**­­­**

**Project 5 – Self Organized Criticality**

Trevor G. Pope & Ben C. Carter

Grand Canyon University

CST-305: Principles of Modeling and Simulation

Professor Ricardo Citro

November 12th, 2023

**Responses and Completed Tasks**

* **Trevor Pope**

Wrote the specific problem solved, did the flowchart, and ReadMe.

* **Benjamin Carter**

Wrote the program, wrote out the mathematical approach.

**System Performance**

This program runs a python plotting library, a python scientific mathematical package, and a python script implementing the Lorenz Function. As the program is in python, it is cross platform, and is able to run on any architecture that supports a python interpreter. The packages that the program uses are based off of C/C++ routines, giving increased performance verses a pure-python implementation.

**Specific Problem Solved:**

The efficiency of allocating memory is affected by repeated file creations and deletions. Eventually this will cause many files to be fragmented because of insufficient space and there will be a critical point where there is too much fragmentation that the system cannot operate anymore because the fragmentation causes save, load, and access times to be very slow. This phenomenon demonstrates deterministic chaos and self-organized criticality. We have gone ahead and implemented a program that uses the Lorenz system to display the fragmentation process over time. This particular system of differential equations demonstrates a file system where x is a file that has a size of 512 KBytes, y is a file that has a size of 256 KBytes, and z is a file that has a size of 512 KBytes.

**Mathematical approach for solving it:**

Systems of differential equations in real life can become extremely complex, involving hundreds or thousands of equations and variables. This vast complexity is too difficult to solve algebraically via a computer system, so computers resort to numerical methods to find solutions. One such way of using numerical methods to find solutions to systems of differential equations is through a Lorenz function. A Lorenz function goes step by step to approximate a solution.

The *r* value is the Rayleigh number, is the Prandtl number. The r value is of interest here as it governs how “far” the simulation can step forward. The greater the number, the more likely it can diverge and become chaotic.

Chaos. The level of “chaos” in a system is measured in three tiers, from lowest to highest chaos:

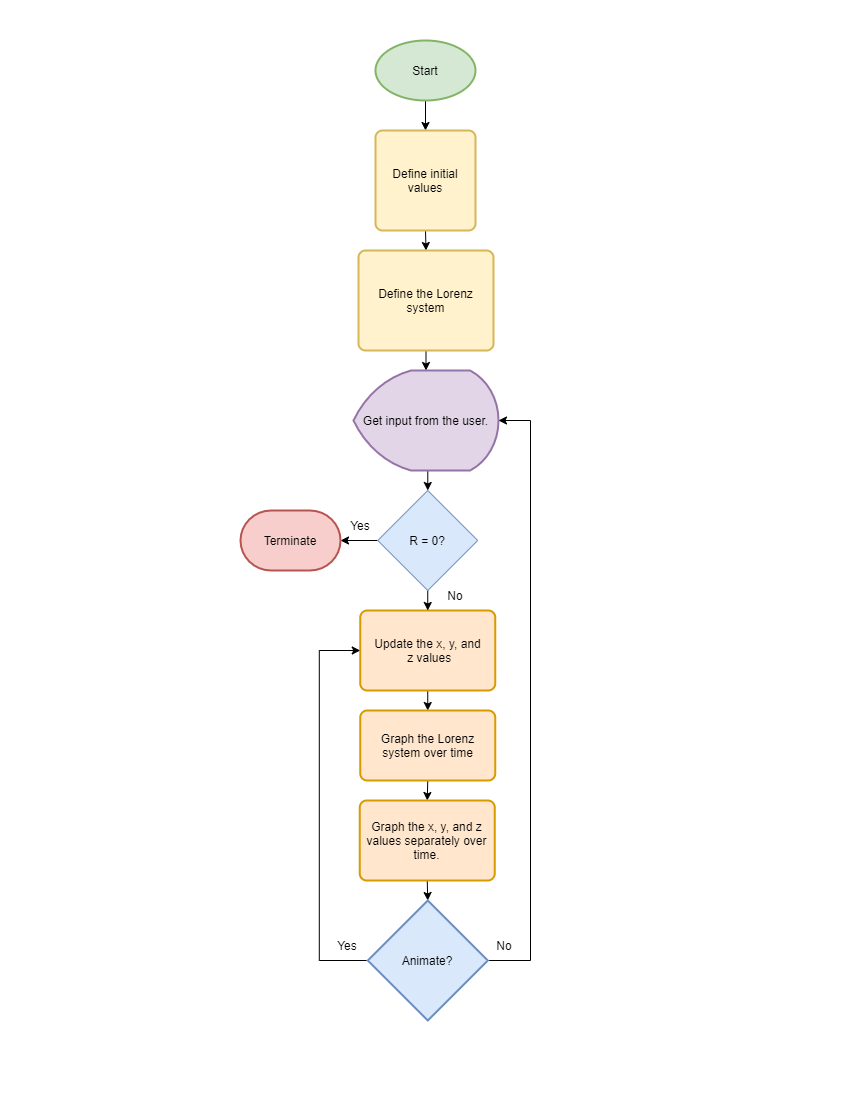
1. Deterministic Model. The solution can be accurately predicted and fully understood point by point.
2. Stochastic Model. The solution cannot be accurately predicted point by point, but it does converge to an equilibrium or attractor. An individual point can’t be calculated by itself, but rather is inferred by the equilibrium. To get individual points, a recursive strategy is required.
3. Chaotic Model. The solution either has no equilibrium/attractor, or has multiple equilibrium/attractors. This makes it impossible to even roughly guess where a point will be. The only way to calculate an individual point is to calculate previous solution points (recursive strategy).

This project tries to find the R value that will make the system turn from a stochastic model to a chaotic mode.

**Algorithm for code:**

1. Declare Lorenz system function.
2. Ask user for r.
3. Update the x, y, and z values step wise.
4. Graph the values on a 3D graph.
5. Graph the values of the x, y, and z each on separate 2D graphs.
6. Repeat (jump to step 2)

**Flowchart:**

****

**Screenshots of the program execution**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a graph

Description automatically generated

A screen shot of a graph

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screen shot of a graph

Description automatically generated

A screen shot of a graph

Description automatically generated A screen shot of a graph

Description automatically generated