The Lorenz Attractor - the Lorenz butterfly

Yi-Jing Sie

Abstract

The goal of this post is to explore the chaotic behavior in the Lorenz attractor. [1] .

1. Introduction

Most people may have heard about the butterfly effect, which states that "The flap of a butterfly's wing in Brazil can set off a cascade of atmospheric events that, weeks later, spurs the formation of a tornado in Texas." [3] Few people know, however, that this idea actually stemmed from the chaotic behavior in the Lorenz attractor[1]. Before we dive into what the Lorenz attractor is, let's talk about The Lorenz system.

The Lorenz system is a simplified mathematical system of three ordinary differential equations developed by Edward Lorenz to model the atmosphere convention :

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

As one can see from above, the Lorenz system represents the change of three quantities (x, y, z) over time. Feeding these equations with a set of well-known parameter values

$$\sigma = 10,\ \beta = \frac{8}{3},\ \rho = 28$$

into the computer to visualize the data, what Lorenz system shows is an elegant curve resembling butterfly wings, known as the chaos behavior of the system, the Lorenz attractor.

Chaotic behavior exists in many natural and artificial systems, such as economics, climate, the stock market, etc. It describes the behavior of models exquisitely sensitive to small changes in initial conditions. That is, even the minuscule disturbance to the starting positions could lead to a completely different system. This kind of behavior causes a quandary of predictability in a real-world setting since we can never get the exact value of any physical measurement.

This beautiful butterfly-wing-like curve never intersects itself at any given time, so it never retraces its path. Instead, it orbits around two critical points

$$(\sqrt{\beta(\rho-1)},\ \sqrt{\beta(\rho-1)},\ \rho-1) \text{ and } (-\sqrt{\beta(\rho-1)},\ -\sqrt{\beta(\rho-1)},\ \rho-1)$$

back and forth forever.

Note that the critical points do not depend on the initial points.In fact, if we apply

$$\sigma = 10,\ \beta = \frac{8}{3},\ \rho = 28$$

to the Lorenz system, the critical points for this Lorenz attractor are $(-\sqrt{72} \approx -8.49, -8.49, 27)$ and $(8.49, 8.49, 27)$.

2. Problem

To demonstrate what Lorenz attractor looks like and its chaotic behavior, I am going to plot some Lorenz attractors starting from different positions for 10000 time steps with a time step size of 0.01.

2.1. Butterfly-like Curve

Let's start with one Lorenz attractor to see the butterfly-like curve. This Lorenz attractor begins at (1, 1, 1), and its critical points are the blue dot at (–8.49, –8.49, 27) and the purple dot at (8.49, 8.49, 27). Figure 1 shows the resulting Lorenz attractor looked from the top and the front.

2.2. Chaotic Behavior

This subsection demonstrates how an infinitesimal change to the initial positions of the Lorenz attractor can result in a completely different system. Figure 2 depicts two Lorenz attractors, one in red and the other in black: The black trajectory represents the Lorenz attractor starting at (1, 1, 1), and the red trajectory represents the Lorenz attractor starting at (1.00001, 1.00001, 1.00001). We can see that the divergence of the Lorenz attractors is conspicuous even with a 1/100000 difference in the x, y, and z-coordinates between the starting points.

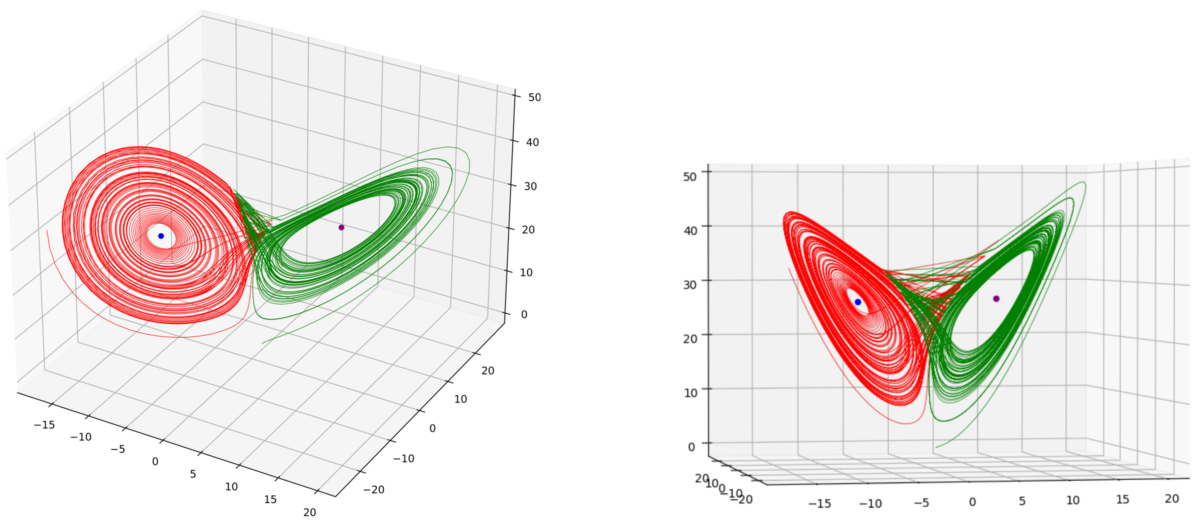

Figure 1: The Lorenz attractor with a starting point at (1, 1, 1)

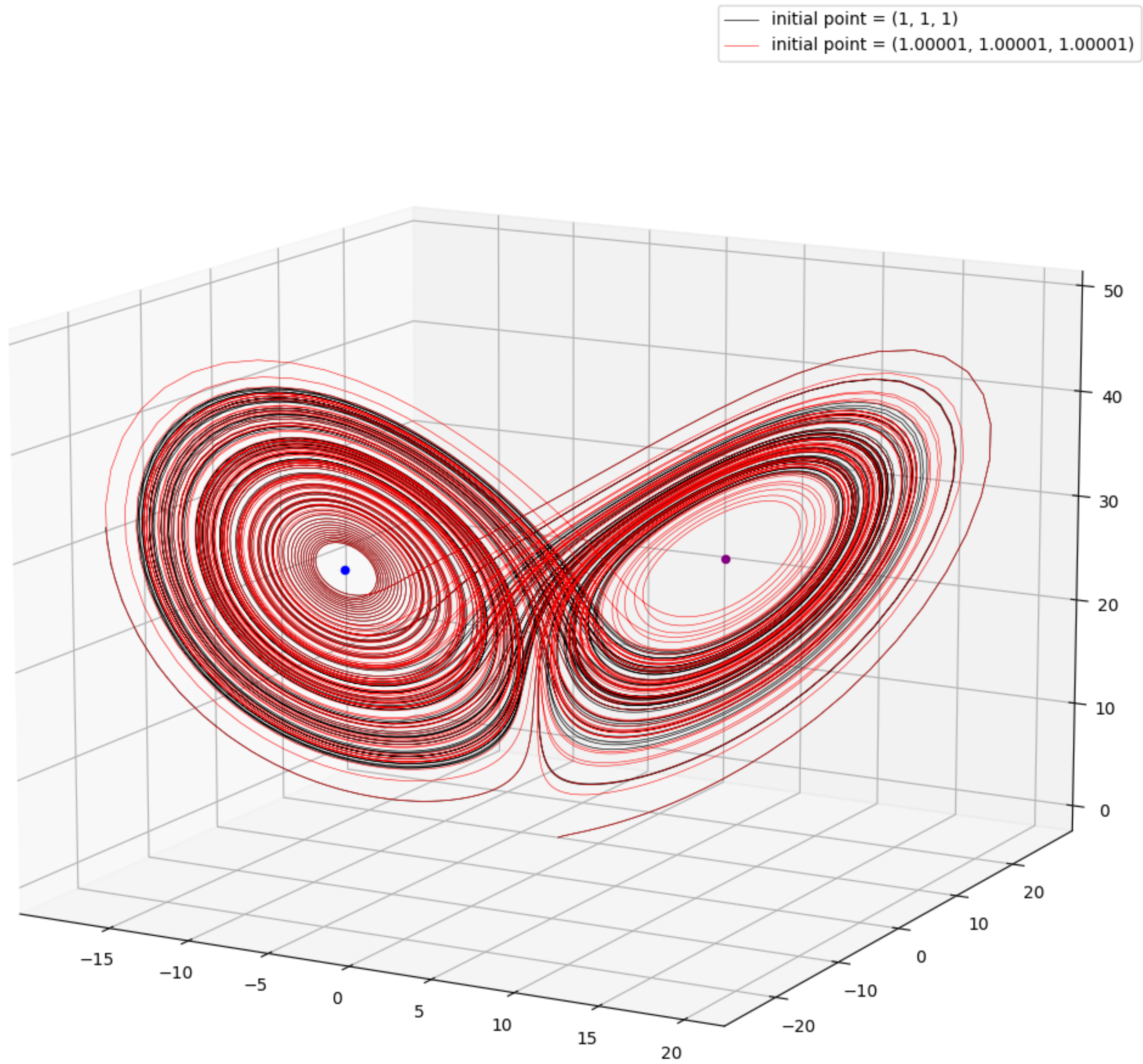

Figure 2: Two Lorenz attractors starting at different initial points

3. Scalability

For the scalability of this project, I first employed ECE HTCondor resources provided by Carnegie Mellon University to run 1,000 simulations concurrently, each of which generates a Lorenz attractor with a perturbed starting point computed by adding random numbers varied between 0 and 0.3 to (1, 1, 1). Each Loren attractor was plotted with

$$\sigma = 10,\ \beta = \frac{8}{3},\ \rho = 28$$

for 10000 time steps where the time step size = 0.01. The results of each simulation were saved to a CSV file. In other words, there would be 1000 CSV files eventually, each of which contains 3 columns and 10000 rows, with the first column representing the x–coordinate, the second the y–coordinate, and the last column the z–coordinate, and each row representing a time step.

Secondly, I utilized PySpark, an interface for Apache Spark in Python, to execute my code on Bridges-2, a Pittsburgh Supercomputing Center's supercomputer[2], computing in parallel the average time steps taken by these 1000 Lorenz attractors while circling each critical point. All of the 1000 Lorenz attractors have the same critical points as those in Figure 1 and Figure 2, that is, the blue dot at (–8.49, –8.49, 27) and the purple dot at (8.49, 8.49, 27).

To determine which critical points that any of the simulated Lorenz attractors is circling at some time step, the distances between the current coordinate and each of the critical points were computed and compared. Below is the resulting table:

The average time steps taken by 1000 Lorenz attractors while circling the points			
Critical point	Mean	Variance	Standard Deviation
(–8.49, –8.49, 27)	5616.93	196316	443.076
(8.49, 8.49, 27)	4383.07	196257	443.009

Figure 3 and Figure 4 are histograms displaying the frequency distribution for the total number of time steps taken by the Lorenz attractors while circling the critical point at (–8.49, –8.49, 27) and at (8.49, 8.49, 27) respectively.

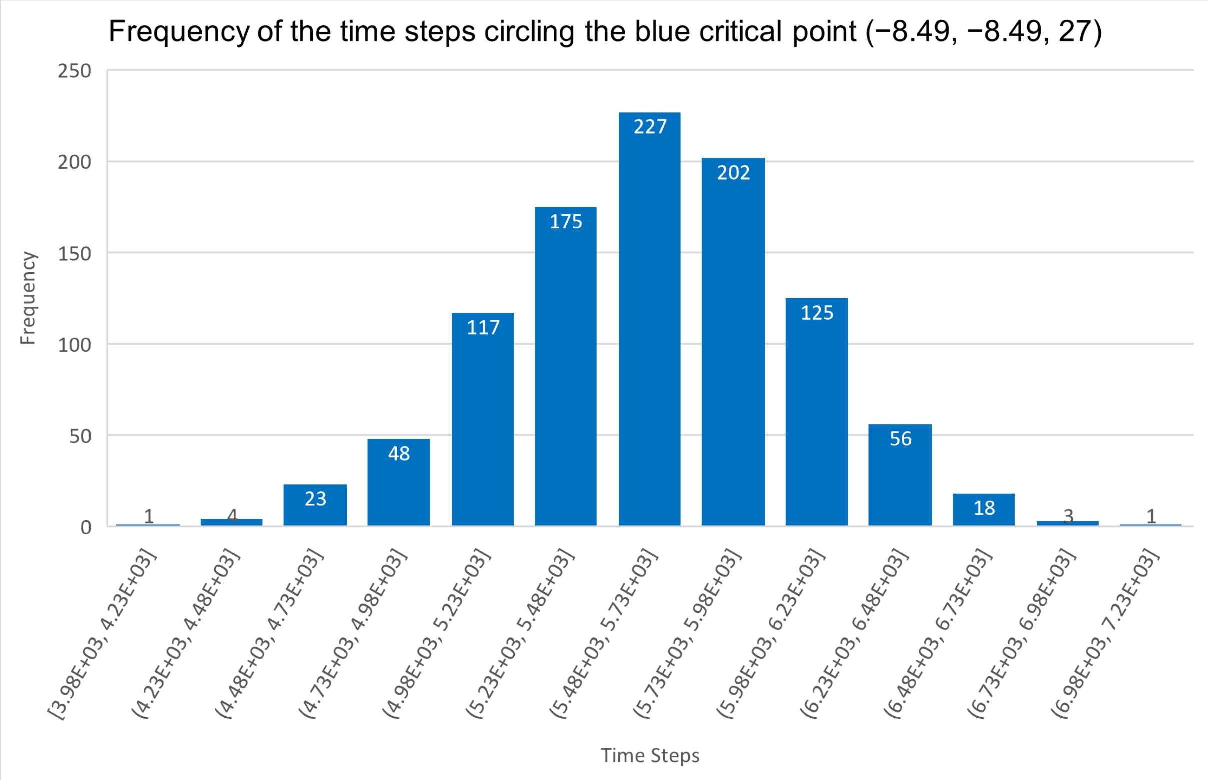

Figure 3: The histogram displaying the frequency of the total number of time steps the simulated Lorenz attractors took while circling the critical point at (–8.49, –8.49, 27) (the blue critical point)

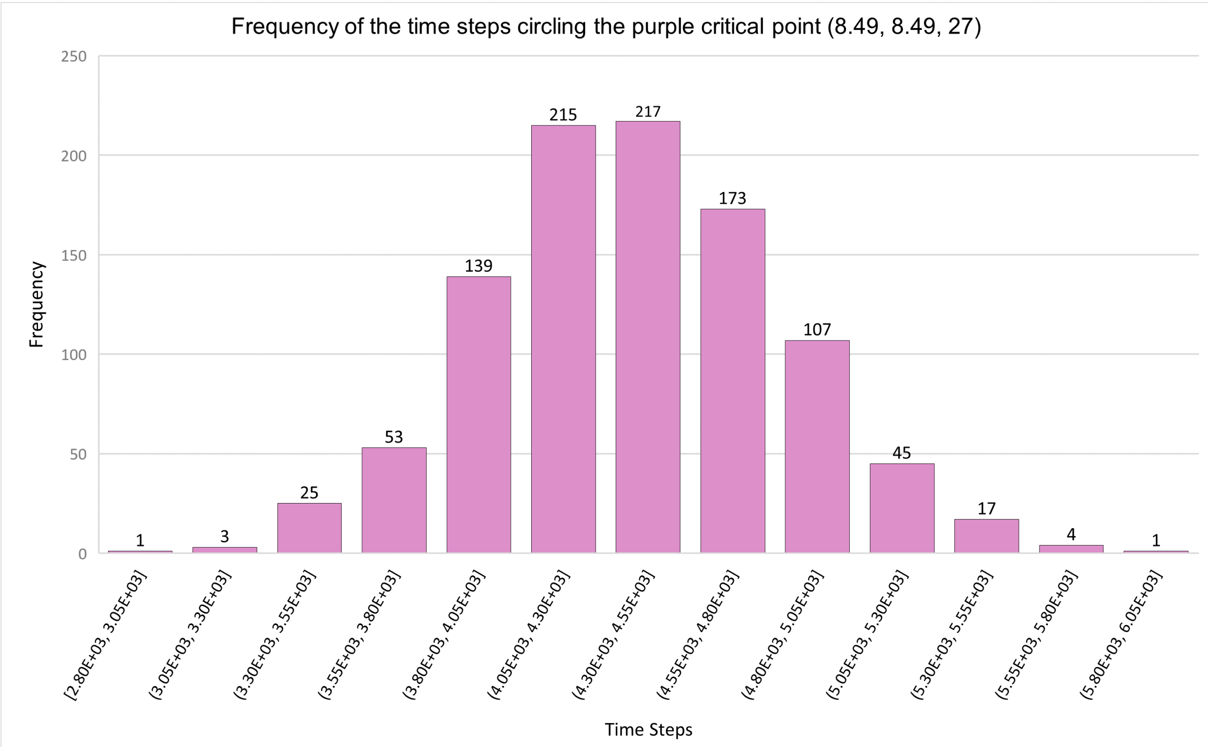

Figure 4: The histogram displaying the frequency of the total number of time steps the simulated Lorenz attractors took while circling the critical point at (8.49, 8.49, 27) (the purple critical point)

References

[1] Edward N Lorenz. “Deterministic nonperiodic flow”. In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141.

[2] *MS Windows NT Kernel Description*. Accessed: 05.10.2021. URL: https://www.psc.edu/resources/bridges-2/.

[3] Natalie Wolchover. *Can a Butterfly in Brazil Really Cause a Tornado in Texas?* URL: https://www.livescience.com/17455-butterfly-effect-weather-prediction.html. (accessed: 05.16.2021).