CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

Calculating Lyapunov Exponents from Time Series

Lejla Dobrić

August, 2022
Supervisor: Mgr. Lynnyk Volodymyr Ph.D.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Dobrić Lejla**	Personal ID number:	**474686**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Branch of study:	**Computer and Information Science**		

II. Bachelor's thesis details

Bachelor's thesis title in English:

Calculating Lyapunov Exponents from Time Series

Bachelor's thesis title in Czech:

Výpočet Ljapunovových exponentů z časových řad

Guidelines:

1. Study various methods for determining Lyapunov exponents from time series.
2. Compare the applications of these methods to time series generated by several chaotic systems using numerical calculations (Matlab/Octave software).
3. Verify the accuracy of calculations of Lyapunov exponents obtained by these methods depending on the length of the analyzed data sets.

Bibliography / sources:

[1] WOLF, Alan, Jack B. SWIFT, Harry L. SWINNEY and John A. VASTANO. Determining Lyapunov exponents from a time series. B.m.: Elsevier BV. July 1985.
[2] BROWN, Reggie. Calculating Lyapunov exponents for short and/or noisy data sets. B.m.: American Physical Society (APS). June 1993.
[3] ROSENSTEIN, Michael T., James J. COLLINS and Carlo J. DE LUCA. A practical method for calculating largest Lyapunov exponents from small data sets. B.m.: Elsevier BV. May 1993.
[4] KANTZ, Holger. A robust method to estimate the maximal Lyapunov exponent of a time series. B.m.: Elsevier BV. January 1994.
[5] WANG, Wen-Xu, Ying-Cheng LAI and Celso GREBOGI. Data based identification and prediction of nonlinear and complex dynamical systems. B.m.: Elsevier BV. July 2016.

Name and workplace of bachelor's thesis supervisor:

Mgr. Volodymyr Lynnyk, Ph.D. Department of Control Engineering FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **21.12.2021** Deadline for bachelor thesis submission: **15.08.2022**

Assignment valid until: **30.09.2023**

Mgr. Volodymyr Lynnyk, Ph.D. Supervisor's signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student's signature

Acknowledgement / Declaration

I would like to express my gratitude to my supervisor Mgr. Lynnyk Volodymyr Ph.D. who guided me throughout this project.

I would also like to thank my family who is my biggest support throughout my academic journey.

At last but not least, I need to thank the Ministry of Education, Youth and Sports of the Czech Republic for giving me this opportunity as a holder in their scholarship program for students from developing countries.

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, August 15. 2022

....

Abstrakt / Abstract

Ljapunovovy exponenty jsou důležité pro charakterizaci atraktoru nelineárního dynamického systému a jeho citlivost na počáteční podmínky. Jinými slovy, Ljapunovovy exponenty nám říkají, kdy je systém chaotický.

V této práci se budeme zabývat dvěma různými metodami odhadu Ljapunovových exponentů z datové řady: Wolfovou a Rossensteinovou metodou. Aplikaci těchto metod porovnáme na datových řadách generovaných několika chaotickými systémy pomocí numerického výpočtu v softwaru Matlab. Přesnost odhadu Ljapunovových exponentů ověříme různými metodami v závislosti na délce analyzovaných datových řad a přidáme k datům šum, abychom viděli, jak si tyto algoritmy povedou s přidáním šumem.

Klíčová slova: Chaos, atraktor, dynamické systémy, Lyapunovy exponenty, Lyapunov spektrum, Wolf metoda, Rosenstein metoda

Překlad názvu: Výpočet Ljapunovových exponentů z časových řad

Lyapunov exponents are important for the characterization of an attractor of a nonlinear dynamic system and their sensitivity to initial conditions. In other words, Lyapunov exponents tell us when the system is chaotic.

In this Theses we are going to study two different methods of estimating the Lyapunov exponents from the data series: Wolf's and Rosenstein's methods. We will compare the application of these methods to the data series generated by several chaotic systems using a numerical calculation in Matlab software. We will verify the accuracy of estimation of the Lyapunov exponents by different methods depending on the length of the analyzed data series and add noise to the data to see how these algorithms will perform with additive noise.

Keywords: Chaos, attractor, dynamic systems, Lyapunov exponents, Lyapunov spectrum, Wolf's method, Rosenstein's method

/ Contents

Tables / Figures

Chapter 1
Introduction

Before the development of computers and technology that would be able to calculate the huge amount of data, chaotic results were explained in a deterministic way of thinking. The thesis of determinism is that all events in the universe are unavoidable. However, if this world would be deterministic, then we would be able to predict every event and make a system that would be able to control everything. Chaotic behaviour was considered to be the impact of disturbances on the systems. Soon after, scientists started to notice that even very small changes in the initial conditions could make different outcomes of the events. Concept of chaos is still relatively new, however widely recognized in mathematics. One of the ways to study dependency of system on initial conditions is Lyapunov exponents. Calculating Lyapunov exponents let us find out if a system shows chaotic behaviour.

1.1 Chaos

The concept of chaos has its beginnings in the 1900s. Henri Poincare studied the problem of object's movement to 3 mutual gravitational forces. Poincare discovered that some orbits could be non-periodical and not constantly increasing or close to a fixed point.

One of the pioneers of discovering chaos was Edward Lorenz. Lorenz worked on weather prediction and simulation. One day, in 1961, Lorenz decided to take a shortcut. Instead, to start the whole program from the beginning, he started it midway through. For initial conditions, he typed numbers from the previous printout. Lorenz noticed that the new run did not give the same results as the previous one, which was something unexpected. The new run should duplicate exactly the same outcome, but that did not happen. After a while, he realized that the computer's memory stored six decimal places as .506127, but on the printout, to save space, it was written only three decimal places: .506. Lorenz discovered, that even small changes in initial conditions were showing different weather results [1].

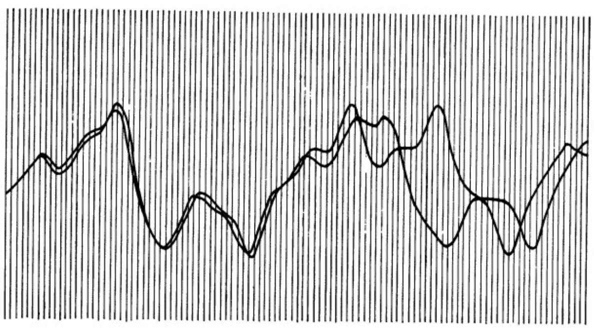

Figure 1.1. Lorenz's 1961 printouts: From nearly same initial points computer for weather prediction shows patterns that, over the time, were getting more apart from each other, until they completely lost all resemblance [1].

In chaos theory these differences in initial conditions and their sensitivity to the outcome is called 'The Butterfly effect', or in more technical terms: 'Sensitive dependence on initial conditions'.

There are many different variations of famous quote for the Butterfly effect, we chose to say it this way: If a butterfly flutters its wings in Beijing, it could cause a hurricane in Florida.

This does not mean that one butterfly is directly connected to the tornado on the other side of the world, but it means that even small changes in initial conditions can cause a series of events that will give us a different outcome. Soon after these discoveries, scientists started to change their way of understanding nature, from the traditional deterministic model to understanding nature as part of the chaos.

There are many ways how we can define deterministic chaos. We will define it as in [2] as unpredictable chaotic behaviour of nonlinear dynamical systems to which are applied deterministic rules.

Dynamical systems are set of equations that evolve over the time. They can be discrete-time (consist of discrete difference equations - maps) or continuous-time (consists differential equations - flows) [3]. Changes of dynamical systems are described with trajectory and orbit. In continuous-time dynamical systems trajectory is a path (progress, line development) that flow takes over the time and orbit is a collection of points (curve) over time evolution. Dynamical systems give us models of real-life events like the spreading of diseases, weather changes, economic recessions, etc.

Chaos is present in many fields e.g. physics, chemistry, economy, astronomy. One of the important problems that all fields have in common is distinguishing deterministic chaos from noise[1]. Detecting and quantifying chaos has become an important challenge.

There have been many approaches to specify chaos e.g. fractal power spectra, entropy, fractal dimension. Unfortunately, based on the research and testing it is proven that most of them fail to characterize chaos on both model end experimental data [4].

There are different models that explain chaotic systems: (1)Henon, (2)Rössler-chaos, (3)Lorenz model, (4)Rabinovich-Fabrikan system, (5)Mackey-Glass, (6)Rössler's hyperchaos. In the next chapter, we will explain some of those models.

1.1.1 The synchronization of chaotic systems

One of the important applications of Lyapunov exponents is in the synchronization of chaotic systems. Let us have two trajectories with very close initial conditions. Over time, they will exponentially separate (diverge). In consequence, chaotic systems initially go against synchronization. This became a significant problem, especially because we can never perfectly know experimental initial conditions. It may seem to us that synchronization of two chaotic systems is impossible, however, with just the right exchange of information of two systems, synchronization can be made

In terms of chaos, we will define *synchronization* as a process, in two or more chaotic systems, that adjust their property of motion to common behaviour. These chaotic systems can be equivalent or nonequivalent. Adjusting can vary from the agreement of trajectories to locking of phases [5]. It is important to mention, that the word *synchronization* does not always describe the same process and needs to be specified for different contexts.

It is shown in [6], that systems will synchronize when Lyapunov exponents for the subsystems are all negative.

[1] Data with a big amount of meaningless information that we call noise.

To sum up, synchronization is relevant phenomena and its properties have been shown in nature as in dynamics of cardio-respiratory system or magnetoen-cephalographic[2] activity of patients with Parkinson's disease.

This is a broad topic to cover, and if you are interested to know more about the synchronization of chaos, we address the reader to Ref. [5]. We will also return to this topic in Chapter 2.5.

1.2 Motivation

Based on the disadvantages of other methods, we can consider that spectrum of Lyapunov exponents can provide us with better diagnostic for chaotic systems. For example, one of the most popular methods used to quantify chaos is the Grassberger-Procaccia algorithm. The popularity of this algorithm can be to its simplicity. Nevertheless, this algorithm has some disadvantages as sensitivity to variations in parameters and it can be unreliable except for long, noise-free time series [7].

Lyapunov exponents are important for the characterization of an attractor of a nonlinear dynamic system and their sensitivity to initial conditions. In other words, Lyapunov exponents tell us when the system is chaotic. Also, they show us the rate of exponential attraction (separation), of the two nearby trajectories with different initial conditions in the phase space (in time evolution). However, calculating Lyapunov exponents can be difficult. Some of the methods that are used for the calculation of Lyapunov exponents can face problems as [7]:

- Unreliable for small data sets
- Can not apply to experimental data
- Numerically unstable
- Demands a lot of computation
- Can be difficult to implement

The objective of this Thesis is to go through different methods of calculating Lyapunov exponents from time series: Wolf's method and Rosenstein's method. We will analyze how algorithms perform on different chaotic systems, under different data sizes and conditions and compare the results of different methods.

1.3 Outline

In Chapter 2 we will define and analyze some of the chaotic systems and go into more detail about the synchronization of chaotic systems. In Chapter 3, we will define and explain Lyapunov exponents and go through two of the methods for calculating Lyapunov exponents: Wolf's method and Rosenstein's method. In Chapter 4, we will compare and analyze the results of those methods under different data sizes and different signal-to-noise ratio (SNR) values. We will see how those methods perform on different chaotic systems and their error rates. Finally, Chapter 5 contains a summary of our conclusions.

[2] Measurement of the magnetic field generated by the electrical activity of neurons.

Chapter 2
Chaotic systems

In this Chapter, we chose chaotic systems based on the historical significance and their use in research in the literature.

Let us first define what is attractor. After considering different definitions we can say that attractor, in dynamical systems, is a state or group of states to which a system is prone to evolve.

2.1 Lorenz attractor

First model of chaos we will mention is Lorenz attractor (1963) [8] - named after meteorologist Edward Lorenz.

Definition of Lorenz attractor is given by three differential, nonlinear equations:

$$\dot{x} = \sigma(x - y) \tag{1}$$
$$\dot{y} = x(\rho - z) - y \tag{2}$$
$$\dot{z} = xy - \beta z, \tag{3}$$

where the equations relate to the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above. Parameters σ, ρ and β represent Prandtl number[1], Rayleigh number[2], and certain physical dimensions of the layer itself. Variables x, y and z evolve over time and they represent convective flow, horizontal temperature distribution and vertical temperature distribution.

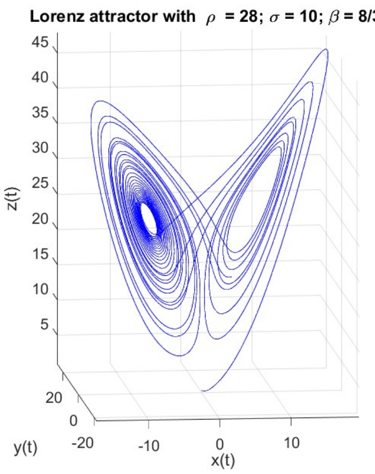

Figure 2.1. Plot of Lorenz attractor with standard values.

[1] Describes the relationship between momentum diffusivity and thermal diffusivity.
[2] Product of the Grashof number, which describes the relationship between buoyancy and viscosity within a fluid.

In Fig. 2.1 we can see how lines are making many curves that form two overlapping spirals that look like butterfly wings. These curves never intercept or go back to their own path. They create a loop, wondering on the one spiral and changing its path to the other side. This plot represents chaos and its randomness and unpredictability.

2.2 Rössler-chaos system

Definition of the Rössler model (1976) [9] is given by these three nonlinear differential equations:

$$\dot{x} = -(y + z) \tag{4}$$

$$\dot{y} = x + ay \tag{5}$$

$$\dot{z} = b + z\,(x - c), \tag{6}$$

where a, b and c are parameters and x, y and z are variables. For parameters Rössler used: $a = 0.20$, $b = 0.20$ and $c = 5.70$. Variables x, y and z evolve with time.

The simplicity of Rössler model lies in the first two equations which are linear and one nonlinear equation. This model has two fixed points called *equilibria* $(F\pm)$. One fixed point $F-$ lies in the centre of the attractor, and the other one $F+$ is outside of the attractor. To find these points, we set all three equations to zero. After solving them we will get coordinates for $F\pm$.

$$(x_\pm, y_\pm, z_\pm) = \left(\frac{c \pm \sqrt{c^2 - 4ab}}{2}, -\frac{c \pm \sqrt{c^2 - 4ab}}{2}, \frac{c \pm \sqrt{c^2 - 4ab}}{2} \right) \tag{7}$$

For Rössler model, we can use linear methods such as eigenvectors, because of their linear properties. However, to fully understand this system, we need to use nonlinear methods such as Poincaré maps and bifurcation diagrams.

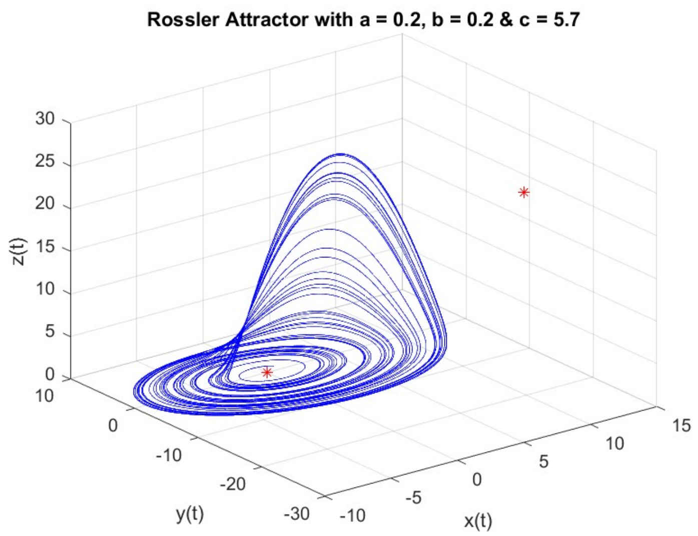

Figure 2.2. Plot of Rössler attractor with standard values. Red points represent the first and second fixed points of the system.

In Fig. 2.2 we can see that one of the fixed points lies in the centre of the attractor and the second point lies far from the attractor. Nonlinear behaviour will happen when the trajectory leaves the xy dimension. The second fixed point influences the attractor and rises the plot to the z-dimension.

2.3 Henon map

Henon map (1976) [10] is a simplified two-dimensional version of the Lorenz model for a discrete-time dynamical system that shows us chaotic behaviour. This model is able to show us the fractal microstructure of a strange attractor. The main purpose of this model is to make the numerical analysis faster and more accurate, to be able to follow the solutions for a longer time.

Henon map is a composition of several maps: folding, stretching, contracting.

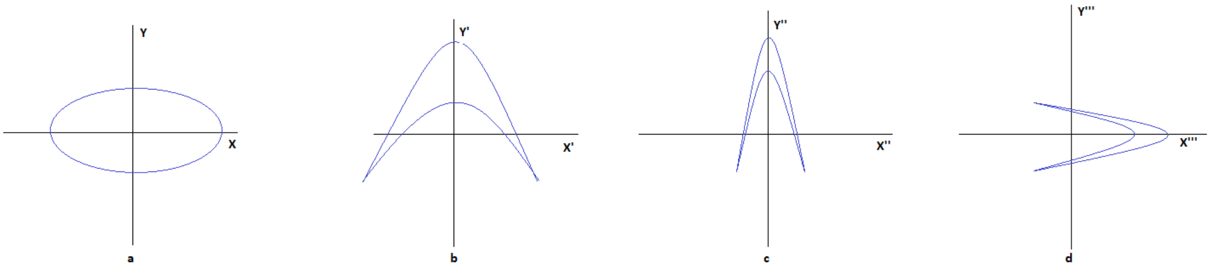

Figure 2.3. Initial area a mapped by T all the way into area d mapped by T'''. Figure b: T' preserves areas; Figure c: Contracts areas; Figure d: T''' preserves areas but reverses the sign.

The Henon map T is composition of $T = T'''T''T'$. Let $x_n = x$, $y_n = y$ and $x_{n+1} = x'''$, $y_{n+1} = y'''$ (mapping will be iterated). Map is defined by equations:

$$x_{n+1} = 1 - \alpha\, x_n^2 + y_n \tag{8}$$
$$y_{n+1} = \beta\, x_n, \tag{9}$$

where parameters $\alpha = 1.4$ and $\beta = 0.3$ show us that map is chaotic. By repeated application of T, succesive points do not always converge towards an attractor but sometimes they diverge to infinity.

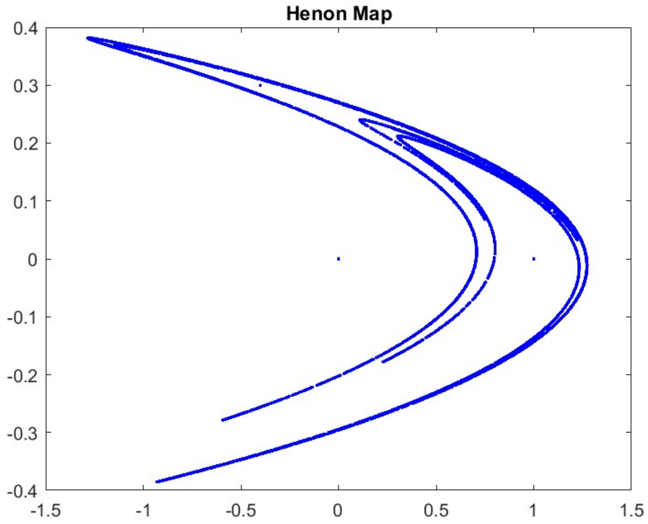

Figure 2.4. Henon map for values $\alpha = 1.4$ and $\beta = 0.3$, $x = 0$, $y = 0$. This plot shows the result of 10000 successive points, starting from the initial points x_0 and y_0.

2.4 Rabinovich-Fabrikan system

Physicists Mikhail Rabinovich and Anatoly Fabrikant, invented a new type of chaotic system, named Rabinovich-Fabrikan (1979) [2]. What is interesting about this system, is that it contains third-order nonlinearities that shows us some unusual dynamics (shapes of wave forms like virtual saddles, double vortex tornado).

Definition of the Rabinovich-Fabrikan system is given by these three nonlinear differential equations:

$$\dot{x} = y\,(z - 1 + x^2) + \gamma\, x \tag{10}$$
$$\dot{y} = x\,(3z + 1 - x^2) + \gamma\, y \tag{11}$$
$$\dot{z} = -2z\,(\alpha + xy), \tag{12}$$

where α and γ are parameters. System dynamics are more sensitive on α parameter than γ. It contains five hyperbolic equilibrium[3] points, one at the origin and four dependent on the parameters α and γ.

[3] Constant solution to a differential equation.

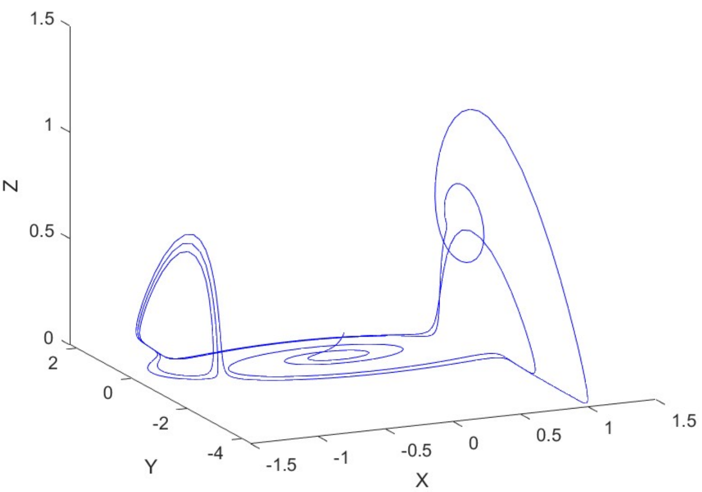

Figure 2.5. Rabinovich-Fabrikant with values $\alpha = 0.1$, $\gamma = 0.98$ and initial values $[0.1, 0.1, 0.1]$. This plot shows that system is chaotic.

Rabinovich-Fabrikant system is not a clearly understood problem that is the subject of only a small number of papers.

2.5 Synchronization types of chaotic systems

Based on the particular coupling configuration, we can divide the synchronization process into unidirectional coupling and bidirectional coupling [5–6].

In the case of unidirectional coupling, the system is constructed with two subsystems that make a drive-response (master-slave) configuration. This means that one of the subsystems (master) evolves voluntarily and leads (drives) the evolution of the other subsystem. As a consequence, the other subsystem (response) is *slaved* and must follow the dynamic of the master. The drive system behaves as chaotic forcing for the response system which produces external synchronization.

In bidirectional coupling, we have a different case. Both subsystems are coupled with each other, expressing mutual synchronize behaviour. For example, this type of process occurs between the cardiac and respiratory systems.

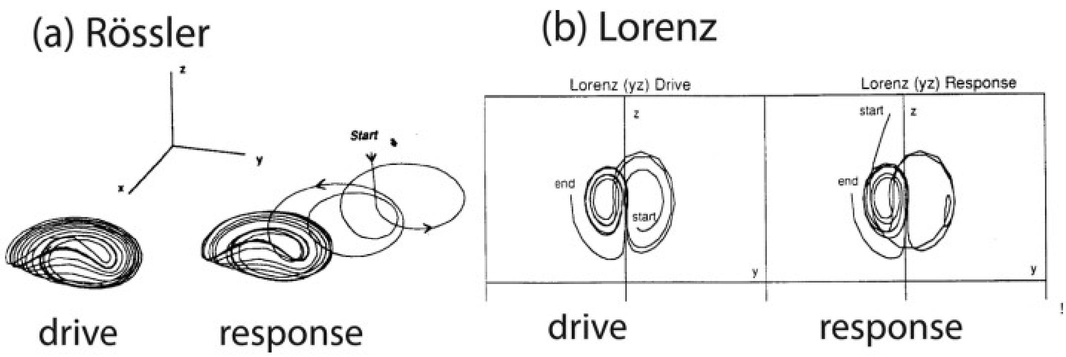

Figure 2.6. Rössler and Lorenz drive-response systems [6].

Complete or identical synchronization (CS) is the simplest one and the first discovered. CS is made up of perfect hooking of the two chaotic systems trajectories. It is accomplished by a coupling signal, where both systems remain in step with each other over time evolution.

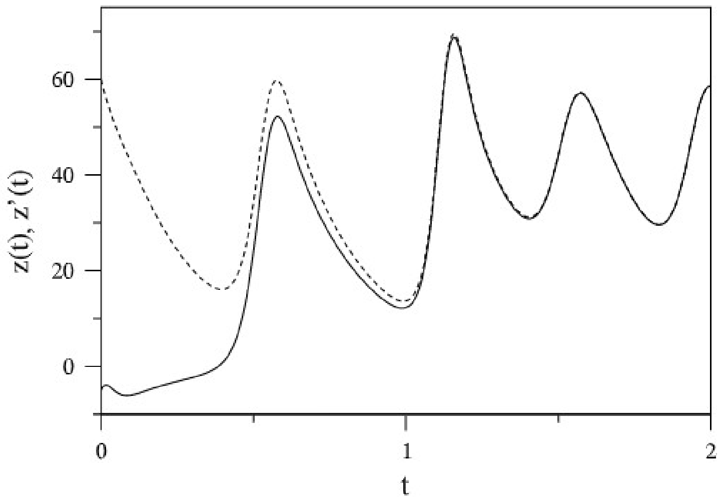

Figure 2.7. Plot of Complete synchronization on Lorenz system [5].

Except for the complete synchronization, for coupled chaotic systems, there are many other synchronization states that have been studied: a) phase and lag synchronization, b) intermittent lag synchronization, c) generalized synchronization, d) imperfect phase synchronization and e) almost synchronization.

Chapter 3
Lyapunov exponents

Lyapunov exponents are the average exponential values of divergence or convergence of two neighbouring trajectories. Using Lyapunov exponents, we can measure this exponential convergence/divergence and characterize it.

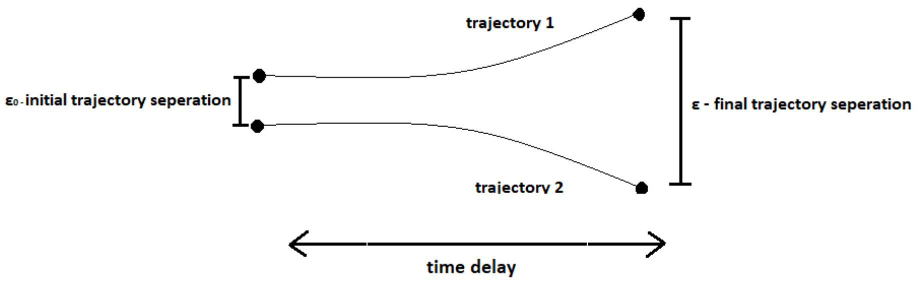

Figure 3.1. A schematic diagram of two separating trajectories [11].

3.1 The Lyapunov spectrum

Let us have n-dimensional infinitesimal [1] sphere of initial conditions in the continuous dynamical system, where n is the number of variables that describe the system. The nature of the flow is misshaped (deformed) and over time (t), the n-sphere evolves into n-ellipsoid.

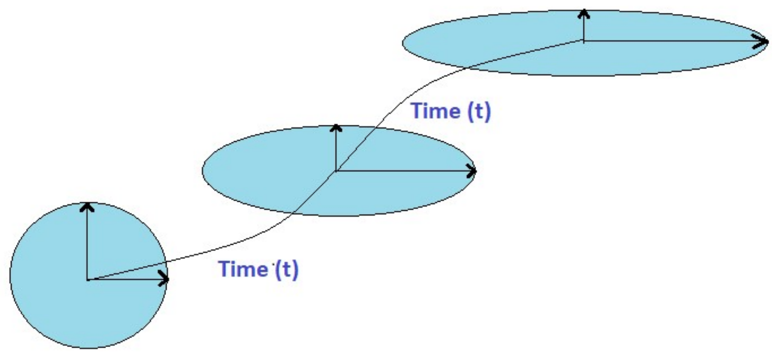

Figure 3.2. Transformation of sphere to ellipsoid over time(t).

[1] An indefinitely small quantity; a value approaching zero.

Ellipsoid principal axes expand or contract by values determined by Lyapunov exponents. That means that we can define Lyaponov exponent in the matter of the length of the principal axis $p(t)$ of the ellipsoid [4, 12].

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)}. \tag{1}$$

We arrange n principal axes of the ellipsoid from largest to smallest

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n,$$

where λ_1 corresponds to the expanding and λ_n to the most contracting principal axe in phase space. We notice that the length of the first principal axis grows as $2^{\lambda_1 t}$. The area defined by the first two principal axes grows as $2^{(\lambda_1+\lambda_2)t}$, the volume defined by the first k principal axes grows as $2^{(\lambda_1+\lambda_2+...+\lambda_k)t}$.

This means that we can define *Lyapunov spectrum* as the exponential growth of a k-volume element, which is given as the sum of the first k exponents. This definition can be useful for use of experimental data [7].

In [13] author proved that every continuous-time dynamical system will have at least one zero exponent, even if it does not have a fixed point. The ellipsoid axes that are expanding and contracting are equal to positive and negative Lyapunov exponents. As the ellipsoid evolves over time, its direction is continuously changing, which means that we can not well define the direction related to the given exponent.

The system that contains at least one positive Lyapunov exponent will present chaotic behaviour. The system with all negative Lyapunov exponents will not be chaotic and it will contain an attracting fixed point or periodic cycle. An example of a non-chaotic system you can see in Figure 3.3.

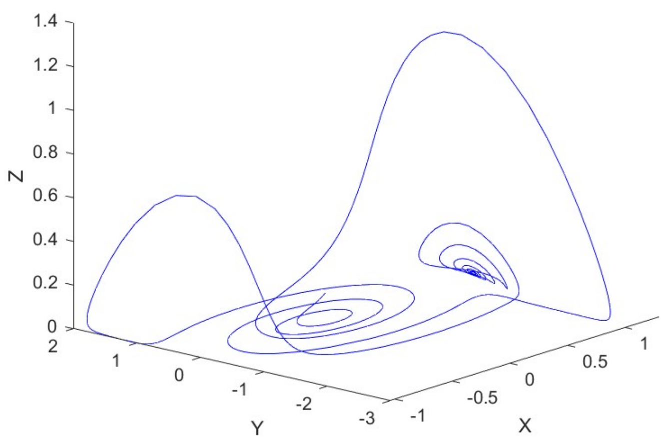

Figure 3.3. Plot of Rabinovich-Fabrikant equations with values $\alpha = 0.1$, $\gamma = 0.5$ and initial values [0.1, 0.1, 0.1]. Example of a non-chaotic system. We can notice attracting fixed point roughly at (x, y, z) = (1.07, −0.4, 0.07) [2].

We can use signs of Lyapunov exponents to categorize the asymptotic behaviour of the system's dynamic. More details you can see in Table 3.1.

Dimension	Dynamics of the attractor	Lyapunov spectrum
1	Chaos	+
	Periodic orbit	-
	Marginally stable orbit	0
2	Periodic motion	0, -
3	Strange attractor	+, 0, -
	A two torus	0, 0, -
	Limit cycle	0, -, -
	Fixed point	-, -, -
4	Rössler hyperchaos	+, +, 0, -
	Hypertortus T^3	0, 0, 0, -

Table 3.1. System's dynamic based of signs of Lyapunov exponents [14].

3.2 Wolf's method

Wolf's method [4] has various versions of its algorithm: 1) fixed evolution time programs for λ_1 and $\lambda_1 + \lambda_2$, 2) variable evolution time programs for $\lambda_1 + \lambda_2$ and 3) interactive programs (used on a graphic machine).

3.2.1 Fixed evolution time program for λ_1

This program is not considered to be sophisticated but it is easy to understand and a good start for understanding this algorithm.

Given the time series $x(t)$, an m-dimensional phase portrait is reconstructed with the time delay method[2]. The point on attractor is defined by

$$\{x(t), x(t+\tau), ..., x(t+[m-1]\tau)\}, \tag{2}$$

then we locate, in the Euclidian sense, the nearest neighbour to the initial point

$$\{x(t_0), ...x(t_0+[m-1]\tau)\} \tag{3}$$

and denote the distance between these two points. We will mark this distance as $L(t_0)$. As time evolves, at time t_1, initial length will progress to length $L'(t_1)$. The length L is generated through the attractor for a short time that is enough for only a small-scale attractor structure that is likely to be examined.

If the evolution of time t is too wide, L' might reduce (shrink) as it passes through the folding region of the attractor (L' is defined by two trajectories). This could lead to inaccuracy of Lyapunov exponent λ_1. The next step is to look for a new data point that sufficiently fulfils two criteria:

- Small separation of $L(t_1)$ from the evolved fiducial point.
- Small angular separation between the evolved and replacement elements.

[2] In experimental data, we usually don't get all variables.

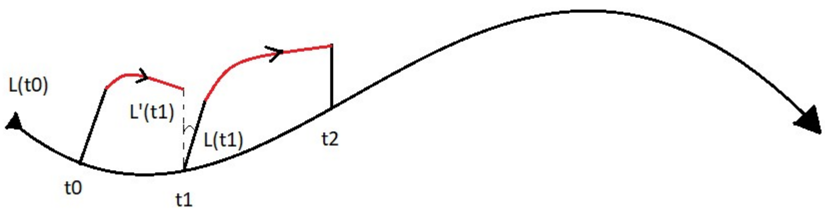

Figure 3.4. Illustration of Wolf's method for evolution time program for λ_1 [15, 4].

If new data points that fulfil these criteria can not be found, we hold on to the points that were used. We repeat this process until the fiducial trajectory has traversed[3] the entire data file. At this point we approximate:

$$\lambda_1 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{L'(t_k)}{L(t_{k+1})}, \tag{4}$$

where M is the total number of replacement steps.

3.2.2 Variable evolution time program for $\lambda_1 + \lambda_2$

This algorithm has similarities to the previous one 3.2.1. However, it is more complex for implementation. We chose three data points: the initial fiducial point and the two nearest points (neighbours). By these points, we define area $A(t_0)$. The area $A(t_0)$ is monitored until it is possible and recommended to do a replacement step. This makes us use various additional input parameters:

- Minimum number of evolution steps before replacement.
- Number of steps to evolve backwards when replacement is shown to be insufficient.
- Maximum length (area) before replacement is attempted.

The evolution time is a variable. Evolution continues until a problem happens. Problems that Wolf's algorithm includes are:

- Principal axis vector grows too quickly.
- The area grows too quickly.
- The area element's skewness[4] exceeds a threshold value.

If any of these problems happen, the trio of data points that are mentioned before, evolve backwards and we try to make replacement. If replacement fails, we will make another step backward to the trio of points and try again. We repeat this process until a trio of data points is extremely close to the previous replacement.

[3] Visiting the elements of the structure and doing something with the data.
[4] Measure of the symmetry of a distribution.

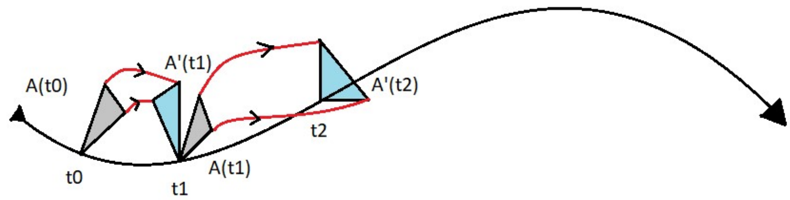

Figure 3.5. Illustration of Wolf's method for $\lambda_1 + \lambda_2$ [15, 4].

Then at this point, we take the best replacement point that is accessible and go forward. At the first replacement time t_1, two points that are not on fudicial trajectory are replaced by two new points to get a smaller area $A(t_1)$. The orientation of $A(t_1)$ in phase space is almost the same as the evolved area $A'(t_1)$.

We repeat this process until the fiducial trajectory has traversed the entire data file. At this point we estimate:

$$\lambda_1 + \lambda_2 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{A'(t_k)}{A(t_{k-1})}, \tag{5}$$

where t_k is the time of replacement step.

3.3 Rosenstein's algorithm

Rosenstein's algorithm [7] is meant to be fast and easy to implement. The first step is to reconstruct the attractor dynamics from single time series using the method of delays. The next step is to find the nearest neighbours (of each point of trajectory).

For the nearest neighbour, we search for the point that minimizes the distance of the specific reference point X_j. This is written as:

$$d_j(0) = \min_{\mathbf{X}_{\hat{j}}} \|\mathbf{X}_{\hat{\mathbf{j}}} - \mathbf{X}_{\mathbf{j}}\|, \tag{6}$$

where $|...|$ symbolizes Euclidian norm and $d_j(0)$ is the initial distance from the $j-th$ point to its nearest neighbour. Here we define limitation to nearest neighbours to have temporal separation greater than the mean period of the time series. For the mean period, we use the mean frequency of the power spectrum[5].

$$\left| \, j - \hat{j} \, \right| > mean\ period. \tag{7}$$

With this, we can look at each pair of neighbours as close initial conditions for different trajectories. Then, the mean rate of separation of nearest neighbours estimates the largest Lyapunov exponent.

[5] We can use other comparable estimate e.g., the median frequency of the magnitude spectrum.

Divergence of $j-th$ pair of nearest neighbour is given by the rate of largest Lyapunov exponent:

$$d_j(i) \approx C_j \exp^{\lambda_1(i.\Delta t)}, \tag{8}$$

where C_j represents the initial separation. When we take the logarithm by both sides of the equation, we get:

$$\ln(d_j(i)) \approx \ln(C_j) + \lambda_1(i.\Delta t). \tag{9}$$

The equation represents a set of parallel lines, where each slope is corresponding to λ_1. Then, using the least-square fit to the line, we can easily calculate the largest Lyapunov exponent. It is defined by:

$$y(i) = \frac{1}{\Delta t} \langle \ln(d_j(i)) \rangle, \tag{10}$$

where $\langle ... \rangle$ represent an average over all values of j. It is important to mention that for small, noisy data sets this process of averaging is crucial to calculate λ accurately. In equation (8) separation of neighbours is normalized, but as you can see in equation (9) this normalization is not necessary for calculating λ_1. By avoiding normalization, we can gain a small computational advantage.

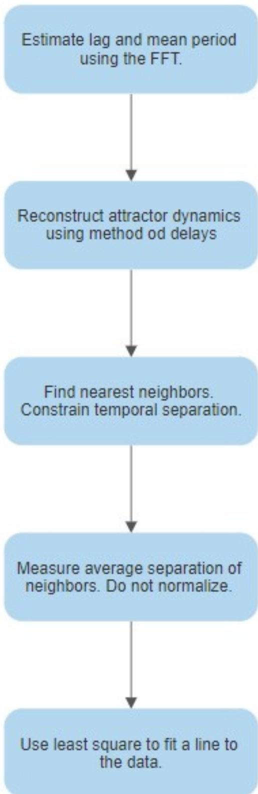

Figure 3.6. Outline of Rosenstein's algorithm [7].

3.3.1 Reconstruction delay

Let us express the reconstructed trajectory, X, as matrix:

$$X = [X_1 X_2 \cdots X_m]^T. \tag{11}$$

Each row of the matrix represents a phase-space vector. X_i represents the time series. For an N-point time series, $X = [X_1 X_2 ... X_n]^T$, the reconstructed matrix is represented as:

$$X = \begin{pmatrix} X_0 & \cdots & X_{(m-1)\tau} \\ \vdots & \ddots & \vdots \\ X_n & \cdots & X_{n+(m-1)\tau} \end{pmatrix}, \tag{12}$$

where m is the embedding dimension, and τ is the reconstruction delay (lag). Matrix X (Mxn) and constants N, M, m and τ are linked as:

$$M = N - (m-1)\tau. \tag{13}$$

To determinate, the right value of reconstruction delay is still an open dilemma. An overly large estimate of the reconstruction delay τ will make the elements of every vector to behave as if they are distributed randomly. On the other hand, underestimating τ would lead to highly correlated elements of the vector. It would lead elements to be concentrated around the diagonal in the embedding space, and insufficiently capture the structure that is perpendicular to the diagonal. Rosenstein's considers the optimal estimation of *lag* can be determined using methods based on auto-correlation function or correlation sum. Furthermore, we get the smallest errors for the *lag* where the auto-correlation function drops to 1 - $\frac{1}{e}$ of its initial value. Finding optimal τ can also be accomplished using the Fast Fourier Transform (FFT) [7].

The optimal value of *lag* is different for every system. For the Henon system, the best results for this algorithm were seen when the value of τ is equal to one. Lorenz and Rössler system performed efficiently for all *lag*s, other than extreme ones [15].

Chapter 4
Numerical results

In this chapter we will compare numerical results we got recreating Wolf's and Rosenstein's algorithms [15–16]. For Wolf's, we used 3.2.1 version of method. We will apply these two algorithms to Lorenz and Rössler systems and see how algorithms perform with different lengths of time series, and with additive noise. Quantitative analyses of chaotic systems can be sensitive to the observation time $t(s)$. For Lorenz system we will use observation time $t(s)$=0.01 and for Rössler system $t(s)$=0.10.

System	Equation	Parameters	t(s)	Expected λ_1
Lorenz[8]	$\dot{x} =$ (x-y) $\dot{y} = x(\ -z) - y$ $\dot{z} = xy- \ z$	$\sigma = 16.0$ $\rho = 45.92$ $\beta = 4.0$	0.01	1.50[4]
Rössler[9]	$\dot{x} = -(y + z)$ $\dot{y} = x + ay$ $\dot{z} = b + z(x - c)$	$a = 0.15$ $b = 0.20$ $c = 10.0$	0.10	0.090[4]

Table 4.1. Chaotic dynamical systems and theoretical values for the largest Lyapunov exponent, λ_1. The sampling period is denoted by Δt [7].

4.1 Runge-Kutta methods

To get numerical results of Lyapunov exponents, we have to find numerical solutions of ordinary differential equations (ODEs). There are many different methods that are used for solving ODEs. These methods can vary in stability, accuracy and simplicity [2]. In this paper, we will use Runge-Kutta methods for solving ODEs.

The Runge–Kutta methods [2, 17], developed around 1900 by the german mathematicians C. Runge and M.W. Kutta, are a group of explicit and implicit iterative methods used to approximately solve ODEs.

The general form [18] is:

$$Y_{n+1} = y_n + hF(T_n, y_n; h), n >= 0, y_0 = Y_0, \quad (1)$$

where $F(T_n, y_n; h)$ can be considered as average slope of the solution on the interval $[t_n, t_{n+1}]$.

One of the explicit and well-known Runge–Kutta method is the fourth order Runge–Kutta method generally referred to as `RK4`.

However, explicit Runge–Kutta methods are not best for solving stiff equations. Their region of absolute stability is small, which is why they are usually unstable. Hence, it is better to use implicit Runge–Kutta methods that are more stable for stiff equations. If you are interested to know more about Runge–Kutta methods, we advise the reader to Ref. [17].

4.2 Time series length

Let us consider the performance of two algorithms for time series of different lengths.

Wolf's algorithm (see results in Table 4.2) performed better for the Rössler system. The Lorenz system was more difficult to test for the fixed observation time because its ill-defined orbital period made it difficult to avoid catastrophic replacements near the separatrix [4].

System	N	m	Callculated λ_1	% error
Lorenz	1000	3	0.362	-75.86
	2000		1.162	-22.53
	3000		1.191	-20.06
	4000		1.942	29.46
	5000		2.147	43.13
Rössler	400	3	0.061	-31.66
	800		0.057	-36.66
	1200		0.036	-59.88
	1600		0.054	-39.77
	2000		0.108	20.00

Table 4.2. Numerical results of Wolf's method for time series of different lengths.

Rosenstein's [7] reported that their algorithm works well for small N, with errors less than $\pm 10\%$ in almost all cases. Also, it is reported that algorithm faces more difficulty with the Rössler system.

In Table 4.3, you can see the results we got implementing this algorithm. For τ we used for Lorenz system value 10, and for the Rössler system we used value 8. We can say that a bigger value of N gives us better results with an error of 2.6% for the Lorenz system, and an error of -4.4% for Rössler system. In our case, for smaller N we got errors less $\pm 15.5\%$ in almost all cases.

As we mentioned, quantitative analyses of chaotic systems can be sensitive to the observation time $t(s)$. Best results are obtained when $t(s)$ is relatively long, and value of N is bigger ($N = 5000$, $t(s) = 0.01\ s$, $N \cdot \Delta t = 50\ s$). However, comparable results can be achieved even when N is smaller. That is why, it is advised in [7], that as long $t(s)$ is small enough (approximately n to $10n$ points [4], to secure a minimum number of points per orbit of the attractor), it is better to reduce the sampling rate (decrease N) and not the observation time $t(s)$.

System	N	τ	m	Callculated λ_1	% error
Lorenz	1000	10	3	0.464	-69.06
	2000			1.281	-14.6
	3000			1.269	-15.4
	4000			1.382	-7.8
	5000			1.539	2.6
Rössler	400	8	3	0.074	-17.7
	800			0.081	-10.0
	1200			0.099	10.0
	1600			0.084	-6.6
	2000			0.086	-4.4

Table 4.3. Numerical results of Rosenstein's method for time series of different lengths.

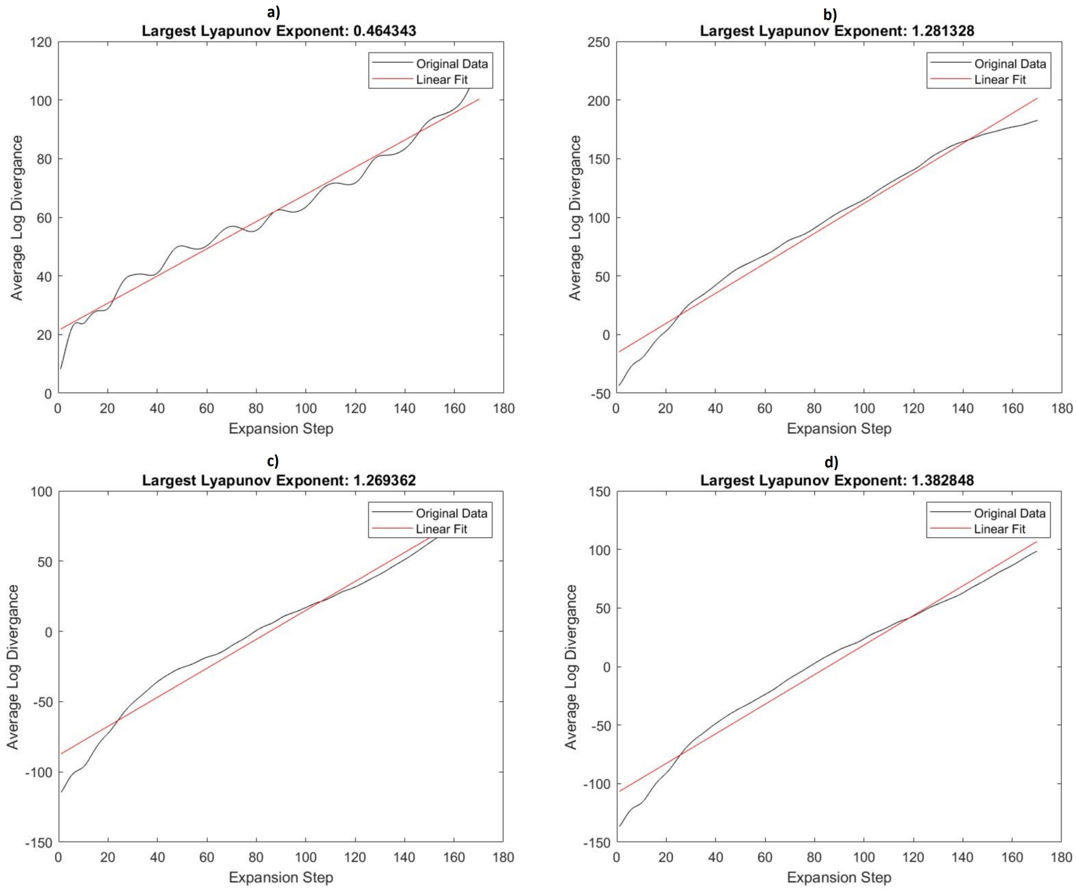

Figure 4.1. Plot of Average Log Divergence versus Expansion Step for the Lorenz attractor for a) N = 1000, b) N = 2000, c) N = 3000, d) N = 4000.

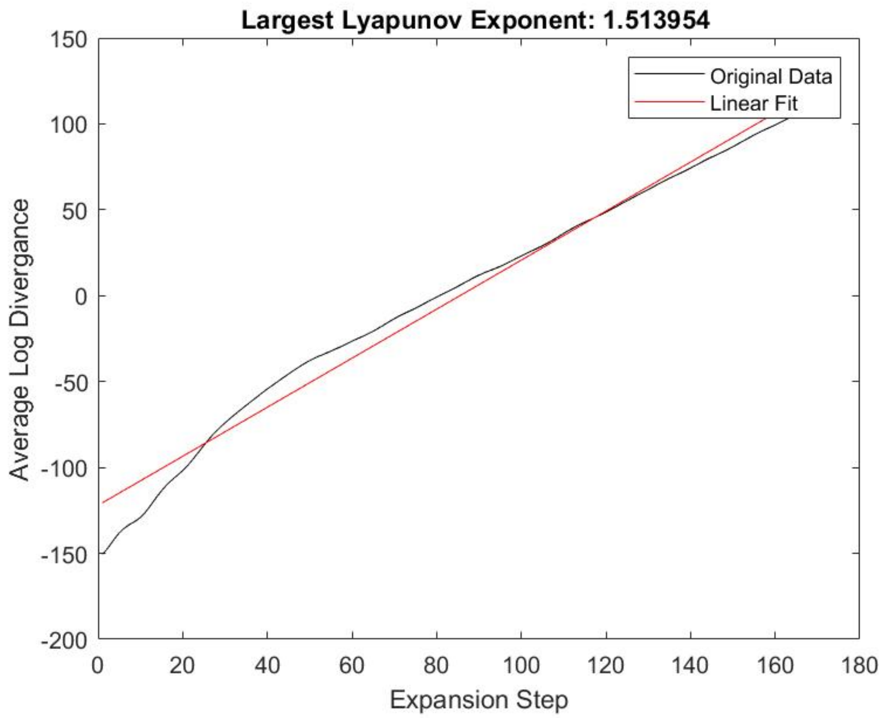

Figure 4.2. Plot of Average Log Divergence versus Expansion Step for the Lorenz attractor for N = 5000.

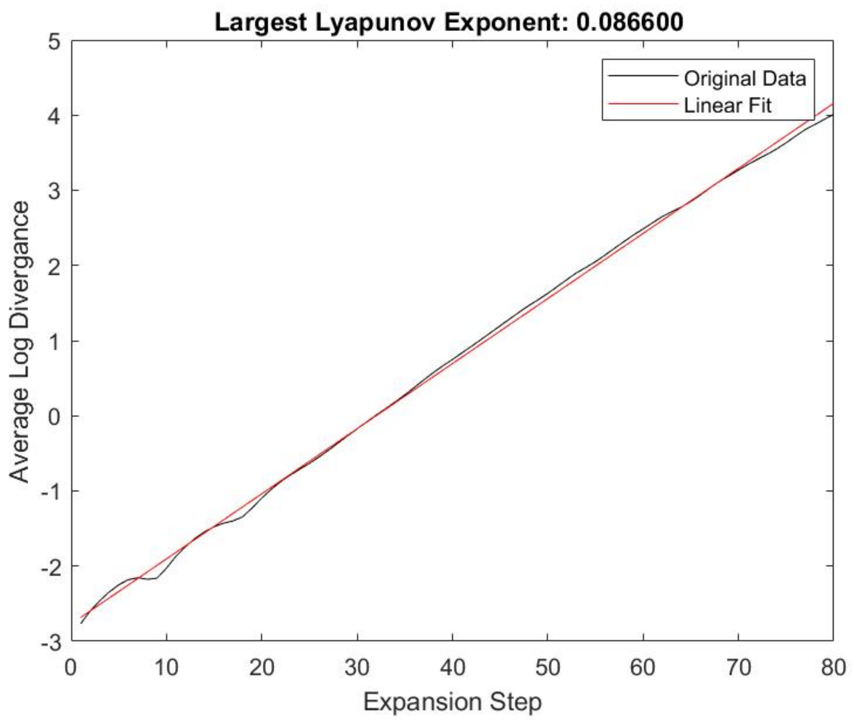

Figure 4.3. Plot of Average Log Divergence versus Expansion Step for the Rössler system for N = 2000.

4.3 Additive noise

In this section, we decided to add noise to our experimental data to see how it will affect the results of these two algorithms. As we mentioned before in 1.1, one of the important problems in dynamical systems is to determinate deterministic chaos from noise. In reality, obtaining data with measurement, without noise is almost unachievable. Removing the noise, while leaving the signal intact leads us to a better ability to detect the chaos. However, the underlying signal could have some frequency content in the stopband or the filter may substantially alter the phase in the passband [7].

We will add a signal-to-noise ratio (SNR). The SNR is defined as the ratio of the power signal to the noise power (pure noise signal, background noise). If SNR is lower than 10, we can consider it to be a high noise. Moderate noise is between 100 and 1000 and SNR greater than 1000 a low noise.

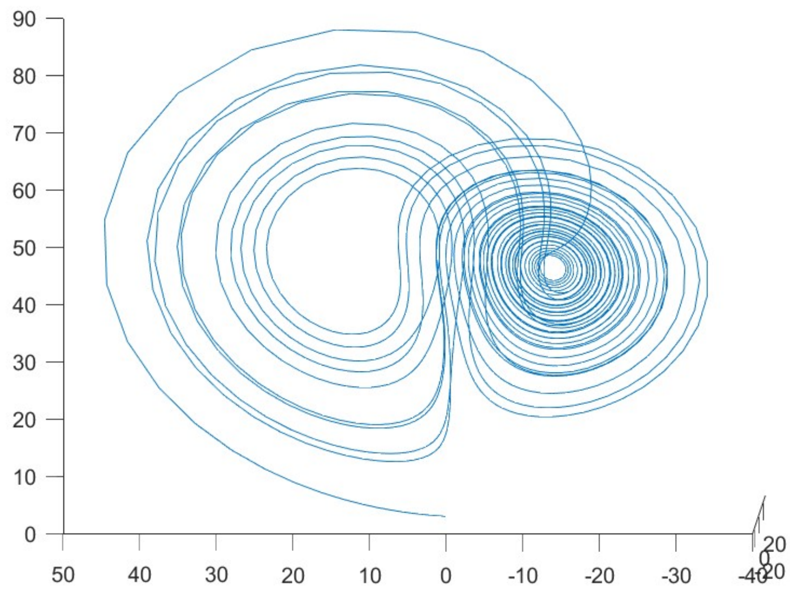

Figure 4.4. Lorenz attractor with additive noise SNR = 1000.

Our results for Wolf's algorithm (see results in Table 4.4) show us the worse results for data with high noise. In our case, Rosenstein's algorithm performed better with noise.

System	N	m	SNR	Callculated λ_1	% error
Lorenz	5000	3	1	8.848	489.8
			10	3.102	106.8
			100	2.147	43.1
			1000	2.146	43.0
			10000	2.141	42.7
Rössler	2000	3	1	2.1689	2308.8
			10	0.1215	35.0
			100	0.10818	20.2
			1000	0.10817	20.1
			10000	0.107	18.8

Table 4.4. Numerical results of Wolf's method with additive noise.

For Rosenstein's algorithm (see results in Table 4.5) we can observe that even with additive noise, we can expect acceptable results for λ_1, excluding the extremely high noise. For data with low noise, the error was smaller then $\pm 10\%$ for both systems. The worst results were for the high noise levels, as mentioned above.

It seems that we cannot estimate the largest Lyapunov exponent in the environments with high-noise, however presence of the positive slope gives us qualitative confirmation of a positive exponent and with this, confirmation of chaos.

System	N	τ	m	SNR	Callculated λ_1	% error
Lorenz	5000	10	3	1	0.536	-64.2
				10	1.196	-22.06
				100	1.374	-8.4
				1000	1.412	-5.86
				10000	1.481	-1.26
Rössler	2000	8	3	1	0.0122	-86,6
				10	0.0259	-72.2
				100	0.0845	-6.11
				1000	0.0849	-5.66
				10000	0.0871	-3.33

Table 4.5. Numerical results of Rosenstein's method with additive noise.

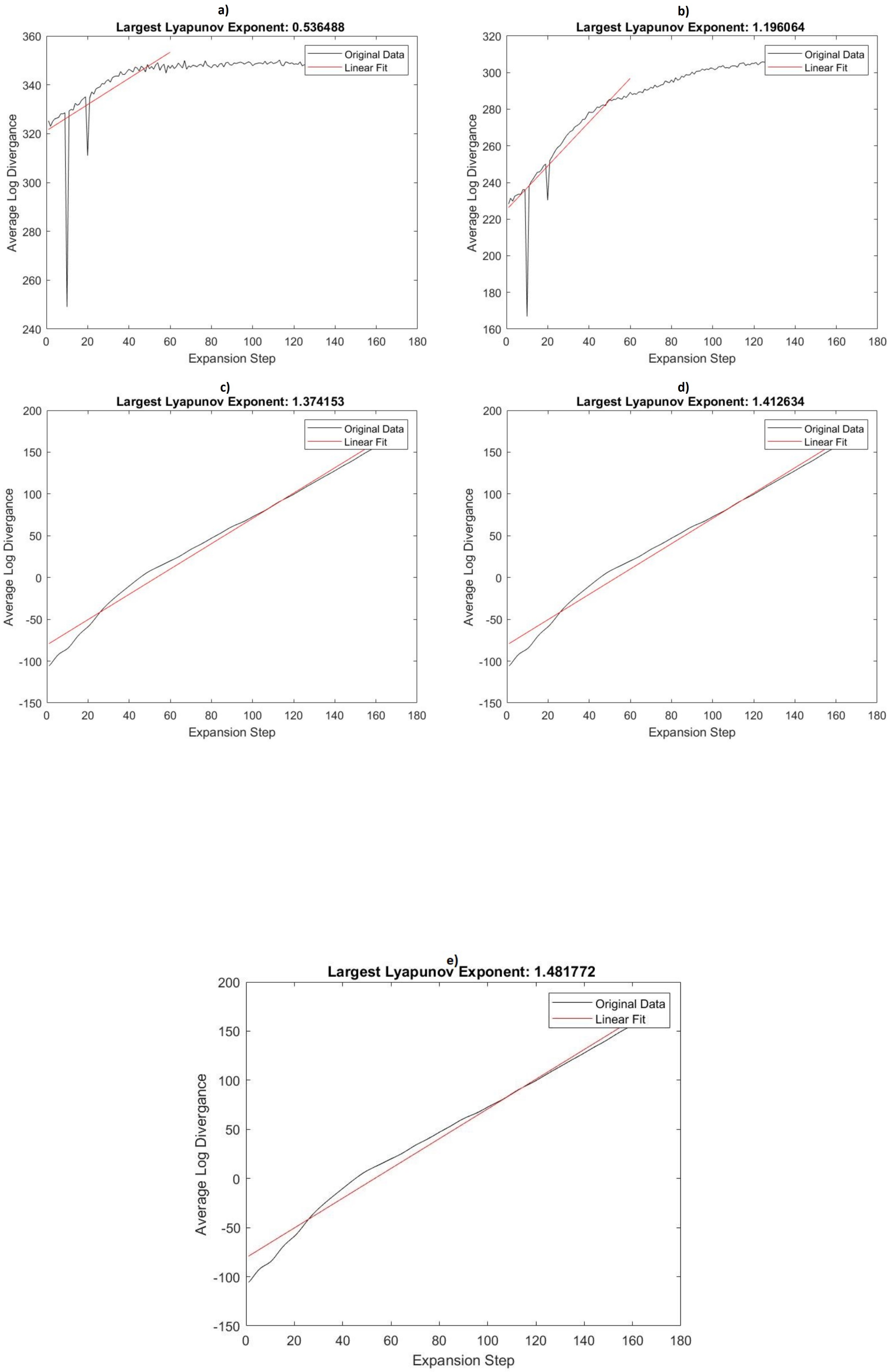

Figure 4.5. Effects of noise level for Lorenz attractor: (a) SNR = 1, b) SNR = 10, c) SNR = 100, d) SNR = 1000, e) SNR = 10000.

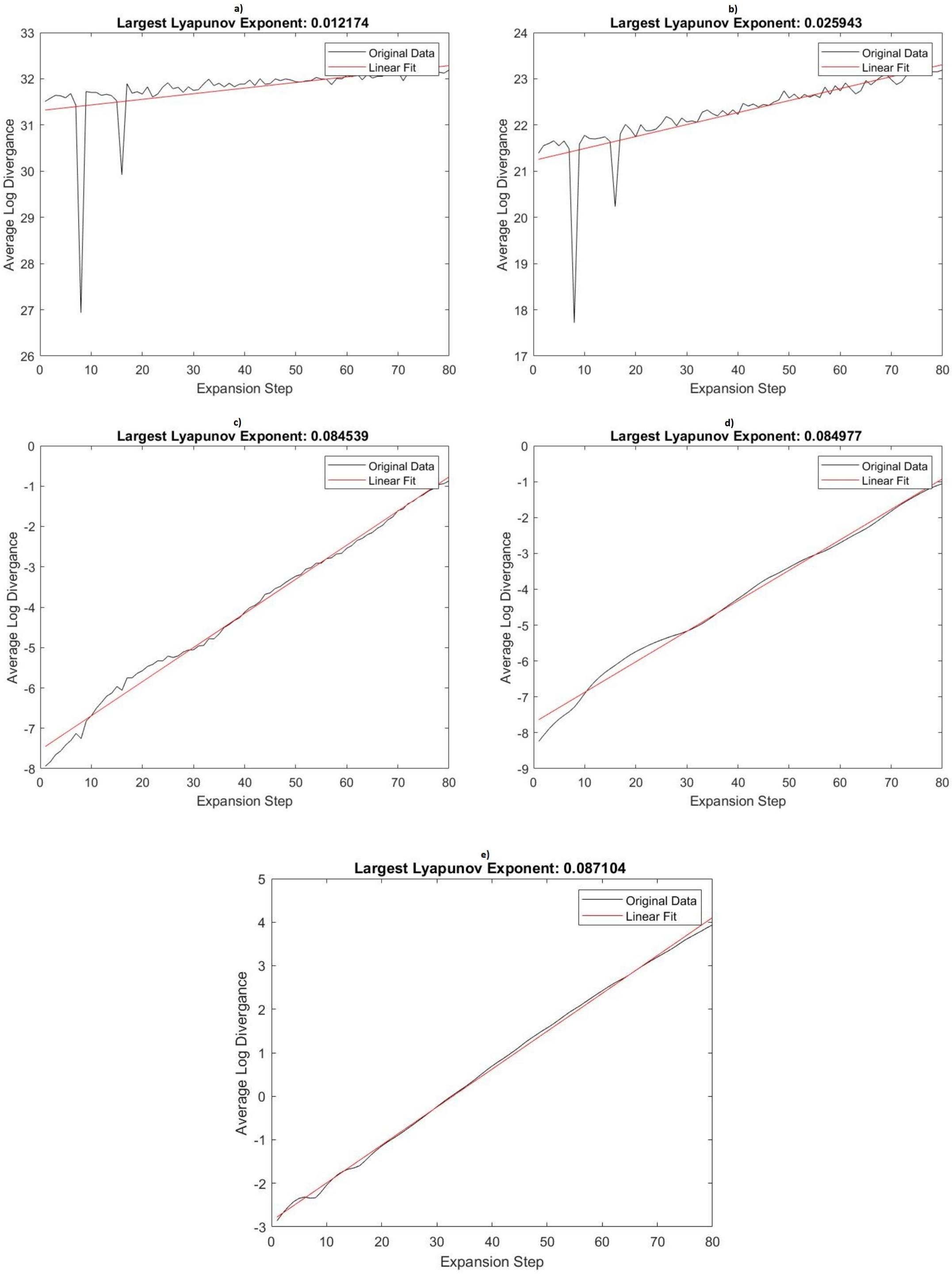

Figure 4.6. Effects of noise level for Rössler system: (a) SNR = 1, b) SNR = 10, c) SNR = 100, d) SNR = 1000, e) SNR = 10000.

Chapter 5
Summary

Lyapunov exponents are important for the characterization of an attractor of a nonlinear dynamic system and their sensitivity to initial conditions. Many of the existing methods for estimating the largest Lyapunov exponent face the difficulties of being unreliable for small data sets, computationally intensive, and relatively difficult to implement.

The aim of this Thesis was to study different methods of estimating the Lyapunov exponents from the data series and to compare the application of different methods on the data series generated by several chaotic systems using a numerical calculation in Matlab software. Then, to verify the accuracy of estimation of the Lyapunov exponents by different methods depending on the length of the analyzed data series and to add SNR to analyze how these algorithms will perform.

Comparing to Wolf's method, Rosenstain's algorithm takes advantage of all available data, which makes this method more accurate for small data sets. Wolf's method focuses is on 'fiducial' trajectory, hence it fails to take advantage of all the available data. However, based on the research [19], even if Rosenstein's algorithm takes advantage of all available data, it does not seem to be better designed to study small data sets as it was originally proposed. Also, Wolf's method requires additional computation. Both methods perform better with a larger size of data set.

Rosenstein's considers that this algorithm can be seen as a better 'predictive' model (prediction in the location of the nearest neighbours, simple delay line, etc.) that requires less computation compared to other predictive methods (e.g., polynomial mapping, neural networks).

Rosenstein's method is considered to be easy to implement. In our opinion, it was easier to implement Rosenstein's method and it seemed more stable than Wolf's.

In our opinion, the results for Wolf's algorithm could be better, based on the other papers and implementations, and our recreation was not as good as expected.

Based on our knowledge of this topic and the small research we did, we can not conclude that either algorithm is superior to the other. In Ref. [19] authors found Rosenstein's algorithm to perform better, however, there are other articles saying that it is yet premature to conclude that either algorithm is superior.

For other interesting methods for calculating Lyapunov exponents, we address the reader to Ref. [20].

Lyapunov exponents are still a popular and up-to-date topic in research, and we would like to refer readers to some of the latest papers that are interesting to read Ref. [21].

References

[1] J. Gleick. *Chaos*. 1988.
http://vattay.web.elte.hu/lectures/ChaosTheory/James%20Gleick%20-%20Chaos.%20Making%20a%20new%20science.pdf.

[2] C. E. Meador. *Numerical Calculation of Lyapunov Exponents for Three-Dimensional Systems of Ordinary Differential Equations*. 2011.
https://mds.marshall.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1105&context=etd.

[3] S. Koshy-Chenthittayil. *Determination of Chaos in Different Dynamical Systems*. 2015.
https://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=3120&context=all_theses.

[4] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*. 1985, 16 (3), 285–317. DOI 10.1016/0167-2789(85)90011-9.

[5] S. Boccaletti, J. Kurths, G. Osipov, D.L. Valladares, and C.S. Zhou. The synchronization of chaotic systems. *Physics Reports*. 2002, 366 (1-2), 1–101. DOI 10.1016/s0370-1573(02)00137-0.

[6] T. L. Carroll L. M. Pecora. Synchronization of chaotic systems. 2015, DOI 10.1063/1.4917383.

[7] M. T. Rosenstein, J. J. Collins, and C.J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*. 1993, 65 (1-2), 117–134. DOI 10.1016/0167-2789(93)90009-p.

[8] E. Lorenz. Deterministic Nonperiodic Flow. *Journal of Atmospheric Sciences*. 1963, 20 (2), DOI 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[9] O.E. Rossler. An equation for continuous chaos. *Physics Letters A*. 1976, 57 (5), 397–398. DOI 10.1016/0375-9601(76)90101-8.

[10] M. Henon. A Two-dimensional Mapping with a Strange Attractor. 1976, DOI 10.1007/BF01608556.

[11] P. S. Addison. Fractals and chaos : an illustrated course. 1997, DOI 0-7503-0400-6.

[12] J. Clinton Sprott. *Chaos and time-series analysis*. 2003.
https://sprott.physics.wisc.edu/chaostsa/.

[13] H. Haken. At least one Lyapunov exponent vanishes if the trajectory of an attractor does not contain a fixed point. *Physics Letters A*. 1983, 94 (2), 71–72. DOI 10.1016/0375-9601(83)90209-8.

[14] M. Sandri. *Numerical calculation of Lyapunov exponents*. Miller Freeman Publications, 1998.
https://venturi.soe.ucsc.edu/sites/default/files/Numerical_Calculation_of_Lyapunov_Exponents.pdf.

[15] T. Gotthans. *Advanced algorithms for the analysis of data, sequences in matlab*. Brno, Czech Republic: 2010.
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=26389.

[16] University of Colorado Department of ComputerScience. *"Wolf 's algorithm for computing Lyapunov exponents from data"*.
https://home.cs.colorado.edu/~lizb/chaos/wolf-notes.pdf.

[17] T. Kroulíková. *Runge–Kutta methods*. Brno, Czech Republic: 2018.
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=174714.

[18] D. Stewart K. Atkinson, W. Han. *Numerical Solution of Ordinary Differential Equations*. 2008.
https://homepage.divms.uiowa.edu/~atkinson/papers/NAODE_Book.pdf.

[19] N. Stergiou F. Cignetti, L. M. Decker. Sensitivity of the Wolf's and Rosenstein's Algorithms to Evaluate Local Dynamic Stability from Small Gait Data Sets. 2011, DOI 10.1007/s10439-011-0474-3.

[20] R. Brown, P. Bryant, and H. D. I. Abarbanel. Computing the Lyapunov spectrum of a dynamical system from an observed time series. *Physical Review A*. 1991, 43 (6), 2787–2806. DOI 10.1103/physreva.43.2787.

[21] A. Krakovska, S. Pocos, K. Mojzisova, I. Beckova, and J. X. Gubas. State space reconstruction techniques and the accuracy of prediction. *Communications in Nonlinear Science and Numerical Simulation*. 2022, 111 106422. DOI 10.1016/j.cnsns.2022.106422.

Appendix A
Enclosure List

CD with source codes in MATLAB:

- Lorenz attractor
- Rossler system
- Henon map
- Rabinovich-Fabrikan system
- Rosenstein method
- Wolf method