USING LOGISTIC MAPS TO SIMULATE PATTERN FORMATION OF Passiflora Incarnata

Ishaan Misra and V. Ramanathan\*

Department of Chemistry, Indian Institute of Technology (IIT-BHU), Varanasi

(\*Author to whom co-respondence must be addressed at [vraman.chy@iitbhu.ac.in](mailto:vraman.chy@iitbhu.ac.in))

Abstract: The change in the color pattern of the petals of *Passiflora incarnata* is studied using the chaos theory in the form of logistic maps and plotted using the corresponding bifurcation diagram. Based on a colorful inspection of the beginning of violet-coloured dots along the filament of the flower's bud stage and the emergence of alternating bands of violet and white colour in the matured bloom, it is possible to deduce that a two-degree model for polynomial mapping can be used to study the temporal oscillations in the flower.

Keywords: *Passiflora-Incarnata,* Logistic maps, Bifurcation diagram, Polynomial mapping, temporal oscillations

* 1. **INTRODUCTION**

The biological world is full of appealing patterns and symmetry (Ball 2016). These can take the form of different-shaped organisms or coloured designs on their skins/surfaces. The emergence of such patterns in nature is of primary interest, not only in the individual circumstances where such patterns are found but also in comprehending a variety of phenomena in both natural and man-made systems. It is unclear at what stage of an organism’s growth a pattern begins to emerge and when it fully sets in. Turing(1952) proposed a mathematical model as part of the reaction-diffusion (RD) theory to demonstrate that morphogenesis could lead to the formation of spatial patterns triggered by random fluctuations. A more general form of the reaction-diffusion equation, derived from its original form proposed by Turing(Murray 2003)

Where u is a vector of chemical concentrations, D is a matrix of diffusion co-efficient and f is the chemical coupling matrix with kinetic parameters p. Considering only two variables A and I to represent the activator and inhibitor in equation(1), we obtain

Various studies have been done to explain temporal oscillations in the Passion flower, *Passiflora Incarnata*(PI)*,* starting from the modified Lotka (Lotka 1910) model. The model comprised a florigen (Chailakhyan 1936), that is responsible for flowering and an antigen (Lang and Melchers 1943) that works against flowering. Although the florigen and the anti-florigen responsible for the temporal oscillations in PI flowering remain to be identified in biochemical terms, the former plays an autocatalytic role and the latter an inhibitory role. In this paper, we have applied the Logistic map model to linearly map the oscillatory violet and white colour pattern in the PI flower.

A logistic map is a polynomial mapping of degree 2, often used to understand complex, chaotic behaviour arising from simple non-linear dynamical equations. Particularly, we look at the bifurcation diagram of a logistic map that shows the values approached asymptotically as a function of the bifurcation parameter in the system.

* 1. **BIFURCATION DIAGRAM-LOGISTIC MAP MODEL**

Logistic maps are mathematical functions that describe the population dynamics of a species in a given environment. They are commonly used to model the growth of a population over time, taking into account factors such as reproduction rates and resource availability. The logistic map is a specific form of a recurrence relation, often used in the context of discrete-time dynamical systems. Mathematically the logistic map model can be written as

(4)

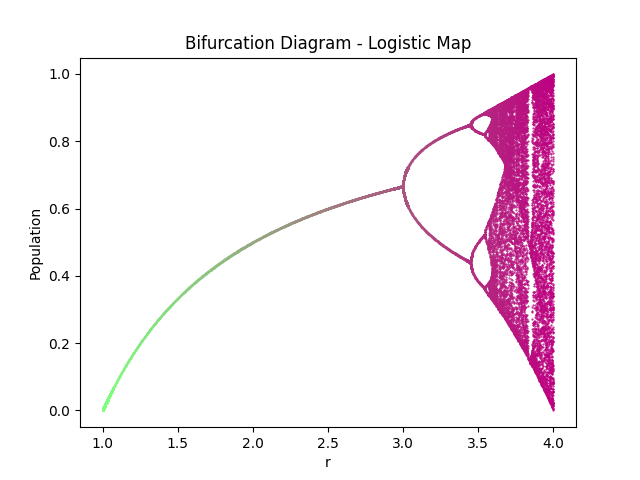
where:

* xn​ is the population at time n,
* xn+1​ is the population at the next time step n+1,
* r is a parameter that represents the reproductive rate or the strength of feedback in the system.

The bifurcation diagram is a powerful tool to explore the behaviour of dynamic systems and is widely used in the study of chaos theory and nonlinear dynamics like logistic maps. It provides insights into the complexity and unpredictability that can arise in seemingly simple mathematical models.

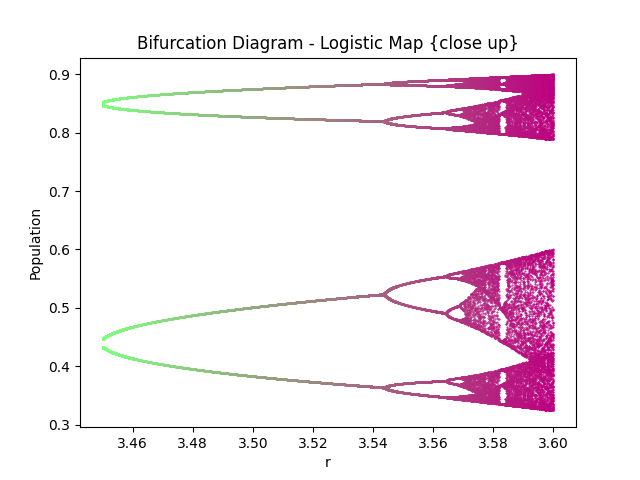
In a bifurcation diagram for a logistic map, r is chosen between 0 and 4. When r is varied between 0 and 4, the logistic map exhibits a variety of bifurcation patterns, including period-doubling bifurcations and chaotic behaviour. As r increases within this range, the system undergoes transitions from stable fixed points to periodic oscillations, and eventually to chaotic behaviour. The Python code(shown in Annexure 1) was developed to demonstrate a general bifurcation diagram for a logistic map that can be run on any compiler online or offline.

Now, if we look at the bifurcation diagram generated by the code(Annexure 1) we observe a repetitive colour pattern in the chaos regions which can be mapped to the development of the Passiflora incarnata bud.



**Figure 1.** General representation of a bifurcation diagram of a logistic map

If we zoom into the bifurcation diagram by changing the range of the parameter r from [1,4] to [3.45,3.6] we get the following pattern.



**Figure 2.** Close-up of a bifurcation diagram of a logistic map

From Figures (1) and (2) we can observe that the more we zoom into the graph the more we observe the same bifurcation pattern occurring multiple number of times leading up to chaos. This feature makes mapping of repetitive patterns like the activator-inhibitor model(filament of PI flower) much easier on the logistic map model.

We use Table 1 from the reference research paper as the basis of our model mapping.

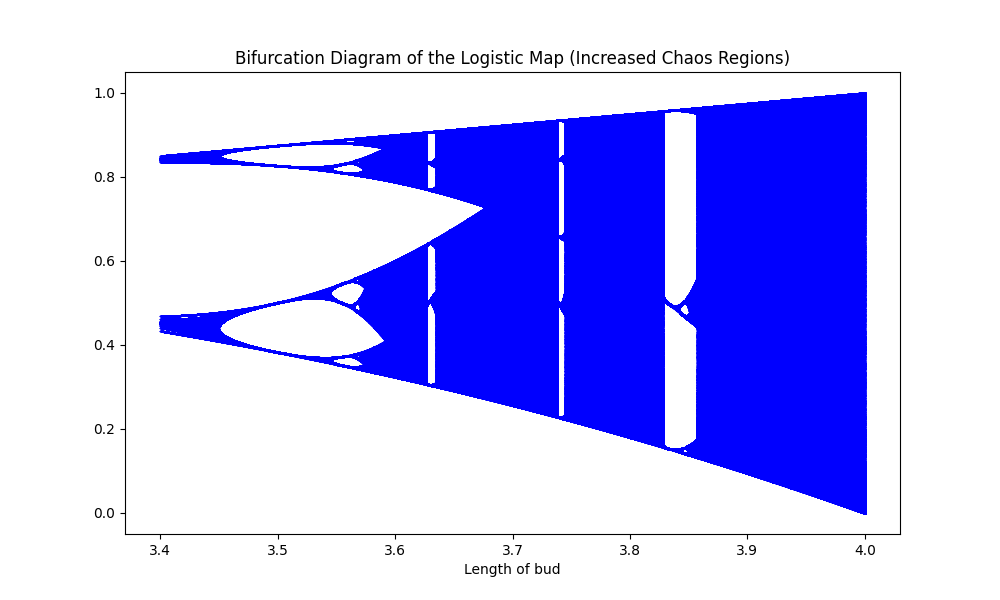
|  |  |
| --- | --- |
| Color of circle | Distance from center |
| V1 | 0.61 |
| V2 | 0.73 |
| V3 | 0.88 |
| V4 | 1.12 |

**Table 1.** Measured distance of the first 4 violet bands from the centre of the flower to the base of the filament.

We tweak our original model to

(5)

This offset of 0.075 in ‘r’ is done manually to optimize the logistic map model to efficiently and accurately map the color bands in the filament. This model considers the first 4 violet bands(Table 1b) by applying the bifurcation diagram visualization as a model to represent the colour pattern in the filaments of the flower as a non-linear dynamical system. On adjusting the widths of the chaotic region to match the violet patch of the small filament(filament in its early stages of development) in the bifurcation diagram, we can extend the logistic map with the same constraints to predict the filament growth and colour pattern. We use the code as shown in Annexure 2 to create a bifurcation map model to map a filament of the PI flower. And measure distances along the length of the PI filament(x-axis) to get the following graph.

****

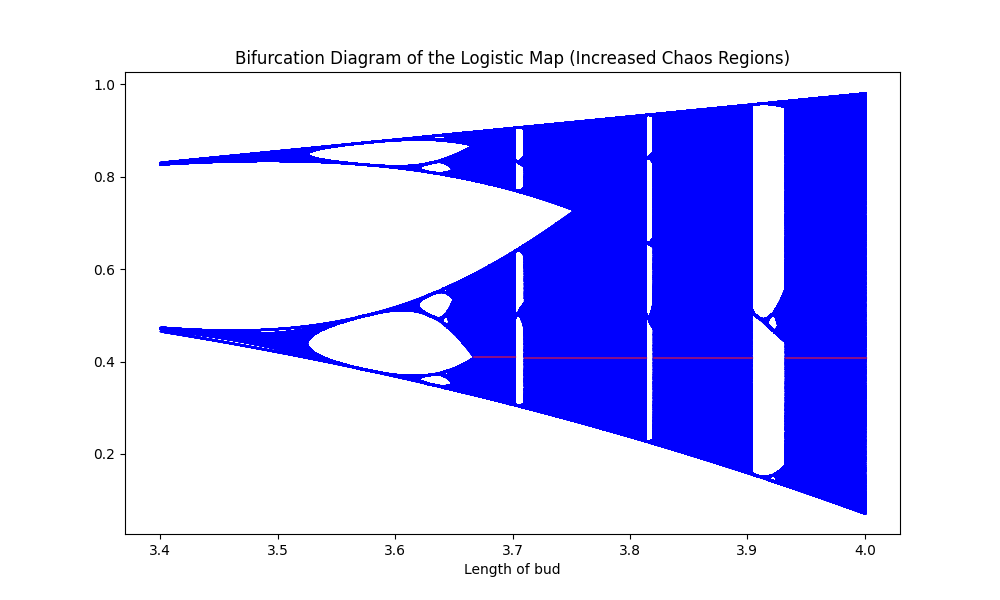
1. (b)

**Figure 3.** Bifurcation diagram mapped to a PI filament. **(a)** The photograph is the top view of the PI flower. (b) Bifurcation diagram of a logistic map to represent the colour pattern in PI flower.

* 1. **OBSERVATIONS**

The chaos region is the width of the color region in the bifurcation diagram, and it is calculated based on the distance of the region from the base to the average centre of the chaos region to mimic the distance calculated from the base of the filament to the violet color band.

We calculate the location of the chaotic region as shown in Figure 4 which resembles the first 4 violet and white colour patterns while ignoring any inconsistencies or connections between the chaotic regions. In V1, although chaos exists before the start of the red line we consider the minimum regions where chaos exists indicating the presence of one color (violet) without any residual white presence. We consider an arbitrary value for the base of the filament at 3.4 units and calculate the distances as follows.



V1

V2

V3

V4

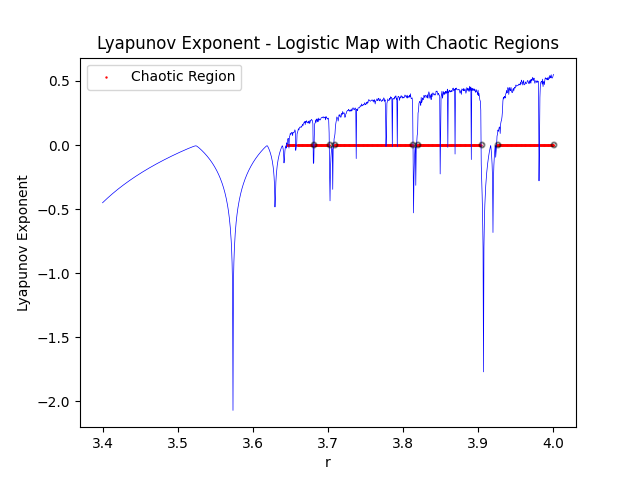
**Figure 4.** Red line marks the chaotic regions under consideration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Color of circle | Chaos Region **Start(S)** | Chaos Region  **End(E)** | Location ={S+E}/2 | Distance of base to chaos region |
| V1 | 3.68169 | 3.70157 | 0.29163000000000006 | 0.47808196721311486 |
| V2 | 3.70814 | 3.81265 | 0.360395 | 0.49369178082191784 |
| V3 | 3.81805 | 3.90333 | 0.46069000000000004 | 0.5235113636363636 |
| V4 | 3.931 | 4 | 0.5624950000000002 | 0.5022276785714287 |

**Table 2.** Measured distance of the first 4 blue chaos bands from the base.

To check our values of Start(S) and End(E) we also use the concept of **Lyapunov Exponent**. The Lyapunov exponent is a concept from the field of dynamical systems and chaos theory. It measures the rate of exponential divergence or convergence of nearby trajectories in a dynamical system. In simpler terms, it quantifies how sensitive a system is to initial conditions. In a chaotic system, small differences in the initial conditions can lead to vastly different outcomes over time. The Lyapunov exponent provides a numerical measure of this sensitivity. A positive Lyapunov exponent indicates chaotic behaviour, as trajectories diverge over time, while a negative Lyapunov exponent suggests convergence. Mathematically, for a system described by a set of differential equations, the Lyapunov exponent is often defined as the average rate of exponential growth or decay of the separation between nearby trajectories. It is represented by the Greek letter lambda (λ). The Lyapunov exponent is a crucial tool in studying and characterizing chaotic behaviour in dynamical systems.

We use the code as provided in Annexure 3 to calculate the Lyapunov exponent of our model and check the accuracy of the points we consider at the start and end in Table 2. The code in Annexure 3 gives us the following graph.



V1

V2

V3

V4

**Figure 5.** Lyapunov exponent for given r

Here we have modified r to r - 0.075, to calculate the Lyapunov exponent for our model. The red line in Figure 5 is the dividing line between the positive and negative exponents. The area above the red line is the chaotic region of our model and the black circle marks on the red lines denote the start and end points of the first 4 violet bands. The difference between start and end values obtained from the Lyapunov exponent curve and the logistic map model have a difference of less than 1e-5 thereby coinciding.

Now that we have the distance of the chaos regions from the base in Table 2 (column 5) and the distances of the violet bands from Table 1, we find a scaling parameter to scale our data. Dividing the values of Table 1 by Table 2(column 5), we get the scaling parameters for the four bands.

|  |  |
| --- | --- |
| Color of circle | Scaling parameter |
| V1 | 2.0916915269348144 |
| V2 | 2.0255552934974124 |
| V3 | 1.9101782109444527 |
| V4 | 1.9911288100338664 |
|  | 2.0046384603526364 |

**Average=**

**Table 3.** Scaling parameter.

We calculate the standard deviation of the scaling parameter which comes out to be 0.06542361897693987. This value is in the range of 1e-2 signifying that the values are tightly wound around the average(having less spread). Therefore we can consider 2.004 as our scaling parameter for our logistic map model to represent the PI flower filament.

* 1. CONCLUSION

This article explores the possibility of using logistic maps as an alternative to the oscillatory equations predominantly used for activator-inhibitor-type systems. It is hereby conclusively demonstrated that the open-source Python-based logistic map model works well for a small filament in a PI flower. Extension of the current study would be rewarding because the bifurcation diagram of a logistic map is fascinating and rich in its displayed behaviour. These pieces of code are general. With a little tweak in the structures of the equation and the parameters involved graph for the desired non-linear dynamical system can be produced. Another interesting project that can be derived from this paper is to generalize this piece of code for other oscillatory systems to observe various reactions.

* 1. **Acknowledgement**

The authors thank the academic infra structure of IIT(BHU) Varanasi.

* 1. **Declarations:**

The authors declare absolutely no conflict of interests.

* 1. **Author Contributions**

VR conceptualized the idea, IM wrote and executed the python code. IM and VR together

wrote the draft.

* 1. **References**

(All values and tables for reference are taken from Reference 1)

1. Bhati AP, Goyal S, Yadav R, N S. Pattern formation in *Passiflora incarnata*: An activator-inhibitor model. J Biosci. 2021;46:84. PMID: 34423786.

**Annexure 1**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  rs=np.linspace(1,4,2000)  N=500  x=.5+np.zeros(N)  endcap=np.arange(round(N\*.9),N)  for ri in range(len(rs)):  for n in range(N-1):  x[n+1]=rs[ri]\*x[n]\*(1-x[n])  u=np.unique(x[endcap])  r=rs[ri]\*np.ones(len(u))  plt.xlabel('r')  plt.ylabel('Population')  plt.title('Bifurcation Diagram - Logistic Map')  plt.plot(r,u,'.',markersize=1,color=[(np.sin(ri/len(rs)/2)+1)/2,1-ri/len(rs),.5])  plt.show() |

**Annexure 2**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Function to generate the logistic map  def logistic\_map(r, x0, num\_iterations):  population = []  x = x0  for \_ in range(num\_iterations):  x = r \* x \* (1 - x)  population.append(x)  return population  # Parameters  num\_iterations = 2000 # Increase the number of iterations  num\_points = 10000 # Increase the number of data points  r\_values = np.linspace(3.4, 4.0, num\_points) # Expand the range of r values  x0 = 0.5 # Initial population  shift\_value = 0.075 # Adjust the shift value as needed  # Generate the bifurcation diagram  bifurcation\_data = []  for r in r\_values:  population = logistic\_map(r - shift\_value, x0, num\_iterations)  bifurcation\_data.extend([(r, x) for x in population])  # Plot the bifurcation diagram  plt.figure(figsize=(10, 6))  plt.scatter(\*zip(\*bifurcation\_data), s=0.1, marker='.', c='blue')  plt.xlabel('Length of bud')  plt.title('Bifurcation Diagram of the Logistic Map (Increased Chaos Regions)')  plt.show() |

**Annexure 3**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  def lyapunov\_exponent(r, x, n):  sum\_lyapunov = 0.0  for \_ in range(n):  x = r \* x \* (1 - x)  sum\_lyapunov += np.log(np.abs(r - 2 \* r \* x))  return sum\_lyapunov / n  # Define the range of r values  r\_values = np.linspace(3.4, 4.0, 1000)  # Calculate the Lyapunov exponent for each r value  lyapunov\_values = [lyapunov\_exponent(r-0.075, 0.5, 1000) for r in r\_values]  # Define a threshold for chaos detection  chaos\_threshold = 0.00 # You can adjust this threshold as needed  # Find indices where Lyapunov exponent crosses the threshold  chaotic\_indices = np.where(np.array(lyapunov\_values) > chaos\_threshold)[0]  # Find the boundaries of each chaotic region  boundaries = np.where(np.diff(chaotic\_indices) > 1)[0] + 1  # Initialize lists to store chaotic regions and their widths  chaotic\_regions = []  chaotic\_widths = []  # Process each chaotic region separately  for i in range(len(boundaries) + 1):  if i == 0:  start\_index = 0  else:  start\_index = boundaries[i - 1]    if i == len(boundaries):  end\_index = chaotic\_indices[-1] + 1  else:  end\_index = boundaries[i]  chaotic\_region = r\_values[chaotic\_indices[start\_index:end\_index]]  chaotic\_regions.append(chaotic\_region)  # Calculate the width of each chaotic region  chaotic\_width = chaotic\_region.max() - chaotic\_region.min()  chaotic\_widths.append(chaotic\_width)  # Plot the Lyapunov exponent with markers indicating the chaotic regions  plt.plot(r\_values, lyapunov\_values, 'b-', linewidth=0.5)  plt.scatter(np.concatenate(chaotic\_regions), [chaos\_threshold] \* len(chaotic\_indices), color='red', marker='.', s=2, label='Chaotic Region')  plt.xlabel('r')  plt.ylabel('Lyapunov Exponent')  plt.title('Lyapunov Exponent - Logistic Map with Chaotic Regions')  plt.legend()  plt.show() |