Proceedings of CONF-CIAP 2026 Symposium: International Conference on Atomic Magnetometer and Applications
DOI: 10.54254/2753-8818/2026.HZ28787

Chaos in the Lorenz System: Using 4th-Order Runge-Kutta Method to Analyze Parameter- and Initial Condition-Dependent Dynamics

Jason Yiping Deng

The Affiliated High School of South China Normal University, Guangzhou, China
dengyp.jason2023@gdhfi.com

Abstract. This paper primarily describes two experiments related to the numerical solution of the Lorenz system using the 4th-order Runge-Kutta method, and verifies the sensitivity of the dynamics of the Lorenz system under different parameters and the sensitivity of the initial value after entering chaos. In the two experiments, the system was observed entering the chaotic system when ρ was between 23 and 25, and a typical butterfly-shaped strange attractor was observed, and the Lyapunov index calculated at 0.9 when the x-value changed from 1e-10, confirming trajectory divergence and chaotic behavior. At the same time, these results also prove the reliability of RK4 as a numerical simulation method.

Keywords: Chaos, RK4, Runge-Kutta method, Lorenz system, Lorenz attractor, Lorenz equations

1. Introduction

Chaotic system is a kind of physical systems with high sensitivity to initial conditions: a slight change of the initial state would result in the occurrence of a very different final result; thus, accurate long-term prediction is extremely difficult. In 1963, while attempting to simulate air convection using a simplified model of thermal convection fluid dynamics on previous researchers, Lorenz considerably reduced the degree of partial differential equations and proposed a group of nonlinear ordinary differential equations known as the Lorenz equations [1], which laid down the foundation of chaos systems currently studied.

The Lorenz system is defined by a set of differential equations describing the movement of systems, and has been given application in the study of chaotic systems like turbulence and climate, with the variables representing the rate of convection, horizontal temperature variation, and the difference between vertical temperature and a linear profile of temperature over time are shown to change accordingly.

In this study, for numerical analysis methods, the fourth-order Runge-Kutta method is selected because it achieves a balance of accuracy and computational efficiency; the precision of the fourth-order Runge-Kutta method is sufficient to ensure the undistorted result of the Lorenz attractor within the proper interval, and in addition, it is also an explicit method with good computational efficiency

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.HZ28787

[2]. In fact, RK4 has been used in the study of Lorenz equations for a long time, and this paper mainly chooses this method with the goal of reproducibility and cost-effectiveness.

From the equations above, which will be mentioned later, people can derive that for certain parameter values, for long-term behavior, the generated chaotic system evolves toward a complicated geometrical entity possessing a Lorenz attractor. It has no periodic behavior and highly depends on the initial values. This corresponds to the behaviour of a chaotic system, that even the slightest variations in initial conditions or parameters cause drastic changes in the output.

This paper briefly introduces the background of the Lorenz equations, parameters' significance, the meanings of its equilibrium points and stability of these points, and the most important thing, the initial value sensitivity and parameter-dependence of chaotic systems. After comparing various numerical solutions of the Lorenz equations, this article adopts the 4th-order Runge-Kutta method to obtain an approximate solution. In order to realize the display and run effect of the Lorenz attractor and observe the changing rules of the Lorenz attractor when the value of the parameter ρ changes, Lorenz attractor was coded and presented through Python and through changing the value of the parameters and initial condition to compare its change in form. The ultimate goal of this paper is to provide data support for understanding the threshold and parameter sensitivity of chaotic systems, and to provide help for beginners of chaotic systems while reproducing the relevant theories of chaotic systems.

2. Literature review

2.1. Chaos theory before Lorenz

Before Lorenz's work with the Lorenz equations introduced chaos theory to the world, great progress had been made. As early as 1695, Leibniz proposed fractional derivatives, paving the way for fractal geometry and even non-integer dimensions [3]. The 19th century was a crucial period for developments in mathematics and physics [3]. Maxwell was the first to recognize that in certain dynamical systems, small changes could lead to significantly larger differences in possible outcomes [3]. In mathematics, Weierstrass, Cayley, Cantor, Poincaré, and Hadamard, contributed respectively with continuous non-differentiable functions, the convergence behavior of Newton's method, the Cantor set with fractal properties, and research on the three-body problem [3]. Among these, Poincaré's pioneering and foundational role was particularly significant, as his work involved bifurcation theory, strange attractors, and Sensitive Dependence on Initial Conditions, which were extensively applied in Lorenz's research [3]. Moreover, even before Lorenz, in 1953, Strotz et al. historically simulated chaotic dynamics for the first time in a computer simulation of the Goodwin model, though they did not recognize the importance of their findings [3].

Building on the research of these predecessors and through his own studies and efforts, Lorenz synthesized all these concepts and proposed the Lorenz equations. He integrated all the elements and explained them within the context of a reproducible computer simulation scenario. Poincaré discovered all these individual elements; however, he did not take them seriously.

2.2. Deterministic nonperiodic flow

This is a groundbreaking, milestone paper and also the theoretical source of the current article. As mentioned in the previous section, Lorenz was the first person in history to integrate the theories and experiments of chaotic systems together, and came up with a complete conclusion.

DOI: 10.54254/2753-8818/2026.HZ28787

The main objectives of this paper were to explore whether deterministic equations could produce nonperiodic solutions and to examine the stability and sensitivity to initial conditions of these solutions [1]. Lorenz simplified Saltzman's convection model to obtain the Lorenz system and used the double-approximation method for numerical simulation, studying long-term behavior by observing trajectories in phase space [1]. In the final results, he found that the system gradually developed into nonperiodic oscillations, with unstable steady-state and periodic solutions existing, while most solutions were nonperiodic [1]. Phase space trajectories exhibited a double-spiral structure and were compressed into a strange attractor [1]. The sequences of maxima of the variables were sensitive to initial conditions and were nonperiodic [1]. Through these results, Lorenz concluded that deterministic equations could produce nonperiodic behavior, and systems satisfying this condition were extremely sensitive to initial conditions, making precise long-term weather forecasting impossible [1].

2.3. Another paper using RK4 to solve the Lorenz equation: RK4 and vectorial RK4-used to analyze a modified form of fractional order Lorenz system

This article, relative to the methods used in the current paper, employs a modified RK4 method to numerically simulate the Lorenz system, investigating the dynamics of a generalized form of the fractional-order Lorenz system using the Caputo derivative definition [4]. Methodologically, the modified RK4 and its vector variant applied here are specifically suited for solving fractional-order differential equations tailored for the fractional-order Lorenz system, preventing the calculation cost adress too high [4].

In terms of conclusions, it demonstrates the effectiveness of this method and also highlights the efficacy of the RK4 method in analyzing complex chaotic systems. Also, though the system studied in this paper is slightly different from it, one is fractional and the other is integer, the change law of dynamic behavior is similar: both have a transition from ordered systems to chaotic systems. In other words, this illustrates the existence of chaotic behavior.

3. Theotical background

3.1. Lorenz equation

The set of Lorenz Equations was given by the following equations (1)~(3) derived by Edward Lorenz [1]:

$$\frac{dy_1}{dt} = \sigma\left(y_2 - y_1\right) \tag{1}$$

$$\frac{dy_2}{dt} = \rho y_1 - y_2 - y_1 y_3 \tag{2}$$

$$\frac{dy_3}{dt} = y_1 y_2 - \beta y_3 \tag{3}$$

These equations can describe a Chaotic system.

DOI: 10.54254/2753-8818/2026.HZ28787

Here we give formulas for variables of where y1 represents convective intensity, y2 indicates horizontal temperature gradient, and y3 denotes the nonlinearity degree of temperature distribution. In addition to the above values, the following represent parameters: Prandtl number, which is the σ; Rayleigh number, which is the ρ; and dimensional parameter, which is the β.

In addition, in this paper, ρ will be studied among all three parameters. It is because the Rayleigh number drives convection; thus, as it increases, the system moves further from equilibrium, y3 starts to oscillate, and the term (ρ-y3) changes constantly, affecting y2 in equation(2) and leading y2 from linear existence to complex oscillation too. Eventually, the system becomes chaotic. However, changing the other parameters, σ and β, only affects the shape of the chaotic attractor or the critical value for chaos, not causing dramatic changes in the system's dynamics like ρ does.

3.2. Lorenz attractor

According to Deterministic Nonperiodic Flow, the attractor is defined as the region where each trajectory must ultimately become trapped in the region where Q<Q1 [1]. For the Lorenz equations, all trajectories are attracted to a set of zero volume, which is a fractal structure, simultaneously non-periodic, sensitive to initial conditions, and singularly hyperbolic [1]. The strange attractor of the Lorenz equations can thus be summarized as possessing the following characteristics: a butterfly-shaped, non-periodic, never-intersecting, fractal structure sensitive to initial conditions; it represents the long-term behavior of the Lorenz system.

3.3. Equilibrium point analysis

To solve for the equilibrium points of the Lorenz equations, it is sufficient to find the steady-state solutions where all three equations (1)–(3) equal zero, namely: the origin (0,0,0) and the non-trivial critical points (β(ρ−1), β(ρ−1), ρ−1) and $\left(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1\right)$, as stated by Lorenz in his paper.

Regarding the stability of these points, through linearization and eigenvalue analysis of the Jacobian matrix at each equilibrium point, it can be concluded that for the Lorenz equations: When 0<ρ<1, the origin is stable, while the other two points do not exist; When 1<ρ<24.74, the origin is unstable, while the other two points are stable; When ρ>24.74, all three equilibrium points are unstable, and the system enters chaos. The value 24.74 comes from the theory of subcritical Hopf bifurcation, when the parameter exceeds this value, the equilibrium points lose stability, as shown in Sparrow's literature [5].

4. Methodology

This paper applies the 4^{th} order Runge-Kutta method as a numerical solution to the Lorenz equation and uses Python to visualize the Lorenz attractors.

4.1. Numerical method selection

Since chaotic systems are extremely sensitive to initial conditions and errors, the numerical simulation method required in this study should have high-order accuracy and certain numerical stability, having an absolutely stable region covering the larger eigenvalues of Lorenz system; thus, basic first-order explicit schemes, such as the Euler method [6] or other methods that only have low-

DOI: 10.54254/2753-8818/2026.HZ28787

order accuracy or poor stability, are excluded. Through certain investigations, the following are several methods that can be used for numerical simulation of chaotic systems.

4.1.1. Adams-Bashforth-Moulton 4th order predictor-corrector (ABM4)

ABM4 includes the results obtained by using the explicit method Adams-Bashforth formula prediction and the implicit method Adams-Moulton formula correction, both of which are fourth-order accuracy. Its formula is as a

predictor equation (4) and a corrector equation (5) [7]:

$$y_{n+1}^{P} = y_n + \frac{h}{24}\left(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right) \tag{4}$$

$$y_{n+1} = y_n + \frac{h}{24}\left(9f_{n+1}^{P} + 19f_n - 5f_{n-1} + f_{n-2}\right) \tag{5}$$

Its advantage is that it has low calculation volume, enhancing efficiency by leveraging and retaining information from preceding computational steps [6], making it less burdensome to the hardware. But the disadvantage is that it needs an additional method to provide it with the first few steps to start, usually with RK4 in the Adams-Bashforth-Moulton method [6], making the code very bloated.

4.1.2. Backward Differentiation Formula 5th order (BDF5)

BDF5 is an implicit iteration method, extremely high in both accuracy and stability among the methods listed in this paper. This method is the most popular tools for the numerical solution of stiff ODEs as they have infinite regions of absolute stability [8] and its encoding is relatively simple, since the SciPy library on the Python platform can help quickly build BDF algorithms. Its formula is as follow equation (6) [9]:

$$\frac{137}{60}\,y_n - 5y_{n-1} + 5y_{n-2} - \frac{3}{10}\,y_{n-3} + \frac{5}{4}\,y_{n-4} - \frac{1}{5}\,y_{n-5} = hf\left(t_n,\ y_n\right) \tag{6}$$

However, for the Lorenz equation in this study, BDF5 seems excessive: the computer experienced stuttering when using this method, but the results were not significantly different from the results of ABM4 or RK4. In other words, as a consequence of these restrictions, BDF-codes are inefficient for high accuracy demands [8].

4.1.3. Runge-Kutta Method 4th order (RK4)

RK4, having 4th accuracy, investigations taken into account both accuracy, efficiency, and stability among most of the numerical solutions of the Lorenz equations. It is derived from a modification of the Taylor Series, and is obtained by finding the four slopes of the line obtained from the function of the differential equation [7]. The following is the operation mode of RK4 [6]:

DOI: 10.54254/2753-8818/2026.HZ28787

$$K_1 = hf\left(t_n, y_n\right) \tag{7}$$

$$K_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{K_1}{2}\right) \tag{8}$$

$$K_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{K_2}{2}\right) \tag{9}$$

$$K_4 = hf\left(t_n + h, y_n + K_3\right) \tag{10}$$

$$y_{n+1} = y_n + \frac{K_1 + 2K_2 + 2K_3 + K_4}{6} \tag{11}$$

This method is beneficial for not required the solution of implicit equations and is self-initiating. Furthermore, its local truncation error is $O(h^5)$ [10], which means that the error of each step is small, so it can give a reasonable solution in relatively large step sizes. All of these make RK4 an efficient method to solve the Lorenz equation.

4.1.4. Conclusion

Overall, RK4 is the most suitable numerical method for this study. The Lorenz system is not a rigid system, and the main factors limiting its simulation are initial values and parameter sensitivity, while RK4 has a better overall advantage than BDF5 in terms of accuracy and efficiency. At the same time, for ABM4, all the shortcomings described above also make it impossible to compete with RK4, which are shown in the table 1 below:

Table 1: Parallel visual comparison table of explicit Euler method, ABM4, BDF5, and RK4

	Euler Method(Ex)	ABM4	BDF5	RK4
Order	1	4	5	4
Local truncation error	O(h^2)	O(h^5)	O(h^6)	O(h^5)
Stability	Very poor	Strong	Moderate	Good
Efficiency (evals per step)	Excellent 1	Very Good 2	High cost Iteration	Good 4
Self-starting	Yes	No	No	Yes

4.2. Code design

This paper uses Python for visualizing the Lorenz attractor by RK4, because Python code can run seamlessly on mainstream operating platforms such as Windows, macOS and Linux; it has the

DOI: 10.54254/2753-8818/2026.HZ28787

powerful visualization capabilities provided by matplotlib; and it is also the most proficient code software for coding.

4.2.1. Overview of the structure

The code mainly contains 5 parts: Lorenz Equations defining part, RK4 numerical integrator, setting of initial conditions and parameters, numerical solution part and 3D visualizing part, as shown in Appendix A. Two libraries, numpy and matplotlib, were imported for numerical computation and 3D plotting respectively. This modular code scheme is clear, and users can adjust the parameters and initial conditions at will to output different attractors to achieve the goals of this project.

4.2.2. Stability verification

For parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, the characteristic butterfly shape of the Lorenz attractor should be visually inspected using a 3D graph generated by Matplotlib, according to Deterministic Nonperiodic Flow [1].

4.2.3. Initial parameter setting

According to historical tradition, that is, the combination of parameters used by Eduard Lorenz in his original paper Deterministic Nonperiodic Flow in 1963: in his article, the long-term trajectory of the Lorenz equation under this parameter highlights the characteristics of its chaotic system; the parameter setting is $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, and the initial condition is set to [1.0, 1.0, 1.0] is a simple and off-origin initial value that can quickly converge to the chaotic attractor. In the two experiments of this project, one of the initial conditions or parameters is adjusted, but the remaining values remain unchanged overall.

4.3. Experiment design

4.3.1. Experiment 1: attractor under different parameters

The first experiment aims to explore the change of the Lorenz attractor when changing Rayleigh number ρ. Run the Numerical simulations of the Lorenz equation with initial conditions of [1.0, 1.0, 1.0], while the parameters are set to $\sigma = 10$, $\beta = 8/3$, $dt = 0.01$, and $t = 0$ to 100, taking different $\rho \in \{0, 13, 23, 25, 28, 50\}$ each time. Then observe the change in the shape of the Lorenz attractor and mark down the equilibrium point. In terms of code, the code shown in Appendix B needs to be inserted in the order of Comments based on the code shown in Appendix A.

4.3.2. Experiment 2: sensitivity of initial condition

The Second experiment aims to explore the change of the result in the Lorenz equations when slightly changing the initial condition. Numerical simulations of the Lorenz equation with two initial conditions of [1.0, 1.0, 1.0] and [1.0001, 1.0, 1.0] are run, while the parameters are set to $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, $dt = 0.01$, and $t = 0$ to 100. Subsequently, calculate the Euclidean distance for each time step; Plot the curve to show the growth trend of the Euclidean distance over time; then plot the logarithmic curve of that curve, and then the linear fitting line of the logarithmic curve in appropriate area, in order to approximate the Lyapunov index, which is the slope of this fitting line,

DOI: 10.54254/2753-8818/2026.HZ28787

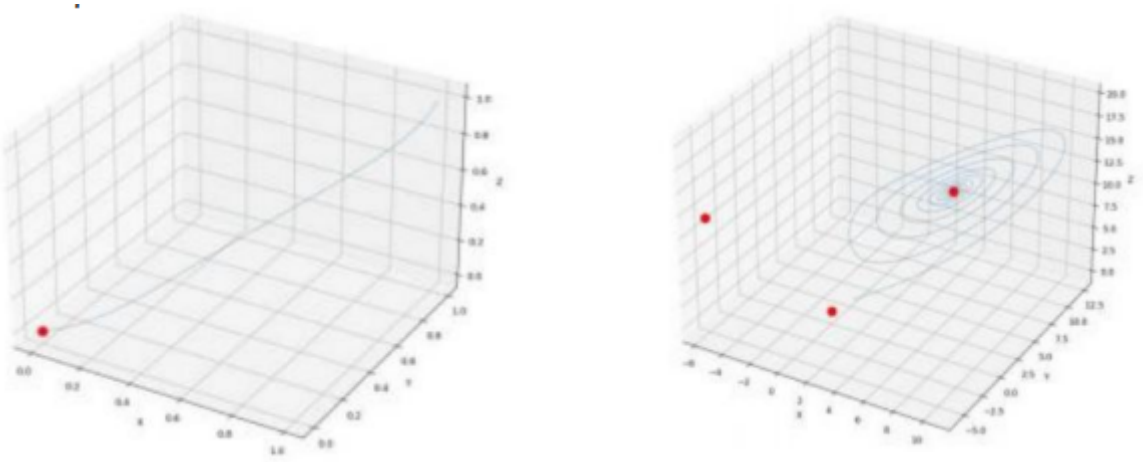
quantifying whether the system has entered chaos. To achieve such steps, the code in Appendix C should be put after the code in Appendix A and run.

5. Result and discussion

5.1. Result of experiment 1

The first experiment shows how changes in the Rayleigh number ρ affect the long-term trajectories of the Lorenz equations, especially at the critical thresholds predicted by bifurcation theory, ρ= 1 and ρ≈24.74. with initial conditions of [1.0, 1.0, 1.0], taking different ρ∈{0, 13,23, 25, 28,50} each time.

From the results in Figure 1, when ρ=0, the system's trajectories quickly converge to the origin (0, 0, 0), with the attractor appearing as a fixed point (0, 0, 0), physically representing the cessation of dynamic motion due to no convection. When ρ= 13 and 23, the system's trajectories spiral and converge to one of the equilibrium points, with the attractor transitioning to a limit cycle around two stable equilibrium points, physically indicating an ordered convection pattern. Finally, when ρ increases to 25 and above (such as ρ=28 and 50), the system enters chaos, with the trajectories forming the characteristic butterfly-shaped fractal structure of the Lorenz attractor, and the attractor exhibits a double-wing structure; at the same time, the size of the attractor increases with the value of ρ .



(a) (b)

DOI: 10.54254/2753-8818/2026.HZ28787

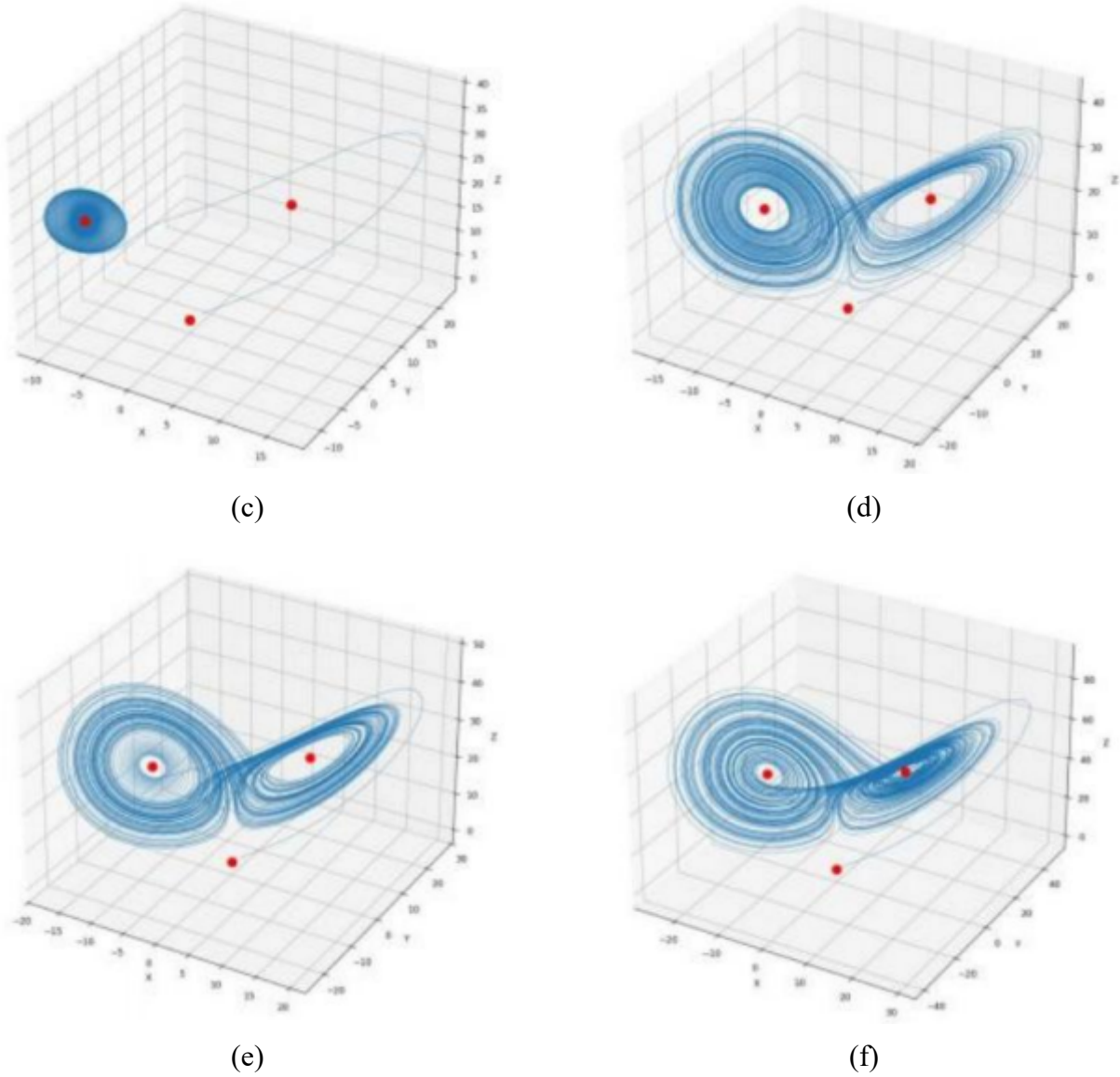


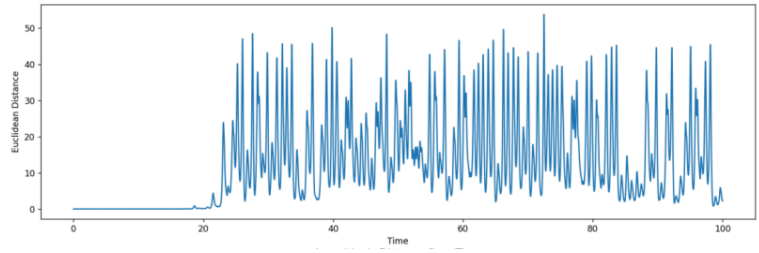
Figure 1. The 3D Numerical simulations of the Lorenz equations' trajectories and equilibrium points. (a) Lorenz attractor σ= 10, ρ=0, β=8/3 (b) Lorenz attractor σ= 10, ρ=10, β=8/3 (c) Lorenz attractor σ= 10, ρ=23, β=8/3 (d) Lorenz attractor σ= 10, ρ=25, β=8/3 (e) Lorenz attractor σ= 10, ρ=28, β=8/3 (f) Lorenz attractor σ= 10, ρ=50, β=8/3

These results clearly demonstrate the transition from a fixed point to a limit cycle and then to a chaotic attractor, almost perfectly matching the phase transitions described in Lorenz's 1963 paper Deterministic Nonperiodic Flow. The images for ρ=23 and 25 show the long-term trajectories transitioning from a limit cycle to a butterfly-shaped fractal structure, which is consistent with the Hopf bifurcation theory at $\rho \approx 24.74$ [5]. The importance of this experiment's results lies in verifying bifurcation theory by showing the complete dynamic phase transition. It reveals how the chaotic attractor's geometric structure forms around unstable equilibrium points and highlights the system's dependence on parameters.

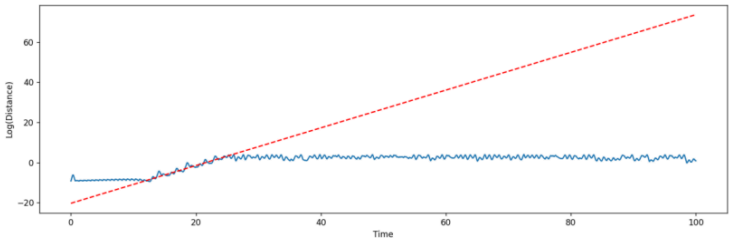
DOI: 10.54254/2753-8818/2026.HZ28787

5.2. Result of experiment 2

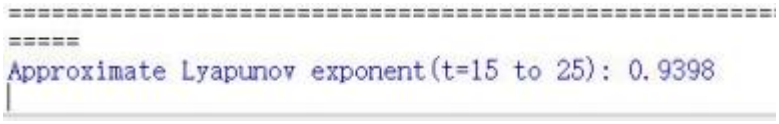
In the second experiment, two sets of trajectory simulations with very close initial conditions were run, and it can be seen that initially, the distance is small, reflecting the closeness of the initial points;



(a)



(b)



(c)

Figure 2. The growth trend of distance between trajectories (a), the logarithmic distance over time (b), and the linear fitting line fit to the log distance (c)

however, as time progresses, the distance grows exponentially and then begins to oscillate sharply, initially indicating the system's sensitivity to initial conditions according to Figure 1. In addition, by taking all the steps mentioned above, the calculated Lyapunov exponent is about 0.9398, which is very close to the theoretical value of 0.9056, according to Sparrows work [5] from chaos theory.

A positive Lyapunov exponent indicates that nearby trajectories diverge exponentially, confirming that the system has indeed entered chaos. Therefore, the results effectively demonstrate the Lorenz system's sensitivity to initial conditions; The significance of this experiment is mainly in verifying the butterfly effect and confirming chaos through the Lyapunov exponent.

5.3. The effect of RK4

The RK4 method achieved excellent results from these two tests by accurately describing the geometric structure of the Lorenz attractor and the system's dependence on its initial condition. The RK4 method not only exhibits high accuracy and stability, but can maintain this property even over

DOI: 10.54254/2753-8818/2026.HZ28787

very long times, thus always representing the correct butterfly shape when depicting the attractor for various values of R and IRI (represented in the previous section). Not only that, but it also accurately matches up the positions of the numerical equilibria. The attractor's form gives further evidence that the RK4 method successfully simulates the long-term system behavior.

The log-distance plot depicted in Figure 1 has a linear trend between time steps 15-25, supporting the associated Lyapunov exponent, which closely matches the theoretical values with only 3.8 percent. Hence, this graph further confirms the results' accuracy; in terms of simulating the Lorenz system, RK4 is proven to be an efficient and precise method for determining long-term system behavior.

5.4. Discussion

In summary, these experimental results comprehensively demonstrate the behavior of the Lorenz system after numerical simulations using the RK4 code on the Python platform, as well as the excellent performance of RK4 itself in handling the Lorenz system (including accuracy and stability). Additionally, it must be noted that these results closely align with theoretical predictions. Experiment 1 illustrates that as ρ increases, the system’s attractor transitions from a stable fixed point to a limit cycle and then to the Lorenz attractor, covering nearly the entire dynamical phase transition of the Lorenz equations, verifying the framework of bifurcation theory and demonstrating the dependence on initial conditions of the system’s long-term trajectories. Experiment 2, by calculating a positive Lyapunov exponent, confirms that the Lorenz system is a chaotic system with extreme sensitivity to initial conditions. These findings validate the theoretical foundation of the Lorenz system.

6. Conclusion and application

In short, the above two experiments have confirmed the sensitivity to the parameters and initial conditions of chaotic systems; they have also demonstrated the value of the bifurcation theorem. Furthermore, it is confirmed that the RK4 method is reliable enough when solving complex equations, such as the Lorenz equation.

Thus, the Runge-Kutta 4th order method can be used to numerically simulated other complex physical systems, such as the Multi-body pendulum,the Three-Body problem, then Duffing Oscillator, even the time-dependent Schrdinger equation. In other word, a non-rigid system composed of continuously derivable ordinary differential equations can be considered to use RK4 to solve. According to the appearance of RK4 on the Chaotic System’s numerical simulation, most of the above questions should be able to perform well.

Also, the Chaotic system itself has a great impact in meteorology, economics, chemistry, and other aspects. As the birthplace of chaos theory, Lorenz used it to state the long-term unpredictability of weather; but on the other hand, chaos theory is conducive to improving short-term predictions, and more in-depth research in the future can allow people to build better weather models. Moreover, in the field of economics, some experts attribute fluctuations such as stock prices to the inherent nonlinear structure of economic systems, which is related to chaotic systems. Even though many studies show that markets are much more complex than deterministic chaos systems, it is a good idea to start from the perspective of this system. Besides, in the field of chemistry, for chemical oscillation reactions, such as the well-known Belousov-Zhabotinsky reaction, when altering some reaction conditions, scientists can observe many behaviors of bifurcation, and get

DOI: 10.54254/2753-8818/2026.HZ28787

closer and closer to the chaotic state, which was quantitively analyze from the perspective of chaos theory.

Finally, this research can also be used as a reference for other beginners who are initially entering the study of chaos theory. It offers an practical path showing the outcomes of Chaotic system. Base on this method example, they can study other more complex systems, through numerical simulation and Python platform simulation, expand the breadth of their physics studies. Within the field of chaos theory, beginners can also explore in depth introducing with RK4 and the Lorenz system.

Therefore, this article can be used as the cornerstone of other research and provide beginners with guidance, which is the greatest value of this paper. They may also advance this study to the fractional order of Lorenz system someday, as mentioned in the literature review.

References

[1] Lorenz, E. N. (2017). Deterministic nonperiodic flow 1. In Universality in Chaos, 2nd edition (pp. 367-378). Routledge.

[2] Ahmad, N., Charan, S., & Singh, V. P. (2015). Study of numerical accuracy of Runge-Kutta second, third and fourth order method. International Journal of Computer and Mathematical Sciences, 4, 111.

[3] Rosser Jr, J. B. (2009). Chaos theory before Lorenz. Nonlinear dynamics, psychology, and life sciences, 13(3), 257.

[4] SALIHI, Y., XHAFERI, M., RASIMI, K., IBRAIMI, A., & SADIKI, F. (2023). RK4 AND VECTORIAL RK4-USED TO ANALYZE A MODIFIED FORM OF FRACTIONAL ORDER LORENZ SYSTEM. Journal ofNatural Sciences and Mathematics of UT, 8(1516), 425-431.

[5] Sparrow, C. (2012). The Lorenz equations: bifurcations, chaos, and strange attractors (Vol. 41). Springer Science & Business Media.

[6] Olatunji, O., & Akeju, A. (2025). Comparative Analysis of Euler and Order Four Runge-Kutta Methods in Adams-Bashforth-Moulton Predictor-Corrector Method. International Journal of Mathematics, Statistics, and Computer Science, 3, 276-293.

[7] Lesnussa, Y. A., & Marsudi, M. (2025). COMPARATIVE ANALYSIS OF NUMERICAL SOLUTIONS OF EULER, RK-4, ABM-4 AND RKCOM4 METHODS OF INITIAL VALUE PROBLEMS IN NONHOMOGENEOUS SECOND ORDER DIFFERENTIAL EQUATIONS. Parameter: Jurnal Matematika, Statistika dan Terapannya, 4(1), 123-140.

[8] Fredebeul, C. (1998). A-BDF: a generalization of the backward differentiation formulae. SIAMjournal on numerical analysis, 35(5), 1917-1938.

[9] Gear, C. W. (1971). Numerical initial value problems in ordinary differential equations. Prentice Hall PTR.

[10] Ascher, U. M., & Greif, C. (Eds.). (2011). A first course on numerical methods. Societyfor Industrial and Applied Mathematics.

Appendix.A

```
import numpy
# 1. Lorenz Equations
def lorenzsys(t, allxyz, sigma, rho, beta):
    x, y, z=allxyz
    return numpy.array([sigma*(y-x), x*(rho-z)-y, x*y-beta*z])

# 2. 4th order Runge-Kutta Method
def RK4(func, t, allxyz, dt, *args):
    k1=func(t, allxyz, *args)*dt
    k2=func(t+dt/2, allxyz+k1/2, *args)*dt
    k3=func(t+dt/2, allxyz+k2/2, *args)*dt
    k4=func(t+dt, allxyz+k3, *args)*dt
    return allxyz+(k1+2*k2+2*k3+k4)/6

# 3. initial Settings
sigma, rho, beta=10.0, 32, 8.0/3.0
t0, tf, dt=0, 100, 0.01
steps=int((tf-t0)/dt)
tp, xp, yp, zp=numpy.linspace(t0, tf, steps), numpy.zeros(steps), numpy.zeros(steps), numpy.zeros(steps)
xp[0], yp[0], zp[0]=allxyz=numpy.array([1.0, 1.0, 1.0])

# 4. Integration
for i in range(1, steps):
    allxyz=RK4(lorenzsys, tp[i-1], allxyz, dt, sigma, rho, beta)
    xp[i], yp[i], zp[i]=allxyz

# 5. 3D Visualization
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(15, 10))
ax=fig.add_subplot(111, projection='3d')
ax.plot(xp, yp, zp, lw=0.5)
ax.set(xlabel='X', ylabel='Y', zlabel='Z', title=f'Lorenz Attractor (σ={sigma}, ρ={rho}, β={beta:.3f})')
plt.show()
```

DOI: 10.54254/2753-8818/2026.HZ28787

Appendix.B

```
#4.5
equi=[[0.0,0.0,0.0]]
if rho>1:
    zeq=rho-1
    xyeq=numpy.sqrt(beta*zeq)
    equi.extend([[xyeq,xyeq,zeq],[-xyeq,-xyeq,zeq]])
#5.5
for i, (x,y,z) in enumerate(equi):
    ax.scatter(x,y,z,s=100,c="red")
```

Appendix.C

```
from scipy.stats import linregress
sigma,rho,beta=10.0,28,8.0/3.0
t0,tf,dt=0,100,0.01
steps=int((tf-t0)/dt)

tp,xp1,yp1,zp1=numpy.linspace(t0,tf,steps),numpy.zeros(steps),numpy.zeros(steps),numpy.zeros(steps)
xp1[0],yp1[0],zp1[0]=allxyz=numpy.array([1.0,1.0,1.0])
for i in range(1,steps):
    allxyz = RK4(lorenzsys,tp[i-1],allxyz,dt,sigma,rho,beta)
    xp1[i],yp1[i],zp1[i]=allxyz
tp,xp2,yp2,zp2=numpy.linspace(t0,tf,steps),numpy.zeros(steps),numpy.zeros(steps),numpy.zeros(steps)
xp2[0],yp2[0],zp2[0]=allxyz=numpy.array([1.0001,1.0,1.0])
for i in range(1,steps):
    allxyz=RK4(lorenzsys,tp[i-1],allxyz,dt,sigma,rho,beta)
    xp2[i],yp2[i],zp2[i]=allxyz

distances=numpy.sqrt((xp1-xp2)**2+(yp1-yp2)**2+(zp1-zp2)**2)
fig2=plt.figure(figsize=(15,10))
ax1=fig2.add_subplot(211)
ax1.plot(tp,distances)
ax1.set(xlabel='Time',ylabel='Euclidean Distance',title='Growth Trend of Distance Between Trajectories')

logno0=numpy.log(distances+0.000000001)
ax2=fig2.add_subplot(212)
ax2.plot(tp,logno0)
inter=(tp>15)&(tp<25)
result=linregress(tp[inter],logno0[inter])
fit=result.intercept+result.slope*tp
ax2.plot(tp, fit,'r--')
ax2.set(xlabel='Time',ylabel='Log(Distance)',title='Logarithmic Distance Over Time')

plt.show()
print(f"Approximate Lyapunov exponent(t=15 to 25): {result.slope:.4f}")
```