. (d) Your commentary on these three plots. What do they tell about how the iterates of the function behave for different values of ?

**Response:**

As the value of ‘a’ increases the period of the plots also increases. If ‘a’ gets big enough, the process leads to chaos and the plots start to subdivide into different subplots.

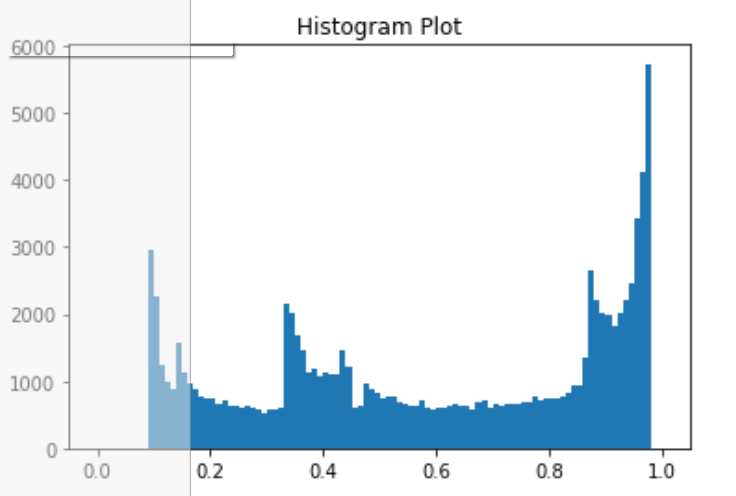






## Equilibrium distribution for logistic map

We have seen that the iterates of the function hop around. The *equilibrium distribution* gives the probability distribution of function values that the system attains.

Modify your code for the previous activity to iterate the logistic map n\_iter times for a single fixed value of *a*, and plot a histogram (using the hist function of matplotlib.pyplot) with bins given by linspace(0,1,n\_bins+1) of the resulting values of y. See below for an example with *a* = 3.9, *n\_bins* = 100, *n\_iter* = 100000.

To hand in

(due Thursday, October 22 at midnight):

* Your code (as .py file)
* In your Word file: One plot for each of the following values of *a*: (a) *a* = 3.64, (b) *a=* 3.68, (c) *a =* 3.72.

Use, *n\_bins = 100, n\_iter = 100000*.

* Explain the relationship between your plots and the bifurcation diagram you created previously.

**Response:**

The histograms here also reveal the same information that the bifurcation plots revealed; chaos is achieved if the ‘a’ gets large enough. It seems clear that as ‘a’ increases from 3.64 to 3.68 to 3.72 more values begin to appear on the plot. For example, in the plot where a = 3.64, no function values were show at region 0.7, but as ‘a’ increases in later graphs the plot becomes more complex and chaotic.

