[redacted]

UNIVERSITY OF LEEDS

You must sign this (digitally signing with your name is acceptable) and include it with each piece of work you submit.

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the UK) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.

I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Student Signature	[redacted]	Date	22-03-23
Student Name	[redacted]	Student Number	[redacted]

[redacted]

SEPARATING CHAOS FROM NOISE

NAME: [redacted]

STUDENT NUMBER: [redacted]

MODULE CODE: MATH3001

MODULE TITLE: Project in Mathematics

TITLE OF PROJECT: Separating chaos from noise

SUPERVISOR: [redacted]

DATE OF SUBMISSION: 23-03-23

[redacted]

SEPARATING CHAOS FROM NOISE

Table of Contents

Ethical Statement:

This work has been carried out appreciating appropriate ethical considerations. The group work, specifically the presentation, was completed successfully and all members acted respectfully. All information that is not my own has been referenced at the end of this report. The research included in this report is based off of the literature surrounding the topic and no in-person data has been recorded, all has been simulated by myself.

Understanding the bigger picture regarding the uses of chaos, benefits of research in this field have led to for example, developments in weather prediction software, however; we must also recognise that the systems considered, however deterministic they are in principle, are still unpredictable. There can be major implications if they provide inaccurate information – for example safety concerns if weather has predicted to be calm yet actually evolves to be inclement. Appreciating these issues as a whole, I believe the execution of this project and its applications have been conducted with the correct sensitivity and do not provide any concerning ethical risk, with any ethical issues considered above.

[redacted]

Direction and aims of the project:

To provide context for the project, I will initially introduce the concept of a chaotic system and the core equations that this research is centred on. Additionally, I will consider what this means in the context of determinism and the implications of non-linearity on predictability. To give more clarity of the real-world applications of chaos theory and examples of such in nature, I aim to provide more examples of chaotic systems and their uses.

I will then begin my analysis by numerically solving the Lorenz system of equations to obtain time series solutions for the system. I will solve these in two different ways, exemplifying the benefits and drawbacks of both methods of numerically solving non-linear ODEs. From this, I will mimic the Lorenz attractor, which I will introduce later; and is analogous with this chaotic system. I also aim to describe the locations of the fixed points of the system and assess their stability. Histograms for the time series will also be plotted to illustrate the nature of the time series and how they are distributed. I aim to explore the parameters of the equations in more detail, elucidating for which values the system behaves in a chaotic way and back this with quantitative proof.

Following on from this I will introduce Brownian noise and its discovery, effects, and applications. I will illustrate Brownian motion in the context of a stochastic differential equation and solve it using the Euler-Marayama scheme; a modified algorithm for solving such differential equations with a randomly added element. The errors associated with each numerical method will also be addressed throughout this report. Now understanding the nature of noise within a system, I will move on to investigate its effects when combined with the Lorenz system of non-linear equations.

Using this appreciation for noise's influence on dynamical systems, I aim to establish the nature of both the noise-like dynamics generated organically from the Lorenz system as a consequence of the non-linear equations it is defined by, and to investigate how much added noise I can introduce to the system whilst still maintaining its core properties. By these properties, I mean whether the shape and characteristics of the power spectral density; a concept I will present later, are still recognisable as I increase the amplitude of the noise.

Succinctly, this project explores the dynamics and behaviour of a non-linear systems of differential equations in 3-dimensional space and illustrates the seemingly random manner in which a simple chaotic system can evolve. An underpinning idea throughout this investigation is that characterising a system by its assumed long-term statistical properties instead of its exact dynamics is an extremely powerful mechanism for describing chaotic systems, and once the system is in this state it can be analysed. In this project I want to both explore the properties of a chaotic system and also uncover how adding different amplitudes of noise can affect it.

[redacted]

Introduction

What is chaos?

When we consider linear differential equations, inputting the same initial conditions into the system will consistently reproduce the same output, and small perturbations from these initial conditions will reflect in the behaviour of this system deviating slightly. Once the system involves terms containing the dependent variable being multiplied together, the system of differential equations is then deemed non-linear, with some of these systems having potential to demonstrate dynamics far from the predictable outcomes of solving linear differential equations. In theory, when simulating the effects of slight adjustments to the initial conditions used to govern the system, such effects should be easily determined, since objectively the system the non-linear equations are describing has a solution, and we know exactly what the initial conditions are. However, using these non-linear equations in real-world applications and experimentally, the unpredictability of the systems comes to light, where repeating investigations with exactly the same initial conditions to infinite accuracy is impossible and thus anticipating the exact behaviour of the system is also impossible.

These seemingly random dynamics give rise to what are understood to be chaotic systems, where extreme sensitivity to initial conditions is displayed and the outcomes are unpredictable (Lorenz, 1963). This chaotic behaviour is only seen involving equations containing non-linearities. Firstly, to appreciate why I will describe such systems as unpredictable even though they are completely solvable, I will first introduce the concept of determinism. If a system is known to be deterministic, we can say with certainty what the output will be if we know the given input. This then raises the question of why we describe a chaotic system to be seemingly unpredictable, and the answer comes from the aforementioned fact that recording initial conditions to infinite accuracy is not achievable. Deviations in these initial conditions that are undetectable experimentally, for example a change of 1 x 10^{-9}ml in a chemical reaction displaying chaotic behaviour, might give a totally unpredictable result (Wegmann and Rössler, 1978). With chaos, changes in initial conditions are amplified so incredibly throughout the system that the determinism of the equations becomes redundant, since the true initial conditions are unknown.

Arguably one of the first people to consider chaos was Poincare, who noted that a small change in the initial conditions of a system can lead to large changes in the result after the system has evolved. (Encyclopaedia.com, 2018). In this current investigation, Edward Lorenz's study of the non-linear dynamics of weather systems will be considered extensively.

Examples of chaos

A core text for this project that has been useful is Strogatz' Nonlinear dynamics and chaos 2nd edition (2015). I find that this is lacking in some day-to-day examples of chaos and hence included below are some examples to introduce the concept that I will be discussing further.

Consider the famous Belousov—Zhabotinsky chemical reaction, which is a reaction where even when the same concentrations of reactants, temperatures and stirring techniques are used, produces a different oscillatory behaviour with each repeat. (Petrov *et al*., 1993). This chaotic oscillatory behaviour which characterises the reaction is exemplified by the fact that there are infinitely many oscillatory solutions to the reactions rate equations (Wegmann and Rössler, 1978). Wegmann and Rössler also note that if you plot the trajectories of the reaction for extremely similar initial conditions they will diverge very quickly (1978). I will revisit this idea in due course and visualise it with my simulations when the trajectories of the Lorenz system are considered.

[redacted]

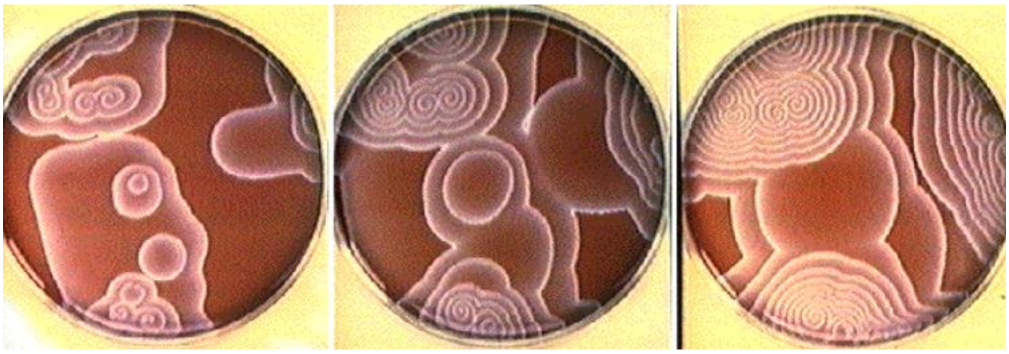


Figure 1| Taken from Gray,C. 2002. Combining the identical reactants in the same conditions can induce propagations in the gels which are completely unpredictable.

For another example, take a double pendulum, where after its release the behaviour over time is unpredictable and can't be replicated – only in simulations with zero noise and identical conditions. Releasing the pendulum from a small angle leads to linear dynamics, but with a larger angle the system becomes completely unpredictable, and its path is impossible to repeat. Even in carefully controlled experimental situations, the smallest amount of variation in the initial conditions will be massively amplified as the system evolves (Cross, 2005).

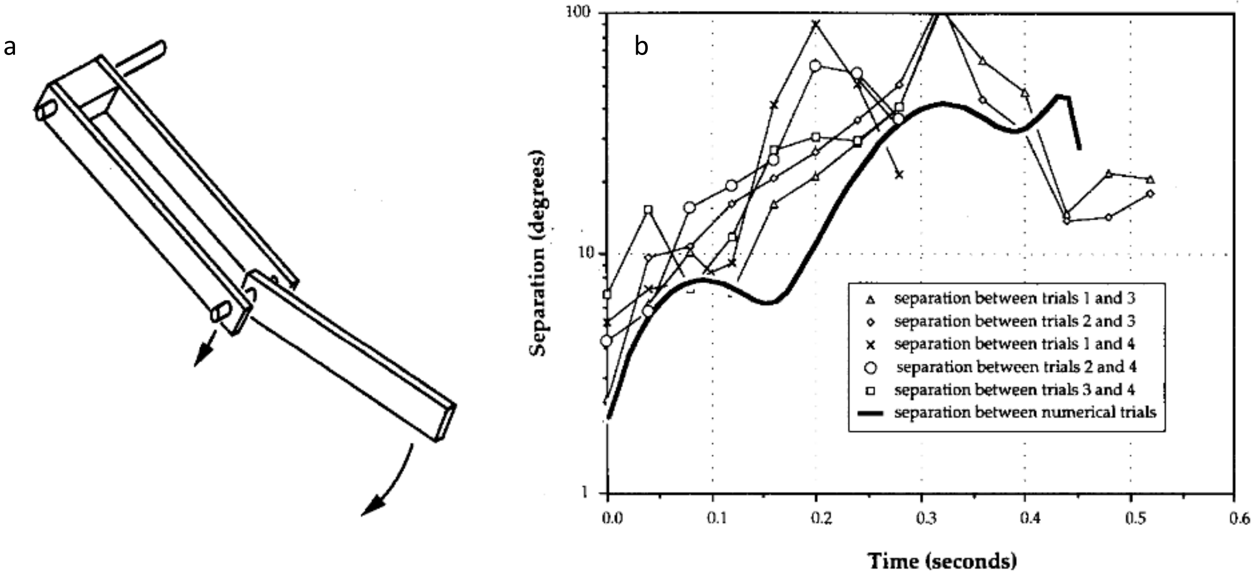


Figure 2| Taken from Shinbrot et al., 1992 – the setup and results of an experiment attempting to quantise the rate of separation of neighbouring trajectories is a simple chaotic system. **a**) The set-up for the double pendulum experiment, consisting of two pendulums in tandem, linked at the middle and both free to swing, both at the join in the centre and at the hinge at the top. **b**) Trials run by releasing the same identical pendulum from the same height with the most accuracy possible. The bold line represents the expected separation between numerical solutions of the system. As illustrated, the 4 trials diverged from the expected calculated trajectories, and all different from one another unpredictably.

In this literature, Shinbrot and colleagues stresses how the unpredictability of the double pendulum is specifically a result of the inability to release the pendulum from identical conditions with the infinitesimal accuracy required for determinism, in this case an accuracy of 1×10^{16} would be required to predict the behaviour of the pendulum for 40 generations (1992). The height of the pendulum and its angle of release were controlled by an electromagnet, yet still repeating the trajectories of the double pendulum was impossible.

In more day-to-day applications and in relation to the main system being investigated in this report, the same explanation can be applied to why we cannot predict the weather accurately to more than a week in advance of when it will happen, since even though initial conditions may seem to match those of a previous day in history that we can then infer future behaviour from, they will not be identical. Though initial conditions may be seemingly indistinguishable, when rounded to for example 5 decimal places, as the system evolves the minute differences in starting conditions will amplify to yield completely different weather. Research into weather patterns and predictions conducted by Edward Lorenz in 1963 uncovered this extreme sensitivity to starting conditions, providing an explanation for why attempting to forecast long-term weather conditions will always result in error, since the system itself and the equations modelling it are inherently unpredictable (Encyclopedia.com, 2018).

The Lorenz System

Governing Equations

Edward Lorenz discussed the idea of chaos and initial condition sensitivity whilst studying weather patterns (Lorenz, 1963). He described the dynamics of weather systems via 12 ordinary differential equations, with 12 variables such as the speed of the wind approaching the system from the west (Krishnamurthy, 2015). He then came to formulate the below three equations after he noticed that his computer simulated weather models looked completely different to each other even though initial conditions were arguably the same. Upon realising that the only difference in two runs of a simulation was that the conditions of one were rounded to 3 decimal places for one attempt, Lorenz realised that the system he was studying demonstrated a significant sensitivity to initial conditions, and this remains a key trait of chaotic systems (Krishnamurthy, 2015).

The equations illustrated below are simplified non-linear equations for fluid flow and are the 3 equations that describe the Lorenz system. The non-linearity is generated by the xy and xz terms:

$$\frac{dx}{dt} = \sigma(y - x) \qquad \frac{dy}{dt} = rx - y - xz \qquad \frac{dz}{dt} = xy - bz$$

Throughout this project, I use the same values for the parameters that Lorenz himself proposed, since it acts chaotically with certainty under these parameters. I will use for simulations σ = 10 , r = 28, b = 8/3 (Lorenz, 1963). The parameters of these equations are known as σ (Prandtl number), r (Rayleigh number) and b.

For completeness, in these equations the x, y and z variables pinpoint the location in phase space where the trajectory is located at time t in the x, y and z directions respectively. These three variables all depend on t, and thus the xz and xy terms in the equations make the system non-linear.

Solving the system numerically:

Having introduced the Lorenz system, I now wish to discuss how I have numerically solved these equations, with the non-linearity in the equations lending themselves to be considered via iterative numerical methods.

Forward Euler Method

Firstly, I explored how to numerically solve the Lorenz system of equations using the Forward Euler method. This capitalises on the understanding that with a small-time increment, the solution to a differential equation can be solved by iteratively using the previous solution, as long as initial conditions are known. It is known as the forward Euler method, since the current value for the function is used to generate the next as opposed

[redacted]
to the backward Euler method, which uses the current value of the function to implicitly find the previous one (Zeltkevic, 1998).

This method can be described by the below equation (Zeltkevic, 1998):

$$y_{n+1} = y_n + hf(y_n\,, t_n)$$

If $y = f(y,t)$, then we can find the next value for y at a point further in time via the equation, the next value that y takes after a time step of h will approximately equal that value of y plus the rate of change of y with respect to time multiplied by the time step h. The smaller the value of h, the more accurate the method will be.

Appreciating that this numerical method does not provide the same accuracy as an analytical solution would, it is useful to consider the error of this method. It is perhaps best visualised by considering the above equation being the first order truncation of the Taylor Series expansion for $y(t_n + h\,)$.

$$y_{n+1} = y(t_n + h\,) \cong y(t_n) + h\,y'|_{t_n} + O(h^2)$$

In the above equation, the $O(h^2)$ represents all terms of order h^2 and higher. In this case, using a small enough time step, terms of order h^2 are small in comparison to terms of magnitude h and thus can be ignored without losing too much accuracy for the value of y_{n+1}. Due to this, the method is known as a first-order technique and has a global error, error between the value calculated and the true value, of order h (Zeltkevic, 1998).

Using this method, I numerically solved the Lorenz system and the individual time series are displayed below, alongside the combined series in 3D:

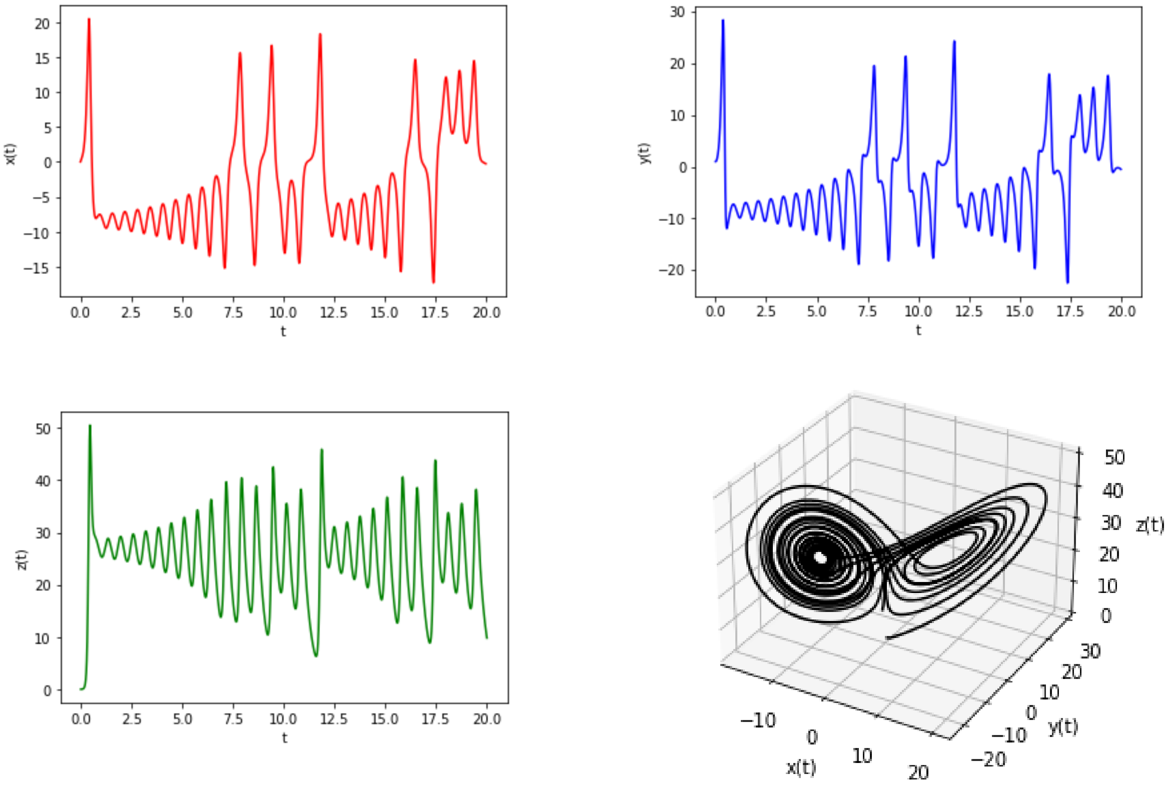


Figure 3 | These plots were generated from a starting point of (0 , 1, 0), run for 20 seconds and had a time step of 0.004 seconds.

Discussing these results, you can see that all three time series are different from one another, note the x and y time series being similar, yet the large spikes are slightly different shapes, being more abrupt for y(t). These

[redacted]
plots illustrate the chaotic nature of the Lorenz system, where the graphs show no periodicity, once a state has been visited by a system - a certain coordinate for (x , y, z) in phase space - it will never return to that.

The idea of phase space is something that I will consider in more depth shortly, when I will talk about the significance of the 3D plot for these time series solutions of the Lorenz system and what I mean when I say a specific 'state' of the system.

Runge- Kutta 4th order

Additionally to the Forward Euler Method, I used another numerical method to solve the Lorenz System; this being the Runge-Kutta 4th order. Using a similar rationale to the prior method, this again is an iterative process to solving non-linear ordinary differential equations, using the current values of x ,y and z to calculate their value after a short time step.

Following this I solved the same equations with identical initial conditions using a slightly more accurate method for solving differential equations numerically – this being the Runge-Kutta 4th order method. As the name suggests, the error of this method is lower than that of Euler's method, with error of order 4. The basis of this being extremely similar to that of Euler's method, but considering 4 values for each of x,y and z and taking roughly an average of all of them, with some estimates weighting more heavily than others.

The y_{n+1} term is defined recursively by the following formula (Romeo, 2020):

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

Note in this case, the $f(y_n , t_n)$ term has been replaced by $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$. This is a weighted average for the rate of change of y with respect to time, the slope of y. Here, values k_2 and k_3 contribute slightly more to the average value of the slope.

The values of k are defined as follows:

$$k_1 = f(y_n , t_n)$$

$$k_2 = f\left(y_n + \frac{h}{2}k_1 \ , \ t_n + \frac{h}{2}\right)$$

$$k_3 = f\left(y_n + \frac{h}{2}k_2 \ , \ t_n + \frac{h}{2}\right)$$

$$k_4 = f(y_n + hk_3 \ , t_n + h)$$

Building on Euler's methods, k_1 is used as a stepping-stone, providing an approximate value for y to be used in a more accurate estimate of y in the later values of k (Butcher, 1996). Using the value of k_1 to then evaluate the value of y at the midpoint of the timestep is a way to increase the accuracy of the estimate. The relative repetition of this to establish k_2 and k_3 generates a much more accurate value for y comparative to Euler's method alone. In 1985, C. Runge first published a paper introducing such methods, suggesting a version of the midpoint rule tailored for ordinary differential equations, and two different arrangements of the trapezium rule. These methods were then developed by K. Heun in 1900, and then W. Kutta in 1901, where W. Kutta developed a method for up to fifth order, but arguably more famously introduced the scheme for 4th order which is commonly used today, the one I have described above (Butcher, 1996).

[redacted]

From these equations, it can be seen that this method is very similar to that of the forward Euler method, but it involves calculating to a better accuracy the rate of change of y with respect to t. The error of this method is of order 4 and as the step size is decreased, the error is quartically reduced (Romeo, 2020).

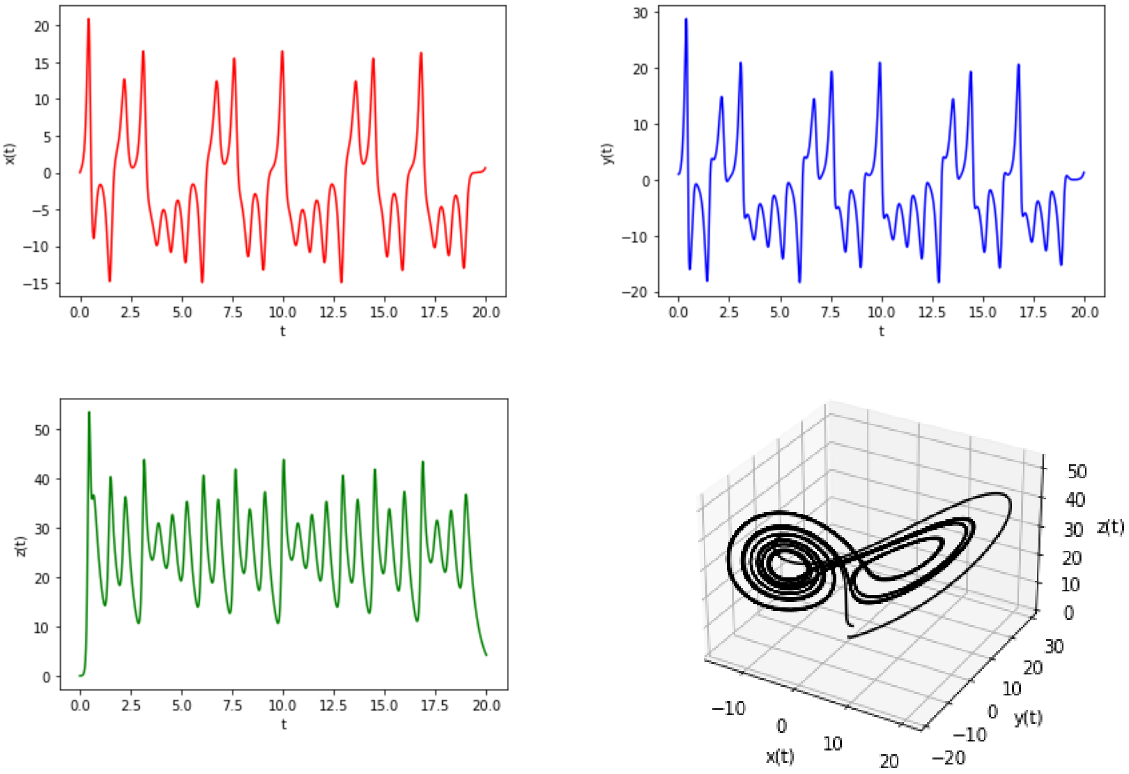


Figure 4| Plots resulting from solving the Lorenz system of equations using the Runge-Kutta 4th order method. These look similar to the outputs of the forward Euler method, and overall seem to represent the same behaviour, yet they do not illustrate the same initial transience – resulting in the 3D plot looking less filled in at the centre compared to the plot for Euler's method. The algorithm has again been run from a starting point of (0, 1, 0) with a step size of 0.004 for 20 seconds.

The System and Attractors

Now that I have discussed how the system can be solved, I would firstly like to illustrate simply the chaotic nature of the Lorenz system regarding its extreme sensitivity to initial conditions. I have generated the below plots under identical conditions, parameter values and overall run time; however, I have changed the starting conditions from (0 ,1, 0) for the left hand graph, to (0.0001, 1, 0) for the right hand graph – in essence shifting the starting point of the trajectory by 0.0001 in the positive x direction. Considering this as a bigger picture; assuming in an experiment we can measure distance we released the chaotic double pendulum from with incredible accuracy using a micrometer, where the error is 0.005mm, and we repeat the experiment twice, a difference of 0.0001 would not be detected and the repeats would seem identical. In a chaotic system, this seeming undetectable error will produce incredible differences in the output as the difference is amplified over time.

Consider my below plots, the trajectories begin to diverge after 15 seconds of runtime. Both of the graphs below are illustrating how the value of x varies with time over the course of 30 seconds, yet after the trajectories have diverged, the behaviour of x is completely different for both systems, even when all other parameters have been kept identical. The difference between the initial starting points appears arbitrary relative to the accuracy that experiments are generally held to, yet the Lorenz system illustrates that chaotic systems are very much sensitive to these changes. This also helps explain the unpredictability of

[redacted]

the Belousov—Zhabotinsky reaction, where with every caution taken to identically repeat the experiment, the timescale of the oscillations and the patterns produced are never the same.

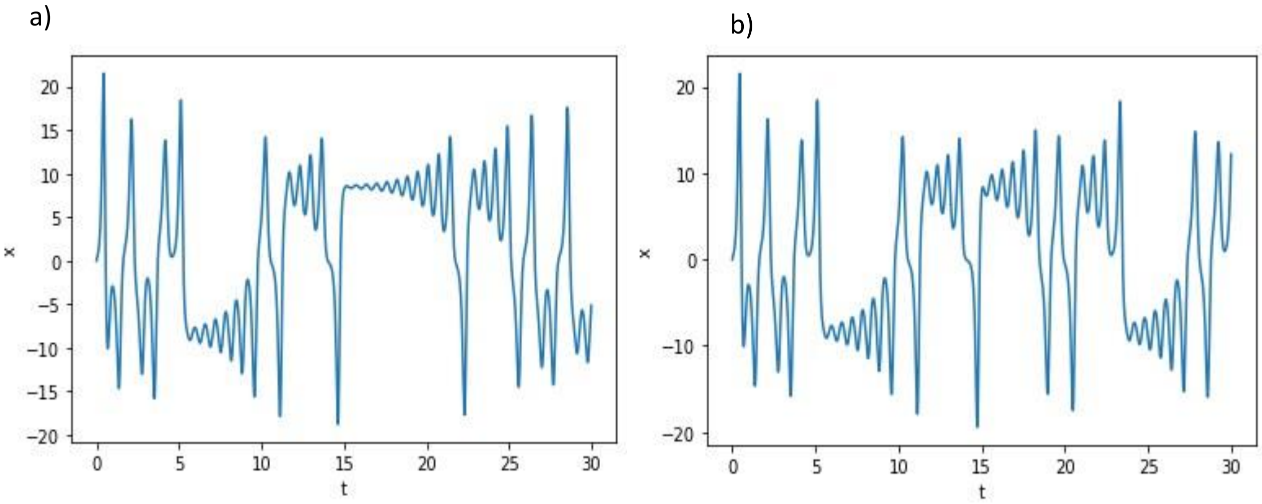


Figure 5| a) Illustrates the time series x(t) for initial conditions (0, 1, 0). **b**) Illustrates the time series x(t) for the initial conditions (0.0001, 1, 0). As you can see, the initial behaviour is very similar, but as the noise is amplified throughout the system the dynamics become completely different, observe around the t= 15 point the time series.

Now considering the x, y and z variables of the system as a whole; when plotted together they create the characteristic butterfly shape resembling the attractor that the system is autonomous with, see **fig 6a**.

An attractor is generated when a chaotic system is plotted within its phase space, this space is a region representing all of possible states of the system (any combination of x, y or z values producing a unique 3D coordinate). More specifically, Lorenz describes that a coordinate in phase space represents a possible instantaneous state of the system (1963). Any point within the phase space can theoretically be used as the initial conditions for X0, Y0 and Z0 (Strogatz, 2015). The trajectories will never cross over and two lines that begin extremely close together will draw out completely different trajectories as time elapses. Since the dynamics are completely aperiodic, the system can never return to a state it has already been in and thus the lines will never cross (Lorenz, 1963).

By observing the below plots, it may seem that the trajectories cross, but this is an illusion of the 3D projection onto the axis. I have generated a two-dimensional plot of an attractor for the Lorenz system, **fig 6b,** from simultaneously plotting six trajectories in a different colour, all starting at a randomly selected point in phase space. This maybe illustrates better that these trajectories do indeed never cross, they just seem to since they move behind each other but at different depths in 3D.

The attractor contains a 'minimal attracting set', containing the minimum number of trajectories required to attract any arbitrarily close trajectories (Il'yashenko, 1991). This attracting set means than any trajectory that starts sufficiently close to the attractor will eventually end up on it and remain on it from then on. So returning to the point that has been mentioned where any location within phase space can be used as the initial conditions to solve the equations for, regardless of starting point, the trajectories will inevitably end up in this attractor and then will never diverge out of it. Each trajectory in the system can potentially fluctuate and has the ability to do so as time progresses, yet the attractor reveals a more predictable long-term form (Encyclopedia.com, 2018). The chaos that the trajectories display, in this case the unpredictability of initial conditions in the real world leading to seeming random weather patterns in the long term, has finite boundaries; the trajectory cannot randomly take any value since it is constrained to the limits of the non-linear equations. However, even with these finite boundaries, the possible evolutions of the system are infinite within this; giving rise to order without predictability, where the system generating the attractor generates many configurations which are extremely similar but never the same (Encyclopedia.com, 2018).

[redacted]

In contrast to conservative systems, the system contracts volume in phase space and is dissipative (Lorenz, 1963). It is for this reason that the flow of all of the solutions can be represented as trajectories forming an attractor. Within the attractor, arbitrarily close trajectories diverge exponentially fast from one another, exhibiting the typical characteristic of chaos, with trajectories beginning close together soon behaving extremely differently (Lorenz, 1963).

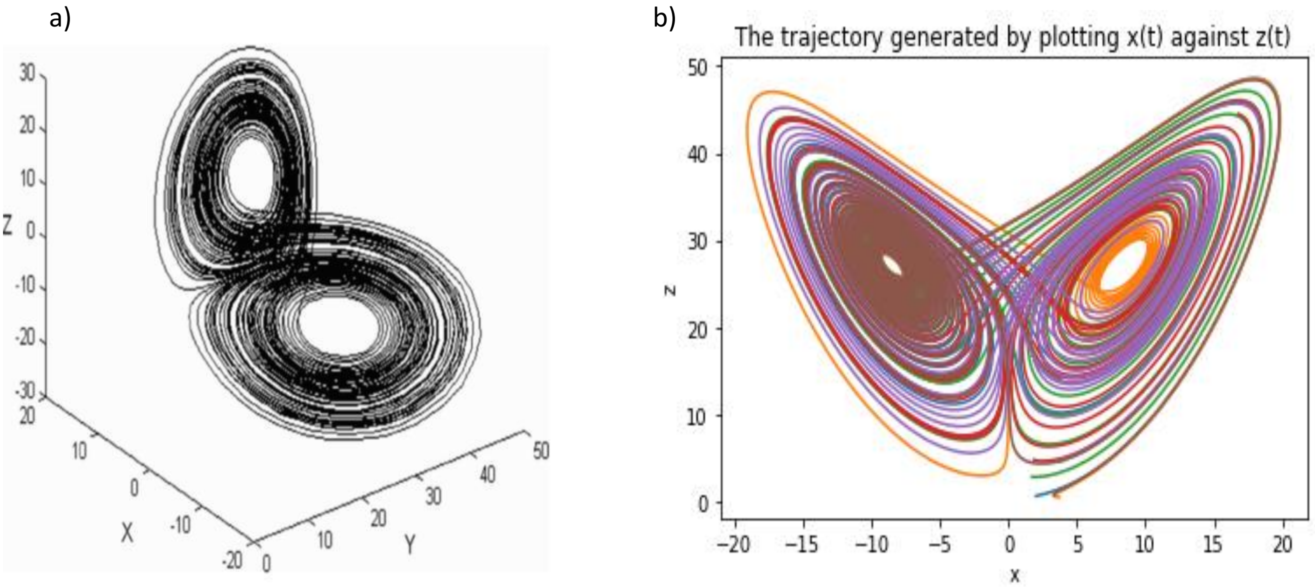


Figure 6 | a) Taken from Wu and Shen, 2009. The chaotic attractor of the Lorenz system. **b)** An attractor that I have generated by plotting multiple trajectories, randomly selecting a point in the phase space and beginning the simulation. Each of x, y, z in the starting coordinate can take a randomly generated value between 0 and 2. Each trajectory has its own colour to illustrate the nature of the attractor. Regardless of the starting point in phase space, the trajectory soon becomes similar to the others. I have plotted this in 2D since it resembles the butterfly attractor which is analogous with chaos. The attractor has been generated by solving the Lorenz system numerically via the Forward Euler method for 20 seconds of run time and a step size of 0.001.

The Lorenz attractor is known as a strange attractor, since it displays both sensitivity to initial conditions and the specific position of the system is never fully known, since two neighbouring trajectories diverge exponentially quickly from each other and never intersect, even though they have once been arbitrarily close to each other (Strogatz, 2015).

The Fixed Points of the Lorenz System

Looking at the attractors in **fig 6**, it can be seen that there are fixed points in the system, for example the spirals in the wings of the 'butterfly'. Now we will look at these in more detail, the equations describing the attractor included below for reference:

$$\frac{dx}{dt} = \sigma(y - x) \qquad \frac{dy}{dt} = rx - y - xz \qquad \frac{dz}{dt} = xy - bz$$

Firstly, a fixed point is defined to be a point in which the values for $\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}$ are 0. In this section I will use the notation x' to refer to $\frac{dx}{dt}$ and similarly for y' and z'. To calculate the fixed points of the Lorenz system, we can set (x' ,y', z') = 0 and solve for this. This is done in detail in **appendix F**.

[redacted]

It can be shown that the fixed points of the Lorenz system are as follows:

$$\boldsymbol{P_1} = (0,0,0), \quad \boldsymbol{P_2} = \left(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1\right), \quad \boldsymbol{P_3} = \left(-\sqrt{b(r-1)}, -\sqrt{b(r-1)}, r-1\right)$$

Where $\boldsymbol{P_1}$ exists for all values of r, and $\boldsymbol{P_2}, \boldsymbol{P_3}$ exist for $r \geq 1$ – where they come into existence at $r = 1$ but are coalesced with the origin (Strogatz, 2015).

The points $(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1)$ Lorenz in his 1963 paper called C+ and C-. As r tends to 1 from above, these fixed points will coalesce with the origin via a pitchfork bifurcation (Strogatz, 2015). This bifurcation will be illustrated shortly.

Stability of the fixed points

To gain a better understanding of the Lorenz system, it will now be useful to discuss not only the existence of these points, but their nature and stability. A fixed point can be classified as one of two categories, stable or unstable (Strogatz, 2015). The point being stable meaning a trajectory starting with initial conditions near the point will remain near the point from then on, an unstable point meaning a trajectory starting near this point will diverge from it. The classification of each point can be done via linear stability analysis, where any non-linear terms in the equations are removed and only the linear elements considered, so in this case the equations to look at are now:

$$x' = \sigma(y-x) \qquad y' = rx - y \qquad z' = -bz$$

Considering this definition of a stable fixed point, it follows that if we add a small increment ε onto the fixed point (0,0,0), such that it becomes (0+ ε, 0+ ε, 0+ ε) for small ε – this trajectory should now remain close to (0, 0, 0) as time evolves and not diverge from it. We can determine by linear stability analysis whether this point is stable or not and whether this trajectory returns to the origin after the introduction of a small perturbation.

Similarly to the derivation of the location of the fixed points, the key results regarding them will be discussed here and the main calculations of them are included in **appendix F and G**.

For r < 1 and the origin

Considering the linearised equations, the z(t) component at the origin can be considered stable since z' is decoupled so z(t) will tend to zero. Representing the linearised system including the small perturbations as the below matrix:

$$\begin{pmatrix}\eta' \\ \vartheta'\end{pmatrix} = \begin{pmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}\end{pmatrix}\begin{pmatrix}\eta \\ \vartheta\end{pmatrix} \text{ with } \eta(t) = x(t) - x^\circ \quad \text{and} \quad \vartheta(t) = y(t) - y^\circ$$

We can note that the Jacobian of the vector field at the fixed point in the x,y plane at (0,0,0) is present in the above equation.

From the literature, it has been identified that certain quantities of the Jacobian matrix of a system can give an indication of the type of fixed point that is being represented. In this case for a 2-dimensional linear stability analysis, and as discussed in detail in the appendices, the trace, determinant and the quantity $\mathrm{Trace}^2 - 4\mathrm{Det}(\mathbf{J})$ will help determine the nature of the fixed point (0,0,0) (Strogatz, 2015).

Noting that $f(x,y) = x'$ and $g(x,y) = y'$, the Jacobian matrix is given by:

$$\boldsymbol{J} = \begin{pmatrix}-\sigma & \sigma \\ r & -1\end{pmatrix}$$

Since the trace of this matrix is negative, but the determinant and the value of $\mathrm{Trace}^2 - 4\mathrm{Det}(\mathbf{J})$ are both positive, we can conclude that the origin is a stable node for r < 1 (Strogatz, 2015).

[redacted]

For r >1 and the origin

If the determinant of **J** is negative then we can immediately conclude that the point is a saddle point, stable in 2 directions and unstable in the third. For r >1, this is the case and hence the origin is a saddle for r >1.

For r > 1 and C+ / C-

When r > 1, we also have the existence of the symmetrical pair of fixed points at C+ and C-. Conducting an analysis similar to that of the calculation in 2D for the fixed point at the origin, we can again derive the Jacobian matrix for the 3-dimensional system at the fixed points C+ and C-. Also by a method reflecting the 2-dimensional linear stability analysis, the Jacobian matrix can assist us with the classification of these fixed points as stable or unstable. In this case however, it is the eigenvalues of this matrix that can help, with negative real parts of the eigenvalues indicating stability, and positive real parts of the eigenvalues indicating instability (Clack, 2006).

We can derive the characteristic equation describing the eigenvalues from the Jacobian matrix for the 3-dimensional system and arrive at the below:

$$\lambda^3 + \lambda^2(1 + b + \sigma) + \lambda b(r + \sigma) + 2\sigma b(r - 1) = 0$$

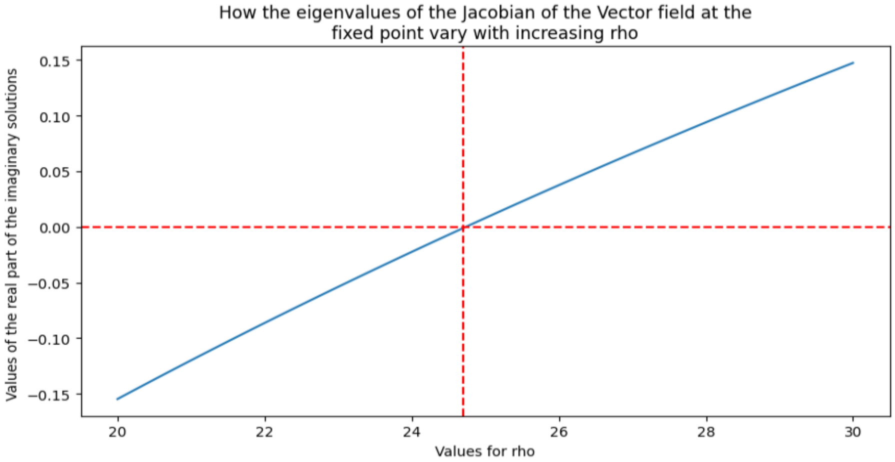
Here I will deviate from the literature in the derivation of the values of r for which the fixed points are stable. Many sources calculate the value of r at which the system becomes unstable via the use of the below result:

$$R_h = \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1}$$

So, using the predefined values of $\sigma = 10$ and $b = 8/3$ for this investigation, we arrive at a value of $R_h = \frac{470}{19} = 27.74\ (2dp)$. However, to stick to a method similar to the 2-dimensional stability analysis, I will now attempt to use the characteristic equation to find the eigenvalues and comment on the stability that way.

I highlight here that there is a lack of evidence in the literature regarding directly solving for the eigenvalues for the Lorenz system. However, to stick to the theme of this investigation – and utilise numerical methods – I believe this is an appropriate way to assess the stability of the fixed points and provide a refreshing way to calculate it. I solved the characteristic equation numerically using code included in **appendix G**, and plotted the real parts of the eigenvalues relating to C+ (since C- is identical just has an opposite signed imaginary part). As illustrated below, this numerical method pinpoints the value at which the eigenvalues become positive and hence the fixed points are stable up until r = 24.7 – agreeing with the result using the prescribed formulas found in the literature.

Hence, the fixed points C+ and C- are stable for $1 < r < 24.74$ and unstable for $r > 24.74$.



How the eigenvalues of the Jacobian of the Vector field at the fixed point vary with increasing rho

Figure 7| Numerically determining the value where the C+ and C- become unstable. The literature indicates that these points become unstable when the eigenvalues of the Jacobian of the vector field become positive. Here it has been pinpointed where these solutions become positive, and this agrees with the literature which obtains this value analytically.

[redacted]

The three fixed points can be summarised by the below bifurcation diagram, illustrating both when the points come into existence and in what regions that are stable and unstable.

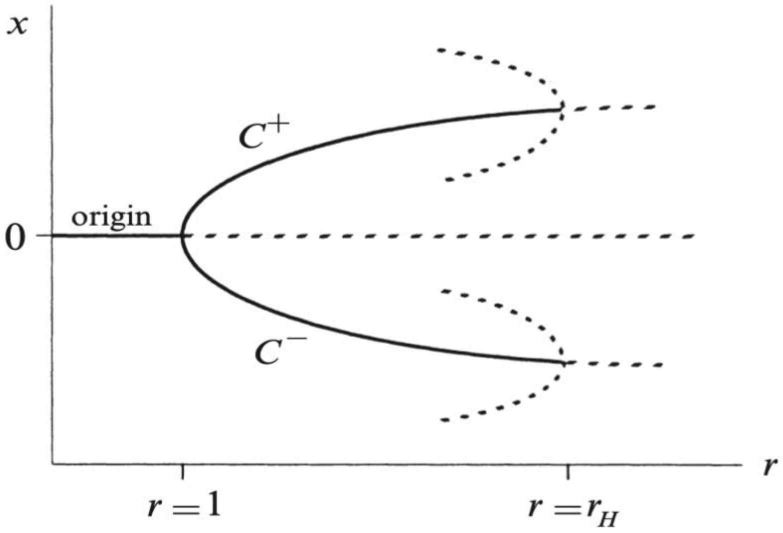


Figure 8| Taken from Strogatz, 2015. This diagram exemplifies the three fixed points discussed above. The bold lines illustrate where the fixed points are stable, and dotted lines show when they become unstable. In this case, rH =24.74. This value is true for both C+ and C-. The point at which the two symmetric fixed points coalesce with the origin is indicated at a value of r = 1, this is the pitchfork bifurcation since the origin becomes an unstable saddle for r>1 and the two symmetric fixed points emerge in the shape of a pitchfork. The two dotted arcs located at r= rh show where the C+ and C- fixed points become unstable.

The Lorenz System and Parameter Influence

I emphasise again the need for appropriate parameters since not all systems that display chaotic behaviour do so for every value of all the parameters it is defined by, for example as discussed above no attractor is present for the Lorenz system for $r < 1$. I illustrate this with the time series that I have inserted below, which clearly not demonstrating unpredictable behaviour as time progresses.

The below plots are the solutions to the Lorenz system of equations. They show the values of x ,y, z (which pinpoint the location of the trajectory in phase space) for a time between 0 and 25 seconds. In these plots the only edit made has been to the Rayleigh number r, being changed to 10 instead of 28 in the simulations.

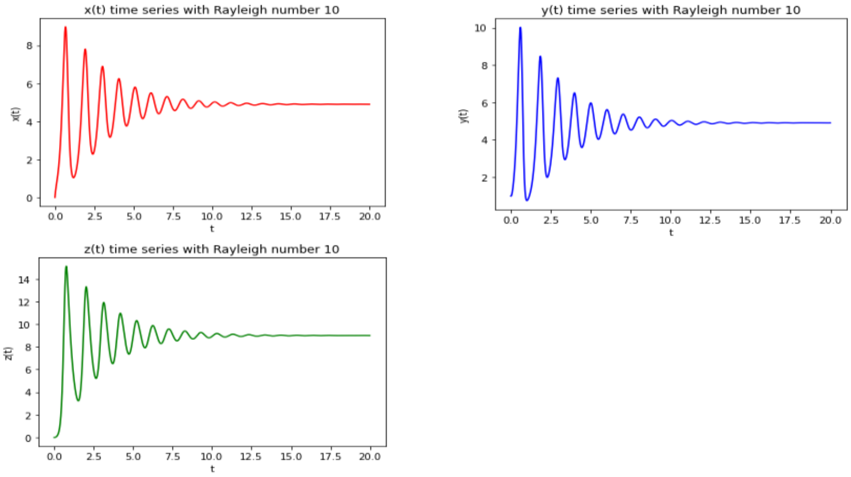


Figure 9 | Time series for the time series x(t), y(t) and z(t) solutions to the Lorenz system of non- linear differential equations using the same parameters except a Rayleigh number of 10 instead of 28.

[redacted]

The series illustrate the lack of chaos exhibited by the system using this different parameter value. The system behaves in a more predictable manner, where any behaviour dampens out by 15 seconds, in terms of phase space, a particle on that trajectory reaches the coordinates approximately (5,5,9) and remains there thereafter.

The origin as a globally stable fixed point for r <1 – **the Lyapunov exponent**

The Lyapunov exponent can be used to illustrate a chaotic system and its sensitivity to initial conditions. In the above example, it has been exemplified how changing the parameters of the Lorenz system can inhibit it displaying chaos. I will now describe with certainty how I know that the system cannot act chaotically for a value for r less than one. This will agree with the fixed point analysis that the origin is a stable node for $r < 1$, where any trajectory will not diverge from the origin.

Whilst studying the literature, Shinbrot and colleagues' work on the double pendulum highlighted a potential indicator of chaos to be the sign of the Lyapunov exponent value of the system (1992). Lyapunov exponents are quantity of a dynamical system; such as the Lorenz system that is currently being considered. The values that the exponents take characterise some quantities of the system; with negative values indicating the system is dissipative, zero values indicating conservative systems and positive ones indicating chaos, the latter positive values only indicating chaos if the dynamical system has an attractor for its behaviour. In this case, it has already been discussed that the Lorenz system has an attractor, so if the value of a Lyapunov exponent can be found to be positive, then they system is displaying chaos.

If we plotted the distance between two neighbouring trajectories as $\delta(t)$ on the vertical axis and time on the horizontal, the Lyapunov exponent can be considered to be the slope of this line. Hence the exponent can be considered a measure of the divergence between arbitrarily close trajectories, with a positive value meaning divergence and thus chaos (Strogatz, 2015).

A dynamical system also does not only have one Lyapunov exponent, instead it has one for every dimension. In this case, the phase space of the Lorenz system is 3 dimensional and thus there are 3 Lyapunov exponents associated with the system (Strogatz, 2015). One is needed to be positive to see chaos.

I will plot how the Lyapunov exponents of the system change with increasing r as evidence as to why not every value for the Lorenz parameters will facilitate chaos. We know that a value of $r < 1$ will not lead to chaos from the stability analysis, and this is exemplified by the Lyapunov Exponent plotted. Where a value of rho less than one leads to negative values for all three Lyapunov exponents and this means that locally to that fixed point (0,0,0) no chaos can be observed.

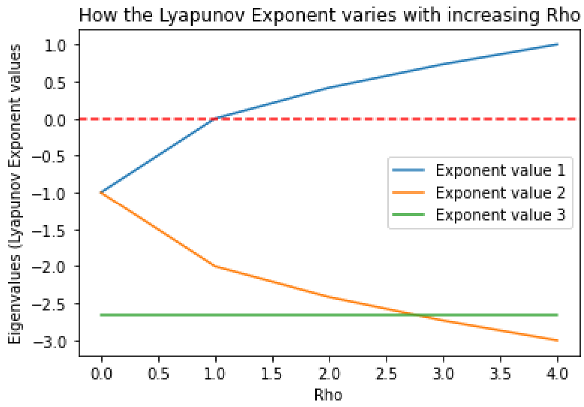


Figure 10| The values of the three Lyapunov exponents of the Lorenz system change over increasing values of r. Only one of the three values is required to be positive in order to facilitate chaos, the above exemplifies that a positive exponent value is only achieved for $r > 1$. The plot was generated by calculating the value of the Lyapunov exponents for a range of increasing values of r, there are three Lyapunov exponents for every iteration since the Lorenz system has three dimensions.

[redacted]

I have also illustrated how the fixed point (0,0,0) acts as a sink for values of rho less than one, **fig 11**, visualising the implications of what we just calculated regarding the Lyapunov exponents. All solutions that have started arbitrarily close to (0,0,0), in this case I have started off 50 trajectories at a random set of (x,y,z) coordinates, are drawn to the origin as seen and no chaos is exhibited. Contrastingly, any value of r giving a positive Lyapunov exponent will produce the analogous attractor of the Lorenz system, **fig 6b**.

The origin is globally stable for $r < 1$, meaning every trajectory will approach the origin as time tends to infinity.

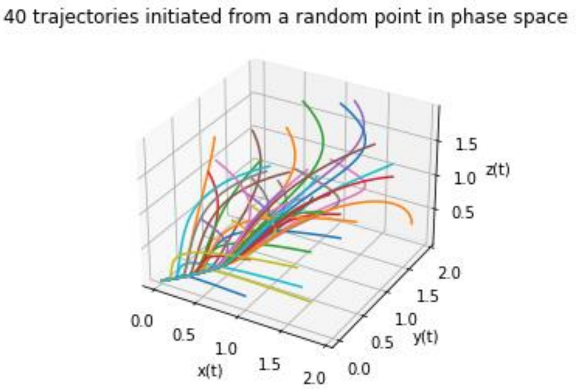


Figure 11| All 40 trajectories, regardless of their initial starting positions in phase space end up at the origin. For a value of r < 1, in this case 0.9, using the same time step and the RK4 method for solving ordinary differential equations, it is clear that the origin is acting as a sink and is globally stable for r < 1.

A note on noise induced chaos

Reflecting on how the values for the Lyapunov exponent around the fixed point (0, 0, 0) suggested that no chaos can be exhibited by the system with a value for rho less than one, I thought it would be interesting to briefly investigate how adding noise influences this dynamic. After confirming with the literature that noise can induce chaotic behaviour, I sought to exemplify that in the Lorenz system (Ellner, 2005). This phenomenon is something that could be explored further and in more depth in the future, noting Ellner and colleagues' comments that noise can have large-scale impacts on non-linear systems which are disproportionate to the amplitude of the noise, in a similar way to how tiny perturbations in initial conditions can give drastically different time series solutions to the Lorenz equations (2005).

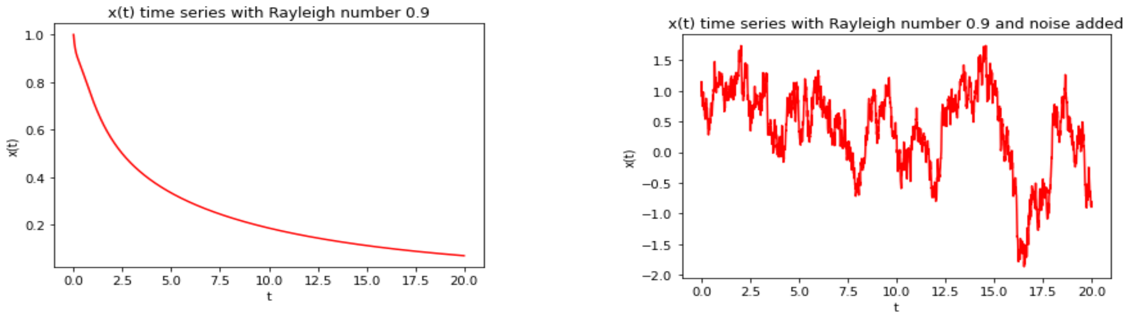


Figure 12| Adding white noise of amplitude 0.01 to the non-chaotic Lorenz system for $r = 0.9$ achieved a chaotic time series.

[redacted]

Introducing Brownian Motion:

Having modelled purely the un-interfered chaotic dynamics of the Lorenz system and having an appreciation for the initial condition sensitivity, we can now build up to consider the effects of adding noise to the system. I will first introduce the concept of adding noise to a differential equation by considering Brownian Motion, this motion being described by a stochastic differential equation – a differential equation incorporating a randomly generated term.

The discovery and evolution of Brownian Motion is described by Smith and Raghav as ‘a collective, even if disjointed, scientific enterprise’ (2022). In 1827 Robert Brown evidenced that pollen grains suspended in water were always in some form of random motion, moving as if they were vibrating (Cohen, 2016). Using a microscope, he then went on to observe this seemingly complex, non-stop motion in a range of materials, from animal tissue to glass and rocks. Since this motion occurred in both liquid and solid states (for example both in the pollen example and in a sheet of glass), it was hypothesised that the explanation for these observations was related to kinetics; yet no kinetic explanation was established for Brownian motion in the nineteenth century (Smith and Raghav, 2022).
Other possible explanations for this persistent random granular motion observed by Robert Brown included surface tension, osmosis and electrical effects; and these proposals were used by many, such as Conybeare, to counter the kinetic explanation. All of these ideas were overwhelmingly lacking in evidence, and doubts around the kinetic theory of heat throughout the nineteenth century influenced the lack of progression to an explanation for Brownian motion (Smith and Raghav, 2022). Finally, Jean Perrin established that Brownian motion was of thermal molecular origin upon the invention of the ultramicroscope, agreeing with Einstein’s 1905 paper regarding small particles suspended within a stationary liquid. The motion observed by Brown was deemed to be due to the action of the water molecules colliding with the suspended particles (Smith and Raghav, 2022).

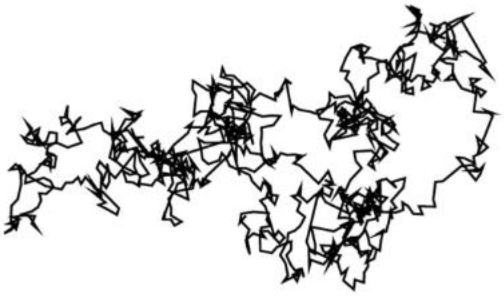


Figure 13| Taken from Cesbron,2017. An illustration of the random motion of a pollen grain observed for 1000 steps. This motion is what Brown would have observed as the pollen grain vibrating. After many years of searching for an explanation for this behaviour, and kinetic theory of heat being investigated more, the motion was deemed to be of thermal molecular origin at the beginning of the twentieth century (Smith and Raghav, 2022).

Notice in the above illustration, the pollen grain doesn’t exhibit an overall net movement in any direction, since the force it is under is acting equally in all directions and has a mean of zero. I will now model this Brownian motion below, the random collisions with water molecules being represented as noise added to a differential equation. The properties of this force, such as it having zero mean, will be considered further now as I model Brownian motion through the use of the Langevin equation.

[redacted]

Langevin Equation

Following the establishment in 1906 that the erotic motion of granules suspended in water is due to the random movement of the atoms in the fluid, Langevin then went on to show in 1908 how to represent the interaction between the particle and the fluid (Pomeau and Piasecki, 2017). Significantly, Langevin split the forces acting upon the suspended particle into two, one regarding the viscosity of the fluid, and the other representing the random force of the molecules colliding with the Brownian particle (Pomeau and Piasecki, 2017). His equation is considered Newtons second law for Brownian motion and the Langevin equation is as follows:

$$m\frac{d^2x}{dt^2} \quad = -6\pi\alpha\eta\frac{dx}{dt} + F_b(t)$$

In this equation, m represents the mass of the suspended particle, with radius α, position $x(t)$, velocity $\frac{dx}{dt}$, and fluid viscosity η. Thus the $m\frac{d^2x}{dt^2}$ term represents the total force acting on the Brownian particle at the instantaneous time t. The coefficient for velocity in the equation represents the friction that particle will experience whilst moving through the liquid, where the exact expression for this coefficient is a result of Stokes law.

Considering the right-hand term of the equation, this represents the random force that the particle experiences as a result of the collisions occurring between it and the fluid atoms. The discussed random force is the term which makes the Langevin equation a stochastic differential equation and gives the effect of background noise (Kuroiwa and Miyazaki, 2014).

Notably, since the action of the force is random, it is impossible to replicate the motion of the particle, however the statistical properties of the force can be assumed if the system is in a statistically steady state, so can be used to simulate a variable with the same statistical properties. These properties are that the force has zero-mean, $F_b\,(t_1\,) - F_b(t_2\,)$ is normally distributed and $F_b\,(t_1\,)$ and $F_b(t_2\,)$ are uncorrelated. The forces being uncorrelated is a result of the frequency of the collisions between the particle and the atoms in the liquid, with approximately 10^7 collisions occurring in 0.0001 seconds. If the force $F_b\,(t\,)$ is a result of the collisions happening instantaneously with the particle, in a small increment of time later, the particle will be being hit by a completely different set of liquid atoms, so the force will be completely different.

Since this is a differential equation, again to solve it I will implement an iterative numerical method. However, to account for the stochastic term in the Langevin equation I will use a slightly modified algorithm.

Euler – Marayama method for the Langevin Equation

In a similar way to the derivation of the Forward Euler method for solving ordinary differential equations, this method is also generated via the truncation of a Taylor series (Bayram et al., 2018). Generating a stochastic expansion up to order 1 yields the below equation which I have them implemented in my project:

$$X(t_{i+1}) = X(t_i) + f\big(X(t_i)\big)\Delta t + g(X(t_i))\Delta W_i$$

Where $X(t_0) = X0$, (Bayram et al., 2018).

I will also plot the probability distribution for the time series resulting from this solution to the Langevin equation, this will illustrate the properties of the random force $F_b\,(t\,)$. To generate this stochastic noise to

[redacted]
incorporate into the Langevin equation as $F_b(t)$, a Weiner process was used since this generates a random array of values with the same properties required.

To get a snapshot of $X(t)$ and to clearly visualise the random motion that the stochastic term is generating, the solution has only been plotted for an overall time length of 3 seconds. In order to confirm that the noise that has been added has a true mean of zero and is gaussian in nature, I let the time series evolve for a long period of time (1000 seconds with a step of 0.01) to plot the probability distribution.

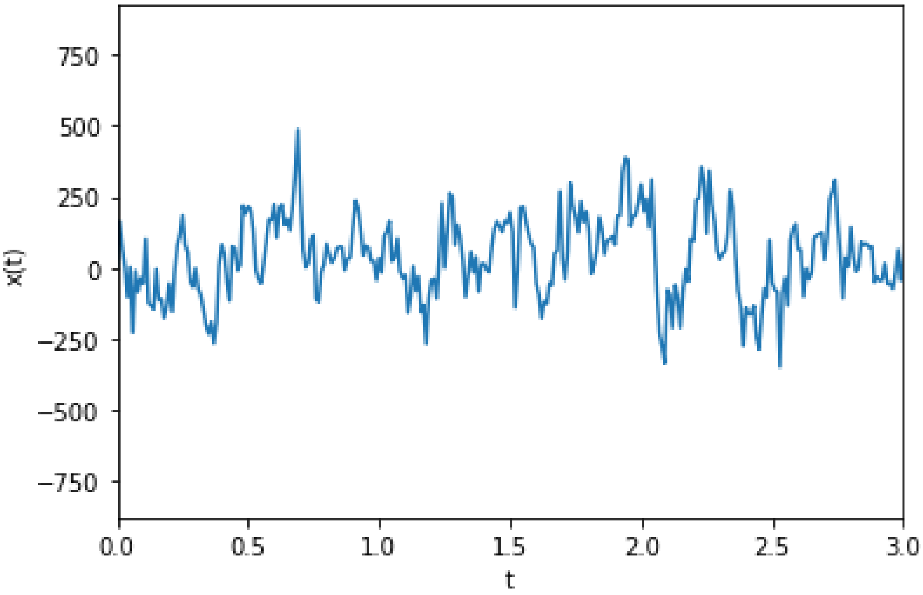


Figure 14|The output from solving the Langevin equation for a duration of 3 seconds. The value of x(t) varies immensely over the course of the three seconds, resembling what the time series would look like if the location of a pollen grain suspended in water were to be observed. Importantly, the random nature of the noise term mimicking the random nature of the interactions between the liquid atoms and the suspended granule means that the time series is unique to the certain random values of noise that have been generated for this simulation. If this were to be run again, a different output would be seen, but would have the same statistical properties such as mean.

Observing the histogram below, as expected, the probability density function for this time series is normally distributed with mean zero, as the noise added was defined to have these properties.

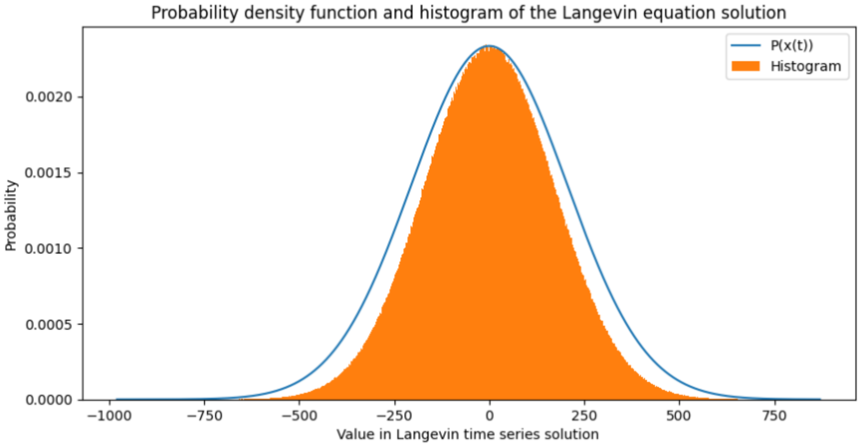


Figure 15| The probability density function and histogram for the time series output from solving the Langevin equation, it has a mean at 0 and is normally distributed as expected given that gaussian noise was added. The simulation was run for 100000 seconds in order to be representative of the properties of the random noise term, these being zero mean and normally distributed.

The histograms of the Lorenz system time series have also been plotted below and are very different to the normally distributed probability density function of the Langevin equation solution.

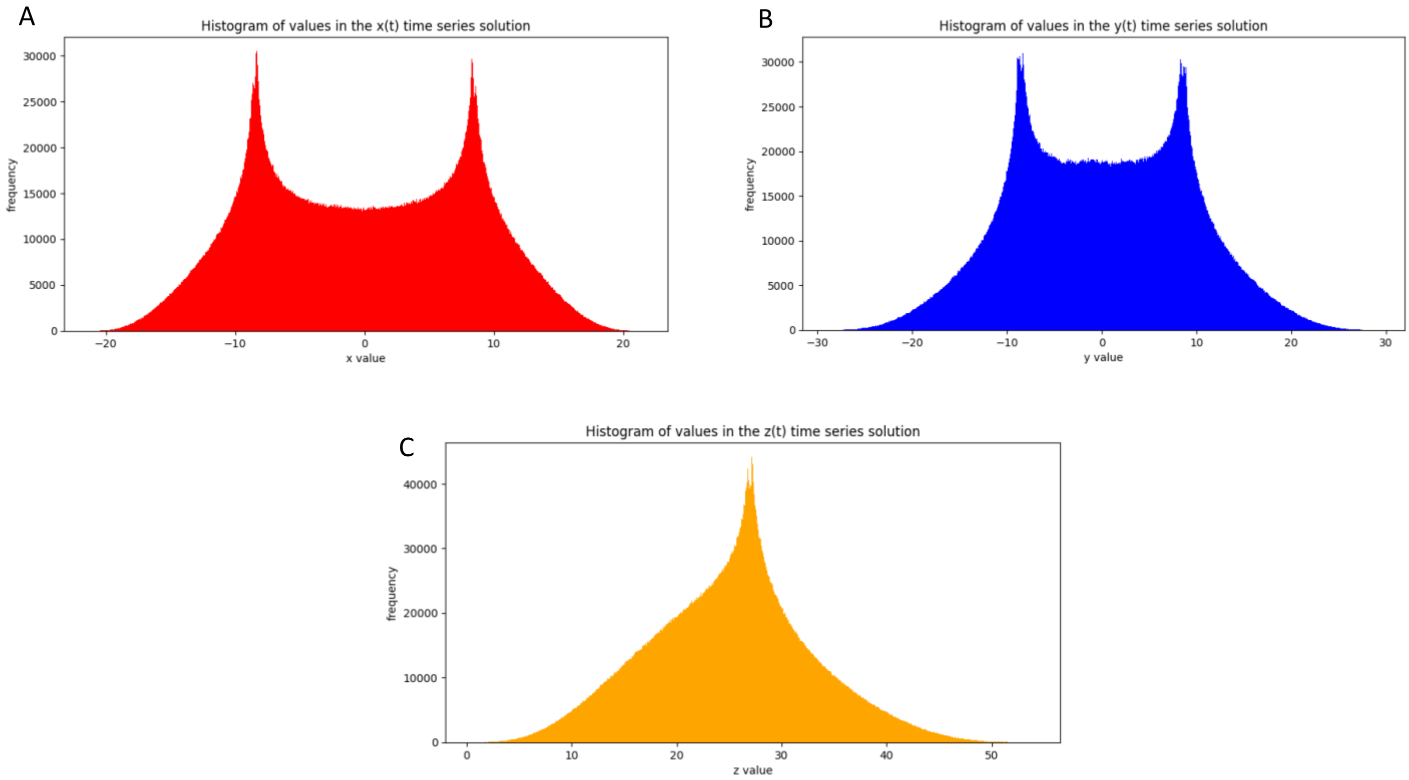


Figure 16| The histograms of the time series solutions to the Lorenz system reveal the distributions of the values as the series evolves. These plots have been generated from a runtime of 100000 seconds with a time step 0.01.

Contrastingly to the PDF of the Langevin equation, it is clear from all three plots that the values in the time series solutions are not normally distributed. There is also a stark difference between **fig 16 a,b** and **fig 16c**, with the latter being unimodal in comparison. In the context of the Lorenz system, these differences illustrate the nature of the time series solutions, with x(t) and y(t) oscillating both above and below zero, yielding two modes and an almost symmetrical histogram about 0, as illustrated in the time series of **fig 4**, whereas the z value is strictly positive so has a single mode.

Overall the rationale behind this section was to illustrate the nature of the noise that I will be adding to the Lorenz system, as well as introducing the iterative numerical method that I will use to solve the Lorenz system once I have added noise to it.

Introducing noise into the system

Now that I have discussed how adding random noise to a differential equation can influence its solution, and having exemplified the nature of this noise, I will now add noise to the previously solved Lorenz system and see how this affects it. In order to see the effects of this noise, I will add white noise to the Lorenz system and then solve it to generate the time series for x, y and z once more. This noise will be added the same way as I have demonstrated in the Langevin equation, with mean zero, normally distributed and force at two different time points uncorrelated.

I will first illustrate the effects at face value of adding this noise, observing what the time series and 3-dimensional plots look like for this system with added noise. I will then take this a step further by analysing the power spectral densities (PSD) of the original Lorenz system and the one with added noise.

[redacted]

A brief look at the effects of white noise on the Lorenz system

Initially, I have added white noise to the Lorenz system and solved it to illustrate the effects on the behaviour of the system as a whole. I have added noise with a mean of zero, variance of 1 and an amplitude of 3.

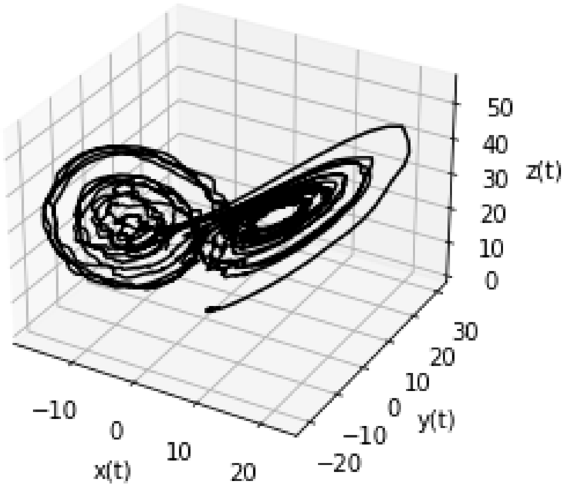


Figure 17| The 3D plot of the Lorenz system solved with added noise illustrates that the noise causes the attractor to become distorted. The overall properties of the system still seem to be present, with the shape of the graph remaining comparable to the previous plots without any noise.

However, just observing the graph isn't sufficient to infer the effects that the noise has had on the properties of the system – and this is where the notion of power spectral densities will become useful. Upon the end of this section, the effects of noise on the Lorenz system and its individual x, y and z time series will have been analysed in better detail.

Considering Power Spectral Densities

Having seen the time series for the non-interfered signal in the time domain, we can assume that it can be plotted in the frequency domain using a Fourier transform, since the signal is a composition of waves at varying frequencies. Since this signal is generated by a chaotic system, it will obviously be aperiodic and therefore I will use a Fast Fourier Transform (FFT).

Interestingly as a side note regarding aperiodicity: the aperiodicity of the system is discussed by Lorenz in his original paper, where he concludes by considering the question of whether chaos theory is applicable to the atmosphere as a whole, since if the atmosphere could be proved to be periodic in comparison, then long term weather predictions could be achieved (1963). If the atmosphere can be proven periodic by integrating for a long enough time and discovering that is has been in the same state twice, then in theory he suggests that the time scale of the weather repeats could be evaluated (Lorenz, 1963).

The FFT is an algorithm by which the discrete time Fourier transform of a signal can be calculated. This is used for aperiodic signals like the time series that we have been considering. Additionally, a PSD will be plotted instead of the simple power spectrum that the FFT returns. The use of this is that the PSD is normalised to provide a value of power per unit frequency, whereas a power spectrum reveals solely the amplitude of the wave at that frequency that is contributing to the signal. This means that the FFT is useful in scenarios where there are a few dominant frequencies in the signal, and PSDs are useful for random aperiodic signals which are generated from a wide variety of frequency components.

I will first rationalise the use of this method through the use of a much simpler example. I will transform the below cosine function into its power spectral density and exemplify the power in this technique in decomposing a signal into its component frequencies. This will both provide a more solid explanation for

[redacted]

exactly what is being calculated when the PSD is computed, and also ensure that the code used to plot the PSD is correct, since the expected PSD from this cosine function is well known.

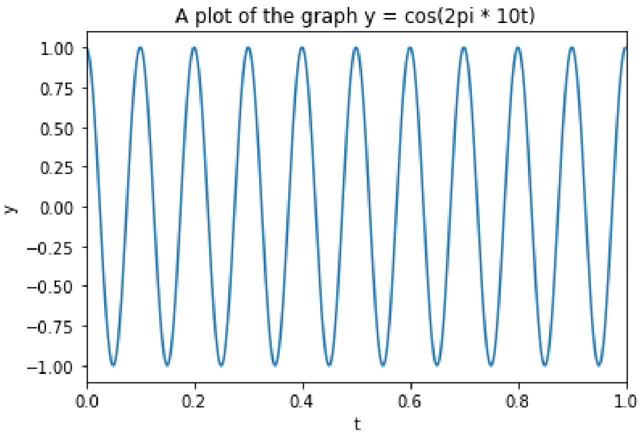


Figure 18| A cosine wave oscillating at a frequency of 10Hz. The aim of the PSD will be to illustrate this with a spike on the x-axis at a frequency of 10, demonstrating that all of the energy in the signal is at 10Hz.

Plotting the PSD for this function should result in a single peak at a frequency of 10Hz. The y- axis units representing power per unit frequency.

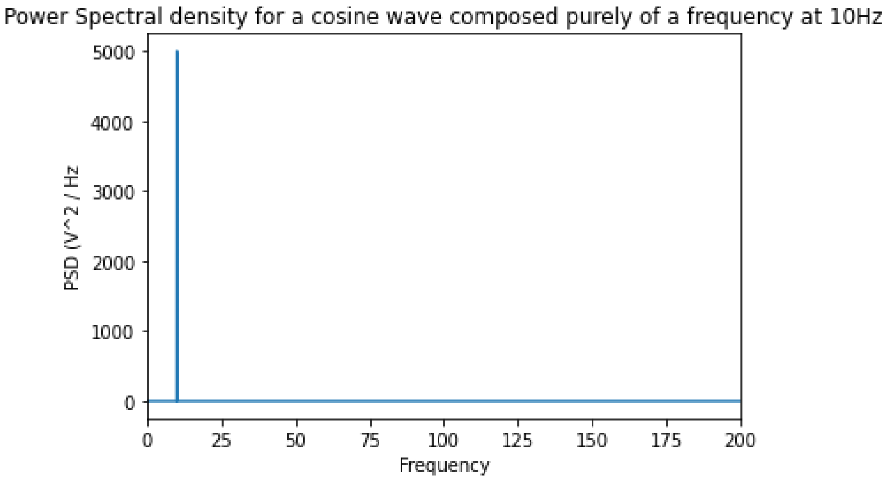


Figure 19| The power spectral density resulting from a cosine wave with a frequency solely at 10Hz. The single peak at a frequency of 10 represents that all the power in the signal is being generated from a wave oscillating at a frequency of 10 Hz.

Hopefully from the above plot it is clear to see the value of the PSD in breaking down a signal. Adding white noise to the cosine function will disturb what the oscillations look like, yet the appearance of the PSD should remain relatively unchanged, since most of the power in the signal is still being generated at the 10Hz frequency, the noise being at a more minimal level across all frequencies.

Plotted below is the PSD of the original cosine wave but with white noise of mean 0, variance 0.01 at amplitude of 1, **fig 20 b**. A key point to emphasise from this exercise is that observing the noisy time series alone cannot inform us much about the nature of the effects of the noise on the signal, only qualitatively that it is having an effect. Plotting the PSD of the signal recovers the background influence over the overall observed time series. Hence, the uses of this PSD extend into our current discussion of the Lorenz System, irrespective of the fact that the time series associated with this are much more complex than the simple cosine wave just discussed.

[redacted]

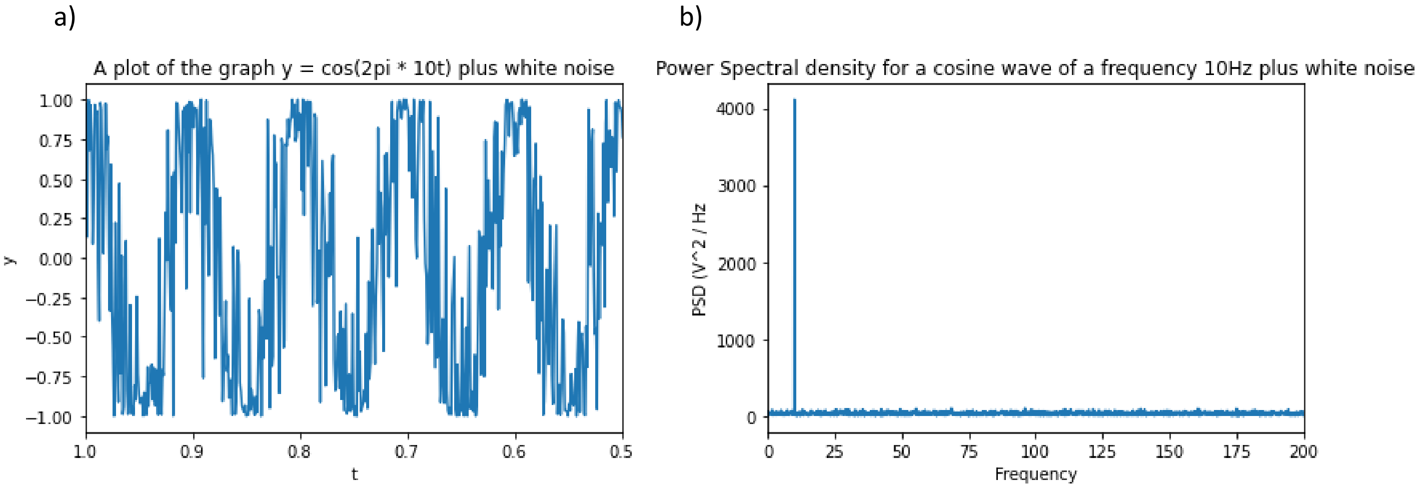


Figure 20| a) How the addition of white noise affects the appearance of the cosine wave, the general form of the wave can still be seen, yet it is more difficult to determine which frequencies are contributing the time series. The plot illustrates how now, not all of the power in the series is being generated at the 10Hz level, there are other frequencies now incorporated that are distorting the appearance of the wave. **b**) Plotting the PSD reveals how the noisy cosine signal is being generated and what frequencies contribute to this. It is obvious that the main contributor is still the oscillations at 10Hz, with all of the noise at a wide range of frequencies at very low power per unit frequency.

Converting the time series of the system into the frequency domain is an invaluable exercise when attempting to quantise the effects of adding noise into this dynamical system.

Power Spectral Densities of the Lorenz System

The process in which the following PSDs have been generated is homologous to the method used for the simple example above. Once more, the noise free PSD will be considered, and then the PSD with added noise. The aim of this section however is different, I am no longer solely identifying the frequency make-up of the signal, but rather attempting to quantify the effects of added white noise – and seeking the point at which the noise dominates the signal.

A useful plot for this analysis is the below, a visualisation of the PSD for pure white noise. The below spectrum is a plot of white noise without any other variables.

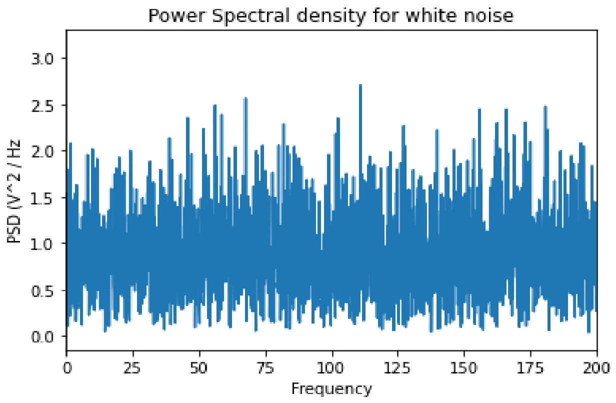


Figure 21| The power within the noise is evenly distributed over all of the frequencies, with no decay. This PSD was generated from a white noise time series with normally distributed noise of zero mean and variance 0.01.

PSDs without noise

Below are the plots for the x, y and z time series of the Lorenz system without the addition of any white noise. I have additionally fitted a least squares regression to the steepest part of the decay of the power, which will be used shortly to characterise the nature of the decay in frequency energy content.

[redacted]

The lines are plotted through the steepest part of the decline to capture most accurately the true value of the slope, avoiding the behaviour at extremely low frequency, between the values of frequency of e^0 and e^1 , characterising the decay.

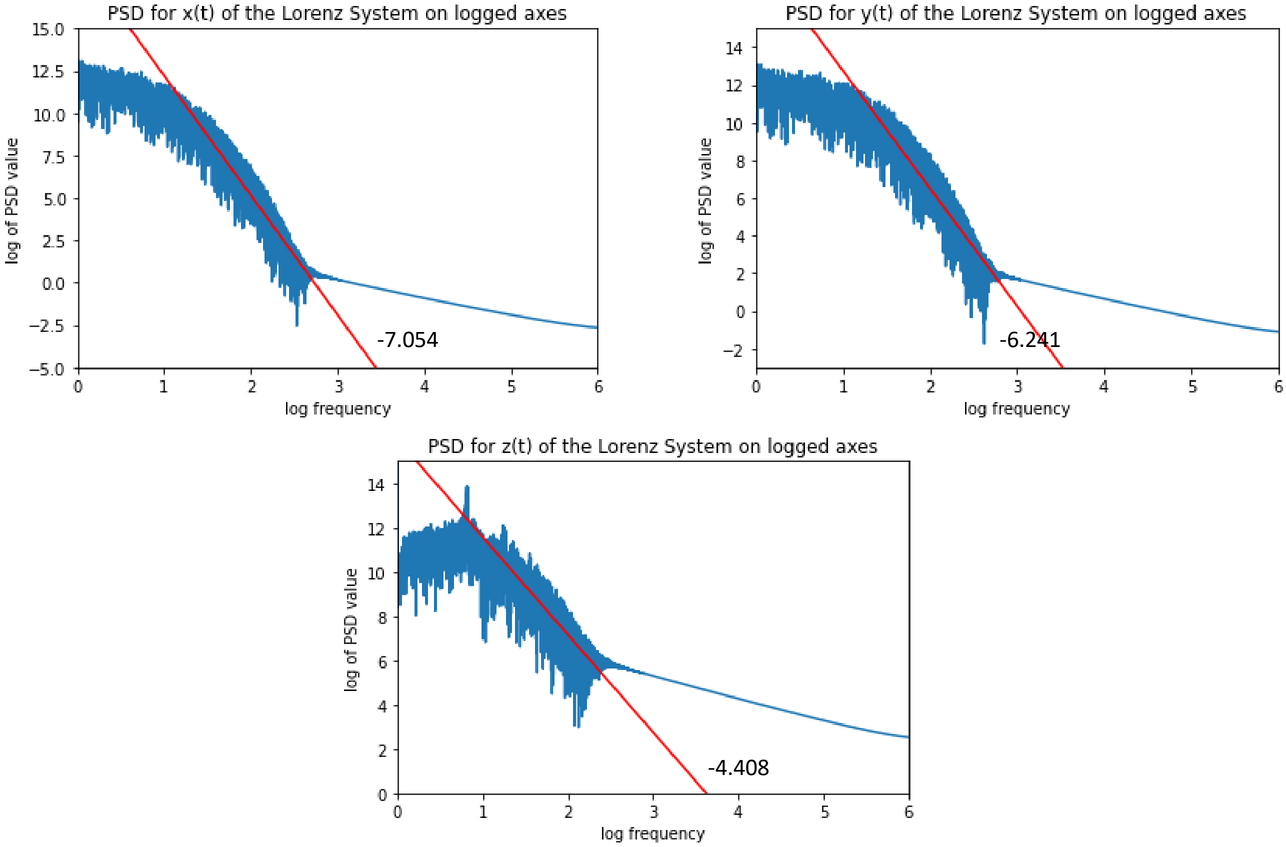


Figure 22| The x, y and z time series' PSDs all follow a power law relating PSD to frequency, since the plots on logged axes result in a linear relationship in a part of the graph. The values for the slopes will be commented on shortly in regard to characterising the nature by which the power decays.

These plots have been generated by solving the Lorenz system using the RK4 numerical method for solving differential equations. This method was chosen over the forward Euler method because its error is much smaller than that of Euler's scheme. Since the aim of plotting these graphs is to quantify how the signal decays over frequency, then the more accurate iterative method was deemed the most appropriate.

Importantly for this exercise, to observe how the power spectrum decays over frequency, it is essential to conclude that the system is in a statistically steady state before taking its power spectrum. This is to ensure that the statistical properties, for example the mean and variance of the time series that this plot is representing, are constant. Since, if they are not then the simulation has not been run for long enough and is still evolving. To ascertain whether the Lorenz system was in a statistically steady state upon decomposition into a PSD, the simulations were run for increasing overall time intervals until the mean and variance of the first half of the time series was equal to the second half of the series. Thus, in the above graphs and in any further calculations on these PSDs, the time series has been run for 1000 seconds to achieve an overall unchanging distribution; the mean and variance were used as indicators that the system is in a statistically steady state. From this, we can be sure that the PSD generated is reflected of the behaviour of the system and the conclusions I make about this PSD can therefore be made with confidence. Additionally, at least a decade of data has been used to fit the slope (in this case frequency values of between 0Hz and at least 100Hz) to ensure accuracy.

The PSDs have been plotted on logged axes for two reasons:

- The frequencies measured are between 0.001Hz and 1000Hz, with the decay of the power mostly occurring between 0.001Hz and around 10Hz. Plotting this on the usual axes results in a region of steep decline followed by a relatively flat line for the rest of the frequencies, this plot would reveal nothing to us about the nature of the steep decay at low frequency, since it cannot be seen on larger axes and will all be concentrated on the left-hand side of the graph.

• In an attempt to quantify the nature of the decay of the signal, I have fitted a line of best fit. The decay of the signal is known to follow a power law, so plotting on logged axes enables simple calculation of the exponent value since the log turns it into a linear relationship. This is exemplified in the below derivation, where S represents the Power Spectral Density, ω represents frequency, A is an arbitrary constant and β determines the rate of decay.

$$S \propto \frac{1}{\omega^{\beta}}$$

$$S = A\omega^{-\beta}$$

$$\log(S) = \log\,(A\omega^{-\beta})$$

$$\log(S) = \log(A) + \log\,(\omega^{-\beta})$$

$$\log(S) = \log(A) - \beta\log\,(\omega)$$

It is clear from the above that the constant A is introduced as a constant of proportionality, it also reveals that the value of β more specifically determines the slope of the decay when the power law relationship between frequency and PSD is plotted on logged axes.

When plotted on these axes, we can calculate the slope of the spectrum to reveal some of the nature of the type of decay that we are seeing. In the case of the Lorenz system and the plots for the x, y, and z time series spectral densities, the aim here is to characterise the rate of decay that we can observe by comparing it to how certain colours of noise decay.

Colours of noise

Assuming that coloured noise follows the power law $S = A\omega^{-\beta}$, the value of β characterises the type of noise that the PSD has been generated from. Even though the Lorenz systems plotted has no additive noise incorporated into it, the way in which the logged PSD vs frequency plot decays resembles the signal of coloured noise, so the aim here is to describe in what way the power decays over frequency.

Table 1| How the colour of noise is determined by the magnitude of the slope of its PSD when plotted on logged axes. The value of β = 0 means there is no decay in power per unit frequency and the noise does not decay over frequency – this is white noise and this is the nature of the noise that will shortly be incorporated into the Lorenz system. Values for β taken from Woolf, 2023.

Value of β	**Colour of noise**
0	White
1	Pink
2	Brown
-1	Blue
-2	Purple
β < -2	Black noise

Noise being classified as various colours is similar to how certain colours of light are generated by specific frequencies; for example white light contains all frequencies of the visible spectrum, and white noise is generated by combining all audial frequencies and is at the same power over all of them. Considering the slope of the decay for the x, y and z solutions, these can be classified as similar to black noise since they are all smaller than -4 (Woolf, 2023). This is referring to the fact that, similar to black light which is a relative

[redacted]
lack of any signal in the visible spectrum, black noise consists of a very small amount of power at a low frequency which decays very quickly – almost a visual representation of silence.

Interpreting these results:

Plotting the decay of the signal on logged axis has helped to reveal both the nature of how the energy decays as frequency increases, but also clearly quantises the amount of decay that occurs, with the highest energies being roughly 10 orders of magnitude stronger than those at higher frequency. I now wish to explain this in the context of the Lorenz System.

In these plots, a value for the maximum frequency measured was 1000Hz, and the simulations run for 1000 seconds with a time step of 0.001. Hence the lowest frequency wave that could be measured was 0.001Hz. The largest values for the PSD being at low frequency is illustrating that the energy is mostly contained in these lower frequencies. Integrating the area under the PSD between these low frequencies reveals the energy contained between them:

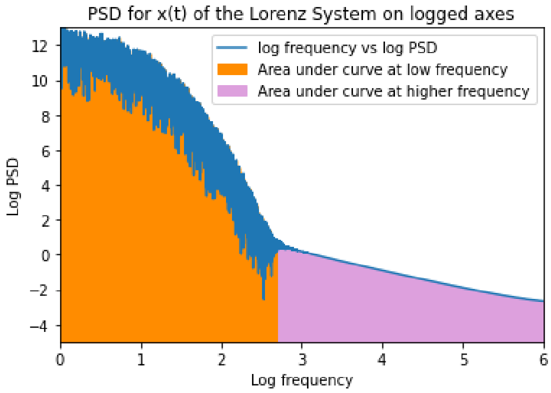


Figure 23| The area under the plot is indicative of the energy contained in certain frequency ranges. The orange region has area of magnitude 19.856 comparative to the pink region that has area of magnitude 10.535. This supports what has previously been said about the lower frequencies being of higher energy.

This steep decay of power as frequency is increased is a visual manifestation of the differing timescales that are occurring simultaneously in the Lorenz system. This has been illustrated below for clarity.

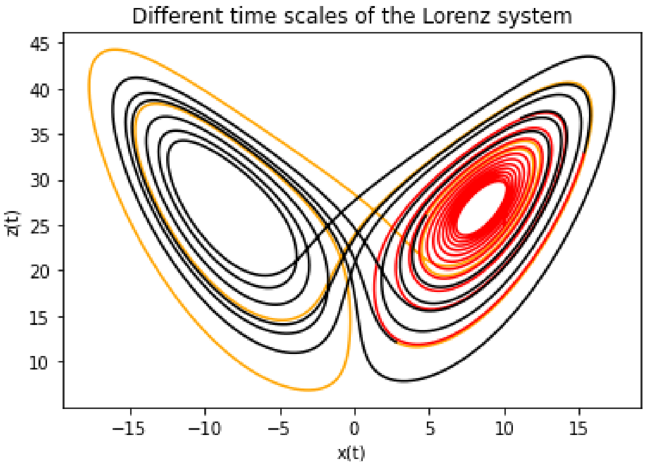


Figure 24| The initial starting point within phase space, and its proximity to a fixed point influences the trajectory that a particle would be on at any specific instantaneous time point. Some trajectories may be traversing the outer 'wings' of the attractor, others may be caught in a loop orbiting a fixed point.

In the system, at any instantaneous point in time, there will be both trajectories moving around the edge of the 'butterfly', illustrated with the orange line, as well as some trajectories moving solely around one wing. Unlike in the simple cosine example where there was energy being injected into the system at a certain frequency, the different timescales within this system are these representations of the frequency of energy.

[redacted]

The longer timescale taken for the trajectory to traverse the entirety of the wings in phase space is much longer than the time taken for the trajectory that is orbiting the centre of one of the wings, meaning that the longer time-period is at a lower frequency comparative to that of the shorter traverse.

PSDs with noise

This noise will then be added to the inputs of the Lorenz system, and the PSDs plotted for the time series solutions of the equations. This will be repeated, adding noise with increasing amplitude, and the effects analysed. The effects are similar for all of the time series, so specifically the x(t) series has been considered here for coherence.

As illustrated by the below plots, as noise is added, the initial steep decay of power over frequency becomes shallower. This is maybe exemplified the most by figure **25d**, where the red and orange lines illustrating the best fit line through the steepest slope and the shallower slope are converging.

It is also interesting to note that contrastingly to **fig 22**, where the initial period of a decay resembling black noise develops into a smooth curve, with the addition of noise, this now represents a decay of $\beta \approx -1$, the characteristic slope of pink noise.

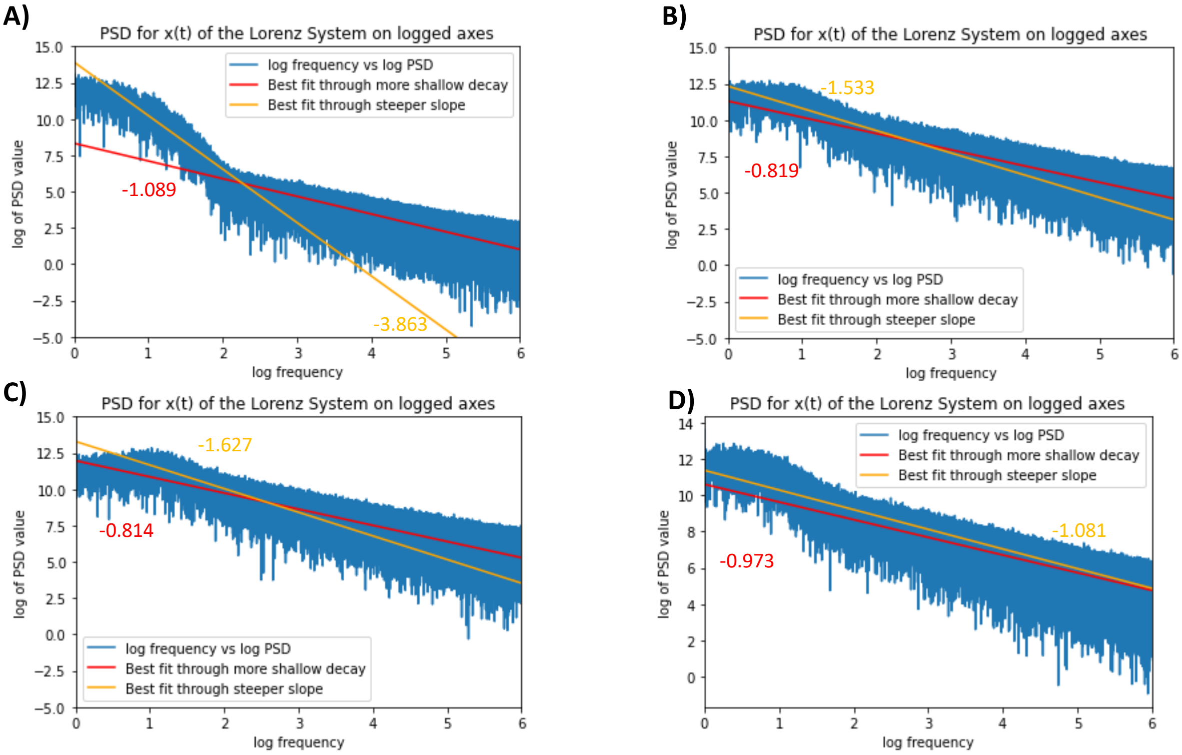


Figure 25| All plots have been generated with noise of variance 0.001 with mean zero. The amplitude has been increased as follows: 1, 50, 100 , 200 for **A** – **D** respectively. For all plots, the slope over the majority of the frequencies is relatively shallow, remaining around a value of -1. The line through the steeper part of the slope initially remains large with small amplitudes of noise (**A**), however the slope becomes less dramatic as more noise is added, plots **B** and **C** illustrate this with a slope of around -1.5. With larger amplitudes of noise, **D**, the best fit line through the steepest part of the graph is very similar to that for the shallower part – implying the noise is overall making the entire spectrum decay at a rate of -1.

From these plots we can see that as the amplitude of the noise is increased, the entire spectrum becomes one of pink noise, with slope roughly –1. Pink noise like this is found commonly in nature (Niklasson, 2019). It is interesting to find that with a large enough influence by the noise, the spectrum reflects that rather than retaining the initial shape without interference. It is also interesting to note that the spectrum becomes that of pink noise, even though the noise added is completely white. This implies that the overall decay of the power in the Lorenz system as frequency increases still has an effect when white noise is added, it is just not as dramatic an effect at very low frequency.

[redacted]

Conclusion:

For the parameter values stated in this report, the Lorenz system consisting of three non-linear differential equations for fluid flow gives rise to chaotic dynamics. Due to the non-linearities in these equations, numerical methods must be implemented to generate time series solutions. Various numerical methods have been investigated for iteratively solving systems of non-linear differential equations and their solutions have been illustrated. Alongside this, Brownian motion has been researched and used as a segue to introduce noise into the Lorenz system. Over the course of this project the time series solutions have been used to generate the strange attractor analogous with the Lorenz system and also have been utilised in the power spectral analysis, allowing the decay of power over frequency to be described in a similar way to coloured noise.

The power spectral densities of the system revealed lots about the nature of the dynamics that the equations generate; with the sharp decay in power as frequency increases exemplifying the different timescales that can be observed within the system at any instantaneous moment, and the rate of decay indicating what type of noise this reflects. The addition of white noise to the system surprisingly didn’t make the entirety of the PSD white for high amplitudes of noise, the nature of the decay in the spectrum being largely pink noise is something that could be built upon further.

Brief self-reflection and possible future research

Overall, this investigation has achieved the aims that were set out in the initial introduction to the project. I have successfully discussed and analysed the Lorenz system as an example non-linear dynamics leading to chaos. Beginning this project with a lack of exposure to the numerical methods that can be used to solve non-linear ordinary differential equations, I feel that I have been successful in researching these and implementing them into my simulations. From this investigation, I have exemplified that the Lorenz system of equations has a strange attractor associated with it for the parameter values σ = 10, r = 29 and b = 8/3. I have enjoyed researching a topic completely new and exploring the system independently. I feel I was successful in my fixed point analysis in the 3D case especially, numerically solving the equation for the eigenvalues in a way that I have not seen in the literature.

Having proven through the use of the Lyapunov exponent that chaos cannot be seen in the Lorenz system for r < 1, it would be interesting to explore more about the briefly considered noise induced chaos for these small values of r. Additionally, it would also be rewarding to attempt to address the ending questions that Lorenz left at the end of his 1963 paper – whether or not the entire atmosphere is chaotic or whether it can be periodic; scope to extend this project further now that I have a base knowledge of chaos would be to specifically consider weather dynamics and atmospheric chaos.

[redacted]

APPENDIX

Numerical iterative methods:

All plots have been generated in Spyder 5.3.3 which is running Python 3.9.5 – 64bit

A – Forward Euler Method

```
import numpy as np
import matplotlib.pyplot as plt
"""
 Define Lorenz attractor with three parameters sigma, beta and rho
These floats are included in the three  non-linear diff eqns that define the attractor

#Introducing Lorenz Equations:
    dx/dt = sigma(y-x)
    dy/dt = x(rho-z)-y
    dz/dt=xy-betaz

These describe the rate of change of x,y,z w.r.t time, t
""″
sigma=10
rho=28
beta= 8/3
#initial conditions:

x0=0
y0=1
z0=0
time_end = 20
time_step =0.01
N = time_end/ time_step

"""
Will solve the equations numerically using Euler's method to produce three graphs,
functions of x,y and z against time
This method will use the idea that Yn = Yn-1 + dy/dt * time_step for all n= 1,..., N
 """
t=np.arange(0.0, time_end, time_step)

x=np.empty(len(t), dtype=object)
y=np.empty(len(t), dtype=object)
z=np.empty(len(t), dtype=object)

x[0]=x0
y[0]=y0
z[0]=z0

def lorenz_attractor_sol ( sigma, rho, beta, time_end, time_step, x0, y0, z0):
     # updating the arrays for each of x,y,z with corresponding values by iteratively
     #finding next value using the one before

   for i in range(0, len(t)-1):
        x[i+1]=x[i] + ( (sigma*(y[i] - x[i])* time_step))
        y[i+1]= y[i] + (((x[i] *(rho - z[i]))-y[i]) * time_step)
        z[i+1]= z[i] + ((x[i]*y[i] - beta * z[i]) * time_step)
   return x, y, z

lorenz_attractor_sol(sigma, rho, beta, time_end, time_step, x0, y0, z0)

""" plotting the graphs for x ,y z against t """
plt.plot(t,y, color = 'r')
plt.xlabel('t')
plt.ylabel('y(t)')
plt.title('y(t) time series')
plt.show()

fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.plot(x,y,z, color = 'black')
plt.xlabel('x(t)')
plt.ylabel('y(t)')
ax.set_zlabel('z(t)')
plt.title('3D projection of the Lorenz System time series solutions x, y and z')

plt.draw()
```

[redacted]

B - RK4 method for numerically solving the Lorenz system

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

# RK4 METHOD

""" Different way to solve first order ODE:
    y1 = y0 +1/6(k1 +2k2 +2k3 +k4)
    k1= h* f(x0, y0)
    k2 = h* f[(x0 + h/2), y0 + 1/2k1]
    k3 = hf[x0 + (½)h, y0 + (½)k2]
    k4 = hf(x0 + h, y0 + k3)
    """
sigma=10
rho=28
beta= 8/3

#initial conditions:

x0=0
y0=1
z0=0

time_end = 20
time_step = 0.01

t=np.arange(0.0, time_end, time_step)

x=np.empty(len(t), dtype=object)
y=np.empty(len(t), dtype=object)
z= np.empty(len(t), dtype=object)
kX1 = np.empty(len(t), dtype=object)
kY1 = np.empty(len(t), dtype=object)
kZ1 = np.empty(len(t), dtype=object)
kX2 = np.empty(len(t), dtype=object)
kY2 = np.empty(len(t), dtype=object)
kZ2 = np.empty(len(t), dtype=object)
kX3 = np.empty(len(t), dtype=object)
kY3 = np.empty(len(t), dtype=object)
kZ3 = np.empty(len(t), dtype=object)
kX4 = np.empty(len(t), dtype=object)
kY4 = np.empty(len(t), dtype=object)
kZ4 =np.empty(len(t), dtype=object)

x[0]=x0
y[0]=y0
z[0]=z0

kX1[0]= time_step * (sigma *( y0-x0))
kY1[0]= time_step * ((x0 *(rho-z0)) - y0)
kZ1[0]= time_step * (x0*y0 - beta*z0)

kX2[0]= time_step* (sigma *( (y0+ 0.5*kY1[0]) - (x0 +0.5 * kX1[0])))
kY2[0]= time_step * ((x0 + 0.5*kX1[0]) * (rho - (z0 + 0.5* kZ1[0]))-((y0 + 0.5*
kY1[0])))
kZ2[0]= time_step * ((x0 + 0.5* kX1[0])* ( y0+ 0.5* kY1[0]) - beta * (z0 + 0.5
```

[redacted]

```
kX3[0]= time_step * (sigma *( (y0+ 0.5*kY2[0]) - (x0 +0.5 * kX2[0])))
kY3[0]= time_step * ((x0 + 0.5*kX2[0]) * (rho - (z0 + 0.5* kZ2[0]))-(y0 + 0.5* kY2[0]))
kZ3[0]= time_step * ((x0 + 0.5* kX2[0])* ( y0+ 0.5* kY2[0]) - beta * (z0 + 0.5 *kZ2[0]))

kX4[0]= time_step * (sigma *( (y0+ kY3[0]) - (x0 + kX3[0])))
kY4[0]= time_step * ((x0 + kX3[0]) * (rho - (z0 + kZ3[0]))-(y0 +kY3[0]))
kZ4[0]= time_step * ((x0 + kX3[0])* ( y0+ kY3[0]) - beta * (z0 + kZ3[0]))

def RK4_sol( sigma, rho, beta, time_end, time_step, x0, y0, z0, kX1, kY1, kZ1, kX2,
kY2, kZ2, kX3, kY3, kZ3 , kX4, kY4, kZ4):
    for i in range(0, len(t)-1):

        kX1[i+1]= time_step * (sigma *( y[i]-x[i]))
        kY1[i+1]= time_step * ((x[i] *(rho-z[i])) - y[i])
        kZ1[i+1]= time_step * (x[i]*y[i] - beta*z[0])

        kX2[i+1]= time_step* (sigma *( (y[i]+ 0.5*kY1[i]) - (x[i] +0.5 * kX1[i])))
        kY2[i+1]= time_step * ((x[i] + 0.5*kX1[i]) * (rho - (z[i] + 0.5* kZ1[i]))-
((y[i] + 0.5* kY1[i])))
        kZ2[i+1]= time_step * ((x[i] + 0.5* kX1[i])* ( y[i]+ 0.5* kY1[i]) - beta *
(z[i] + 0.5 *kZ1[i]))

        kX3[i+1]= time_step * (sigma *( (y[i]+ 0.5*kY2[i]) - (x[i] +0.5 * kX2[i])))
        kY3[i+1]= time_step * ((x[i] + 0.5*kX2[i]) * (rho - (z[i] + 0.5* kZ2[i]))-
(y[i] + 0.5* kY2[i]))
        kZ3[i+1]= time_step * ((x[i] + 0.5* kX2[i])* ( y[i]+ 0.5* kY2[i]) - beta *
(z[i] + 0.5 *kZ2[i]))

        kX4[i+1]= time_step * (sigma *( (y[i]+ kY3[i]) - (x[i] + kX3[i])))
        kY4[i+1]= time_step * ((x[i] + kX3[i]) * (rho - (z[i] + kZ3[i]))-(y[i]
+kY3[i]))
        kZ4[i+1]= time_step * ((x[i] + kX3[i])* ( y[i]+ kY3[i]) - beta * (z[i] +
kZ3[i]))

        x[i+1]= x[i] + 1/6 * ( kX1[i] + 2 * kX2[i] + 2 * kX3[i] + kX4[i])
        y[i+1]= y[i] + 1/6 * ( kY1[i] + 2 * kY2[i] + 2 * kY3[i] + kY4[i])
        z[i+1]= z[i] + 1/6 * ( kZ1[i] + 2 * kZ2[i] + 2 * kZ3[i] + kZ4[i])

    return x, y, z

RK4_sol( sigma, rho, beta, time_end, time_step, x0, y0, z0, kX1, kY1, kZ1, kX2, kY2,
kZ2, kX3, kY3, kZ3 , kX4, kY4, kZ4)

fig = plt.figure()
ax = fig.add_subplot(projection='3d')

ax.plot(x,y,z, color = 'black')
plt.xlabel('x(t)')
plt.ylabel('y(t)')
ax.set_zlabel('z(t)')
plt.label('The 3D plot of the three time series')
plt.draw()
plt.show()

plt.plot(x,y)
plt.xlabel('x')
plt.ylabel('z')
plt.title('The trajectory generated by plotting x(t) against z(t)')
plt.show()
```

[redacted]

C - Euler Marayama Scheme for Langevin Equation

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

time_end = 1000
time_step = 0.01

N = time_end / time_step

t=np.arange(0.0, time_end, time_step)

pi= 3.142
a = 0.001
m = 0.001
eta = 1
X0 = 0

np.random.seed(1)

dW = np.sqrt(time_step) * np.random.randn(int(N)) # weiner process


x=np.empty(len(t), dtype=object)
x[0] = X0

for i in range(0, len(t)-1):
     x[i+1]=x[i] + ((-6*pi*a*eta)/m) * x[i] * time_step  + (1/m)* dW[i]

plt.plot(t,x)

plt.xlabel('t')
plt.ylabel('x(t)')
```

[redacted]

D - Generating a PSD

```
fs=1000
time_end = 1000
time_step =1/1000

# RK4 method as per Appendix B to return time series for x, y, z

RK4_sol( sigma, rho, beta, time_end, time_step, x0, y0, z0, kX1, kY1, kZ1, kX2,
 kY2, kZ2, kX3, kY3, kZ3 , kX4, kY4, kZ4)

spectrum = np.fft.fft(x)  # calculates FFT of the time series x(t)

time_range = np.linspace(1, fs, len(t)) # defining x axis

#plotting PSD
plt.plot(np.log((time_range)),np.log(np.abs(spectrum)))
```

[redacted]

E – Line of best fit through PSD and integration under the curve

```
def func(x, a, b): # define the equation I want scipy.optimize to return i.e y= mx+c
    y = a*x + b
    return y

#fitting regression using the indexes of the lists which plot the points corresponding
to the decay

# indexes [0:950000] correspond to line through the shallow decay
alpha = scipy.optimize.curve_fit(func, xdata = np.log(time_range[0:950000]), ydata =
np.log(np.abs(spectrum)[0:950000]))[0]

print(alpha)
plt.plot(np.log(time_range), (alpha[0])*np.log(time_range) + alpha[1] , 'r')


#indexes [6000:20000] correspond to steepest decay in the x(t) time series without noise

beta = scipy.optimize.curve_fit(func, xdata = np.log(time_range[6000:20000]), ydata =
np.log(np.abs(spectrum)[6000:20000]))[0]

print(beta)
plt.plot(np.log(time_range), beta[0]*np.log(time_range) + beta[1], 'orange')

plt.xlabel('log frequency')
plt.ylabel('log of PSD value')
plt.title('PSD for x(t) of the Lorenz System on logged axes')


plt.legend(['log frequency vs log PSD','Best fit through more shallow decay' , 'Best
fit through steeper slope' ])

plt.show()

#returns same graph of the PSD for the x(t) time series but with two slopes included

from numpy import trapz
# Compute the area using trapezoidal rule, indexes used to split the region of steepest
decay up from the overall PSD

area = trapz(np.log(time_range)[0:14000],np.log(np.abs(spectrum)[0:14000]), dx=0.001)
print("area =", np.abs(area))

area  =  trapz(np.log(time_range)[14000:800000],np.log(np.abs(spectrum)[14000:800000]),
dx=0.001)
print("area =", np.abs(area))

plt.fill_between(np.log((time_range))[0:14000], np.log(np.abs(spectrum))[0:14000],y2=-
5,color= "darkorange")

plt.fill_between(np.log((time_range))[14000:800000],np.log(np.abs(spectrum))[14000:800
000],y2=-5,color= "plum")

plt.xlabel('Log frequency')
plt.ylabel('Log PSD')
plt.legend(['log frequency vs log PSD', 'Area under curve at low frequency', 'Area under
curve at higher frequency' ])
plt.show()
```

[redacted]

F - Calculating the fixed points and their stability

F i) – Calculation

Fixed points occur when the value of x’, y’ and z’ = 0

Hence;

$$0 = \sigma(y - x) \quad [1]$$
$$0 = rx - y - xz \quad [2]$$
$$0 = xy - bz \quad [3]$$

From equation [1]:

$$y = x \quad [4]$$

Subbing [4] into [3]:

$$0 = x^2 - bz \quad [5]$$

Letting $x = 0$,

$$bz = 0$$
$$z = 0$$

And since $y = x$,

$$y = 0$$

Hence, a fixed point occurs at **(0, 0, 0)**

Instead letting $z = r - 1$,

It follows from [5] that:

$$x^2 = b(r - 1)$$
$$x = \pm\sqrt{b(r - 1)}$$

And since $y = x$,

$$y = \pm\sqrt{b(r - 1)}$$

Hence the remaining two fixed points are at:
$\boldsymbol{(\sqrt{b(r-1)}, \sqrt{b(r-1)}\ , r-1)}$ and $\boldsymbol{(-\sqrt{b(r-1)}, -\sqrt{b(r-1)}\ , r-1)}$

F ii) Linear Stability analysis

2D linear stability analysis: The premise of this exercise is that the behaviour of the system close to the fixed points can be described by the linear terms in the system.

Linearising the equations by removing the non-linear terms;

$$x' = \sigma(y - x) = f(x, y) \quad [6]$$
$$y' = rx - y = g(x, y) \quad [7]$$
$$z' = -bz = h(x, y) \quad [8]$$

Conducting this stability analysis, the premise is that adding a small perturbation to the fixed point and analysing the rate of change of the perturbation with respect to time will reveal the nature of the point – i.e whether it is stable or not.

In this case, at the point $(x^\circ, y^\circ, z^\circ)$ = (0, 0, 0), eqn [8] is decoupled so $z(t) \rightarrow 0$ and is stable.

In regard to the other two equations, let’s introduce a small perturbation such that $\eta(t) = x(t) - x^\circ$ [9] and $\vartheta(t) = y(t) - y^\circ$ [10] . We aim to analyse the rate of change of this perturbation with respect to time to see whether the fixed point is stable or not.

[redacted]

Considering [9] first,

$$\eta' = \frac{d(x - x^\circ)}{dt} = \frac{dx}{dt} - \frac{dx^\circ}{dt}$$

Since $\frac{dx^\circ}{dt} = 0$ as it is at a fixed point,

$$\eta' = \frac{dx}{dt} = f(x, y)$$

So in the case of adding an arbitrary deviation from the point,

$$\eta' = f(x^\circ + \eta\ , y^\circ + \eta)$$

$$\eta' = f(x^\circ, y^\circ) + \eta\left(\frac{\partial f}{\partial x}\right) + \vartheta\left(\frac{\partial f}{\partial y}\right) + O(\eta^2)$$

Where the partial derivatives are evaluated at x°, y° and the above equation is the result of a Taylor expansion

Since this is at a fixed point, then $f(x^\circ, y^\circ)$ is zero, and so:

$$\eta' = \eta\left(\frac{\partial f}{\partial x}\right) + \eta\left(\frac{\partial f}{\partial y}\right) + O(\eta^2)$$

Where terms of order η^2 or larger have not been considered

Similarly, for eqn [10]:

$$\vartheta' = \vartheta\left(\frac{\partial g}{\partial x}\right) + \vartheta\left(\frac{\partial g}{\partial y}\right) + O(\vartheta^2)$$

Representing this in matrix form:

$$\begin{pmatrix} \eta' \\ \vartheta' \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} \begin{pmatrix} \eta \\ \vartheta \end{pmatrix} \qquad [11]$$

Ignoring any higher order terms as they are small in comparison the size of the perturbations

In [11] we can see that this includes the Jacobian matrix, **J**, of the vector field at $f(x^\circ, y^\circ)$, and this technique will also be used when analysing the stability of the three-dimensional case. Now that we have the Jacobian of the vector field at the fixed point, we can use this matrix to determine its stability (Strogatz, 2015).

In the case of the linearised equations [6] and [7], this becomes:

$$\begin{pmatrix} \eta' \\ \vartheta' \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma \\ r & -1 \end{pmatrix} \begin{pmatrix} \eta \\ \vartheta \end{pmatrix}$$

A key idea that I will now be using in stability analysis is that the eigenvectors of the Jacobian matrix will indicate the nature and stability of the fixed point.

Finding the eigenvectors using $(J - \lambda I)\begin{pmatrix} \eta' \\ \vartheta' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$,

The characteristic equation for the eigenvalues being:

$$det\begin{pmatrix} -\sigma - \lambda & \sigma \\ r & -1 - \lambda \end{pmatrix} = (\sigma + \lambda)(1 + \lambda) - r\sigma$$
$$= \lambda^2 - \lambda(-(1 + \sigma)) + \sigma(1 - r) = 0$$

[redacted]

Note that I have put the above in that form to note certain characteristics of the equations, this being that the coefficient of the $-\lambda$ term is the trace of **J** and the final term is the value of det(**J**). These quantities will help us comment on the stability of the fixed point, with the sign of the determinant, trace and the sign of $\mathrm{T}^2 - 4\mathrm{Det}(\mathbf{J})$ indicating what type of fixed point we are dealing with (Strogatz, 2015).

If the determinant is negative we have a saddle point. If the determinant is positive but $\mathrm{T}^2 - 4\mathrm{Det}(\mathbf{J})$ is negative then we have a spiral but if $\mathrm{T}^2 - 4\mathrm{Det}(\mathbf{J})$ is positive we have a node. The sign of T indicates the stability of these spirals and nodes, with $\mathrm{T} > 0$ indicating it is unstable and $\mathrm{T} < 0$ indicating stability (Strogatz, 2015).

Hence, the value of $\mathrm{T}^2 - 4\mathrm{Det}(\mathbf{J})$ is $(1+\sigma)^2 - 4\sigma(1-r) = (\sigma-1)^2 + 4\sigma r$

Since the origin as a fixed point exists for both $r < 1$ and $r > 1$, we must consider these two cases separately.

$\boldsymbol{r < 1}$ **:**

Then Det(**J**) will be positive and $\mathrm{T}^2 - 4\mathrm{Det}(\mathbf{J})$ is positive also, so we have a node. Since T is negative then **the origin is is a stable node for $r < 1$**. Any small perturbation will result in the trajectory being pulled back into the origin. **This has been proved to be globally stable via the use of the Lyapunov exponent in the text – no strange attractors for $r < 1$.**

$\boldsymbol{r > 1}$ **:**

Then Det(**J**) will be negative and **we have a saddle point at the origin for $r > 1$** , so in 3D space this means it is stable in two directions and unstable in the third (Strogatz, 2015).

What about the symmetrical points that emerge for $r > 1$:

Again, similarly to the previous case for the origin, we will consider the Jacobian matrix of the vector field at the fixed point $f(x°, y°, z°)$. In this case, consider perturbations in the x direction as $\eta(t) = x(t) - x°$ [9] , y direction as $\vartheta(t) = y(t) - y°$ [10] and the z direction as $\theta(t) = z(t) - z°$ [12] . We are not considering the linear stability here, but using the original equations for the Lorenz system including the non-linearities, hence :

$$x' = \sigma(y - x) = f(x, y) \quad [13]$$
$$y' = x(r - z) - y = g(x, y) \quad [14]$$
$$z' = xy - bz = h(x, y) \quad [15]$$

$$\begin{pmatrix} \eta' \\ \vartheta' \\ \theta' \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & \frac{\partial g}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{pmatrix} \begin{pmatrix} \eta \\ \vartheta \\ \theta \end{pmatrix} \quad [16]$$

$$\begin{pmatrix} \eta' \\ \vartheta' \\ \theta' \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma & 0 \\ r - z & -1 & -x \\ y & x & -b \end{pmatrix} \begin{pmatrix} \eta \\ \vartheta \\ \theta \end{pmatrix} \quad [17]$$

Consider the fixed point C+ = $(\sqrt{b(r-1)}\ , \sqrt{b(r-1)}\ , r-1)$

[17] becomes:

$$\begin{pmatrix} \eta' \\ \vartheta' \\ \theta' \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & -\sqrt{b(r-1)} \\ \sqrt{b(r-1)} & \sqrt{b(r-1)} & -b \end{pmatrix} \begin{pmatrix} \eta \\ \vartheta \\ \theta \end{pmatrix}$$

Once again finding the eigenvalues to comment on the stability of the fixed points:

$$det \begin{pmatrix} -(\sigma+\lambda) & \sigma & 0 \\ 1 & -(1+\lambda) & -\sqrt{b(r-1)} \\ \sqrt{b(r-1)} & \sqrt{b(r-1)} & -(b+\lambda) \end{pmatrix} = 0$$

$$= -(\sigma+\lambda)[(1+\lambda)(b+\lambda)+b(r-1)] - \sigma[-(b+\lambda)+b(r-1)]$$

$$= -(\sigma+\lambda)[\lambda^2+b\lambda+\lambda+br] - \sigma(-2b-\lambda+br)$$

$$= -\lambda^3-\lambda^2(1+b+\sigma)-\lambda(br+b\sigma+\sigma-\sigma)+2\sigma b-2br\sigma$$

$$= -\lambda^3-\lambda^2(1+b+\sigma)-\lambda b(r+\sigma)+2\sigma b(1-r)$$

$$= \lambda^3+\lambda^2(1+b+\sigma)+\lambda b(r+\sigma)+2\sigma b(r-1)$$

Using our values of b and σ,

$$0 = \lambda^3+\frac{41}{3}\lambda^2+\frac{8}{3}\lambda(r+10)+\frac{160}{3}(r-1)$$

We aim to find a range of values for r, if they exist, where the points C+ and C- are stable. From the literature I have determined that there is a region of values where these fixed points are stable, and another region where they are unstable (Strogatz, 2015). If the solutions have negative real values then we can deem these to be stable points, if they have positive real parts then we deem them unstable (Clack, 2006).

Numerically solving this, I have deemed these real roots to occur for $1 < r < 24.7$, which agrees with the literature on this topic. Hence the fixed points C+ and C- are unstable for < 24.7.

[redacted]

G - Plotting the stability of the C+ and C- points

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



x=sympy.Symbol('x')

values = np.linspace(20,30,100)


real_values=[]
for i in values:
    poly = x**3 + 41/3*x**2 + 8/3*x*(i+10) + 160/3*(i-1)
    roots = sympy.solve(poly,x)
    real_values.append(sympy.re(roots[1]))

plt.plot(values,real_values)
plt.axvline(24.7, color='r', linestyle='--')
plt.axhline(0, color='r', linestyle='--')
plt.xlabel('values for r')
plt.ylabel('values of the real part of the imaginary solutions for the cubic
equation')
plt.title('How the eigenvalues of the Jacobian of the Vector field at the fixed
point vary with increasing rho')
plt.show()
```

[redacted]

References

Bayram, M., Partal, T. and Orucova Buyukoz, G. 2018. Numerical methods for simulation of stochastic differential equations. *Advances in Difference Equations*. **2018**(1).

Butcher, J.C. 1996. A history of Runge-Kutta methods. *Applied Numerical Mathematics.* **20**(3), pp. 247 – 260.

Cesbron, L. 2017. On the derivation of non-local diffusion equations in confined spaces. *Department of Pure Mathematics and Mathematical Statistics Centre for Mathematical Sciences.* [no pagination].

Clack, C.T.M. 2006. *Dynamics of the Lorenz Equations.* [Online] . [Accessed 22 March 2023]. Available from: https://www.vibrantcleanenergy.com/wp-content/uploads/2017/07/Dynamics-of-the-Lorenz-Equations.pdf

Cohen, L. 2016. A Review of Brownian Motion Based Solely on the Langevin Equation with White Noise. *Springer Proceedings in Mathematics & Statistics*. **177.** [no pagination].

Cross, R. 2005. A double pendulum swing experiment: In search of a better bat. *American Journal of Physics.* **73**(330), pp.330-339.

Encyclopedia of Management. 2023. *Chaos Theory.* [Online]. [Accessed 20 March 2023]. Available from: https://www.encyclopedia.com/management/encyclopedias-almanacs-transcripts-and-maps/chaos-theory

Ellner, S. P, Turchin, P and Roos, A. D. 2005. When Can Noise Induce Chaos and Why Does It Matter: A Critique. *Oikos.* **111**(3), pp. 620–31.

Gray, C. 2002. An Analysis of the Belousov-Zhabotinskii Reaction. *Rose-Hulman Undergraduate Mathematics* **3** (1) . [no pagination].

Il'yashenko, Ju. S. 1991. The concept of minimal attractor and maximal attractors of partial differential equations of the Kuramoto–Sivashinsky type. *Chaos: An Interdisciplinary Journal of Non Linear Science.* **1**(168).

Krishnamurthy, V. 2015. Edward Norton Lorenz. *Resonance*. **20**(3),pp.191–197.

Kuroiwa, T. and Miyazaki, K. 2014. Brownian motion with multiplicative noises revisited. *Journal of Physics A: Mathematical and Theoretical*.. **47**(1),p.012001.

Lorenz, E.N . 1963. Deterministic Nonperiodic Flow . *Journal of Atmospheric Sciences.* **20**, pp.130 - 141.

Niklasson, L. 2019. *Low intensity natural sounds and pink noise's effect on attention.* Bachelor thesis in Psychology, Umea Universitet.

Petrov, V., Gáspár, V., Masere, J. and Showalter, K. 1993. Controlling chaos in the Belousov—Zhabotinsky reaction. *Nature*.. **361**(6409),pp.240–243.

Pomeau, Y and Piasecki, J. 2017. The Langevin equation. *Comptes Rendus Physique.* **18**(9-10), pp. 570-582.

Romeo, G. 2020. *Mathematics for dynamic economic models, Elements of Numerical Mathematical Economics with Excel*. London: Academic Press.

Smith, G.E and Raghav, S. 2020. The Historical Background: Brownian Motion as of 1905. *Brownian Motion and Molecular Reality*, OXFORD STUDIES IN PHILOS SCIENCE SERIES, pp. 88–C3.N97

[redacted]

Shinbrot, T., Grebogi, C., Wisdom, J. and Yorke, J.A. 1992. Chaos in a double pendulum. *American Journal of Physics*.. **60**(6), 491–499.

Strogatz, S.H .2015. *NONLINEAR DYNAMICS AND CHAOS* **.** 2nd ed. Florida : Westview Press.

Wegmann, K. and Rössler, O.E. 1978. Different Kinds of Chaotic Oscillations in the Belousov-Zhabotinskii Reaction. *Zeitschrift für Naturforschung A*.. **33**(10), 1179–1183.

Woolf, P. 2023. *2.5: Noise Modeling - White, Pink, and Brown Noise, Pops and Crackles.*[Online]. [Accessed 18 March 2023]. Available from: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/02%3A_Modeling_Basics/2.05%3A_Noise_modeling-_more_detailed_information_on_noise_modeling-_white%2C_pink%2C_and_brown_noise%2C_pops_and_crackles

Wu, X.-J. and Shen, S.-L. 2009. Chaos in the fractional-order Lorenz system. *International Journal of Computer Mathematics*. **86**(7),pp.1274–1282.

Yoshimoto, M and Kurosawa, S. 2000. Chaos control using noise. *Materials Science and Engineering: C* . **12**(1-2), pp.63-66.

Zeltkevic, M. 1998. *Forward and Backward Euler Methods*. [Online].[Accessed 13 March 2023]. Available from: https://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node3.html