**CST-305: Benchmark – Project 5 – Self-Organized Criticality**

Jordan Scott and Angel

Grand Canyon University

CST-305: **Principles of Modeling and Simulation Lecture & Lab**

Ricardo Citro

March 24, 2024

**CST-305: Benchmark – Project 5 – Self-Organized Criticality**

**Responsibilities:**

We both worked on the development in the Lorenz’s code and both took part in working on the documentation.

**System Performance in Context:**

The efficiency of memory allocation for storing files can be affected by the creation and deletion of files on the storage device, potentially leading to fragmentation and gaps. This could pose challenges in the runtime for saving, loading, and reading files, possibly resulting in system slowdowns.

**Specific Problem Solved:**

A model will be developed to illustrate the dynamic nature of a system that exhibits the chaos and attributes of a self-organized file system. Through this, we can gauge the durability of the file system and predict when it will fall below the critical threshold.

**Mathematical Approach:**

To approach this, we will use the Lorenz system, where the system of differential equations has 2 nonlinear terms given by

Where all the parameters are assumed to be positive and  **>** b + 1. This system is X’ = L(X) to use this in context of 3D, we can have the parameters equal to file sizes such as

x = file size of 100MB (video file)

y = file size of 5MB (Photo file)

z = file size of 350MB (Video file)

In order to analyze the system, we must find the equilibria. To do this we use

Such that the latter 2 parts of the equilibria exists when r > 1, and linearizing the system we get a matrix of

Y’ = Y

In the origin, the eigen values of the matrix shown above are -b and

Where the eigenvalues are negative when 0 <= r < 1  
  
We used 3 files that were in Megabytes.

X: The first file was a video file was a 4K video and took up 100MB of space.  
Y: The second file was a photo file with 5MB of space.

Z: The third file was another video file and it was shot at the same resolution (4k) but was a little longer and took up 350MB of space.

**Implementation in Code:**

This code implements the Lorenz system, a mathematical model describing a chaotic, deterministic dynamical system. We utilize NumPy for numerical computations and Matplotlib for plotting.

We first define the `lorenz` function (system's dynamics through a set of differential equations) based on parameters “x”, “y”, “z”, “r”, “s”, and “b”. Next it does the calculation for “x\_dot”, “y\_dot” and “z\_dot” and returns all of those values. The simulation is conducted over a specified number of time steps, with initial values set for the state variables “x”, “y”, and “z”.

The user is prompted to input three different values of “r”, corresponding to low, medium, and high values. For each `r` value, the system's evolution is computed and plotted individually for the `x`, `y`, and `z` axes, as well as in 3D space to visualize the Lorenz attractor. This allows for the exploration of the system's behavior under various conditions and parameter values, providing insight into its chaotic dynamics.

**Flowchart:**

A screenshot of a computer

Description automatically generated

**Screenshots:**

**Non-Chaotic:**

**A screenshot of a graph

Description automatically generatedA group of graphs showing different sizes and numbers

Description automatically generated with medium confidence**

Lorenz Attractor with r value of 5.   
  
Here we see that the system is not very chaotic. The 3d graph shows the spiral but, on the x(t), y(t), and z(t) as time moves forward the system is stable due to the flat line on the graphs.

**Semi-Chaotic:**

A graph of an attraction

Description automatically generated

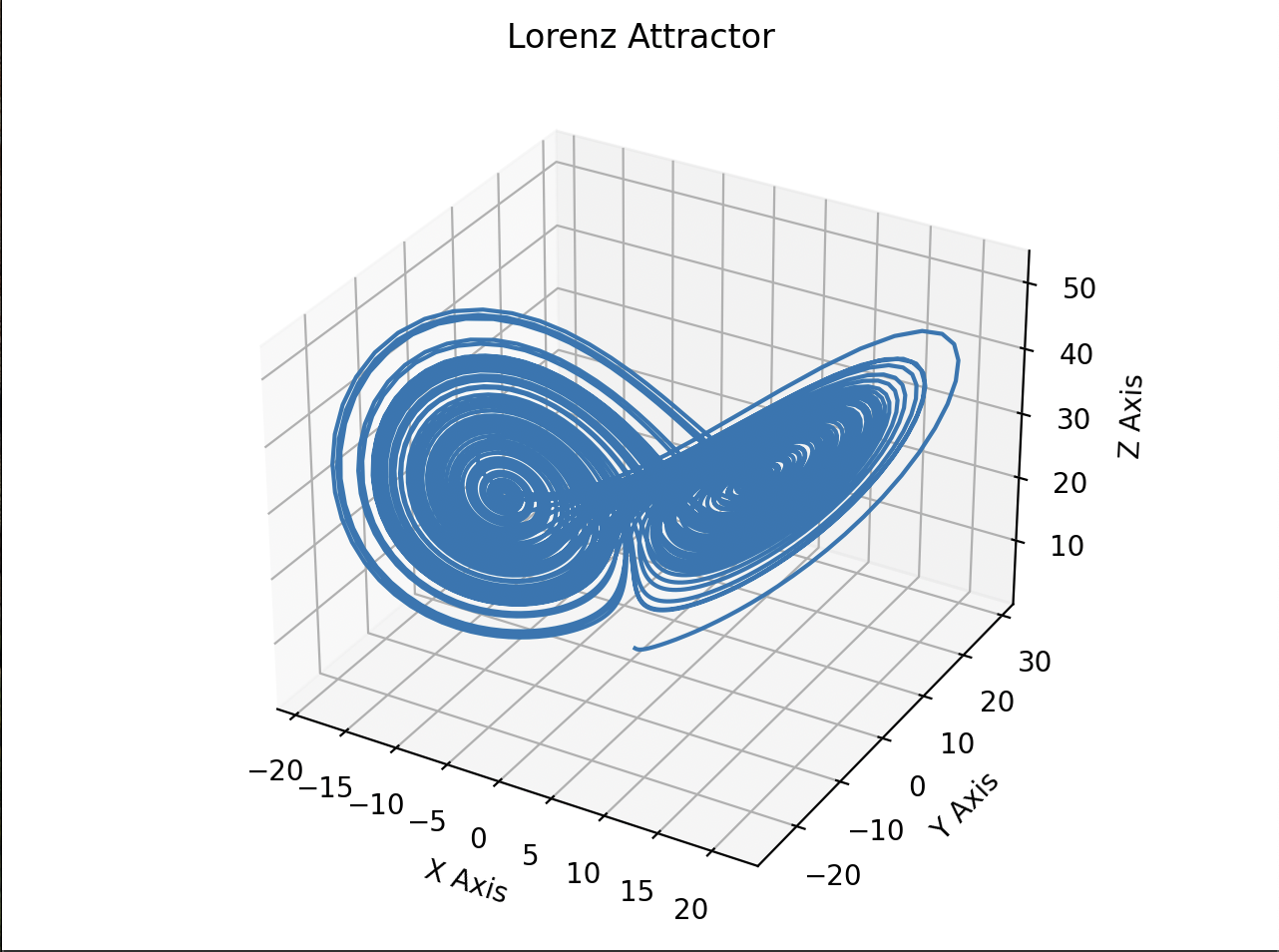
**A group of graphs showing different types of data

Description automatically generated with medium confidence**

**Lorenz Attractor with r value of 10.**

**The system got a little more chaotic, but it is still not fully chaotic. The 3d graph shows the line spiraling towards the center but the x(t), y(t), and z(t) graphs say something different. At the beginning the system jumps around but as time goes on the line goes flat, stabilizing.**

**Chaotic:**

****

**A group of blue lines

Description automatically generated**

Lorenz Attractor with r value of 28.

Here the system is completely chaotic. The 3d graph shows two spirals that are merged. On the x(t), y(t), and z(t) graphs, the jumps are erratic and over time they do not stabilize and are

In summary, as the r value rises, the system transitions into chaos due to the spiral converging into another spiral. The graphs of x(t), y(t), and z(t) exhibit increased chaotic behavior with higher r values, as the system's response becomes increasingly erratic over time.

GitHub link: <https://github.com/angel-vlzqz/Modeling-and-Simulation/tree/main/projects/project%205%20CLC>