Dragos Pop (2618006)

# Assignment 1 - Logistic map

## Concise method description

The function implemented in Matlab computes the logistic map $x_{n+1} = \lambda x_n(1 - x_n)$ for a sequence of lambda values in a demanded interval and plots the orbit diagram.

## Function syntax

Given the required name of the Matlab file, the function can be called by *Assignment_1_Pop()*, specifying the right arguments. These are:

- *lambdaMin* = the lower bound of the interval; should be larger than 0 and smaller or equal to lambdaMax
- *lambdaMax* = the upper bound of the interval; should be larger or equal to lambdaMin and smaller or equal to 4
- *N (optional)* = the number of lambda values in the interval [lambdaMin, lambdaMax] for which the "attractor" is computed. In case N is not specified by the user in the input, it takes a default value of 1000 (i.e. the function works even if the input is only lambdaMin and lambdaMax)

Accordingly, calling *Assignment_1_Pop(3.4, 4, 1000)* is the same as calling *Assignment_1_ Pop(3.4, 4),* since N is optional, and returns the picture below. This resembles Figure 10.2.7 from the book *Strogatz S. H., Nonlinear Dynamics and Chaos*, which indicates that the function performed successfully.

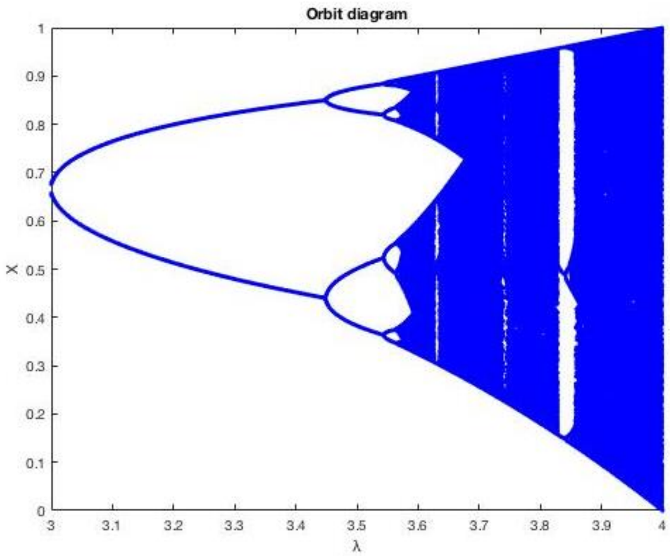


Figure 1 – Orbit diagram of logistic map on [3.4, 4], N = 1000

## Restrictions

The program has the following requirements in order to work effectively:

1. The input has to satisfy the given input condition: $0 \leq \lambda_{min} \leq \lambda_{max} \leq 4$
2. The initial population $x_0$ has to belong to the interval (0, 1) instead of [0,1] because when it takes the value 0 or 1, the value yields a steady state 0, as the image below shows. Moreover, one can see by looking at the function $x_{n+1} = \lambda x_n(1 - x_n)$ that $x_0$ actually belongs to (0, 0.5] since the terms are mirrored in relation to 0.5. For instance, $x_0$ = 0.2 gives the same result as 0.8, because 1-0.2 gives 0.8 and vice-versa. After trying the function multiple times, it was concluded that as long as the initial population is part of (0, 0.5], regardless of its value, it produces the same result.

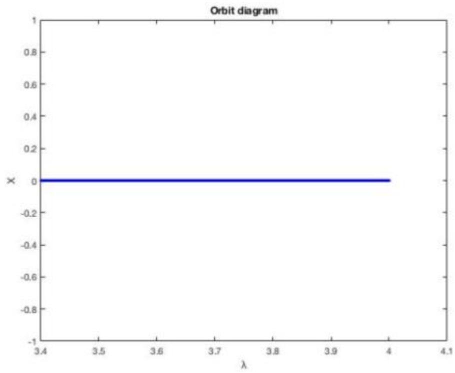


Figure 2 – Orbit diagram of logistic map when the initial population $x_0$ is 0 or 1 reaches a steady state 0

## Program design

In the beginning, the N parameter is set to optional, followed by the implementation of the input condition $0 \leq \lambda_{\min} \leq \lambda_{\max} \leq 4$, which stops the program and warns the users if not respected. Next, the initial values are assigned and the data structured initialized. In order to store values in 2 dimensions, x is initialized as a matrix of n rows and N+1 columns when lambdaMin is different from lambdaMax or n rows and 1 column otherwise, while the values lambda takes are stored as a vector.

At this point, for every lambda, an orbit of size 600 is computed, but only the last 300 considered, according to the theory in Chapter 10.2 of *Strogatz S. H., Nonlinear Dynamics and Chaos*, in order to “allow the system to settle down to its eventual behavior”, iteratively using the result from the previous operation. Lastly, the values are plotted against the respective lambda value that was used in their computation.

## Figure interpretation

The illustration on the right was generated after calling Assignment_1_Pop(0, 4, 1000). Along the horizontal axis it has the values of lambda and on the vertical one, the values of ($x_{301}$, $x_{302}$, …, $x_{600}$). From it, one can distinguish 3 parts. First, when lambda is smaller than 1, the population goes extinct (steady state 0) because it is continuously decreased by lambda. In the second part, from lambda ranging from 1 to 3, the population reaches a non-zero steady state that increases together with lambda. Lastly, when lambda takes values in between 3 and 4, it is visible that the population starts to oscillate. Moreover, from the Figure 4, one can better see that the population bifurcates in 2 directions at $\lambda = 3$, following a period-2 cycle until it reaches approximately 3.45 when it starts a period-4 cycle. The same pattern seems to continue further on, up until the point where the details are impossible to be reflected in the picture. Consequently, it is fair to state that as lambda grows larges, so does the chaos.

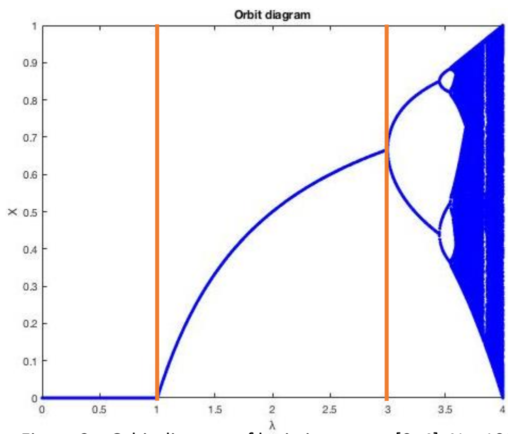


Figure 3 – Orbit diagram of logistic map on [0, 4], N = 1000 has three parts (0, 1), (1, 3) and (3, 4)

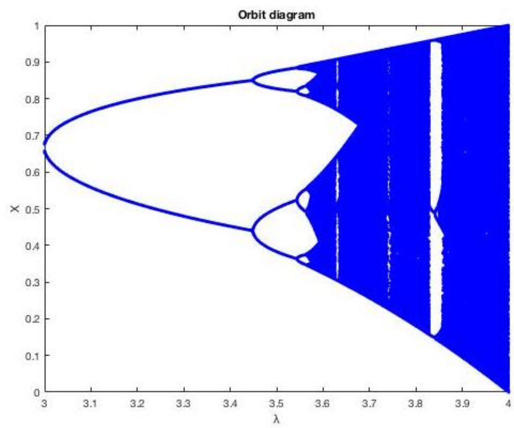


Figure 4 – Orbit diagram of logistic map on [3, 4], N = 1000 oscillation