ROBOTIX

Software Domain Recruitment Document

JC NAIDU R
Software Domain Head

May 20, 2023

---

1 The growth/decay of population of animal species.

1.1 Description

Suppose we want to model the population of rabbits. We have x population this year, x represents the percentage of theoretical maximum population. Now let us have a simple assumption that this population grows every year by the growth rate r. We will have a simple equation for the population model given by

$$x_{n+1} = rx_n$$

But this assumption would imply that growth will be exponential (which is not true as x will grow larger than 1). We will take into account of the constraints of the environment and modify it as follows

$$x_{n+1} = rx_n(1 - x_n) \tag{1}$$

The introduction the factor $(1-x_n)$ makes sure that when x becomes maximum, the population is constrained. This term limits the population from growing exponentially.
If we take any particular value of $r \in (1, 2)$ we will observe that x reaches an equilibrium after enough iterations regardless of the initial population. Plot a graph between any arbitrary x (not 0 0r 1 obviously) and any $r \in (1, 2)$. You will find that x reaches an equilibrium. Now change the value of initial x. You will observe that irrespective of the initial x, it will always reach a particular equilibrium value for a given r. How is the equilibrium value related to r? Now plot $x_{equilibrium}$ against r. Increase the value of r above 2. What will happen now?

1.2 Things to submit

a) Plot between r and $x_{asymptotic}$ when $0 < r < 4$.

- Start with a vector of r values, and a vector of x values
- Perform many (how many?) iterations on the whole vector of x values, so that each place in the vector is updated according to its appropriate r.
- Plot the resulting x values vs. r values.
- Submit a *.png* file.

b) Code used to to generate the plot.

1.3 Things to note

1. Make sure that r is dense enough in the graph so that it can be assumed continuous for human eye.
2. Any language of choice can be used to plot the graph but Python or Matlab should be preferred.
3. If the values are not reaching an equilibrium, take asymptotic values. Your graph should look something like this:

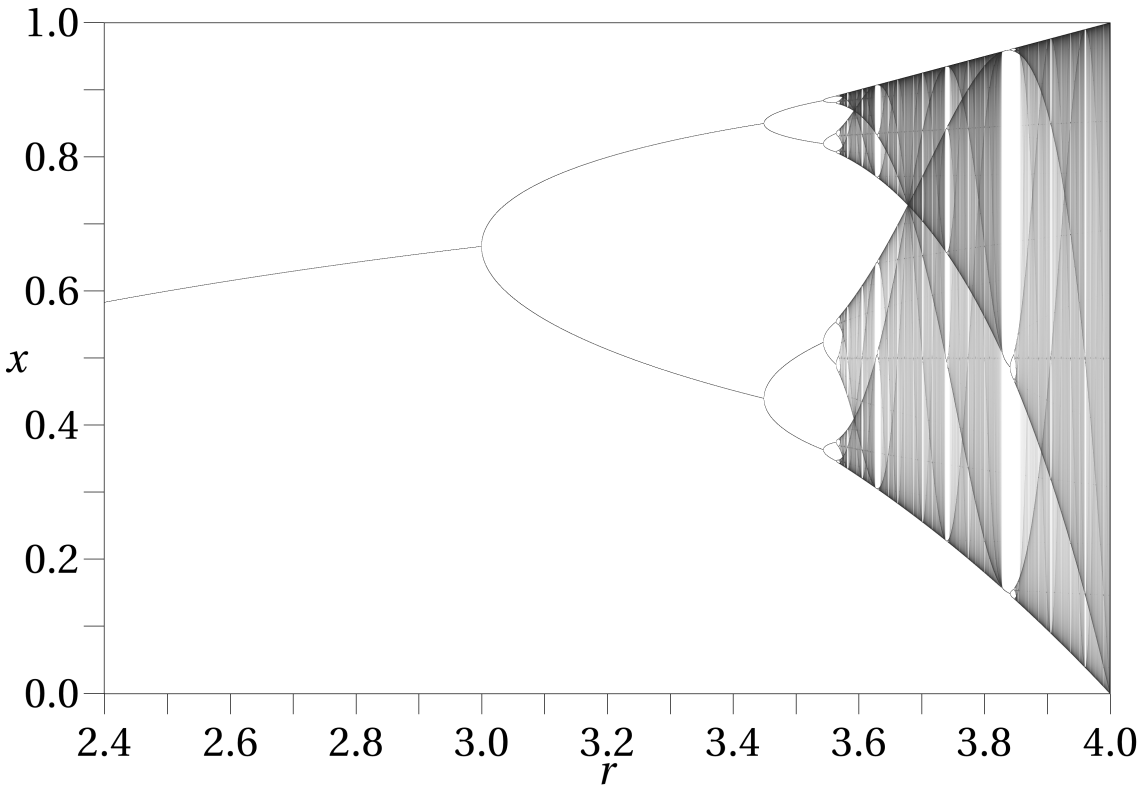

Figure 1: Bifurcation diagram for logistic map

4. Plagiarism would result in immediate disqualification. You will have to explain each bit of your code in a *viva*.

2 Conway Game of Life

2.1 Description

The world consists of a 2D grid. Each cell in the grid can be "alive" or "dead". At each step the cells are updated according to the following rules:

- A dead cell will become alive if it has exactly 3 live neighbors (each non-boundary cell has 8 neighbors in this grid).

- A live cell will die unless it has 2 or 3 live neighbors.

We use a matrix to hold the grid. A cell is "alive" if the relevant matrix element is 1 and "dead" if 0. Each step is referred as a generation. In each generation:

1. Figure out how many live neighbors each cell has.
2. Update the grid.
3. Plot the grid.

2.2 Things to Submit

Code used to generate the plot

2.3 Things to note

1. Since we have limited computation power, limit the grid size to $100by100$ during the computation.
2. The location of the initially "alive" cells and the generation given as input. The code should plot the input generation's plot.
3. Refer to this Wikipedia article for more details https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
4. The language used must be Python or Matlab.

3 Help Batman catch Joker

This problem statement yet to be prepared. Basically creating bitmap animation to chase an object.

3.1 Description

The Gotham city is falling apart and Batman has to catch Joker who created this menace. You will implement a chasing algorithm for Batman to chase and catch the Joker who is movingly randomly.

1. Build a graphics subsystem for simulation. You have to start by constructing a 64 by 48 black image and creating a *.bmp* reader to show the image.
2. Next step is to put the bitmap data including it's header data and pixel data into one continuous stream of bytes.
3. Test the working with test cases like
    - Clearing the image to a colour.

- Drawing polygons.
- Implement a function to translate a vector of polygon vertex points by a constant x and y. You have to figure out creating continuous images and showing them in a window to create animation.

4. Create a triangle to represent Batman and test its rotation and translation.
5. Create a triangle to represent Joker and implement a basic coin flip algorithm to determine its motion. Flip a coin every 5 seconds.The outcome of the coin flip will determine the direction Joker would take a turn.
6. Both Batman and Joker move with constant speed.
7. Produce the animation with at least 12 frames per second.

3.2 Things to submit

A folder with all the files including:

- A README file on how to compile and run.
- A list of all required libraries and dependencies.

3.3 Things to note

- Candidate has the full freedom to make appropriate changes to the problem statement with prior intimation (or clear description of changes made and reason for the change in README file).
- This is an open ended problem statement and can be solved in multiples. Submit whatever you have done even if not fully working.

Resources

The software domain involves learning a lot on our own. So it is expected from you to search on internet and learn. Some basic resources were given.

How to submit

A git repository of the solutions need to be made and the repository link is to be submitted.

A word from problem setter

These problems were curated with inspiration from various university courses. Even though they seem complicated at first, with enough patience and thinking they are merely implementing simple algorithms. Please refrain from plagiarism. In case of any confusion, please contact me on +91 6303 49 7076 or Whatsapp me or mail cs21btech11050@iith.ac.in.