EPQ — ARTEFACT

Investigating the Sensitivity of Chaos Theory and the Lorenz System in Atmospheric Modelling

---

Candidate Name: Haolin Guo
Candidate Number: [0000]
Centre Name: [Your school or college]
Centre Number: [00000]
Supervisor: Lisa Bray
Examination Board: [AQA / Edexcel / OCR / CAIE]
Word Count: 3,569
Submission Date:

Contents

Overview

This artefact is a twin-trajectory explorer for the Lorenz system. It consists of a Java program, LorenzLyapunov.java, that integrates the Lorenz equations and prints the results as CSV data, and a browser page, lorenz_viewer.html, that reads that CSV data and produces three linked plots and a numerical result. The separation of calculation and visualisation allows the plots to be redrawn without recalculating the simulation and provides an audit trail that can be independently examined.

The artefact runs the same equations twice from starting states that differ in one variable, then measures how fast the two runs diverge. It produces the largest Lyapunov exponent, from which a doubling time and a predictability horizon follow. The sections below describe the artefact in the order in which a user meets it: the simulation that generates the data, the figures that display it, the controls, the explanations offered to the user, the output numbers, and the underlying mathematics.

1. The interactive simulation

The simulation has two steps. First, the Java code integrates two trajectories, A and B, using a self-coded fourth-order Runge–Kutta method with a fixed step of $dt = 0.001$ over fifty model time units. Because the step is fixed, both trajectories are evaluated at the same instants. The Java code stores the positions of both trajectories and the Euclidean distance between them at each integration step, writing all eight columns to a file named lorenz_twin.csv.

Before any output is written, the program performs a self-check by evaluating the derivative at the two nonzero fixed points, where it should be zero, and prints the result. An incorrectly typed equation is therefore detected before any trajectory is calculated.

The browser page is then opened and one CSV is dropped onto it. Parsing, fitting and plotting are done locally, with no uploaded data or server calculation. Each member is inspected separately; the eight retained CSV files are then compared in the report's ensemble table. What the browser displays for an individual member is described below.

2. Visualisation

The argument is illustrated by three figures, which should be viewed sequentially: the first figure shows the action of the system, the second figure illustrates the problem, and the third figure quantifies the problem.

Figure 1 — Lorenz attractor (rotate / zoom)

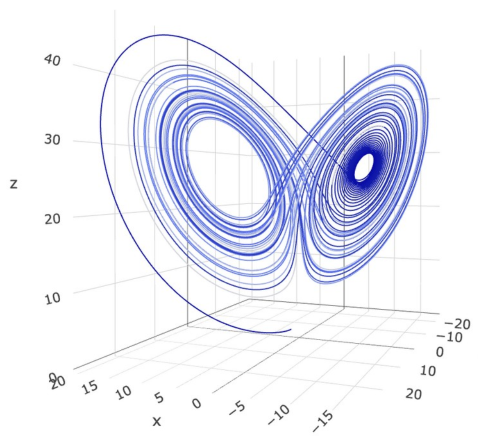

Figure 1. Bounded but never repeating. The trajectory remains within a finite region of phase space but never repeats and never intersects itself. The two-lobed structure is best seen by rotating the plot. This is the behaviour of a single run; the point of the artefact is what happens when two are run together.

Figure 2: Two trajectories from almost-identical starts

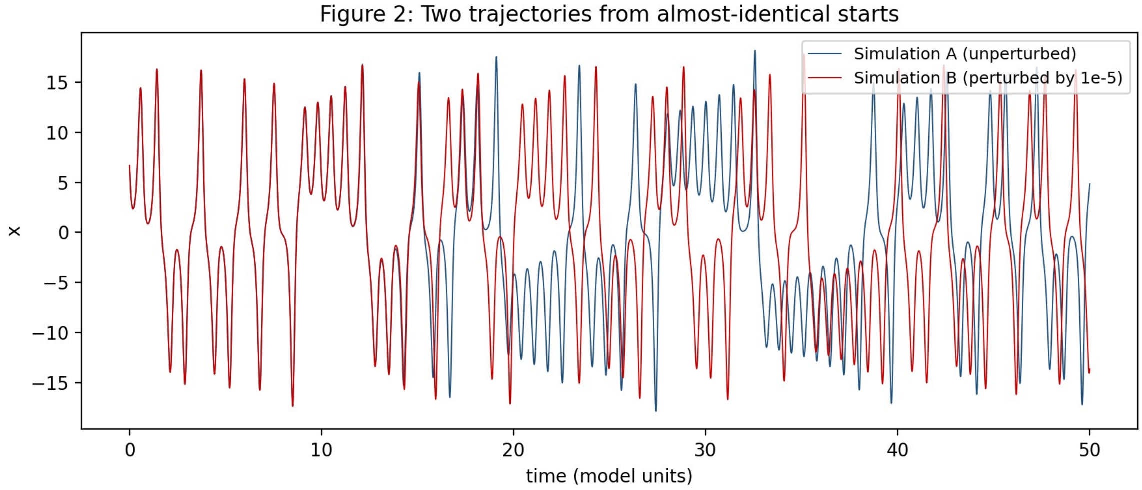

Figure 2. Two runs from almost identical starts. Run A is drawn in blue, Run B in red. For a long stretch only one colour is visible because the two curves lie on top of each other; in the run shown they remain indistinguishable for about twelve model time units. Nothing is added to the system at the moment they part.

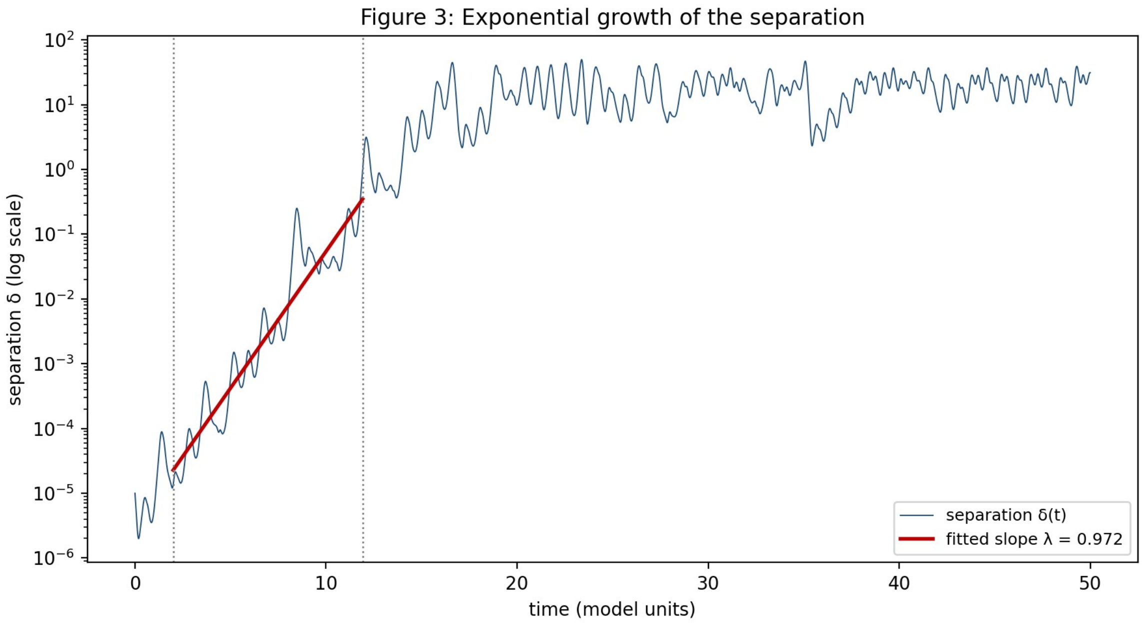

Figure 3. Logarithmic axis separation. A straight line in this figure indicates that the separation is increasing by a constant factor in each equal time interval, which is what $\delta(t) \approx \delta_0 e^{\lambda t}$ *describes. The dotted lines indicate the fitting window, the red line is the linear fit, and its slope is λ. Because that window determines the result, the reader is given control of it.*

3. Controls

This page displays the two numbers that determine the fit window, since the measurement depends on them and a reader should be able to test that dependence rather than take it on trust. They default to the Java rule, FIT_T_MIN = 2 and FIT_DELTA_STOP = 1, so the result shown can be checked against the printed result. Modifying them produces a visible re-fit in the browser: an earlier start includes the alignment transient, and a higher stop includes more of the saturated tail. Changing the Java fitting rule itself would require the constants to be edited and the run repeated.

The current Java version takes spin-up, σ, ρ, β and δ_0 as command-line arguments, so a documented alternative run requires no change to the source code. Each CSV begins with a machine-readable record of the run: the parameters, seed, state after spin-up, dt, perturbation and fit rule. The browser displays those values above the figures, removing any uncertainty about what produced the result while retaining the classical values as the default comparison case.

4. Explanations for the user

Each figure has an explanatory note alongside it, rather than a list of captions at the end, so that a reader unfamiliar with the report can follow the argument as the page unfolds. Three of these notes do specific work. The note alongside Figure 2 states that nothing is injected at the moment the curves separate, because a first-time reader tends to assume that something must have been. The note alongside Figure 3 explains why the axis is logarithmic before the fitted line is shown. The note alongside the results explains why the fit excludes both ends of the curve.

The artefact stops at the calculation of the Lorenz system. It displays the fitted exponent for a member and the timescales derived from it, while the broader atmospheric interpretation[1] is left to the report. This distinction matters because an idealised model cannot on its own determine a numerical predictability limit for the atmosphere.

5. Numerical outputs and calculations

These figures show that the two runs diverge; the following numbers indicate the speed and certainty of this divergence.

For each loaded file the page shows the number of rows, the initial separation, the fit window, the fitted exponent, the doubling time and the predictability horizon. It also shows the complete run configuration read from the CSV header: parameters, initial state, spin-up, dt, perturbation and fit rule. A quoted result can therefore be traced back to the data that produced it.

The exponent is calculated by performing a least squares fit of ln δ against t over a window that is predetermined rather than selected by visual inspection: the fit begins at the first point after $t = 2$ and ends at the first point where δ reaches 1. R^2 is reported with every exponent so that the linearity of the line is quantified rather than assumed, and any member with a poor fit is evident rather than being averaged away unnoticed.

Quantity	Symbol	Value	Source	Note
Largest Lyapunov exponent	λ	0.833 ± 0.124	measured	ensemble of 8 members

[1] Shen, Pielke Sr. and Zeng, ‘One Saddle Point and Two Types of Sensitivities within the Lorenz 1963 and 1969 Models’ (2022).

Published comparison	λ	0.906	Sprott[2]	within 1 s.d. of the mean
Doubling time	$\frac{\ln 2}{\lambda}$	0.83	derived	model time units
Predictability horizon	t	≈ 14	derived	$\delta_0 = 10^{-5}$ to $\delta = 1$
Fit quality	R^2	0.91	measured	member started on the attractor
Fit quality, no spin-up	R^2	0.75	measured	λ falls to 0.441

Reporting R^2 is not a theoretical exercise. It is what revealed the spin-up problem: a run from the initial condition (1, 1, 1) yields λ = 0.441 with $R^2 = 0.75$, and it is the low R^2 that signals a bent line rather than just a small exponent. The same code, the same rule, but a pair started on the attractor yields 0.972 with $R^2 = 0.91$.

The CSV is the audit trail: it holds both trajectories, their separation and the full configuration record. The supplied plotting script can reproduce the three static figures from it, while the browser viewer re-fits and explores the same data locally. Each member filename includes its spin-up value, so ensemble runs cannot silently overwrite one another.

6. Educational content: the mathematical framework

The material in this section appears in the artefact beside the figure it explains rather than collected at the end, so that a reader meets each definition at the point where it is needed.

The Lorenz system is a system of three coupled ordinary differential equations given by the following:

6.1 Where the equations come from

These equations were derived by Edward N. Lorenz[3] from the physics of Rayleigh–Bénard convection: a shallow layer of fluid heated from below and cooled from above. For weak heating the fluid stays still and conducts heat; for stronger heating it overturns in rotating rolls. Building on the work of Saltzman[4], Lorenz expanded the full flow, which has infinitely

[2] Sprott, Chaos and Time-Series Analysis (2003).
[3] Lorenz, ‘Deterministic Nonperiodic Flow’ (1963).
[4] Saltzman, ‘Finite Amplitude Free Convection as an Initial Value Problem—I’ (1962).

many degrees of freedom, into a series of modes and truncated all but the three most important, resulting in the above system.

This remains, however, a toy model. Three numbers cannot represent a real fluid, much less the atmosphere: the model has no geography, no moisture, and no seasons. Its simplicity is its virtue. It is nearly the simplest possible system that retains the essential nonlinearity of convection, so any unexpected behaviour it exhibits cannot be attributed to complexity. If unpredictability arises in this model, it does so for fundamental reasons.

6.2 The variables

The three variables describe the state of the convecting fluid at a particular time: x represents the intensity of the convective motion, that is, the overturning of the fluid; y represents the temperature difference between the ascending and descending currents; and z represents the departure of the vertical temperature profile from linear.

Two characteristics of these equations are worth noting. First, they are deterministic: if the exact values of x, y and z are known at a particular time, the equations determine their values at all subsequent times, and there is no random variable anywhere. Second, they are nonlinear: the terms xz and xy represent the multiplication of two variables. This is significant because a linear system cannot exhibit chaos, so any chaotic behaviour in this system must arise from these two product terms.

6.3 Phase space and trajectories

The state of the system can be represented by a single point in a three-dimensional space with axes x, y and z, known as phase space. As time evolves, this point moves, tracing a curve known as a trajectory. The equations give a velocity to every point of the space, and determinism has a geometric meaning: through any point there passes one and only one trajectory, so trajectories cannot cross. This is why Figure 1 is a three-dimensional plot rather than three separate graphs against time.

6.4 The attractor

An attractor is the set to which the trajectories settle after the transient dynamics have decayed. Simple systems have simple attractors: a point for a damped pendulum, a closed loop for a steady oscillation. The Lorenz system has a strange attractor. Its trajectories are confined to a bounded region of phase space shaped roughly like a butterfly, but they never repeat and never self-intersect.

6.5 The largest Lyapunov exponent

The largest Lyapunov exponent, λ, quantifies the sensitivity to initial conditions. If two trajectories are initially separated by a small distance δ_0, their separation after time t will be roughly $\delta_0 e^{\lambda t}$. A positive value of λ is the hallmark of chaos, and the magnitude of λ sets the predictability timescale, with errors doubling approximately every $\frac{\ln 2}{\lambda}$ time units. This single number is what the artefact exists to measure.

6.6 The parameters and the onset of chaos

The classical values of $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$, studied by Lorenz[5], are used in this project. The first two values are fixed properties of the water-like fluid and the cell geometry, and ρ is the control parameter that varies the strength of the heating. As ρ increases, the system passes from stillness ($\rho < 1$) through steady convection to chaos: stability analysis shows that the steady states lose stability at $\rho = \frac{\sigma(\sigma + \beta + 3)}{\sigma - \beta - 1} \approx 24.74$ for these values[6], beyond which the trajectory wanders indefinitely without settling. Choosing $\rho = 28$ places the system inside this chaotic regime; the value is standard in the literature, allowing the simulations in this project to be checked against published results.

7. User testing and changes

The CSV viewer was used without instruction by three testers. Their results are in the project folder. The questionnaire does not test the explanations of λ or doubling time, only loading the CSV and reading Figure 2, so that will be the next question for the usability test.

Tester	Observed use	Feedback on test items and/or test directions	Comments / change made
Samantha	Independently completed the CSV loading process; could not reliably detect any deviation.	The zoom controls were fixed at a default centre.	Scroll zoom with the mouse pointer and fit-window controls.
Emily	Was able to load the	The plot jerked and the	Continuous panning,

[5] Lorenz, 'Deterministic Nonperiodic Flow' (1963).
[6] Strogatz, Nonlinear Dynamics and Chaos, 2nd edn (2015).

	CSV on her own but lost the original location as she zoomed in and out.	visible crossings changed as she moved around.	pointer-centred zoom, and a hover read-out of the gap.
Terry	Independently loaded the CSV; gave 18.38 for the requested point.	No additional suggestion was recorded.	The vague answer supported the guidance in Figure 2 and the gap display added after the first two tests.

All three testers loaded the CSV without a hint, but none identified the requested separation point clearly and consistently. The resulting changes therefore target navigation and interpretation rather than CSV loading: the old centre-based zoom buttons were removed, panning no longer snaps on release, and the viewer now displays the difference between the two curves on hover. These are recorded observations from the tests, not a claim that the artefact has passed a full usability validation.

8. Using the interactive explorer

This is the sequence of use for the artefact itself, beginning with a reproducible baseline and then allowing the three outputs to be inspected and the fitting choice tested without altering the stored CSV. The commands below are run from the 03_Code folder, where the Java file and browser viewer are supplied together.

Compile the simulator with javac LorenzLyapunov.java and generate the standard member with java LorenzLyapunov 30. The program prints its fixed-point check, configuration, fit window, λ, R^2, doubling time and predictability horizon, then writes lorenz_member_spinup30.csv. Open lorenz_viewer.html and drop that CSV onto the marked area; the page performs its analysis locally and uploads nothing.

Stage	User action	What is visible?
1	Generate or select a CSV member.	A named CSV with the run record and eight data columns.
2	Drag it onto the browser page.	The summary and run-configuration panels will appear above the plots.
3	Rotate and zoom Figure 1.	One bounded two-lobed trajectory in the x-y-z phase space.
4	Pan or zoom Figure 2; hover over a curve.	Both trajectories and their varying x-gap; the tooltip shows the gap in common decimals.
5	Read Figure 3 from left to right.	It shows the log-scale separation curve and the red fitted line only over the selected growth interval.
6	Change the two fit-window inputs and select Re-fit λ.	A recalculation for the loaded data; restoring $t > 2$ and $\delta \geq 1$

		returns the recorded default.

The fit window is all the viewer controls for the CSV currently displayed. Altering the spin-up, σ, ρ, β or δ_0 requires a new Java experiment, run as java LorenzLyapunov [spinUp] [sigma] [rho] [beta] [delta0]. The order of the arguments matters: an alternative run is new raw evidence, not a modification of the existing data. For instance, java LorenzLyapunov 45 10 28 2.6666666667 1e-5 generates a 45-time-unit spin-up member using the classical parameters.

9. Data record and supplied output

This section allows an assessor to inspect what the artefact actually produces, rather than relying on a description of it. The raw data files shown here are generated by the same Java source included in the package.

9.1 What every generated CSV records

Each current CSV begins with a comment line carrying the run record, followed by the data header t, xA, yA, zA, xB, yB, zB and delta. The comment line does not affect the numerical data; it allows the browser to report the exact configuration that generated the loaded file.

Recorded field	Why it is useful in the artefact
σ, ρ, β and dt	Denote the Lorenz system and the fixed time grid used for both trajectories.
Seed and state after spin-up	Displays the initial state of the measured twin experiment, not just the original seed.
Spin-up and δ_0	Establish a connection between the individual member file and its time of formation and initial difference.
Fit t-minimum and δ-stop	Allows the displayed browser fit to be compared with the Java measurement rule.
t, xA, yA, zA, xB, yB, zB, delta	All coordinates used in the plot as well as the Euclidean distance, provided for independent reproduction or verification of the results.

Each member file contains 50,001 numerical rows from t = 0 to t = 50 inclusive at dt = 0.001, and its filename records its spin-up period so that ensemble members cannot silently overwrite one another. The separate Python script is retained only to turn completed CSV data into static PNG figures; it is not part of the numerical solver.

9.2 The eight supplied members

The values below are the direct numerical outputs of the eight retained CSV members. They belong here as evidence generated by the artefact; the justification for the measurement and the meaning of its limitations are discussed in the report.

Member	Spin-up	λ	R^2	Doubling time
1	30	0.972	0.9121	0.71
2	35	0.853	0.8664	0.81
3	40	0.910	0.8799	0.76
4	45	0.827	0.8700	0.84
5	50	0.866	0.8464	0.80
6	55	0.681	0.7462	1.02
7	60	0.615	0.7415	1.13
8	65	0.937	0.9355	0.74
Mean		0.833 ± 0.124		0.83

For the standard member the Java and browser calculations agree when the viewer uses the recorded default fit rule, $t > 2$ and $\delta \geq 1$. Either boundary can be moved deliberately to show why the displayed slope depends on the chosen growth interval; this changes the browser analysis only and leaves the original CSV intact. In the table above, the mean doubling time is calculated as ln 2 divided by the mean exponent, which is slightly less than the mean of the eight individual doubling times (0.85).

9.3 What the R^2 column shows

Reading down the R^2 column reveals a pattern in the artefact's own output. The two members with the lowest exponents, 0.615 and 0.681, are also the two with the lowest fit quality, 0.742 and 0.746, while every member above $R^2 = 0.80$ has an exponent between 0.827 and 0.972. The artefact makes this inspectable rather than merely asserted: loading member 7 and then member 8 into the viewer shows the difference in straightness directly on Figure 3, over the same fit window. What the pattern means for the reported result is argued in the report; what the artefact contributes is that the evidence for it is visible in the interface and recoverable from the supplied CSV files.

10. Artefact submission contents

The runnable artefact is delivered as a small, inspectable bundle rather than just a document. Each of the following components is retained so that a later assessor can reproduce the visible output and trace it back to the raw data.

The following is a breakdown of the components and their functionality:

Component	Functionality
LorenzLyapunov.java	Compile and run the fixed-step twin-trajectory experiment, including alternate documented settings.
lorenz_viewer.html	Load a CSV, inspect the configuration, change the displayed fit window and explore all three figures.
Eight raw CSV files	Check the numerical columns and reproduce the supplied individual-member plots and values.
Static PNG figures and HOW_TO_RUN.md	See the intended output quickly and follow the exact route to a fresh run.

The artefact therefore demonstrates the mechanism, the controls, the numerical record and the resulting visualisations. The report serves a different purpose: it explains the rationale behind the design choices, evaluates the evidence against the research literature, and qualifies what this idealised Lorenz-system result can claim about real weather forecasting.

11. Known limitations of the artefact as software

These are properties of the artefact itself rather than of the physics it measures. Each is recorded so that what the artefact cannot do is stated here rather than discovered in use.

Limitation	Why it was accepted, and what removing it would take
The viewer loads Plotly from a public content-delivery network, so the page needs an internet connection the first time it is opened.	Keeping Plotly external keeps the artefact one small HTML file that can be emailed or copied to a memory stick. Bundling a local copy would remove the dependency at the cost of a far larger file.
One member is displayed at a time; the ensemble is assembled by hand in the report.	Each CSV holds 50,001 rows, and loading eight at once slows the page noticeably. Reading only the eight recorded λ values into a single overlay would be the next feature to add.
A malformed or non-Lorenz CSV is not validated; the page assumes the eight expected columns.	Every supplied file is produced by the included Java source, so the assumption holds for the submitted package. A column-header check would make the

	page safe for files from elsewhere.
The fit window is the only quantity the browser can change; σ, ρ, β, spin-up and δ_0 require a new Java run.	This is deliberate. An altered parameter is new raw evidence, not a re-reading of existing evidence, so it should produce a new CSV rather than silently change what is on screen.
The browser repeats the same least-squares fit as the Java program rather than checking it independently.	Agreement between them confirms that the two implementations read the same window; it is not an independent test of the method. That would need the renormalising procedure of Wolf et al.[7]

None of these prevents the reported result from being reproduced from the supplied files.

They define the limits of what the artefact can be asked to show.

[7] Wolf et al., ‘Determining Lyapunov Exponents from a Time Series’ (1985).