A PROJECT REPORT

ON

Logistic Mapping and Chaos

SUBMITTED BY:

Bhavesh Gupta

ENROLLMENT NO: 03217711922

B.TECH 2nd YEAR

SUBJECT: Probability, Statistics and Linear Algebra

ARTIFICAL INTELLIGENCE & DATA SCIENCE (AI&DS) (A)

SEMESTER-3

SCHOOL OF ENGINEERING & TECHNOLOGY

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES – TECHNICAL CAMPUS

PITAM PURA, NEW DELHI, DELHI 110034

UNDERTAKING

I declare that the work presented in this project titled "**Logistic Mapping and Chaos**", submitted to School Of Engineering & Technology, Vivekananda Institute Of Professional Studies – Technical Campus for B.Tech 1st Year Project for Probability, Statistics and Linear Algebra is not plagiarized or submitted the same work for the award of any other examination.

DATED: 26th December, 2023

NAME: BHAVESH GUPTA

ENROLLMENT No: 03217711922

PLACE: SCHOOL OF ENGINEERING & TECHNOLOGY,

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES – TECHNICAL CAMPUS

PITAMPURA, DELHI 110034

INDEX

INTRODUCTION

Chaos theory, a branch of mathematics and physics, explores the unpredictable and complex behavior that can emerge in deterministic systems. The logistic map, a simple mathematical model of population growth, is a classic example that exhibits chaotic dynamics under certain conditions. In this project, we embark on a journey to delve into the intriguing world of chaos by investigating the bifurcation diagram and Lyapunov exponents of the logistic map.

Our primary objective is to visually and quantitatively explore the intricate patterns and chaotic regions that arise in the logistic map as we vary a key parameter, r, known as the control parameter. By systematically examining the bifurcation diagram, we aim to unveil the population dynamics that occur at different r values.

Additionally, it aims to show the sensitivity of the logistic map to initial conditions by calculating and visualizing Lyapunov exponents.

Methodology:

The bifurcation diagram provides a panoramic view of the logistic map's behavior as r changes, offering insights into the transitions from periodicity to chaos. Concurrently, Lyapunov exponents serve as quantitative measures of the system's sensitivity to initial conditions.

By combining these two approaches, we aspire to gain a comprehensive understanding of the logistic map's chaotic nature.

THEORETICAL BACKGROUND

The Logistic Map:

The logistic map is a simple mathematical model that describes population growth over discrete time steps. It is defined by the recurrence relation:

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

Here, x_n represents the population proportion at time step n, and r is the control parameter that influences the dynamics of the system. The logistic map exhibits a wide range of behaviors, from stable periodic oscillations to complex, chaotic patterns, depending on the value of r.

Bifurcation Diagram:

A key tool for visualizing the complexity of the logistic map is the bifurcation diagram. This diagram illustrates how the system's behavior changes as the control parameter r varies. It reveals the emergence of periodic windows, where the system exhibits regular oscillations, as well as chaotic regions characterized by intricate and seemingly random patterns.

Lyapunov Exponents:

Lyapunov exponents are numerical measures of the exponential divergence or convergence of trajectories in a dynamical system. For the logistic map, positive Lyapunov exponents indicate sensitivity to initial conditions and are indicative of chaotic behavior. The Lyapunov exponent quantifies the rate at which nearby trajectories in the system diverge, highlighting the unpredictability inherent in chaotic systems.

System Requirements

Software used for this project is not extensive and basic system requirements for running any python code is sufficient:

- Modern Operating System:
  - Windows 7 or 10
  - Mac OS X 10.11 or higher, 64-bit
  - Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture). ARM CPUs are not supported.
- 4 GB RAM
- 5 GB free disk space

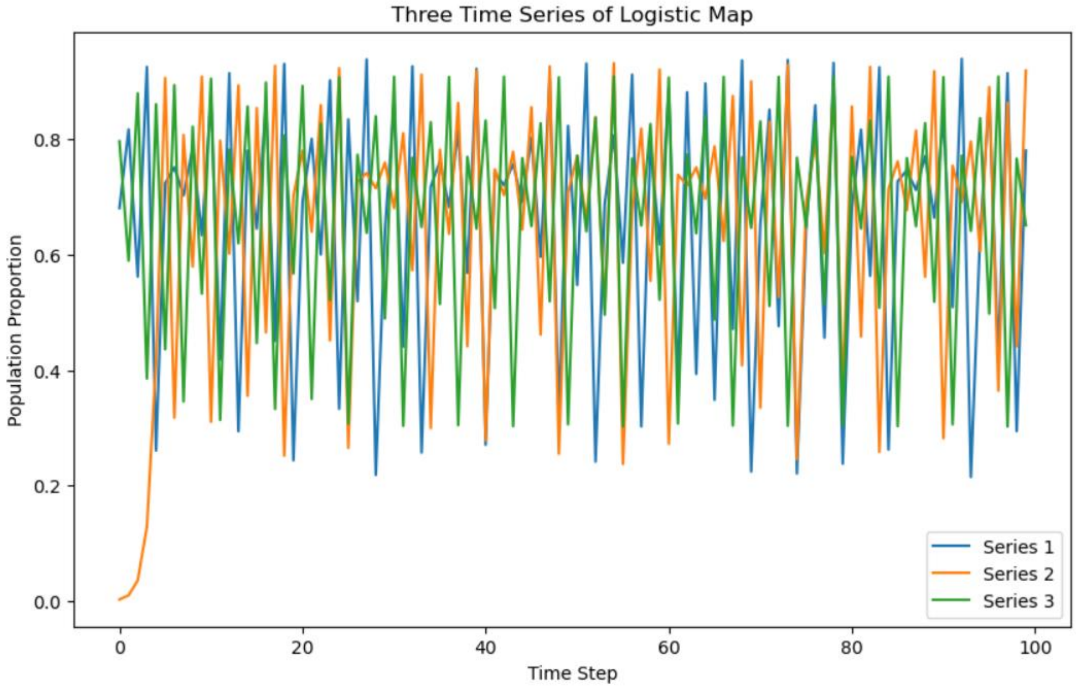
Most users will find that any computer bought in recent years will meet (and usually exceed) these hardware requirements.

DEMONSTRATION OF THE PROJECT:

Demonstration 1: Senitivity of Logistic maps to initial conditions

The following python code randomly selects r values within range of 3.5-4.0 introducing variability in the behavior of each time series. Initial population size is also taken at random between 0 and 1.

```
import matplotlib.pyplot as plt
import numpy as np

def logistic_map(r, initial_condition, num_iterations):
    time_series = [initial_condition]
    for _ in range(num_iterations - 1):
        x_n_plus_1 = r * time_series[-1] * (1 - time_series[-1])
        time_series.append(x_n_plus_1)
    return time_series

num_series = 3
num_points = 100
initial_conditions = np.random.rand(num_series)

# Generate and plot 3 time series
plt.figure(figsize=(10, 6))
for i in range(num_series):
    r_value = np.random.uniform(3.5, 4.0)  # Randomly choose an r value
    time_series = logistic_map(r_value, initial_conditions[i], num_points)
    plt.plot(time_series, label=f'Series {i + 1}')

plt.legend()
plt.xlabel('Time Step')
plt.ylabel('Population Proportion')
plt.title('Three Time Series of Logistic Map')
plt.show()
```

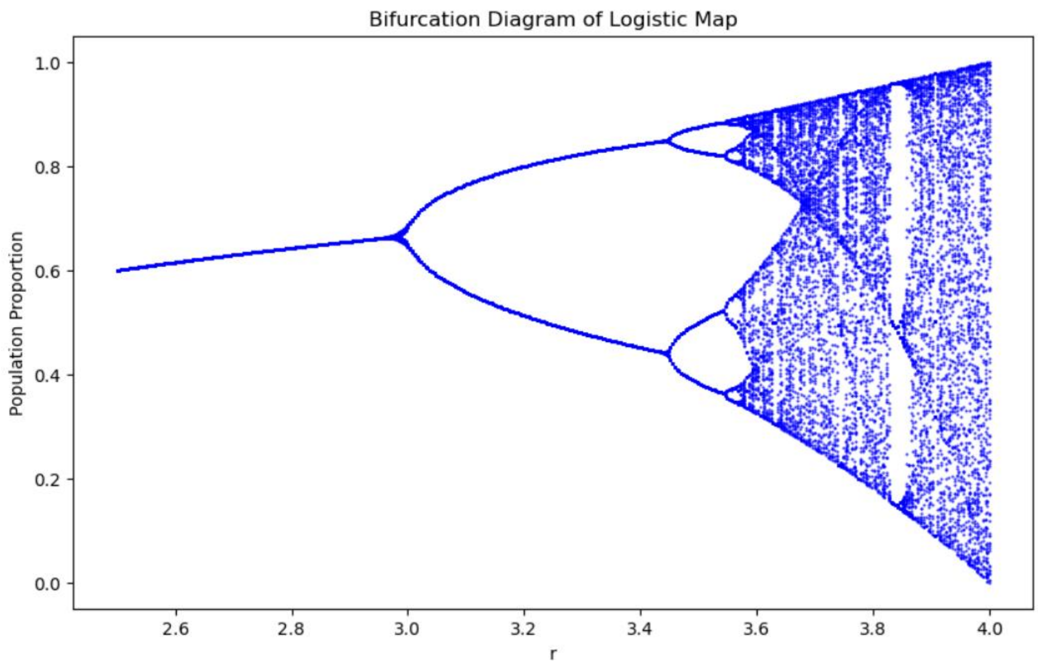


Inference: The time series plot showcases variation of population size over time for r values of range 3.5-4.0.

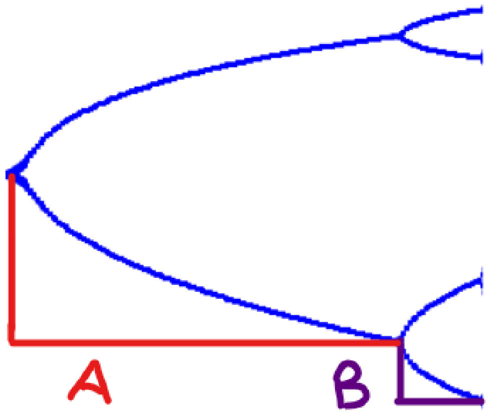
Demonstration 2: Bifurcation diagram and verification of the Feigenbaum Constant.

```
import matplotlib.pyplot as plt

def logistic_map(r, initial_condition, num_iterations):
    time_series = [initial_condition]

    for _ in range(num_iterations - 1):
        x_n = time_series[-1]
        x_n_plus_1 = r * x_n * (1 - x_n)
        time_series.append(x_n_plus_1)

    return time_series

def bifurcation_diagram(min_r, max_r, num_points, num_iterations_per_point):
    r_values = []
    population_values = []

    for point in range(num_points):
        r = min_r + (max_r - min_r) * point / (num_points - 1)
        time_series = logistic_map(r, initial_condition,
num_iterations_per_point)

        # Record the last portion of the time series to avoid transient
behavior
```

```
        r_values.extend([r] * (num_iterations_per_point // 2))
        population_values.extend(time_series[num_iterations_per_point // 2:])

    return r_values, population_values

# Parameters for the bifurcation diagram
min_r = 2.5
max_r = 4.0
num_points = 400
num_iterations_per_point = 200

# Perform the bifurcation diagram simulation
r_values, population_values = bifurcation_diagram(min_r, max_r, num_points,
num_iterations_per_point)

# Visualization
plt.figure(figsize=(10, 6))
plt.title("Bifurcation Diagram of Logistic Map")
plt.scatter(r_values, population_values, s=1, color='blue', marker='.')
plt.xlabel("r")
plt.ylabel("Population Proportion")
plt.show()
```



Inference: The Feigenbaum constant is calculated by calculating the ratio of 2 successive bifurcation lengths and equals to 4.66.

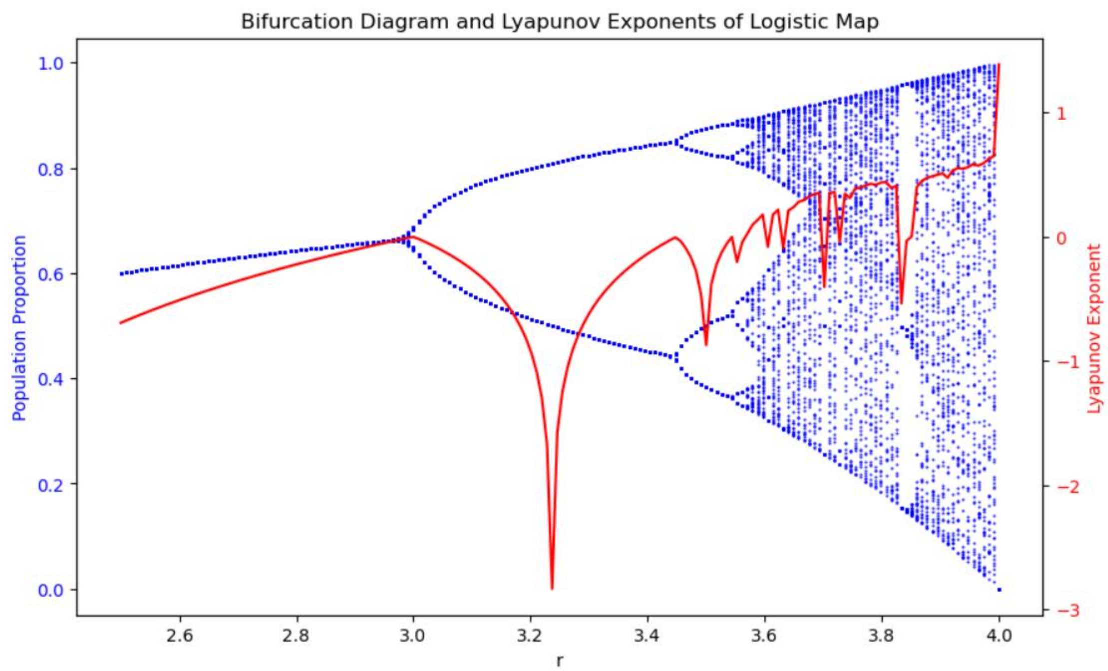


In the case of our generated plot this equals A/B where A is estimated to be 4.35 cm and B was estimated to be 0.93 cm giving the final value as 4.67 ≈4.66

Demonstration 3: Lyapunov Exponent and chaos.

```
import matplotlib.pyplot as plt
import numpy as np

def logistic_map(r, initial_condition, num_iterations):
    time_series = [initial_condition]

    for _ in range(num_iterations - 1):
        x_n = time_series[-1]
        x_n_plus_1 = r * x_n * (1 - x_n)
        time_series.append(x_n_plus_1)

    return time_series

def lyapunov_exponent(r, initial_condition, num_iterations,
num_lyapunov_iterations):
    x = initial_condition
    sum_lyapunov = 0

    for _ in range(num_iterations):
        x = r * x * (1 - x)

    for _ in range(num_lyapunov_iterations):
        derivative = r - 2 * r * x
        sum_lyapunov += np.log(np.abs(derivative))
        x = r * x * (1 - x)

    return sum_lyapunov / num_lyapunov_iterations

# Parameters for the bifurcation diagram
min_r = 2.5
```

```python
max_r = 6.0
num_points = 400
num_iterations_per_point = 200
initial_condition = 0.5
num_lyapunov_iterations = 1000

# Calculate bifurcation diagram
r_values, population_values = [], []

for point in range(num_points):
    r = min_r + (max_r - min_r) * point / (num_points - 1)
    time_series = logistic_map(r, initial_condition,
num_iterations_per_point)
    r_values.extend([r] * (num_iterations_per_point // 2))
    population_values.extend(time_series[num_iterations_per_point // 2:])

# Calculate Lyapunov exponents for each r
lyapunov_exponents = [lyapunov_exponent(r, initial_condition,
num_iterations_per_point, num_lyapunov_iterations) for r in r_values]

# Visualization
fig, ax1 = plt.subplots(figsize=(10, 6))

# Bifurcation diagram on the left y-axis
ax1.set_xlabel("r")
ax1.set_ylabel("Population Proportion", color='blue')
ax1.scatter(r_values, population_values, s=1, color='blue', marker='.')
ax1.tick_params(axis='y', labelcolor='blue')

# Lyapunov exponent on the right y-axis
ax2 = ax1.twinx()
ax2.set_ylabel("Lyapunov Exponent", color='red')
ax2.plot(r_values, lyapunov_exponents, color='red')
ax2.tick_params(axis='y', labelcolor='red')

plt.title("Bifurcation Diagram and Lyapunov Exponents of Logistic Map")
plt.show()
```



Inference: The Lyapunov exponent is a measure of the average rate of divergence or convergence of nearby trajectories in a dynamical system. In the context of the bifurcation diagram for the logistic map, it provides insights into the sensitivity to initial conditions and the degree of chaos in different regions.

Regions of Chaos:

High positive Lyapunov exponents are indicative of chaotic behavior.

These regions correspond to parameter values(r) where the system exhibits sensitive dependence on initial conditions and chaotic trajectories.

Bifurcation Points:

Bifurcations are observed to hold peak exponent values orchestrating higher values of chaos.

CONCLUSIONS

In conclusion, the logistic map, and its bifurcation diagram, coupled with an analysis of the Lyapunov exponent, has provided intriguing insights into the complex and fascinating world of nonlinear dynamics and chaos theory.

The behavior of the logistic map under varying parameters, particularly focusing on the bifurcation diagram to identify regions of stability, bifurcation points, and chaotic dynamics. The graphical representation of the bifurcation diagram revealed captivating patterns, including period-doubling cascades and self-similar structures, showcasing the rich and intricate nature of the logistic map's behavior.

The Lyapunov exponent, a key measure of sensitivity to initial conditions, was employed to quantify the system's divergence or convergence. The analysis of Lyapunov exponent values across different r values highlighted regions of stability and instability, with high positive values indicative of chaotic behavior. This aligns with the theoretical expectations and further emphasizes the chaotic nature of the logistic map in specific parameter regimes.

The logistic map, a simple yet powerful model, demonstrated its ability to exhibit a wide range of behaviors, from stable periodicity to chaotic trajectories. The findings underscore the universality of chaos in nonlinear systems and contribute to our understanding of the intricate dynamics that can emerge from seemingly straightforward mathematical models.

While this study provides valuable insights, it acknowledges certain limitations, such as the numerical approximations employed. Additionally, variations in initial conditions or parameters could lead to different behaviors, warranting further investigation.

In retrospect, this project has been a captivating journey allowing to witness the beauty of complexity arising from seemingly deterministic equations. This project has left me with a heightened appreciation for the intricate dance between stability and chaos, and the profound implications this has for our understanding of the natural world.

REFERENCES:

Matplotlib documentation: https://matplotlib.org/stable/index.html

Logistic map and Bifurcation diagram – Chaos: Making a New Science (book by- James Gleick)

Lyapunov Exponent: https://en.wikipedia.org/wiki/Lyapunov_exponent

Numpy Documentation: https://numpy.org/doc/1.26/

Lyapunov Exponents for the Logistic Map:
https://demonstrations.wolfram.com/LyapunovExponentsForTheLogisticMap/