October 2018

Investigation of chaos in biological systems

Navaneeth Mohan
*The University of Western Ontario*

Supervisor
Dr. Geoff Wild
*The University of Western Ontario*

Graduate Program in Applied Mathematics

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Navaneeth Mohan 2018

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Dynamic Systems Commons, Non-linear Dynamics Commons, and the Numerical Analysis and Computation Commons

Recommended Citation
https://ir.lib.uwo.ca/etd/5727

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca, wlswadmin@uwo.ca.
Abstract
Chaos is the seemingly irregular behavior arising from a deterministic system. Chaos is observed in many real world systems. Edward Lorenz’s seminal discovery of chaotic behavior in a weather model has prompted researchers to develop tools that distinguish chaos from non-chaotic behavior. In the first chapter of this thesis, I survey the tools of detecting chaos namely, Poincaré maps, Lyapunov exponents, surrogate data analysis, recurrence plots and correlation integral plots. In chapter two, I investigate blood pressure fluctuations for chaotic signatures. Though my analysis reveals interesting evidence in support of chaos, the utility such an analysis lies in a different direction that I point to the reader. In chapter three, I investigate a simulation of predator-prey interactions. My analysis casts doubt on some of the claims laid by past researchers, and I prompt future researchers to probe some specific questions that I have outlined in this thesis.
Contents

Abstract I

List of figures IV

Acknowledgement VI

1 Dynamical analysis of time series. 1

1.1 Introduction ................................................. 1

1.2 The Logistic map ........................................... 4

1.2.1 Determinism ............................................. 6

1.2.2 Sensitive dependence on initial condition .................. 8

1.2.3 Detecting Non-Linearity .................................. 12

1.2.4 Boundedness .............................................. 14

1.2.5 Aperiodicity ............................................. 14

1.3 Chaos in experimental time series. .......................... 17

1.3.1 Computing Lyapunov exponent for experimental time series ... 18

2 Chaos in physiological data. 23

2.1 Introduction ................................................ 23

2.2 Deterministic structure in blood pressure recordings .............. 24

2.3 Signatures of aperiodicity .................................... 25

2.4 Sensitivity to initial conditions, and non-linearity ............... 25

2.5 Non-integral dimension ....................................... 27

2.6 Discussion ................................................ 28

3 Chaos in population dynamics. 31

3.1 Introduction ................................................ 31

3.2 The Model .................................................. 32

3.3 Implementing the model ..................................... 35

3.4 Results .................................................. 36

II
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Chaotic fluctuations in Canadian lynx population.</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Chaos and randomness are indistinguishable</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Comparison between Poincar plots of chaotic and random system.</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Bifurcation diagram of Logistic map.</td>
<td>7</td>
</tr>
<tr>
<td>1.5</td>
<td>Following the divergence of nearby trajectories to compute the Lyapunov exponent.</td>
<td>9</td>
</tr>
<tr>
<td>1.6</td>
<td>Convergence of the Lyapunov Exponent for the Logistic map.</td>
<td>10</td>
</tr>
<tr>
<td>1.7</td>
<td>Generating surrogate data for Logistic map.</td>
<td>13</td>
</tr>
<tr>
<td>1.8</td>
<td>Surrogate data analysis of Logistic map and linear time series.</td>
<td>14</td>
</tr>
<tr>
<td>1.9</td>
<td>Recurrence plot of periodic, chaotic and random time series.</td>
<td>16</td>
</tr>
<tr>
<td>1.10</td>
<td>Correlation integral for periodic, chaotic, and random time series.</td>
<td>17</td>
</tr>
<tr>
<td>1.11</td>
<td>Schematic diagram of algorithm to compute Lyapunov exponent of experimental time series.</td>
<td>19</td>
</tr>
<tr>
<td>1.12</td>
<td>Convergence of the Lyapunov exponent of the Henon map.</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>Power spectrum of blood pressure recordings.</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Poincare map of blood pressure time series.</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Recurrence plot of blood pressure recordings.</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Surrogate data analysis of blood pressure recordings.</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Correlation integral of blood pressure recordings.</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Snapshot of the simulated predator-prey dynamics.</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Predator population dynamics for homogeneous and diffusive predator movement.</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Predator population dynamics for alternate cases of diffusive predator movement.</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>Predator population dynamics for alternate cases of homogeneous predator movement.</td>
<td>37</td>
</tr>
<tr>
<td>3.5</td>
<td>Final state of the simulated lattice for homogeneous and diffusive predator movement.</td>
<td>38</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>3.6 Spatial scaling relationship for homogeneous and diffusive predator movement</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>3.7 Poincare map of predator population for homogeneous and diffusive predator movement</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>3.8 Surrogate data analysis of predator population dynamics for homogeneous and diffusive predator movement</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>3.9 Recurrence plots of predator population dynamics for homogeneous predator movement</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>3.10 Correlation integral plots for homogeneous predator movement</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>3.11 Recurrence plots in the case of diffusive predator movement at small scales of observation</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>3.12 Recurrence plots in the case of diffusive predator movement at large scales of observation</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>3.13 Correlation integral plot for diffusive predator movement</td>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>
Acknowledgement

Dr. Geoff Wild, for his patience and guidance.
AJ, for ensuring my sanity.
Leon Glass and Michael Guevara, for discussing at length why I was wrong.
Jack Johnson, for lighting up the long days.
Chapter 1

Dynamical analysis of time series.

1.1 Introduction

This thesis will focus on Chaos theory and its relevance to certain biological systems. Chaos theory is a branch of mathematics which deals with the study of irregular behavior that appears to be random, but is not. To characterize a system as chaotic implies the system exhibits irregular behavior though it follows a set of deterministic rules. Since it’s discovery in the 1800s, chaotic behavior has been identified and studied in a wide range of phenomena such as populations dynamics, epidemiology, astrophysics, weather and stock markets[52, 71, 31, 50].

![Figure 1.1: Lynx fur returns of the Northern Department, Hudson’s Bay Company, 1821-1934. These cyclic fluctuations show signs of chaos[11]. Data from [10].](image)

Chaotic systems are necessarily non-linear. Unlike linear systems, the study of non-linear systems involve qualitative description. This is owed to the To investigate the stability of states, we wish to determine the attractor - i.e a set of states the dynamics tends towards - and the dependence of the attractor to the parameter values.
Apart from population dynamics and epidemiology, chaos has been identified in other biological systems. The identification of chaos in the physiological activity of the human heart has been debated [19]. Chaos is certainly relevant in describing the electrical activity of human heart [49] leading to the suggestion that chaotic models are relevant in modeling cardiac arrhythmias.

Due to the appearance of chaos in natural systems, many different methods to identify and predict chaos have been developed. This is challenging, because at first glance, a chaotic system may not appear any different from a noisy system as shown in figure 1.2.

![Figure 1.2: A chaotic system (red) and a random system (blue) are seemingly indistinguishable at first glance. The chaotic system was generated from the Logistic map.](image)

The red time series was generated from the Logistic map (a chaotic system)

\[ x_{t+1} = r x_t (1 - x_t) \]  

while the green time series is a random string of numbers in the range [0, 1].

To distinguish between a chaotic and noisy system, we can look for four characteristic features. They are:

1. **Deterministic, non-linear, and low-dimensional**: The dynamics of the system follows a deterministic rule, i.e., at each instant of time there is unambiguously only one state that the system can take. This is unlike a stochastic system where the system’s state at each instant of time is specified by a distribution. Further, this deterministic system must be low-dimensional, and non-linear. The additional condition is to exclude infinite-dimensional linear systems that are capable of exhibiting chaotic fluctuations.
2. **Aperiodic long-termed behavior**: The state of the system in the phase space never settles down into a stable configuration. The solution to the nonlinear equation may oscillate, but not in a periodic manner.

3. **Sensitive dependence on initial conditions**: This is an essential feature of chaotic systems, poetically expressed by Lorenz as the *butterfly effect*. It means that small changes to the initial conditions of the system are exponentially amplified and deviate the system from its expected trajectory. Thus it is impossible to predict the dynamics over long times.

4. **Boundedness**: The solutions to the system stay within a finite range and do not approach ±∞.

---

### Chaos in turbulence

Chaos is easily characterized in deterministic models that are stationary, autonomous, noise-free, and low-dimensional. However, natural systems of interest are neither noise-free, nor stationary. This poses a problem in distinguishing chaotic from stochastic dynamics. Of particular interest is turbulence. Turbulence is defined as the irregular motion of particles in a fluid. Unlike laminar flow, the different vertical layers of the fluid undergo mixing. An essential feature of turbulent flow is that particle velocity varies significantly and irregularly in both space and time. Common examples of turbulent flow are atmosphere and ocean currents, and the flow in boat wakes and around aircraft-wing tips.

Turbulence can be modeled as a solution of the Navier-Stokes equation for the fluid velocity $u_i$

$$\frac{\partial u_i}{\partial t} = -\frac{\partial (u_i u_k)}{\partial x_k} - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial (x_k)^2} + F_i$$

and the Poisson equation for the pressure $p$

$$\frac{1}{\rho} \frac{\partial^2 p}{\partial (x_i)^2} = -\frac{\partial^2 (u_i u_k)}{\partial x_i \partial x_k}$$

Here $\rho$ is the density, $\nu$ is the kinematic viscosity, and $F_i$ is the time-independent external force. The solution is turbulent at high velocities, and these solutions are sensitive to their initial conditions. In conjunction with its low-dimensional $(d = 4)$, non-linear, aperiodic dynamics, this is evidence for chaos. However, chaos is difficult to confirm in natural turbulent systems due to the presence of noise. Natural systems are not free from stochasticity. While this may mask the inherent chaotic features in a chaotic natural system, it does not weaken the argument for chaos in natural systems because stochasticity and chaos can be distinguished.
Chaotic systems are predictable over short time intervals however, their long-term evolution is unpredictable due to our limited measurement accuracy combined with the system’s sensitivity to initial conditions. Stochastic systems on the other hand are inherently unpredictable.

Following these characteristics, chaos has been defined as aperiodic bounded behavior in a deterministic system that exhibits sensitive dependence on initial conditions. Most methods of detecting chaos in a system involve basic non-linear time series analysis. In the following section we will illustrate these four characteristics in a time series generated by the Logistic map.

1.2 The Logistic map

The Logistic map in equation 1.1 is a simple non-linear recurrence map. The time series generated from the Logistic map has been used to model population growth, with \( x_t \) as the population at time \( t \) and \( r \) as the population growth rate. The Logistic map is also of interest in this thesis because it exhibits chaotic behavior for certain range of \( r \) values. Here we shall demonstrate some of the methods and arguments to illustrate the four chaotic features of the Logistic map.

\[ \omega - \text{limit sets and the Poincaré-Bendixon theorem.} \]

We shall briefly examine the continuous version of the Logistic map

\[ \frac{dx}{dt} = r x(1 - x), x(0) = x_0 \]

This equation has the solution

\[ x(t, x_0) = \frac{x_0}{x_0 + (1 - x_0)e^{-rt}} \]

Unlike its corresponding difference equation, the Logistic differential equation does not exhibit any chaotic behavior. In the long-time limit, the solutions tend to \( \lim_{t \to \infty} x(t, x_0) = 1 \). Or, in the language of sets, we say that the \( \omega \)-limit set of
the differential equation is a singleton set of 1
\[ \omega(x_0) = 1 \]

Properties of \( \omega \)-limit sets

- **Existence**: The \( \omega \)-limit set of a bounded orbit is non-empty.
- **Closure**: An \( \omega \)-limit set is closed.
- **Invariance**: if \( x_1 \in \omega(x_0) \), then \( x(t, x_1) \in \omega(x_0) \).
- **Connectedness**: The \( \omega \)-limit set of a bounded orbit is connected.
- **Transitivity**: If \( z \in \omega(x_0) \) and \( y \in \omega(x_1) \), then \( z \in \omega(x_1) \).

The Poincaré-Bendixon theorem classifies the possible limit sets of planar systems. It states that for any system
\[
\frac{dx(t)}{dt} = f(x), \quad x(0) = x_0
\]
If \( x \in U \subseteq \mathbb{R}^2 \), and \( x(t) \leq \infty \forall t \), then
- \( \omega(x_0) \) is an equilibrium, or
- \( \omega(x_0) \) is a closed orbit, or
- If \( x \in \omega(x_0) \) then \( \omega(x) \) is an equilibrium.

According to the Poincaré-Bendixon theorem, chaotic trajectories are not possible on 2 dimensional planar systems. Only 3+ dimensional systems or a 2 dimensional system on a non-planar geometry can accommodate chaotic trajectories. As an example consider the phase-portrait for the 2-d system.

\[
\frac{dU}{dt} = V \\
\frac{dV}{dt} = -U(1-U) + 2V
\]

The nullcline \( 2V = U(1-U) \) passes through stable node \((0,0)\), and saddle point \((1,0)\).

Since the solution trajectories to this system do not intersect, on a 2-d planar system, the solutions can tend to either (i) a stable-node or, (ii) a stable-limit cycle or, (iii) they can approach infinities. Either of these outcomes would deem the trajectory non-chaotic.
1.2. THE LOGISTIC MAP

1.2.1 Determinism

Consider the time series generated from the Logistic map shown in figure 1.2. The dynamics generated from the Logistic map can resemble a string of random numbers. To uncover any of the deterministic patterns hidden behind this generated time series we must visualize the phase space of the system by embedding the time series in a higher dimension. In figure 1.3 both time series have been embedded into a 3 dimensional state space by plotting \( x_t \) versus \( x_{t+1} \) versus \( x_{t+2} \). The noise data is distributed throughout the space, which is characteristic of an infinite dimensional system. On the other hand, the Logistic data exhibits a strange attractor that twists and curls in 3 dimensional space. The folding and stretching pattern is due to the chaotic system’s sensitive dependence on intial conditions and it’s boundedness. Such plots are called Poincare plots. Plotting the Poincare plot is one of many ways to possibly uncover any deterministic structure hidden in the time series.

![Figure 1.3: A chaotic system (red) and a random system (blue) are clearly distinguishable when embedded in a higher dimension.](image)

The time series in figure 1.3 and 1.2 was generated for a single value of \( r = 4 \), a regime where the Logistic map exhibits chaotic behavior. To identify chaos in other ranges of \( r \) we plot a bifurcation diagram. For \( 0 \leq r \leq 1 \), the value of \( x_t \) approaches 0, independent of initial conditions. For \( 1 \leq r \leq 3 \) \( x_t \) approaches \( r/(1-r) \) independent of initial conditions. Beyond \( r = 3 \), the value of \( x_t \) oscillates between multiple values. In general the period cycle of oscillation increases with \( r \).
1.2. THE LOGISTIC MAP

The bifurcation diagram in figure 1.4 was generated numerically by iterating over a range of \( r \) values and counting the different \( x_t \) values. At certain values of \( r \) the diagram shows a forking in the values that \( x_t \) could take. This depicts that the system oscillates between twice as many points than before. As the values of \( r \) increase, the bifurcations increase until the system is oscillating between infinitely many points. This is known as period-doubling path to chaos, and is a characteristic feature of chaotic maps. The time-series seen above was generated for \( r = 4 \) when the system is oscillating with a period of infinite length, thus appearing to be random.

<table>
<thead>
<tr>
<th>Period doubling path to chaos in the Logistic map</th>
</tr>
</thead>
<tbody>
<tr>
<td>The bifurcation diagram in figure 1.4 was generated numerically and depicts the period-doubling path to chaos. We follow \cite{8} to prove the period-doubling path to chaos.</td>
</tr>
<tr>
<td>The steady states ( x^* ) of the Logistic map for ( x_t \in [0, 1] ) are given by.</td>
</tr>
</tbody>
</table>
| \[
| x_{t+1} = f(x_t) = rx_t(1 - x_t) \\
| x^* = 0 \text{ and } x^* = 1 - 1/r \text{ for } r > 1 |
| The second steady state assumes \( r > 1 \) |
| The curve of fixed points we are interested in is \( x^* = 1 - 1/r \). Let us redefine the dynamical variable as \( \zeta_t = x_t - x^* \) so that our curve of fixed points occur at \( \zeta^* = 0 \). |
In the redefined variable, the Logistic map is

\[ \zeta_{t+1} = -\zeta_t(r\zeta_t + r - 2) \]

The conditions for a period-doubling bifurcation at \((0, r^*)\) are

\[ f(0, r^*) = 0, f^2(0, r^*) = 0, f^2_{\zeta}(0, r^*) = 0, f^2_{\zeta, r}(0, r^*) \neq 0, f^2_{\zeta, \zeta, \zeta}(0, r^*) \neq 0. \]

Computing these quantities for the Logistic map we find

\[ f(0, 3) = 0, f_{\zeta}(0, 3) = -1, f^2(0, 3) = 0, f^2_{\zeta}(0, 3) = 0, f^2_{\zeta, r}(0, 3) = -2, f^2_{\zeta, \zeta}(0, 3) = -108. \]

Hence, a period-doubling bifurcation has been verified at \(r^* = 3, x^* = 1 - 1/r^* = 2/3\).

As we increase \(r\), new steady states of order 2 emerge. These steady states are given by

\[ x_{t+2} = x_t \]

\[ \Rightarrow rx_{t+1}(1 - x_{t+1}) = x_t \]

\[ \Rightarrow r^2x_t(1 - x_t)(1 - rx_t + r^2 x_t^2) = x_t \]

Apart from \(x^* = 0\) and \(x^* = 1 - r^{-1}\), two new steady states have emerged

\[ x^*_{\pm} = \frac{1}{2}(1 + r^{-1}) \pm r^{-1}\sqrt{(r - 3)(r + 1)} \text{ for } r > 3 \]

Now considering the second iterate and fourth iterate i.e \(f^2(x_t, r)\) and \(f^4(x_t, r)\) we perform a similar analysis as for the period 2 bifurcation, and verify that a period 4 bifurcation occurs at \(r^* = 3.449, x^* = 0.813\) and \(r^* = 3.449, x^* = 0.476\).

### 1.2.2 Sensitive dependence on initial condition

As Lorenz noticed in his seminal simulation [35], small changes to initial conditions can diverge exponentially until the original trajectory is no longer recognizable. This sensitivity to initial conditions is quantified by computing the **Lyapunov Exponent**.

The Lyapunov Exponent, \(\lambda\) is a measure of the long-time rate of divergences between two nearby trajectories. Let \(Z_t\) and \(Z'_t\) be two trajectories initiated close to each other.

\[ |\delta Z_t| = |Z_t - Z'_t| \]

\[ \lim_{t \to 0} |\delta Z_t| = 0 \]

\[ \lim_{t \to \infty} |\delta Z_t| = e^\lambda |\delta Z_0| \]

Hence, the Lyapunov Exponent can be defined as follows

\[ \lambda = \lim_{t \to \infty} \lim_{\delta Z_0 \to 0} \frac{1}{t} \log \frac{|\delta Z_t|}{|\delta Z_0|} \]

(1.2)
On a chaotic trajectory the two nearby points diverge exponentially which translates into a positive Lyapunov exponent.

For a chaotic map such as the logistic map in equation (1.1) we describe a procedure to compute the largest Lyapunov exponent. The essence of the procedure is to compute the derivative along the direction of maximum expansion and average its logarithm over the trajectory. The method is as described in [59].

1. Choose two nearby initial conditions \( R_0 \) and \( R_0 + \Delta R_0 \). \( \Delta R_0 \) is chosen much smaller than the scale at which the flow occurs. In double precision we consider \( |\Delta R_0| = 10^{-10} \).

2. Iterate both initial conditions by one time-step.

\[
\begin{align*}
R_0 &\rightarrow R_1 \\
R_0 + \Delta R_0 &\rightarrow R_1 + \Delta R_1
\end{align*}
\]

3. Before iterating the trajectories further, rescale the second trajectory to \( R_1 + (\Delta R_1 / \Delta R_0) \Delta R_0 \). This ensures that the trajectories remain close while letting the directions orient to that of maximum expansion.

4. Add the quantity \( \lambda_n = \log(\Delta R_n / \Delta R_0) \) to a running average.

5. Repeat steps 2-4 until the running average converges.

Figure 1.5: Numerical method for computing the largest Lyapunov exponent. Two nearby trajectories initially \( \Delta R_0 \) apart, diverge by \( \Delta R_t \) at the \( t^{th} \) iteration. At each iteration we rescale their divergence to \( \Delta R_0 \) and add keep tab of the sequence of divergence \( \Delta R_t \).

The figure was recreated from [59].

Applying this method to the Logistic map (eq. (1.1)) with \( r = 4, x_0 = 0.1, \Delta x_0 = 10^{-10} \), we observe the convergence to the Lyapunov exponent of \( \lambda = 0.6934 \) as shown in figure 1.6.
1.2. THE LOGISTIC MAP

This result is consistent with the value of $\lambda = 0.6934$ as mentioned in [15]. The positive value of Lyapunov Exponent for the Logistic map confirms the exponential divergence of initially close points.

![Figure 1.6: Convergence of the Lyapunov Exponent for the Logistic map to $\lambda = 0.6934$ using the numerical procedure of section 1.2.2.](image)

An improved algorithm for computing the Lyapunov Exponent

The algorithm described in the above section to compute the Lyapunov Exponent of an iterative map has a major shortcoming [12] for recurrence maps of $d > 1$ dimension. Though the algorithm accounts for the divergence of the initial conditions by rescaling the $\Delta R_n$ to $\Delta R_0$ at each time step, there is an additional singularity - in a chaotic system, each vector $R_n$ tends to fall along the local direction of most rapid growth. Thus the space spanned by the $R_n$ vectors become indistinguishable. Therefore, apart from rescaling the $\Delta R_n$, we should also realign them every few time steps. This can be achieved by performing a Gram-Schmidt re-orthonormalization procedure on the vector frame.

To compute the Lyapunov exponent for $d > 1$ dimensional recursive maps, a more elaborate algorithm described in [72] (developed independently by [5] and by [55]) is presented here. Here, [72] define the Lyapunov exponent according to the deformation of initial conditions from an infinitesimal n-sphere into an n-ellipsoid during the long-term evolution along the trajectory. The $i^{th}$ one-dimensional Lyapunov
1.2. THE LOGISTIC MAP

The Lyapunov exponent is defined in terms of the lengths of the ellipsoidal principal axis $p_i(t)$

$$\lambda_i = \lim_{t \to \infty} \log \frac{p_i(t)}{p_i(0)}$$

Hence, to calculate the Lyapunov exponent, we measure the long-term deformation of the n-ellipsoid. The linear extent grows as $2^{\lambda_1 t}$, the area defined by the first two principal axes grows as $2^{\lambda_1 t + \lambda_2 t}$ and the volume defined by the first three principal axes grows as $2^{\lambda_1 t + \lambda_2 t + \lambda_3 t}$ and so on.

To implement the procedure, we use a phase space plus tangent space approach. Consider a principal axes of orthonormal states anchored to the trajectory. Thus the origin of the principal axes evolve according to the nonlinear recursive map. The principal axes themselves evolve according to the linearized recursive map. This construction of principal axes can be considered as a linear perturbation of the nonlinear trajectory, thus enabling us to explore the nearby states of the trajectory.

Starting from a post-transient initial condition on the $n$-dimensional recursive map, we define a set of principal axes $\{v_1, v_2, ..., v_n\}$ Each $n$-dimensional vector $v_i$ evolves according to the linearized equations. After one iteration of the linearized equations, the $n$ vectors tend to align along the direction of the trajectory.

This is fixed by using GSR, which produces the following orthonormal axes $\{v'_1, v'_2, ..., v'_n\}$

$$v'_1 = \frac{v_1}{||v'_1||}$$

$$v'_2 = \frac{v_2 - \langle v_2, v'_1 \rangle v'_1}{||v_2 - \langle v_2, v'_1 \rangle v'_1||}$$

$$v'_n = \frac{v_n - \langle v_n, v'_{n-1} \rangle v'_{n-1} - ... - \langle v_n, v'_1 \rangle v'_1}{||v_n - \langle v_n, v'_{n-1} \rangle v'_{n-1} - ... - \langle v_n, v'_1 \rangle v'_1||}$$

To find the first Lyapunov Exponent, we observe the growth of $v_1$ which grows as $2^{\lambda_1 t}$.

Now we demonstrate the above procedure for the following 2-dimensional map.

$$x_{t+1} = 1 - ax_t^2 + y_t$$

$$y_{t+1} = bx_t$$

(||)
This map is known as the Henon map \[24\]. For the classical values of \(a = 1.4, b = 0.3\) the Henon map is chaotic. We observe in the adjacent figure the convergence to the Lyapunov Exponent \(\lambda = 0.4183\).

1.2.3 Detecting Non-Linearity

When looking for deterministic signatures in our time-series, we should consider the nature of determinism. Chaotic system of finite dimension are necessarily non-linear. Linear systems can exhibit chaos only in infinite dimensions. The presence of linearity would indicate that the system is not indeed low-dimensional and/or chaotic. So to verify whether the dynamics is linear or not, we begin our investigation by assuming that the dynamics is linear, and then proceed to refute it. This is known as null hypothesis rejection \[8\]. The hypothesis we are trying to refute is the null hypothesis of linear dynamics; the given time series was produced by a linear dynamical system.

To test whether the time series is inconsistent with the null hypothesis, we generate an ensemble of surrogate data that is consistent with the null hypothesis. In this case, that would surmount to generating a time-series from a linear stochastic process. For this purpose, the power spectrum would prove to be useful. If the surrogate data and the original time series share an identical power spectrum, they will also share linear properties. Thus, any measures of linearities would fail to distinguish between the them. Only a non-linear measure, such as Lyapunov exponent, can distinguish between the original time series and the surrogate data. To generate the surrogate data, we perform the following steps \[66\] .

1. Compute the amplitudes \(A(\omega)\) and phases \(\phi(\omega)\) of the Fourier transform for the original time series.

2. For each \(\omega\) replace the phases \(\phi(\omega)\) with random numbers between 0 and \(2\pi\). This generates a new set of phases \(\phi_{\text{rand}}(\omega)\)

3. Compute the inverse Fourier transform for \(A(\omega)\) and \(\phi_{\text{rand}}(\omega)\). The time series thus generated is the surrogate data.

The generated surrogate data has the same amplitudes \(A(\omega)\) as the original time series,
implying they have the same power spectrum. Statistically, the surrogates will share similar linear properties as the original time series.

By generating an ensemble of surrogate data (figure 1.7), and computing a discriminating statistic for all of them, we can verify how likely is original data to be linear. An example of a discriminating statistic is the Lyapunov Exponent. If the original data’s Lyapunov Exponent is a statistical outlier among the surrogate data’s Lyapunov Exponents, then the null hypothesis has been refuted.

We demonstrate this technique for two time series. First, we generate a time series from the Logistic map which we already know is non-linear. Then, we repeat the technique for a time series generated by a linear stochastic process

$$X_{t+1} = 4 + 0.94X_t + \nu_t$$  

(1.3)

where $$\nu_t$$ is Gaussian noise. By comparing the Lyapunov exponents of the original time series and its surrogates in histogram 1.8, we observe that the the Logistic map’s Lyapunov exponent is 0.693, while the surrogate’s Lyapunov exponent lies in the range of $$1.75 \leq \lambda \leq 2.0$$. This clearly refutes our null hypothesis, thus confirming the non-linearity of the
1.2. THE LOGISTIC MAP

Figure 1.8: 150 surrogate data were generated for the Logistic map (left). The surrogate data (black) have a Lyapunov exponent in the range of [1.8,1.9] while the Logistic map has a Lyapunov Exponent (red) of 0.693. None of the surrogate data have a Lyapunov exponent in the neighborhood of the original data. The large difference between the ensemble’s Lyapunov exponent and the original Lyapunov exponent refutes the null hypothesis and thus confirms the non-linearity of the Logistic map. 50 surrogate data were generated (right) for time series generated from model equation 1.3. The model time series (red) has a Lyapunov exponent of 1.32, and all 50 surrogates (black) have a Lyapunov exponent in the range of [1.28, 1.34] thus confirming that the original time series was generated by a linear process.

Logistic map. Meanwhile, the linear time series of equation 1.3 has a Lyapunov exponent of 1.31, which lies within the range of its surrogates $1.28 \leq \lambda \leq 1.34$.

1.2.4 Boundedness

Even when the Logistic map oscillates between infinitely many values, it always oscillates within the range $[0, 1]$ for any initial condition $x_0 \in [0, 1]$ and $r \in [0, 4]$.

1.2.5 Aperiodicity

Unlike a periodic system, a chaotic system’s oscillations are aperiodic. To verify this, we construct the recurrence plots of the Logistic map for different $r$ values. A recurrence plot measures “How close” and “How often” a time series approaches a previous value. The procedure for constructing a recurrence plot is:

1. Embed the time series as a time-delayed vector

$$X_t = \{ x_t, x_{t-h}, x_{t-2h}, \ldots x_{t-(p-1)h} \}$$

where $p$ is the embedding dimension, and $h$ is the embedding delay.
2. For any two embedded vectors $D_i$ and $D_j$, compute the distance between them

$$
\delta_{i,j} = |D_i - D_j|
$$

3. Pick a distance $d$ and find all pairs of points $(i, j)$ such that $\delta_{i,j} \leq d$. Construct a plot with $i$ on the horizontal axis and $j$ on the vertical axis. If $\delta_{i,j} \leq d$ then place a point on the coordinate $(i,j)$.

In figure 1.9, the recurrence plot for a periodic, a chaotic and a noisy system is shown. For a periodic time series of period $T$, $\delta_{i,j} = 0$ for every $|i - j| = nT$. This results in a series of diagonal stripes. The recurrence plot of a chaotic system will exhibit a more complicated pattern but the chaotic system does exhibit some signs of periodicity. For white noise, the recurrence plot shows no structure.

The number of dots on the recurrence plot tells how many times the trajectory came within distance $d$ of a previous value. What is important is not the dot counts, but how the dot counts change with $d$. The correlation integral $C(d)$ is defined to be the fraction of pairs $(i,j)$ where $D_i$ and $D_j$ are closer than $d$ for $i \neq j$

$$
C(d) = \lim_{N \to \infty} \frac{1}{N^2} \sum_{i,j=1; i \neq j}^{N} \Theta(d - |D_i - D_j|) \tag{1.4}
$$

where $\Theta(x)$ is the Heaviside function. This definition of correlation integral was first suggested by [23].

Naturally, as $d$ is increased, more dots appear in the recurrence plot and $C(d)$ increases. The pattern of dots that constitute the correlation integral is unique to periodic, chaotic, and noisy time series. Thus plotting the correlation integral is a method of identifying chaotic time series.

An important result obtained from the correlation integral is the correlation dimension. The correlation dimension is closely related to the fractal dimension of a time series, and can identify the presence of an attractor.

The correlation dimension is most easily explained for a random time series. Consider a random time series embedded in 3 dimensions and scattered in a 3 dimensional volume such as in figure 1.3. Choose a reference point and count the number of points that fall within a 3 sphere of radius $d$. As $d$ increases, the number of points within this sphere increases in a power law that is proportional to $d^3$. For a random time series embedded in $p$ dimension, the $C(D)$ increases as $d^p$.

$$
C(d) = Ad^p
$$

$$
\log C(d) = \log A + p \log(d)
$$

Hence, for a random time series, the correlation dimension $\nu = p$. For a chaotic time series that lies on an attractor of dimension $\nu$, embedded in dimension $p$, the embedding
1.2. THE LOGISTIC MAP

Figure 1.9: Recurrence plot of a periodic (left), chaotic (center), and random (right) time series. The periodic time series was generated by the Logistic map for parameter $r = 3.5$. The diagonal stripes are separated by 2 implying a periodicity of 2. The plot was constructed for $p = 2, d = 0.01$. The chaotic time series was generated by the Logistic map for parameter $r = 4$. Plot was constructed for $p = 2, d = 0.2$. The recurrence plot of the random time series was constructed for $p = 2, d = 0.2$. The periodicity of the random time series is not apparent due to the randomness of the data.
produces a structure that is not uniformly scattered in the phase space. As long as $p > \nu$, the embedded time series will have the same topological properties as the original attractor. By constructing the correlation integral for this embedded time series and plotting it’s log-log curve, we can find the attractor’s correlation dimension.

Figure 1.10: Log-Log plot of Correlation integral for periodic, chaotic and random time series. The Logistic map generates a periodic time series for $r = 3.45$ and a chaotic time series for $r = 3.5969$. The two series have strikingly different Correlation integral plots. The random time series has a steeper slope than chaotic time series. Slope of chaotic time series is $\nu = 0.496$, which agrees with the result from [23]. Slope of random time series is $\nu = 1.94$. All plots were constructed for $p = 2$.

### 1.3 Chaos in experimental time series.

In the previous section we reviewed the methods of identifying chaos in a time series generated by a deterministic non-linear recursive map. However, these methods have limitations because time series derived from experiments are not generated by a recursive map. It is also futile to attempt fitting a recursive map onto weather and stock market data. The methods from our previous section would not suffice to identify chaos in an experimental time series.

However, on that note, we should recognize that not all was futile. Among the four characteristics of chaotic time series - determinism, aperiodicity, sensitivity to initial con-
1.3. CHAOS IN EXPERIMENTAL TIME SERIES.

ditions and boundedness - our earlier methods to identify determinism and aperiodicity are still valid. The methods involved embedding the time series onto a higher dimension, and computing the correlation dimension. Given a large time series, these methods work just as well for experimental time series as they did for generated time series.

It is the other two characteristics - sensitivity to initial conditions and boundedness - that require special attention. The first one involves computing the lyapunov exponent for the time series. As for boundedness, it is not possible to establish a bound for all times outside the range of experimental data. In the following section we shall outline a method of computing the Lyapunov exponent for an experimental time series.

1.3.1 Computing Lyapunov exponent for experimental time series

To compute the Lyapunov exponent from an experimental time series, we follow the algorithm developed by Wolf et.al. \[72\]. The Lyapunov exponent is defined according to the deformation of the initial conditions from an infinitesimal n-sphere into an n-ellipsoid during the long-term evolution along the trajectory.

Instead of working with the original time series, we reconstruct the phase space by embedding the time series as a time-delayed vector

\[ \mathbf{X}_t = \{x_t, x_{t-h}, x_{t-2h}, \ldots x_{t-(p-1)h}\} \]

Where \( p \) is the embedding dimension, and \( h \) is the embedding delay. According to Taken’s theorem \[63\] the time series from the original phase space \( \{x_t\} \) and the reconstructed phase space \( \{\mathbf{X}_t\} \) lie on the same attractor.

Assuming the existence of at least one positive exponent, we compute the largest lyapunov exponent by monitoring the divergence between a single pair of nearby orbits. The procedure as described in \[72\] is:

1. Choose the initial point \( \mathbf{X}_{t_0} \). Say it’s nearest neighbor is the \( t_i \)th instant, \( \mathbf{X}_{t_i} \). The Euclidean distance between the two points is \( L_{t_0} = |\mathbf{X}_{t_0} - \mathbf{X}_{t_i}| \). The temporal separation \( |t_0 - t_i| \) should be at least one mean orbital period because points much closer than one orbital period are characterized by a zero lyapunov exponent. \( \mathbf{X}_{t_0} \) is called the fiducial point and \( \mathbf{X}_{t_i} \) is called the non-fiducial point.

2. A suitable time-step of evolution \( \Delta t \) is chosen and the two points are evolved.

\[ \mathbf{X}_{t_0} \rightarrow \mathbf{X}_{t_0+\Delta t} \]
\[ \mathbf{X}_{t_i} \rightarrow \mathbf{X}_{t_i+\Delta t} \]

The time-step \( \Delta t \) is short enough such that the two trajectories don’t intersect as they pass through a fold in the attractor.
3. At a $\Delta t$ time forward the distance between the two points has increased to

$$L'_{t_0+\Delta t} = |X_{t_0+\Delta t} - X_{t_i+\Delta t}|.$$ 

4. After a few iterations, when the separation $L'_{t_0+\Delta t}$ becomes large, the two trajectories might intersect as they pass through a folding region of the attractor. This would cause an underestimation of the Lyapunov exponent. Hence, we replace the non-fiducial point $X_{t_i+\Delta t}$ with a point closer to the fiducial point. Due to the finite amount of data, this replacement procedure is unlikely to preserve orientation. We can best hope for a replacement point that minimizes the orientation change. If an adequate replacement point is not found, we continue the procedure with the points that were being used.

5. The replacement point is a distance $L_{t_1}$ away from the fiducial point. The steps 2-4 are repeated until the end of the fiducial trajectory is reached.

6. At each time-step $t_n$ the quantity $\log \left( \frac{L'_{t_n+\Delta t}}{L_{t_n}} \right)$ is added to a running average and at the end of the trajectory the quantity

$$\lambda_1 = \frac{1}{t_M - t_0} \sum_{n=0}^{M} \log \left( \frac{L'_{t_n+\Delta t}}{L_{t_n}} \right)$$

is estimated. Where $M$ is the total number of replacement steps.

Figure 1.11: The diagram represents the evolution of two nearby points along the fiducial and the non-fiducial trajectory. Every $\Delta t$ steps their divergence is adjusted by replacing the non-fiducial point. Figure recreated from [59].

To test the validity of this routine, we will generate a time series from the following 2 dimensional recursive map

$$x_{t+1} = 1 - ax_t^2 + y_t$$
$$y_{t+1} = b x_t$$

(1.6)

This map is known as the Henon map [24]. For the classical values of $a = 1.4, b = 0.3$, the Henon map is chaotic.
1.3. CHAOS IN EXPERIMENTAL TIME SERIES.

The generated time series will model an experimental data set. In figure 1.12 we see the Lyapunov exponent converging to $\lambda = 0.4215$ consistent with our earlier result in section 1.2.2.

![Figure 1.12: Convergence of the Lyapunov exponent of the Henon map to $\lambda = 0.421$ using the procedure described in section 1.3.1](image)

Distinguishing chaos from multi-periodicity

Multi-periodic systems are a class of systems that are neither recognized as random, chaotic or periodic. They can produce seemingly chaotic time series.

In the figure we observe the time series generated by a sum of four distinct periodic functions.

$$X_t = \Sigma_{k=1}^{2} Cos\left(\frac{a_k}{2\pi} t\right) + Sin\left(\frac{b_k}{2\pi} t\right)$$

The Poincaré return maps show a helical structure. This could be due to the multiple periodicities underlying the system.
1.3. CHAOS IN EXPERIMENTAL TIME SERIES.

The recurrence plots for the multi-periodic system reveals some non-trivial structures that is not periodic. However it is unclear whether this is evidence of chaos.

The correlation integral plots clearly show signatures of the underlying periodic framework of the system, and do not show any signs of chaos.

Finally, the surrogate data analysis reveals that the time-series can be generated by a linear model.

These results show that our tests can distinguish between chaotic and multi-periodic systems.

Remarks

We concluded this chapter with the demonstration of an algorithm to compute the Lyapunov exponent of an experimental time series. In tandem with the tools we developed earlier - Poincare maps, recurrence plots, correlation integral plots, and surrogate data analysis - we are now armed to investigate any time series for chaotic signatures.

It should be kept in mind that none of these tools, by themselves, can confirm the presence of chaos in a time series. This is not due to the limitations of the methods, but rather the limitation of our definition of chaos. There is not one particular characteristic of chaos that can conclude our investigation, but rather a collection of tell-tale signatures that can increase our confidence.

In the following two chapters, we shall use these tools to investigate two different time series of biological interest. In chapter two, we will perform a novel inspection of a time series of blood pressure recordings for any chaotic signatures. In chapter three, we probe
a previously raised question of chaotic signatures in predator-prey interactions.
Chapter 2

Chaos in physiological data.

2.1 Introduction

The detection of chaos and non-linearity in real-life systems has been an active area of research since Lorenz’s seminal work [35]. Chaotic signatures have been identified in biological [58, 70, 56] and physiological systems [51, 54]. Of particular interest within the academic and medical circles is the supposed presence of chaos in cardiac cycles such as heart rate variability (HRV) [34, 69, 44]. Electrically stimulated in-vitro cardiac cells have shown non-linear behavior, including period-doubling bifurcations [49, 11].

Chaotic models have been proposed to reproduce the many dynamical features of cardiac cycles [38, 65], however the very presence of chaos in cardiac cycles is debated [21]. Researchers have performed dynamical analysis of cardiac data, only to produce ambivalent results. Some report that HRV is low-dimensional [4, 7], while others do not [10, 29]. The detection of determinism in HRV [37, 38] is also questionable because stochastic (i.e non-deterministic) models have been proposed to account for the characteristics of HRV [3, 25]. Quantification of HRV is of clinical significance [39] because an absolutely regular HRV is an indicator for the onset of hypertension [54]. For this reason, dynamical analysis of cardiac cycles needs to be performed.

Using the methods we developed in the previous chapter, we will look for chaotic signatures in cardiac cycle recordings. The dynamical analysis of cardiac cycles could detect signatures of cardiac abnormalities. However, we will not use HRV recordings. HRV changes with breathing rate - increasing during exhalation, and decreasing during inhalation - hence HRV is non-stationary. Since, non-stationarity is a non-linear process, this could interfere with our attempt in detecting chaos. For this reason, I believe HRV recordings are not the most reliable data to analyze when looking for chaotic signatures.

A more stable measurement of cardiac cycles is blood pressure recording, because blood pressure changes by only 40% even if cardiac output may increase by as much
2.2. DETERMINISTIC STRUCTURE IN BLOOD PRESSURE RECORDINGS

as five-fold [43]. For this reason, it is more useful to investigate the dynamics in blood pressure fluctuations.

The specific mammal in examination is the Dahl salt-sensitive (SS) rat. The Dahl SS rat suffers from salt-sensitive hypertension due to dysfunction of its baroreflex control system - the mechanism that helps maintain blood pressure. The data was collected to develop a mathematical model and estimate physiological parametric differences between a low and high salt diet in rats. The authors published their data on a free online database of recorded physiological signals. [22].

Figure 2.1: The inset shows the time series of blood pressure recordings over an interval of 120 seconds. Sampling rate is 10 ms. The power spectrum analysis of the time series reveals a dominant low frequency term of $\sim 6$ hz.

The time series data was collected for 120 seconds, sampled every 10 ms. As seen in the inset in figure 2.1, the time series exhibits non-stationarity for the first $\sim 50$ seconds of data recording. For this reason, I exclude the first half of the data recordings in all investigations.

2.2 Deterministic structure in blood pressure recordings

The power spectrum of the time series reveals a dominant low frequency oscillation in the time series (figure 2.1). This could be attributed to the periodic oscillations in the blood pressure. However, the Poincare map (figure 2.2) of the time series reveals a deterministic structure unlike that of a periodic time series.

To further probe this time series, we shall construct the recurrence plot and estimate the correlation integral.
2.3 Signatures of aperiodicity

The recurrence plots have similar structure at all sampled time intervals. There are 12 diagonal lines in the 2 second interval, implying a periodicity of $\sim 6\text{Hz}$, in agreement with our earlier power spectrum analysis. The periodicity in blood pressure, which is easily observed when recording the pulse, is driven by the regularity heart beat. Whether this periodicity is affected by the salt diet has not been investigated. The discontinuity of the diagonal lines indicate some 'gaps' in the periodicity. These 'gaps' are not due to noise. Though they show some semblance to a periodic oscillator with two harmonic frequencies, that is not the case in heart beat’s periodicity. Thus, it points to an aperiodic dynamical feature, perhaps non-stationarity.

2.4 Sensitivity to initial conditions, and non-linearity

To detect any non-linearities in the system, we generated surrogate data using the procedure described in section 1.2.3. The Lyapunov exponent was computed for the surrogate data and the original data.
2.4. SENSITIVITY TO INITIAL CONDITIONS, AND NON-LINEARITY

Figure 2.3: The recurrence plots for 2 second intervals sampled from different instances across the time series. All plots were constructed for an embedding dimension of $p = 1$ and embedding delay of $d = 2ms$

Figure 2.4: Lyapunov exponent for the surrogate data (black) and the original data (red). $\lambda_{data} = 0.285$. 70% of the surrogates have a Lyapunov exponent in the range $\in [0.4, 0.5]$. The data overlaps with only 15% of the surrogates

As seen in the histogram in figure 2.4, the original data has a positive Lyapunov exponent of $\lambda_{data} = 0.285$ while the surrogate data has a mean Lyapunov exponent of $\langle \lambda_{Surr} \rangle \approx 0.401$. This suggests a low likelihood that the data was generated from a linear model. 70% of the surrogate data have Lyapunov exponents in the range of $0.4 \leq \lambda_{Surr} \leq 0.5$, while $\sim 85\%$ of the surrogates have a Lyapunov exponent larger than $\lambda_{data}$. Thus, with $\sim 85\%$ confidence, we can say that the original data’s dynamics has a non-linear origin to it.
2.5. NON-INTEGRAL DIMENSION

Refuting a periodic oscillator model of blood pressure fluctuations.

The presence of a dominant frequency of $6h_\text{z}$ prompts the investigation of modeling the blood pressure fluctuations as a periodic oscillator. Here we consider a $6h_\text{z}$ oscillator mixed with Gaussian noise to model the measurement error. The model time-series was created to have a similar mean and variance as the original blood pressure time-series. However, on investigating further we see that this model fails on at least two counts.

The Poincaré plot of this model time-series is a thick band as expected from a noisy periodic oscillator. It does not exhibit any non-trivial structure as seen in fig:2.2.

Surrogate data analysis reveals that the time series has a linear structure, unlike the non-linearity observed in fig:2.4.

These results show that the dynamics in the blood pressure fluctuations are more complex than a mono-period oscillator. Further, from our earlier investigation of multi-periodic systems in section 1.3.1 we observe that blood pressure fluctuations do not show signatures of multi-periodicity.

2.5 Non-integral dimension

To uncover the underlying dimension of the BP’s dynamics, we construct correlation integral plots. The two sets of correlation plots were made for different values of embedding.
2.6. DISCUSSION

dimension \((p)\) and embedding delays \((h)\), while maintaining one of those parameters constant. For a given \(h\) and \(p\), the plots have an approximately fixed slope throughout the entire length of the data set. For \(h = 4\) time-steps (i.e 0.04s) and \(2 \leq p \leq 6\), the slopes range from \(1.6 \lesssim \nu \lesssim 2\). For \(p = 3\) and \(2 \leq h \leq 6\), all slopes are \(\nu \approx 2\). From their characteristics we gather that the system is neither periodic, nor random. More interestingly, the correlation dimension suggests a low dimensional deterministic structure.

2.6 Discussion

Analysis of the time series reveals deterministic signatures in the dynamics of BP. The presence of an attractor was confirmed by plotting the Poincare map. However, this deterministic structure could also be due to the inherent periodicity in the BP as evidenced from the Power Spectrum.

The periodic nature of BP was investigated by plotting recurrence maps. The recurrence map revealed discontinuous diagonal stripes. The broken diagonal pattern is unlike that of a noisy periodic time series. It should be kept in mind that blood pressure fluctuations are not periodic. Perhaps, a chaotic time series, or a non-stationary, periodic time series would produce such a recurrence plot. In either case, the presence of low dimensional determinism \((1.6 \leq \nu \leq 2)\) was confirmed by the correlation plots. The largest Lyapunov exponent was estimated to be positive \((\lambda_1 \approx 0.285)\), which improves our confidence that the time series is aperiodic. Further, the surrogate data analysis suggests an 85% chance of non-linear origin to the dynamics.

Past research on dynamical analysis of BP recordings \([69, 33, 37]\) have revealed similar results as ours. Wagner et.al\([69]\) studied the effect of baroreceptor denervation on the long term (4 hours) dynamical signatures of BP fluctuations in conscious dogs. Their results revealed a deterministic structure in the BP fluctuations of the control subjects. All control subjects’ BP fluctuations had a high Lyapunov exponent \((\lambda_1 = 1.85 \pm 0.18)\) and correlation dimension \((\nu = 3.05 \pm 0.23)\) while denervated subjects exhibited “less chaotic” and “more predictable” features \((\lambda_1 = 0.74 \pm 0.08, \nu = 1.74 \pm 0.2)\). Further, surrogate data analysis promised \((p−value < 10^{-5})\) an underlying non-linear structure for all control subjects, and for all but one of the denervated data set. The loss of complexity that accompanies denervation is hypothesized as an inability of the handicap system to adjust to external fluctuations.

A similar study involving baroreceptor intervention was performed by Lovell et al. \([37]\) on seven conscious dogs to analyze shorter durations (20 mins) of BP recordings. A deterministic phase space structure was observed in both, control and intervention group. In agreement with Wagner et al., the Lyapunov exponent was higher in the control group \((\lambda_1 = 0.40\pm0.04)\) than the intervention group \((\lambda_1 = 0.261\pm0.03)\). Surrogate data analysis
Figure 2.5: (Top row) The correlation integral for embedding dimensions $2 \leq p \leq 6$ sampled from different time intervals for the same embedding delay of $h = 4$ time steps (0.04s). (Bottom row) The correlation integral for embedding delay $2s \leq h \leq 6s$ sampled from different time intervals for the same embedding dimension of $p = 3$. 
revealed a non-linear structure in the dynamics. However, the control group had a smaller correlation dimension ($\nu = 0.3.46 \pm 0.38$) than the intervention group ($\lambda_1 = 4.59 \pm 0.23$), which contradicted Wagner et al.’s results. This was attributed to the different procedure of baroreceptor intervention. While Wagner et al. intervened anesthetically, Lovell et al. surgically isolated the carotid sinus region.

Kinnane et al’s [33] analysis of the blood pressure time series of white rabbits revealed results similar to Wagner et al’s. Their method of intervention involved removal of the arterial baroreflex loop. This was followed by dynamical analysis which revealed a lower Lyapunov exponent in the intervened system. The correlation dimension and Poincare map of the control subjects were comparable to Wagner et al. Surrogate data analysis indicated ($p$-value 0.05) a non-linear structure, however only 20 surrogate data were constructed. The rather small sample of surrogates casts doubt on the statistical significance of their analysis.

The above three research articles were mentioned to indicate that a vast number of researchers have investigated chaotic signatures in BP. These investigations were performed with the intention of identifying chaotic signatures in cardiac cycles under different physiological conditions. These researchers have produced a spectrum of results ranging from strong to nil evidence for chaos in cardiac cycles. However, conclusive evidence is still lacking on whether cardiac cycles are truly chaotic or not. Further, counter-arguments to the case of chaos in cardiac cycles have been made [21]. Thus, characterizing the chaotic nature of cardiac cycles is challenging. Further, identifying whether cardiac cycles are chaotic or not is not evidently useful.

However, as demonstrated by my investigations, analyzing cardiac cycles through the lens of non-linear dynamics can produce interesting results. I propose that instead of focusing on whether cardiac cycles are chaotic or not, future research would benefit by identifying the distinguishing dynamical features of healthy and pathological BP recordings. If abnormalities in the heart or vascular system translate into unique dynamical features in the BP fluctuations this knowledge could supplement current diagnostic methods. Further, such a line of research will also guide the development of cardiac system modelisation by constraining the possible set of models that are proposed.
Chapter 3

Chaos in population dynamics.

3.1 Introduction

Predation - the act of one organism killing another for the purpose of nourishment - is a predominant mechanism through which natural selection acts. In its broadest definition, predation includes grazing as well. It is best described as an evolutionary arms race between predators and prey who evolve to develop better predation and predator-evasive tactics respectively. Predation can alter species diversity and population composition. Predation, in one form or the other, is essential for the redistribution of resources in an ecosystem.

Predator-Prey systems can exhibit complex non-stationary dynamics, such as stable synchronous oscillations. Mathematical models of these systems can recreate the patterns observed in nature. However, these models are usually non-spatial models that do not account for such relevant length scales in the dynamics as predator-mobility, prey-mobility, predator’s feeding-range and prey herd size.

The relevance of length scales in ecological system has been of interest for at least a century. However, it was especially during the last four decades that the relevance of scale in ecosystems was recognized. The study of length scales is, perhaps, most relevant to ecology in the context of developing conservation strategies because biodiversity is an inherently spatial feature. For this reason, the study of length scales has inspired novel methods of investigating biodiversity richness and, the spatial characteristics thereof.

My motivation to investigate spatial models arises from several predator-prey models that have been developed by past researchers. A common theme across all models is the allusion to the existence of a critical length scale determined by the predator’s mobility. At this scale, deterministic dynamics is most strongly observed, though the nature of this determinism was not discussed. The authors also hypothesized that
analysis below or above this critical length scale would produce vastly different results. However, quantitative analysis of this dynamics requires to be performed.

In this chapter we will re-investigate the spatially explicit predator-prey model developed by [12], and analyze the relevance of length scale to dynamics. In doing so, I hope to uncover the nature of the deterministic dynamics the authors have alluded to multiple times. The methods of investigation developed in chapter 1 will reveal whether the deterministic dynamics is chaotic or not.

3.2 The Model

The model we use was developed by De Roos et al. [12], and reflects key assumptions of the Rosenzweig-MacArthur model [8]. We simulate a community of predators and prey inhabiting a $128 \times 128$ lattice with periodic boundary conditions. At any instant of time, a node of the lattice can be in one of four basic states: (i) empty patch, (ii) occupied by one prey only, (iii) occupied by one predator only, (iv) occupied by one prey and one predator. A snapshot of the lattice’s initial distribution is shown in figure 3.1.

![Figure 3.1: Snapshot of the simulated predator-prey dynamics. The community exists on a square $128 \times 128$ lattice with periodic boundaries. Each node of lattice is classified as either empty (green), containing one prey (yellow), containing one predator (red), or containing one prey and one predator (white)](image)

The Rosenzweig-MacArthur Model

The Rosenzweig-MacArthur model is similar to a Lotka-Volterra type of predator-prey interaction. In the absence of predators, a logistic growth and death is assumed for prey population. A linear predation rate is assumed, which translates to a linear predator reproduction rate, and an additional linear prey death rate.
Predator death is assumed linear as well. Further, the model accounts for predator handling time and foraging time. As a result, it accounts for the predator’s limited mobility.

Following the notation of [12], the set of non-linear equations that constitute the model:

\[
\frac{dF}{dt} = r_m F \left(1 - \frac{F}{K}\right) - \frac{aF}{1 + a\tau_h} C
\]

\[
\frac{dC}{dt} = \epsilon \frac{aF}{1 + a\tau_h} C - \mu C
\]

where, \(F\) and \(C\) are prey and predator densities respectively, \(r_m\) is the prey birth-rate, and \(K\) is the maximum carrying capacity for prey population. \(a, \tau_h, \epsilon,\) and \(\mu\) are the search rate, handling time, conversion efficiency, and death rate of predator respectively. \(a\) is the area scanned by a predator in unit time, and hence has the dimension of area/time.

\(\frac{aF}{1 + a\tau_h}\) is the number of prey killed by the predator in unit time. This is a Holling type-II functional response. The Holling type-II functional response assumes that food foraging and food processing are mutually exclusive, and models predator intake as a rectangular hyperbola. The predator’s kill rate attains a maximum of \(1/\tau\) at infinite prey density. The model exhibits large amplitude oscillation for low values of \(\mu\) and high values of \(K\).

The predator’s type-II response function dictates that for increasing search rates, and decreasing handle times, the prey killed increases asymptotically. For zero handle time the model reverts back to a type-I response function.
During a given iteration of the simulation, the state of each node is updated. Although nodes are updated in a random order, the basic updating procedure simulates five events in a fixed order.

The first simulated event is predator movement. We consider two versions of predator movement. In one version of the simulation, predators move **homogeneously** throughout the lattice. We simulate homogeneous movement by choosing a random destination for each predator from among all nodes in the lattice. If a predator is already found at the chosen destination, a new choice is made, and this continues until a node without a predator is found. In the second version, predators move in a **diffusive** manner. In this case, we choose a random destination for each predator from among its four nearest-neighbor nodes. If a predator is already found at the chosen destination, no predator movement occurs.

The second event is prey reproduction. Prey do not move, but the prey population can expand as prey individuals reproduce into nearby nodes. In our simulation, a prey individual attempts to reproduce with probability $P_{rg}$. If reproduction is attempted, then one of the four nearest-neighbor nodes is chosen at random. If the chosen node is already occupied by a prey individual, then reproduction fails. By contrast, reproduction is successful whenever no prey is found occupying the chosen node; and, in this case, a newborn prey individual is placed on the chosen neighboring node. Overall, the way in which prey reproduction is simulated implies that the growth of the prey population is negatively density dependent.

The third event is consumption of prey by predators. Consumption can only occur when a predator and prey occupy the same node. Moreover, consumption can only occur if at least $T_h$ time steps (iterations) have passed since a predator’s most recent meal. In biological terms, $T_h$ represents the time it takes a predator to handle, and ultimately consume, a prey individual. A positive handling time suggests that the predator should show a Type II functional response to increases in prey density.

The fourth event is predator reproduction. A predator attempts to reproduce $R_{mh}$ times, provided it has consumed at least $F$ prey individuals since its last set of attempts. For each attempt at reproduction, a parent chooses one random node from among its four nearest-neighbours. If the chosen node is not occupied by a predator, an offspring is placed there. If the chosen node is occupied by a predator, the parent chooses one random node from among its eight next-nearest neighbours. This process continues either until a node that is not occupied by a predator is found, or until the $m^{th}$ nearest neighbourhood is considered by the parent ($m = 4$ in our simulations). If the node, selected at random from the $m^{th}$ nearest neighborhood, is occupied by a predator, then the attempt at reproduction is aborted. Of course larger value of $m$ reduces the extent to which predator reproduction is density-limited.
The fifth, and final, event is predator death. Each predator dies with density-independent $P_d$.

3.3 Implementing the model

On the $128 \times 128$ lattice, an empty patch is specified by only one variable - its position. A prey as well is specified only by its position. However, a predator is specified by three variables - position, time since last meal, and number of meals consumed since last reproduction instance. To keep account of all these variables across the lattice, at all times, the lattice is programmed in Python’s class structure. Each node of the lattice has six attributes: `node.prey`, `node.pred`, `node.meals`, `node.meals`, `node.thand`, `node.nbrx`, `node.nbry`. If the node is (not) occupied by a prey, then `node.prey = 1(0)`. If the node is (not) occupied by a predator, then `node.pred = 1(0)`. A node that is occupied by a prey and a predator has `node.prey = 1` and `node.pred = 1`. If `node.pred = 1`, then the node can attain two more features. `node.meals \in [0, F]` keeps count of the number of meals the predator has consumed and `node.thand \in [0, T_h]` keeps count of the time since the predator last consumed a meal. `node.nbrx` is a list of x-coordinates of the node’s neighbours. `node.nbry` is a list of y-coordinates of the node’s neighbours. The choice of such a class structure is a computationally efficient method to keep track of the various states of the lattice.

The parameter values were set identical to the choice of De Roos and colleagues [12]. Particularly, prey growth rate $P_{rg} = 0.75$, predator handle time $T_h = 2$, predator litter size $R_{mh} = 2$, predator’s feeding cap $F = 4$, and predator death rate $P_d = 0.06$. The initial conditions were set by distributing the prey and predators uniformly on lattice. 20% of the lattice was allotted for predators, another 20% for prey, and the other lattice spots remained empty. The simulation was equilibrated for 1000 time-steps to stabilize transients and then iterated for a further 2048 time-steps to acquire data.

Specifically in the case of diffusive predator movement, alternate versions of the simulation were investigated. In one version, the initial conditions of prey and predator density were set to 50% and 20% respectively to investigate the effect of abundant prey. Then, the initial conditions of prey and predator density were set to 20% and 50% respectively to investigate the effect of deficient prey.

In another version, the boundary conditions were changed to Neumann boundary conditions. Finally, a larger lattice of $256 \times 256$ was simulated to consider the finite-size effect of the lattice.

The Python codes for simulating the different versions of the simulations have been detailed in Appendix.
3.4 Results

In accordance with results of De Roos et al.\cite{12}, the time-averaged population densities are approximately identical regardless of homogeneous or diffusive predator movement. However, the dynamics of the population differ markedly, even though the simulation parameters were kept identical ($P_{rg} = 0.75$, $T_h = 2$, $R_{mh} = 2$, $F = 4$, $P_d = 0.06$). As seen in figure 3.2, when predator mortality is sufficiently low, and predator movement is homogeneous, the prey and predator population undergo stable oscillatory dynamics. For the same parameter values in the case of diffusive predator movement, the predator population is stabilized against fluctuations. This stabilization is attributed\cite{12} to statistical averaging across the patchy population distribution. The simulations were also performed with different boundary conditions and initial conditions. The results of the dynamics are plotted in figure 3.3 and 3.4.

In figure 3.5, the clustering of individuals is evident. This points to a critical length scale that separates regions of different dynamical phases. Above this critical scale, population dynamics in different regions of the lattice are uncoupled from one another and tend to be out of phase. Regions of high density in one area, are balanced by lower densities elsewhere, and an apparent steady-state is the result. The patchiness of population distribution can be quantified by analyzing the lattice across different length-scales. We scan windows of varying sizes and compute the variance of the population density.
3.4. RESULTS

Figure 3.3: Alternate cases of diffusive predator movement were simulated. The case of Neumann boundary conditions (Diffusive NBC) exhibits a smaller mean population and a larger amplitude of fluctuations compared to the other cases. This could be due to the boundary restricting predator reproduction. Different initial conditions do not change the long-term dynamics because the system settles into the same equilibrium after the initial transient period.

Figure 3.4: Alternate cases of homogeneous predator movement were simulated. Apart from a phase shift, the oscillatory dynamics exhibit the same amplitude and frequency. Different initial conditions do not change the long-term dynamics because the system settles into the same equilibrium after the initial transient period.
3.4. RESULTS

Figure 3.5: The final state of the lattice, depicting the spatial distribution of predators and prey. (left) For homogeneous predator movement. (right) For diffusive predator movement. In the case of diffusive predator movement, it is evident that the individuals cluster.

Figure 3.6: Relationship between the window area of observation, and the variance in predator number per window area within the window, for homogeneous and diffusive predator movement. In the case of the diffusive predator movement, a characteristic hump occurs at a window area of \( \approx 4000 \) i.e for window side \( \approx 63 \) lattice points. The variance was computed over 1000 simulations.
3.4. RESULTS

against window area, with the variance being rescaled by the inverse of window area. In the case of homogeneous predator movement, the variance scales quadratically as a function of window area. This testifies to the accuracy of our computation. In the case of diffusive predator movement, a characteristic hump is observed at a window size of \( \approx 63 \times 63 \). This shall be further investigated in later sections.

### Variance for Diffusive Predator Movement

The plateauing of the variance in the case of diffusive predator movement can be understood as follows. At a critical window size of \( \ell^* \times \ell^* \) the dynamics becomes strongly deterministic, i.e all windows of size \( \ell^* \times \ell^* \) have identical variance. If we now consider a window \( \ell \times \ell \) where \( \ell = m\ell^* \), then this \( \ell \times \ell \) window contains \( m^2 \) copies of the \( \ell^* \times \ell^* \) window, suggesting the relation \( N_{\ell \times \ell} = m^2 N_{\ell^* \times \ell^*} \), where \( N_{\ell \times \ell} \) is the predator population within the \( \ell \times \ell \) window. Moreover, since the dynamics within each of the \( m^2 \) windows are uncouples, we can argue that \( \text{Var}(N_{\ell \times \ell}) = m^2 \text{Var}(N_{\ell^* \times \ell^*}) \). This gives us:

\[
\frac{\text{Var}(N_{\ell \times \ell})}{\ell^2} = \frac{\text{Var}(N_{\ell^* \times \ell^*})}{\ell^*^2} = C^* \quad (♦)
\]

From the above equation it follows that at windows larger than \( \ell^* \times \ell^* \) the \( \text{Var}(X_{\ell \times \ell})/\ell^2 \) tends to a constant value.

#### 3.4.1 Signatures of deterministic dynamics

The Poincare map was plotted for the time series of population dynamics in the case of homogeneous and diffusive predator movement. For homogeneous predator movement, the Poincare map shows the distinctive circular pattern that is seen in periodic time series. For diffusive predator movement, the Poincare map has filled out the entire space, signifying noisy dynamics. Though these Poincare maps were constructed for the population dynamics observed at the lattice size, a similar pattern is observed at all window sizes.

#### 3.4.2 Non-linearity in the system

To quantify the extent of non-linearity in the system, surrogate data was generated for the dynamics at different spatial scales. Following this, the Lyapunov exponent was computed for the original data and the surrogate data.

In particular, we will focus on the interesting length scales of \( \approx 63 \times 63 \) lattice points.

For homogeneous predator movement, the time series has a positive Lyapunov exponent at windows \( 68 \times 68, 72 \times 72 \) and \( 76 \times 76 \). The apparent non-zero Lyapunov exponent for a periodic system could be due to noise. At these windows, the surrogate data analysis
3.4. RESULTS

Figure 3.7: The Poincare map plotted in the case of (left) homogeneous predator movement, and (right) diffusive predator movement.

reveals a strong non-linear origin to the data, with a $p$-value $\leq 0.03$. thus refuting the null-hypothesis.

For diffusive predator movement, the time series shows non-chaotic signatures across all windows. At windows $58 \times 58$, $62 \times 62$, and $76 \times 76$, the surrogate data analysis has failed to refute the null-hypothesis with a $p$-value $\geq 0.35$. At windows $68 \times 68$ and $72 \times 72$, the surrogate data analysis reveals a $p$-value of $\approx 0.05$.

3.4.3 Signatures of aperiodicity

To detect any signatures of periodicity in the time series, we constructed recurrence plots across different length-scales of observation.

From the recurrence plots in figure 3.9, we observe that in the case of homogeneous predator movement, across all length scales, the recurrence plots exhibit five diagonal lines across the 400 time steps plotted. This implies a periodicity of $\approx 80$ time steps. At small length scales, the dynamics is dominated by noise, thus blurring the diagonal lines. The periodic structure is more pronounced at length scales one-tenth of the lattice size.

The recurrence plots were all constructed for an embedding dimension of 4 and an embedding delay of 6. To confirm that the deterministic component of the dynamics is identical at all length scales, we construct correlation integrals across different length scales.

The correlation integral in figure 3.10 identifies periodicity for small and large window sizes. Further, all plots appear to have the same slope ($4 \leq \nu \leq 5$), signifying identical correlation dimension across all length scales. This is as expected in the case of homogeneous predator movement, because a well-mixed dynamics is scale independent.

We perform a similar analysis for the case of diffusive predator movement. As seen
Figure 3.8: 500 surrogate data were generated for the time series of homogeneous (top row) and diffusive (bottom row) predator movement across different window sizes. In the case of homogeneous predator movement, we see non-chaotic signatures for windows $58 \times 58$ and $62 \times 62$, but chaotic signatures for windows $68 \times 68$, $72 \times 72$, and $76 \times 76$. In the case of diffusive predator movement, the time series shows non-chaotic signatures across all windows.
3.4. RESULTS

Figure 3.9: Recurrence plots for the case of homogeneous predator movement, as observed through different window sizes. (left) Window size of $12 \times 12$. The diagonal stripes are separated by $\approx 80$ timesteps, implying a periodicity of $\approx 80$ timesteps. (center) At a window size of $58 \times 58$ we observe a similar periodicity of $\approx 80$ timesteps. (right) For a window size of $76 \times 76$, the periodic structure is clear. All plots were constructed for embedding dimension of 4, and embedding delay of 4.

Figure 3.10: Correlation integral plots in the case of homogeneous predator movement. The plots reveal periodicity at all windows of observation, and similar dimensionality to the dynamics.
3.5. DISCUSSION

Figure 3.11: Recurrence plots in the case of diffusive predator movement, as observed through different window sizes of (left) $8 \times 8$, (center) $12 \times 12$ (right) $16 \times 16$. At the scale of $8 \times 8$ and $12 \times 12$, noise is a dominating factor. This can be inferred from the regular clustering of points at the four corners of the plot. At a larger scale of $16 \times 16$, the pattern is characteristically different, without any signatures of noise. All plots were constructed for embedding dimension of 4, and embedding delay of 6.

In figure 3.11, at small length scales of observation, the recurrence plot shows noisy signatures. Any deterministic dynamics is washed out by noise.

However, more interesting dynamics is observed at larger length scales as seen in figure 3.12. Especially near the $63 \times 63$ window size.

The recurrence plots point to interesting dynamics near the $63 \times 63$ window size. To investigate this neighbourhood, we plot the correlation integral as shown in figure 3.13. The embedding delay was 4 time steps and the embedding dimension was 5.

Across all window sizes, we observe a similar pattern. For small values of scanning distance $d$, the dynamics shows signatures of periodicity. At larger values however, they show a correlation dimension of $\nu \approx 2.4$.

3.5 Discussion

In this chapter we investigated a predator-prey model for chaotic signatures. The aim was to distinguish between the deterministic dynamics observed in homogeneous and diffusive predator movement. Our results show that mobility of predators at small scales have implications across the lattice.

In the case of homogeneous predator movement, the dynamics shows periodicity at all length scales, as inferred from the Poincare map, the recurrence plots, and the correlation integrals. However, the Lyapunov exponent was observed to be positive at length scales
3.5. DISCUSSION

Figure 3.12: Recurrence plots in the case of diffusive predator movement, as observed through different window sizes of (left) $58 \times 58$, (center) $68 \times 68$ (right) $76 \times 76$. At all 3 length scales, the recurrence plots show similar patterns that are characteristically different from the case of homogeneous predator movement. All plots were constructed for embedding dimension of 4, and embedding delay of 4.

Figure 3.13: The correlation integral plotted for a range of window sizes near the $63 \times 63$ window size, where the characteristic hump was observed in figure 3.6.
close to half the lattice size. This was unexpected as periodic time series have a dominant Lyapunov exponent of $\lambda = 0$. The Lyapunov exponent observed at window sizes $68 \times 68$, $72 \times 72$, and $76 \times 76$ are comparable to the Lyapunov exponent of the Logistic map in the chaotic regime. This positive Lyapunov exponent is attributed to the dynamical noise inherent in the system.

In the case of diffusive predator movement, a characteristic length scale was observed by plotting the population’s variance across different length scales. However, further investigation failed to confirm any special features of the dynamics. In fact, the Poincare map did not reveal any deterministic features, even though correlation integral plots point to low-dimensional deterministic dynamics. Surrogate data analysis of the dynamics near the characteristic length scale failed to confirm chaotic signatures even though recurrence plots show interesting patterns that are neither periodic, nor random. My methods of investigation are inconclusive. Alternative investigation should be performed to classify the nature of the dynamics.

Past researchers have performed similar investigations into identifying a characteristic length scale of an ecosystem. However, to the best of my knowledge, they did not investigate beyond detecting the signatures of a characteristic length scale. Whether this length scale produced any interesting dynamical features is still unclear.

The original work by De Roos et al. [12] had identified a characteristic length scale of a similar range. Their analysis demonstrated that predator mobility gives rise to the characteristic length scale. However, their analysis did not confirm their claim that the dynamics above and below this length scale are characteristically different. While my results do not refute that of De Roos et al. it does refute their claim that the dynamics above and below the characteristic scale is different. Further, it should be noted that De Roos et al. did not perform any non-linear analysis.

Rand and Wilson [11] had developed a similar simulation that involved diffusive predator and prey movement. They used this simulation to identify a characteristic length scale defined as the scale which maximizes the deterministic features of the dynamics. This was verified by computing the dimensionality of the system, and observing that much of the dynamics was confined to a four dimensional attractor. This contrasted with my result because no deterministic structure was observed within the Poincare map for diffusive dynamics. Further, they demonstrated that nearby trajectories diverged exponentially as evidence of chaotic dynamics. However, that was the only test for chaos they performed. Further tests should have been performed. Specifically, surrogate data analysis should have been performed.

Stone [61] performed a completely deterministic simulation of predator-prey interaction. Predators diffused towards regions of high prey density. Birth and death were determined by the individual’s age, population density. Further, the individuals were
modeled to exhibit diverse behavior that mimic insects. The ensuing population dynamics was tested for chaotic signatures by computing the Lyapunov exponent and correlation dimension. Both measures, $\lambda_1 = 0.04, \nu = 4.7$, were in support of chaotic dynamics at the scale of the lattice. The high dimension could be attributed to the various behavioral states that the model accounts for. This simulation was perhaps the most promising by far because it excluded the effect of noise. Thus, the positive Lyapunov exponent is attributed to purely deterministic sources. As a test for non-linearity, a special case of the simulation data was fit to the non-linear Ricker Logistic difference equation. However, this cannot be take as conclusive evidence for non-linearity. The goodness of the fit should be quantified using a distance metric. Surrogate data analysis should have also been performed. Further, this deterministic simulation should be investigated across different length scales.

My analysis shows that predator-prey simulations with limited predator mobility does show a characteristic length scale. However, whether this length scale is of any relevance for investigating ecosystems is not clear. Non-linear dynamical analysis did not reveal any distinguishable dynamical features between homogeneous and diffusive predator movement.

The purpose of non-linear time series analysis should be to identify the characteristic length scale, rather than detect chaotic signatures. Whether the dynamics is chaotic or not is irrelevant because such knowledge does not translate into handling real world ecosystems. If however, the dynamics is not characteristically different around the critical length scale, refuting previous researcher’s claims, then non-linear dynamical analysis would prove futile in understanding spatial characteristics from time series.

However, the system exhibits different statistical properties above and below this length scale. The ecosystem simulation exhibits many parallels with physical systems undergoing phase transitions. Hence, future research should instead focus on analyzing the spatial characteristics from a statistical perspective. Classifying the statistical properties above and below this length scale would aid in identification of the characteristic scale of real world ecosystems.
Chapter 4

Conclusions

This thesis has reviewed and demonstrated various techniques of detecting chaos in a time series. In order detect chaos, one must realise what are the characteristics of chaos. This in itself can be a challenge, because there is no single, universally accepted definition of chaos.

Following the words of Strogatz [62], we have a well-rounded definition of chaos. According to this definition, a system is considered chaotic if it is \textit{deterministic} (as opposed to random), \textit{non-linear, low-dimensional, aperiodic}, and \textit{sensitive to initial conditions}.

To confirm the presence/absence of any of these five traits in a time series, we have a specific set of computational tools that we can deploy. Poincare maps can detect deterministic structures in the phase space of the trajectory. Recurrence plots can suggest the extent to which the time series is periodic. Correlation integral plots can reveal the dimensionality of the system. Computing the Lyapunov exponent is a measure of the trajectory’s sensitivity to initial conditions. Finally, surrogate data analysis provides a confidence measure of the system’s non-linearity. It should be noted that none of these tools can individually confirm the presence/absence of chaos, but only collectively.

These computational tools were the backbone of this thesis. The algorithm of Grassberger and Procaccia [23] was used to estimate the correlation integral and the algorithm of Wolf et al. [72] was used to estimate the Lyapunov exponent. All algorithms were executed in Python language, and applied to model and experimental data sets.

The model data sets, particularly the time series generated by the Logistic and Henon maps, confirmed the correctness of my Python algorithms. Proceeding on that note, I investigated two experimental time series of biological origin.

The first time series was a recording of blood pressure fluctuations. Similar themes of research had been previously pursued by many researchers but, it had produced ambivalent results. My search for chaotic patterns within these blood pressure fluctuations revealed very interesting results. While some of the results are still open to interpretation,
and require further investigation, particularly in the context of non-stationarity, I have good confidence that blood pressure fluctuations are indeed chaotic. The Poincare map in section 2.2 revealed a deterministic structure within the blood pressure fluctuations which encouraged my investigations. Following this, the recurrence map in section 2.3 showed signatures of aperiodicity. In section 2.4 I computed the Lyapunov exponent to estimate the divergence of nearby trajectories and also performed a surrogate data analysis to measure the extent of non-linearity in the system. Both results were in support of chaotic signatures. The correlation dimension computed in section 2.5 converged to a non-integral value in the range of $1.6 \leq \nu \leq 2$.

However, on that note, I would like to mention that classifying blood pressure fluctuations as chaotic or not, in my opinion and others [21], does not carry much utility. A more fruitful line of research for future researchers to consider would be to identify pathological signatures within these physiological recordings. Past research has found dynamical properties that distinguish healthy and diseased heart rate [20] and even predict the onset of lethal cardiac arrhythmias [57].

The second time series I investigated was a simulated predator-prey interaction based on a model developed by De Roos et al. [12]. It has been suggested in past literature that limited individual mobility can give rise to characteristic length scales. This characteristic length is the scale at which the system exhibits maximum determinism. It was claimed that observations below and above this length scale would reveal different dynamical features. However, this claim was not supported by any evidence from the authors, and neither was it supported by my analysis presented here.

Rand and Wilson [1] had performed a similar simulation of mobility limited prey and predator on a lattice of size $150 \times 150$. Using a similar variance analysis as presented in figure 3.6 they identified a characteristic scale of the order of 100. However, the dynamical implications of this length scale was not presented. They demonstrated that at the scale of the lattice the dynamics could be reduced to four dimensions, and that nearby trajectories diverged exponentially. This was considered to be evidence for chaotic dynamics at the scale of the lattice although, further analysis should have been performed, in particular surrogate data analysis, because my analysis did not reveal any conclusive evidence for chaos in the dynamics.

The purpose of my investigation was to identify the dynamical aspects of the critical length scale. Following that, real world ecosystems could be investigated based on census recordings to identify the critical length scale of the system. Knowing the critical length scale of an ecosystem would help in developing conservation strategies. However, from my investigations, I conclude that the dynamical aspects of this critical length scale are difficult to uncover. Instead, a more fruitful pursuit would be to investigate the statistical aspects of the time series above and below this critical length scale.
However, during my analysis I also discovered that many of the claims of past researchers are debatable and require further support, or reconsideration of the basis for the claims for the benefit of ecologists and conservation strategists.
Chapter 5

Appendix

A. Lyapunov exponent program for non-linear maps

The following Python code was implemented to compute the Lyapunov exponent of a non-linear map such as the Logistic map \[^{[1.1]}\] or Henon map \[^{[A]}\]

A.1 Generating Data

```python
import numpy as np

def fun(x,y): #Henon map
    return ( 1 - 1.4 * x**2 + y, 0.3 * x )

(u,v) = (0.1, 0.2) #Initial conditions

# burn in trajectory
for t in range(10000): #Burning in the trajectory
    (u,v) = fun(u,v)

# collect raw data
nPts=2048 # mess with this to change estimate
x = np.empty(nPts,dtype=float)
for t in range(nPts):
    x[t] = u
    (u,v) = fun(u,v)
```

A.2 Create Embedded Data and Parameter Instantiation

```python
# create embedded data
```
dim = 2
tau = 2
z = np.array([])
for n in range(dim):
z = np.append(z, np.roll(x, -n*tau))
z = np.reshape(z, (dim, nPts))

# prune off useless data and transpose
z = z[:,0:-(dim-1)*tau]
z = np.transpose(z)

# time steps and duration
evolv = 1
dt = 1.0

sum = 0.0
nIter = 0
nPts = np.shape(z)[0]
scalmn = 1e-08
scalmx = 1e-01  # mess with this to change estimate
anglmx = np.pi/24  # mess with this to change estimate

ind0 = 0  # points to fiducial trajectory
dInit = 1e38  # start with ridiculous large number

A.3 Estimating Lyapunov Exponent

# find nearest nbr to initial pt of fiducial trajectory
# but make sure not too close in space or time
for i in range(10, nPts):
d = np.linalg.norm(z[ind0,:] - z[i,:])
if (d < dInit) and (d > scalmn):
dInit = d
ind1 = i

for t in range(nPts - evolv):
# get coordinates of evolved points
pt0 = np.copy(z[ind0+evolv,:])
pt1 = np.copy(z[ind1+evolv,:])
# compute final separation between pair and update the
# exponent ingredients

dFinal = np.linalg.norm(pt0-pt1)

sum += (np.log2(dFinal)-np.log2(dInit))/(evolv*dt)
nIter += 1

# search for replacement

flag = 0
for amult in 2**np.arange(3):
    anglmx = amult * anglmx
    for zmult in range(1, 6):
        thetaMin = np.pi
        for i in range(nPts - evolv):
            # reject candidates too close in time to the
            # fiducial trajectory
            if np.abs(ind0+evolv - i) < 10:
                continue

            # compute distance between candidate and fiducial
            dNew = np.linalg.norm(z[ind0+evolv, :]-z[i, :])
            # reject candidates too far or too close in space
            if (dNew > zmult * scalmx) or (dNew < scalmn):
                continue

            # compute angular change
            cosTheta = np.dot(pt0-pt1, pt0-z[i, :])/(dFinal*dNew)
            cosTheta = np.clip(cosTheta, -1., 1.)
            theta = np.arccos(cosTheta)
            # reject point if not smallest angular change so far
            if (theta >= thetaMin):
                continue

            thetaMin = theta
            d2 = dNew
            ind2 = i
            # currently, ind2 is best candidate we can find
            # and d2 is its distance from fiducial trajectory
            if (thetaMin <= anglmx):
                # then ind2 is good enough
flag = 1
break
# if thetaMin > anglmx, look at more distant points
if (flag == 1):
    # we have found a suitable replacement
    break

ind0 = ind0 + evolv
if (flag == 0):
    # stick with the old secondary trajectory
    ind1 = ind1 + evolv
    dInit = dFinal
else:
    # use the suitable candidate found above
    ind1 = ind2
    dInit = d2

print sum/nIter

B. Lyapunov exponent program for experimental time series

The code below was implemented in Python to estimate the Lyapunov Exponent of an experimental time series.

B.1 Create Embedded Data and Parameter Instantiation

```python
import numpy as np

nPts=4096  # mess with this to change estimate
x = bp[-nPts:]
x = x - np.mean(x)

deM = 1
dt = 1.0  # ten millisec

dim = 3
tau = 2
```
z = np.array([])
for n in range(dim):
    z = np.append(z, np.roll(x, -n*tau))
z = np.reshape(z, (dim, nPts))

# prune off useless data and transpose
z = z[:, 0:-(dim-1)*tau]
z = np.transpose(z)

sum = 0.0
nIter = 0
nPts = nPts - dim*tau  # not all data is useful for scanning
scalmn = 1e-02
scalmx = 1.0  # mess with this to change estimate
anglmx = np.pi/24  # mess with this to change estimate

ind0 = 0  # points to fiducial trajectory
dInit = 1e38  # start with ridiculously large number

B.2 Estimating Lyapunov Exponent

# find nearest nbr to initial pt of fiducial trajectory
# but make sure not too close in space or time
for i in range(10, nPts - evolv):
    d = np.linalg.norm(z[ind0, :] - z[i, :])
    if (d < dInit) and (d > scalmn):
        dInit = d
        ind1 = i

for t in range(nPts - evolv):
    # get coordinates of evolved points
    pt0 = np.copy(z[ind0+evolv, :])
    pt1 = np.copy(z[ind1+evolv, :])

    # compute final separation between pair
    # and update the exponent ingredients
    dFinal = np.linalg.norm(pt0 - pt1)
    sum += (np.log2(dFinal) - np.log2(dInit)) / (evolv*dt)
nIter += 1
# search for replacement
flag = 0
for amult in 2**np.arange(3):
    anglmx = amult * anglmx
    for zmult in range(1, 6):
        thetaMin = np.pi
        for i in range(nPts - evolv):
            # reject candidates too close in time to fiducial trajectory
            if np.abs(ind0+evolv - i) < 10:
                continue
            # compute distance between candidate and fiducial
            dNew = np.linalg.norm(z[ind0+evolv, :]-z[i, :])
            # reject candidates too far or too close in space
            if (dNew > zmult * scalmx) or (dNew < scalmn):
                continue
            # compute angular change
            cosTheta = np.dot(pt0-pt1, pt0-z[i,:])/(dFinal*dNew)
            cosTheta = np.clip(cosTheta, -1., 1.)
            theta = np.arccos(cosTheta)
            # reject point if not smallest angular change so far
            if (theta >= thetaMin):
                continue
            thetaMin = theta
            d2 = dNew
            ind2 = i
            # currently, ind2 is the best candidate we can find
            # and d2 is its distance from fiducial trajectory
        if (thetaMin <= anglmx):
            # then ind2 is good enough
            flag = 1
            break
    if (flag == 1):
        # we have found a suitable replacement
        break
# if no suitable point and no way to continue secondary
# trajectory then should continue through the main loop
# and start a new fiducial trajectory
if (ind1 + evolv > nPts - evolv) and (flag == 0):
    continue

ind0 = ind0 + evolv
if (flag == 0):
    # stick with the old secondary trajectory
    ind1 = ind1 + evolv
    dInit = dFinal
else:
    # use the suitable candidate found above
    ind1 = ind2
    dInit = d2

print "Lyapunov Exponent is" % (sum/nIter)

C. Simulating Predator-Prey interactions

The simulation was implemented in Python. The code for the various steps in the simu-
lation are given below

C.1 Parameter specification and Data Collection

```python
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt

# model parameters
Prg = 0.75
Rmh = 2
Pd = 0.06
F = 4
Th = 2

# simulation parameters
dim = 128
```
Pr_prey = 0.2
Pr_pred = 0.2
max_nbhds = 4
max_iter = 3048

# data collection
n_windows = 7
nprey_homo = np.zeros(max_iter, dtype=int)
npred_homo = np.zeros(max_iter, dtype=int)
npreyeaten_homo = np.zeros(max_iter, dtype=int)

windowsize = [4,32,64,128,1024,4096,16384]

n_windows = len(windowsize)
windowx,windowy = np.random.choice(np.arange(dim)),
np.random.choice(np.arange(dim))

PreyWindows_homo = np.zeros((max_iter,n_windows),dtype=int)
PredWindows_homo = np.zeros((max_iter,n_windows),dtype=int)

C.2 Auxillary functions

#To compute Temporal Autocorrelation
def TAC(x,max_delay):
    maxiter = len(x)
    xbar = np.mean(x)
    n = len(x)
    xvar = np.var(x)
    autocorr = np.zeros(max_delay)
    for k in range(maxiter//2):  #for a delay of k
        AC = 0
        for t in range(1,n-k):
            AC+= (x[t]-xbar)*(x[t+k]-xbar)
        autocorr[k] = AC/xvar
    return autocorr

#To sample different window sizes
def sample_window(mynode,i,j,max_nbhds):
    prey_cnt = 0
    pred_cnt = 0
    for n in range(max_nbhds):
for y in range(-n,n+1):
x = n - np.abs(y)
if x == 0:
    prey_cnt += mynode[np.mod(i+y,dim),j].prey
    pred_cnt += mynode[np.mod(i+y,dim),j].pred
else:
    prey_cnt += mynode[np.mod(i+y,dim),
                        np.mod(j+x,dim)].prey
    pred_cnt += mynode[np.mod(i+y,dim),
                        np.mod(j+x,dim)].pred

return prey_cnt, pred_cnt

C.3 Instantiation of lattice with periodic boundary conditions

# construct node object
class node:
    pass

# initialize lattice
print "initializing"
lattice = np.array([[]])

for i in range(dim * dim):
    lattice = np.append(lattice, node())
lattice = np.reshape(lattice, (dim,dim))

for i in range(dim):
    for j in range(dim):
        # initialize state of the node
        lattice[i,j].prey = np.random.binomial(1, Pr_prey)
lattice[i,j].pred = np.random.binomial(1, Pr_pred)
lattice[i,j].meals = np.random.choice(np.arange(F))
lattice[i,j].thand = np.random.choice(np.arange(Th))
        # establish topology
        lattice[i,j].nbry = [[]]*max_nbhds
        lattice[i,j].nbrx = [[]]*max_nbhds

for n in range(max_nbhds):
    tmpy = []
C.3 Instantiation of lattice with Neumann boundary conditions

```python
# Instantiating the lattice for Neumann boundary conditions.
for i in range(dim):
    for j in range(dim):
        # initialize state of the node
        lattice_base[i,j].prey = np.random.binomial(1, Pr_prey)
        lattice_base[i,j].pred = np.random.binomial(1, Pr_pred)
        lattice_base[i,j].meals = np.random.choice(np.arange(F))
        lattice_base[i,j].thand = np.random.choice(np.arange(Th))
        # establish topology
        lattice_base[i,j].nbry = [ [] for _ in range(max_nbhds) ]
        lattice_base[i,j].nbrx = [ [] for _ in range(max_nbhds) ]
        for n in range(max_nbhds):
            tmpy = []
            tmpx = []
            for y in range(-n, n+1):
                x = n - np.abs(y)
                if x == 0:
                    tmpy.append((i+y))
                    tmpx.append(j)
                else:
                    tmpy.append((i+y))
                    tmpx.append((j+x))
                    tmpy.append((i+y))
            lattice[i,j].nbrx[n] = tmpx
            lattice[i,j].nbry[n] = tmpy
```
C.4 Predator movement

Homogeneous movement with periodic/Neumann boundary conditions

```python
# establish a list of all nodes in lattice
long, lat = np.meshgrid(np.arange(dim), np.arange(dim))
lat = np.reshape(lat, np.size(lat))
long = np.reshape(long, np.size(long))
order = np.arange(dim*dim)

# iterate
for iter in range(max_iter):
    print "iteration\n" + str(iter+1),
    print "of\d" % max_iter

# update predator state
for k in order:
    i, j = lat[k], long[k]
    lattice[i,j].thand += 1

# Homogeneous predator Movement
np.random.shuffle(order)
for k in order:
    i, j = lat[k], long[k]
    if lattice[i,j].pred == 1:
        inew, jnew = np.random.choice(np.arange(dim)),
        np.random.choice(np.arange(dim))
        while lattice[inew,jnew].pred == 1:
            inew, jnew = np.random.choice(np.arange(dim)),
            np.random.choice(np.arange(dim))
    if lattice[inew,jnew].pred == 0:
        lattice[inew,jnew].pred = 1
        lattice[i,j].pred = 0
        lattice[inew,jnew].meals = lattice[i,j].meals
        lattice[inew,jnew].thand = lattice[i,j].thand
```
Diffusive predator movement with periodic boundary conditions

```python
# Diffusive predator Movement
np.random.shuffle(order)
for k in order:
i, j = lat[k], long[k]
if lattice_diff[i,j].pred == 1:
bh = 1
l = np.random.choice(np.arange(len(lattice_diff[i,j].nbrx[bh])))
inew, jnew = lattice_diff[i,j].nbry[bh][l], lattice_diff[i,j].nbrx[bh][l]
if lattice_diff[inew,jnew].pred == 0:
lattice_diff[inew,jnew].pred = 1
lattice_diff[i,j].pred = 0
```

Diffusive predator movement with Neumann Boundary conditions

```python
# predators move diffusively
np.random.shuffle(order)
for k in order:
i, j = lat[k], long[k]
if lattice_diff_NBC[i,j].pred == 1:
bh = (1)
inew, jnew = 10000,100000
while inew < 0 or jnew < 0 or
     inew > dim-1 or jnew > dim-1:
    # print('entering predmove loop')
l = np.random.choice(np.arange(len(lattice_diff_NBC[i,j].nbrx[bh])))
inew, jnew = lattice_diff_NBC[i,j].nbry[bh][l], lattice_diff_NBC[i,j].nbrx[bh][l]
if lattice_diff_NBC[inew,jnew].pred == 0:
lattice_diff_NBC[inew,jnew].pred = 1
lattice_diff_NBC[i,j].pred = 0
```
C.5 Prey reproduction, prey consumption and predator death

```python
# prey reproduce
np.random.shuffle(order)
for k in order:
    i, j = lat[k], long[k]
    if lattice[i,j].prey == 1:
        if np.random.binomial( 1, Prg ) == 1:
            nbhd = 1
            l = np.random.choice( np.arange(lattice[i,j].nbrx[nbhd]))
            inew, jnew = lattice[i,j].nbry[nbhd][l], lattice[i,j].nbrx[nbhd][l]
            if lattice[inew,jnew].prey == 0:
                lattice[inew,jnew].prey = 1

# predators consume prey
prey_eaten = 0
for k in order:
    i, j = lat[k], long[k]
    if lattice[i,j].prey == 1
        and lattice[i,j].pred == 1
        and lattice[i,j].thand > Th:
            lattice[i,j].prey = 0
            lattice[i,j].meals += 1
            lattice[i,j].thand = 0
            prey_eaten +=1
npreyeaten_homo[iter]=prey_eaten

# predators reproduce
np.random.shuffle(order)
for k in order:
    i, j = lat[k], long[k]
```
if lattice[i,j].pred == 1 and lattice[i,j].meals >= F:
    lattice[i,j].meals = 0
for birth in range(Rmh):
    for nbhd in range(1,max_nbhds):
        l = np.random.choice( np.arange( len(lattice[i,j].nbrx[nbhd])) )
        inew, jnew = lattice[i,j].nbry[nbhd][l],
                     lattice[i,j].nbrx[nbhd][l]
        if lattice[inew,jnew].pred == 0:
            lattice[inew,jnew].pred = 1
            lattice[inew,jnew].meals = 0
            lattice[inew,jnew].thand = Th
            break

# predator death
for k in order:
    i, j = lat[k], long[k]
    if lattice[i,j].pred == 1:
        lattice[i,j].pred -= np.random.binomial(1,Pd)
Bibliography

[1] Using spatio-temporal chaos and intermediate-scale determinism to quantify spatially extended ecosystems.


Navaneeth Mohan

EDUCATION

University of Western Ontario, Canada
MSc. in Applied Mathematics
Cumulative Grade Point Average: 8.8/10
August 2016 - July 2018 (expected)

Chennai Mathematical Institute, India
MSc. in Physics
Cumulative Grade Point Average: 8.30/10
August 2014 - July 2016

Chennai Mathematical Institute, India
BSc. (Hons) in Physics
Cumulative Grade Point Average: 7.48/10
August 2011 - July 2014

ACADEMIC PROJECTS

Time-Series Analysis of Physiological Data
University of Western Ontario, London
Advisor: Prof. Geoff Wild
- Obtained data from multiple sources within PhysioNet Data Bank
- Analyzed physiological data for statistical patterns.
- My investigations reveal that blood-pressure fluctuations are chaotic.
- This result has the potential to improve cardiac diagnostic tools.
- Wrote code in Python to perform statistical data analysis.
Sept 2016 - Present

Time-Series Analysis of Ecological Data
University of Western Ontario, London
Advisor: Prof. Geoff Wild
- Ran large scale simulations of a mathematical model of an ecosystem.
- Wrote Python code to analyze simulation results.
- My analysis revealed that many long-standing assumptions in existing ecosystem models need to be revisited.
Sept 2016 - Present

Molecular dynamics simulation of glass
Institute of Mathematical Sciences, Taramani
Advisor: Dr. Satyavani Vemparala
- Study the mechanical properties of glass in collaboration with an experimental group at IIT-Madras.
- Ran large-scale simulations in computer clusters to mimic the conditions of glass preparation.
- Wrote code in Python to analyze results. My analysis directed future experimental research.
May 2015 - April 2016

Identification of Transcription factor binding sites
Institute of Mathematical Sciences, Taramani
Advisor: Dr. Rahul Siddharthan
- Developed Python code for identifying binding sites on RNA strand.
- Algorithm used Bayesian Inference with the prior probabilities set by high throughput data.
- Learned Python packages, and algorithms in Dynamic Programming and Optimization.
May - July 2012

RELEVANT COURSES
Advanced Mathematical Biology, Statistics, Monte-Carlo Methods, Advanced Python, Data Analysis with R,
**JOURNALISM & COMMUNICATIONS**

**News Producer at Radio Western**  
**May 2018- Present**  
- Journalist: Cover local events, record audio interviews, and capture photographs. Report back to the radio station by the prescribed deadline.  
- News Reader: Research local and campus news, write a script, read and record said script, and air the news within a tight deadline of two hours.  
- Honed my skills in interviewing, photography, script writing, and recording/editing on Adobe Audition.  
- Worked individually or as a team when required.

**Freelance contributor at Western News**  
**March 2018- Present**  
- I have been actively working with Dr. Jason Winders, Director, Editorial Services to cover stories.  
- Learned to research an issue, find a story, validate the evidence, pitch the story, and meet deadlines in print media.  
- High quality of journalism expected at campus newspaper.

**Co-host/Producer/“Headhunter” for Gradcast**  
**July 2017- Present**  
- Gradcast is a radioshow/podcast aimed at sharing the life and research of graduate students at University of Western Ontario.  
- Interview researchers outside of my research expertise, thus requiring me to adapt to every researcher.  
- Prepare a script to interview said researchers.  
- Record/edit on Adobe Audition and upload to different podcast distributors.  
- Network on campus, cold-call researchers, and recruit guests for the show.  
- Honed my communication vocabulary, elevator pitch, and interview skills.

**Editor/Contributor at Arrogantgenome**  
**July 2017- Present**  
- The Arrogantgenome is a blog founded by Dr. David Smith featuring science articles by University of Western Ontario’s students.  
- My task was to proofread, understand the science, and paraphrase the content in the vernacular.

**PROFESSIONAL EXPERIENCE**

**Mathematics and Physics Tutor**  
**October 2017 - Present**  
- Mathematics and Physics tutor for high school and elementary students.  
- Broadened my vocabulary of analogies/metaphors for explaining difficult mathematical concepts.

**Waiter at Massies Fine Indian Cuisine**  
**May 2017 - August 2017**  
- Worked 3 days a week at lunch hours during summer term (May-August).  
- Greet customers, set tables, prepare drinks, and serve customers.  
- Make quick decisions to offer speedy customer service while maintaining a calm and patient demeanor.

**Teaching Assistant**  
**September 2016 - Present**  
- Taught 1st year Calculus.  
- Prepare and deliver lectures to a class of over 100 students, and grade their quizzes.  
- This experience has taught me methods and habits to engage a large audience.

**Data Science Internship at EvenRank**  
**May - July 2015**  
- EvenRank is a Job Searching Portal with access to a large database.  
- My responsibility was to analyze data and obtain statistical information about job trends.  
- Wrote code to perform exploratory data analysis on JSON files, extract specific information, and visualize said information as histograms.
PROFESSIONAL DEVELOPMENT

Projected Futures Summer School in Science Journalism July 2018
- Performed street interviews of the general populace on concerns of science.
- Improved my writing skills, interview skills, story pitches, and media presence.
- Learned to write about scientific articles outside of my research expertise.
- I was the only international student to be awarded a tuition waiver.

Basic Life Support Skills Certified by American Heart Association to provide essential First Aid.

Language of Difficult Conversations An eight hour certified workshop on conflict resolution offered by the Teaching Support Centre at University of Western Ontario.

Teaching in the Canadian Classroom A sixteen hour interdisciplinary certified course covering a range of communication tools and inter-personal skills for teaching. Offered by the Teaching Support Centre at University of Western Ontario.

VOLUNTEER WORK

Organizer, Western Conference in Science Education 2017
My tasks included organizing the presentation room and introducing speakers at WCSE 2017. Tested my abilities in communication, multi-tasking and meeting deadlines.

Mechanic, Purple Bikes September 2016 - Present
Fix bikes and offer speedy customer service at Purple Bikes, the campus bike shop.

Teacher, Pudiyador March 2015 - April 2016
- Pudiyador organizes after-school programs for children of lower income families.
- Volunteered at Pudiyador to conduct hands-on science activities.
- Tested my ability to handle a classroom and teach in a vocabulary designed for elementary school children.

AWARDS
Canadian Mathematical Society President’s Award for poster presentation.

COMPUTER SKILLS
- Programming Languages: Python, Scipy, Numpy, SAS, R
- Scientific: Matlab, Mathematica
- Scripting: \LaTeX, HTML, CSS, Javascript
- Database Management: MongoDB, PyMongo

LANGUAGES
English(fluent); French(basic); Tamil(fluent); Malayalam(intermediate); Hindi(basic).

EXTRACURRICULAR
Perform at amateur comedy/music open-mics.