Grand Canyon University

Project 5 – Self-Organized Criticality

Ryan Scott and Diego Guerra

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

Dr. Ricardo Citro

November 13, 2022

Introduction:

As a result of the creation and deletion of files over time, the fragmentation of a file system increases. Once a file system reaches a critical point, it will be unable to operate anymore and will perform slower and take longer to access files. System performance will be affected and file access times will be prolonged. Using the Lorenz equation we learned about in class, we can simulate and graph what that critical point would look like. Simply put, as a variable in the simulation gets closer to the critical point, the graph becomes more unstable.

Code for Lorenz Function:

```
import numpy as np
import matplotlib.pyplot as plt


dt = 0.01   # Step size
num_steps = 10001   # Steps
xs, ys, zs = np.empty(num_steps), np.empty(num_steps), np.empty(num_steps)  # Creates empty space
xs[0], ys[0], zs[0] = (.69, .420, .1337)   # NICE initial conditions!


def Lorenz(x, y, z, r, s=10, b=2.667):
    x_dot = (y - x) * s             # Calculates dx/dt
    y_dot = (r * x) - y - (x * z)   # Calculates dy/dt
    z_dot = (x * y) - (b * z)       # Calculates dz/dt

    return (x_dot, y_dot, z_dot)
```

With:

- Three-dimensional point of interest **x**, **y**, **z**.
- Lorenz attractor parameters **s**, **r**, and **b**.
- Initial conditions of (.69MB, .42MB, 1.337MB)

We will find the derivatives of x, y, and z in relation to time by creating a function that takes in x, y, z, and r variables and passing constants into a Lorenz function in Python. The derivative values of x, y, and z will be added to an array and plotted in three dimensions with respect to t.

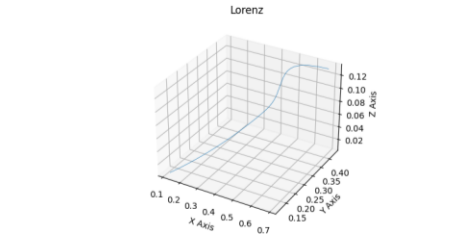
Gather Points Generated by R-Value:

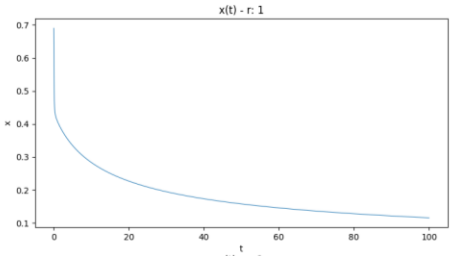
```
def R_TIME(r):
    for i in range(num_steps - 1):
        x_dot, y_dot, z_dot = Lorenz(xs[i], ys[i], zs[i], r)
        xs[i + 1] = xs[i] + (x_dot * dt)
        ys[i + 1] = ys[i] + (y_dot * dt)
        zs[i + 1] = zs[i] + (z_dot * dt)

    lin = np.linspace(0, 100, num=10001)
    plotGraphs(xs, ys, zs, lin, r)


# Test values
R_TIME(1)   # Normal
R_TIME(10)  # Entropy
R_TIME(20)  # Insanity!

# Try your own values!
while True:
    R_TIME(int(input("Enter an R value: ")))
```

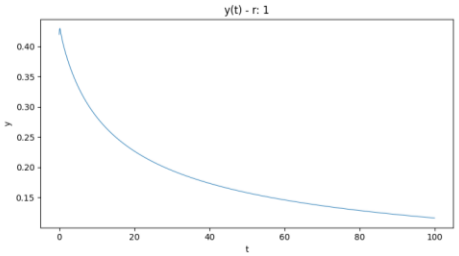
Graph the Calculated Values:

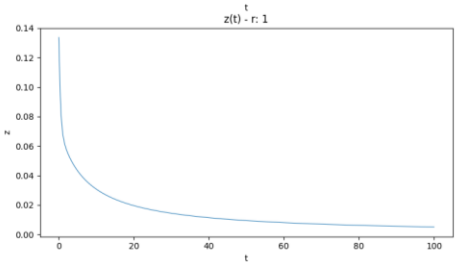
```
def plotGraphs(plot_x, plot_y, plot_z, lin, r):
    fig = plt.figure()  # Displays all graphs in the same window

    graph = fig.add_subplot(2, 2, 1, projection='3d')  # Create 3D graph
    graph.plot(plot_x, plot_y, plot_z, linewidth=0.5)  # Graph X, Y, Z in 3D
    graph.set_xlabel("X Axis")
    graph.set_ylabel("Y Axis")
    graph.set_zlabel("Z Axis")
    graph.set_title("Lorenz")

    graph = fig.add_subplot(2, 2, 2)         # Create 2nd graph
    graph.plot(lin, plot_x, linewidth=0.75)  # Plot time and X values
    graph.set_xlabel("t")
    graph.set_ylabel("x")
    graph.set_title(f"x(t) - r: {r}")

    graph = fig.add_subplot(2, 2, 3)         # Create 3rd graph
    graph.plot(lin, plot_y, linewidth=0.75)  # Plot time and Y values
    graph.set_xlabel("t")
    graph.set_ylabel("y")
    graph.set_title(f"y(t) - r: {r}")

    graph = fig.add_subplot(2, 2, 4)         # Create 4th graph
    graph.plot(lin, plot_z, linewidth=0.75)  # Plot time and Z values
    graph.set_xlabel("t")
    graph.set_ylabel("z")
    graph.set_title(f"z(t) - r: {r}")

    plt.show()
```

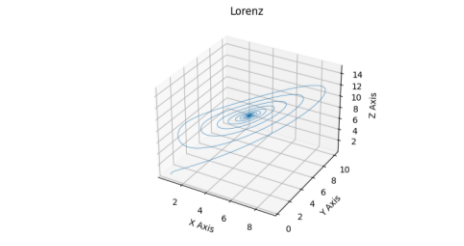
Graphs:

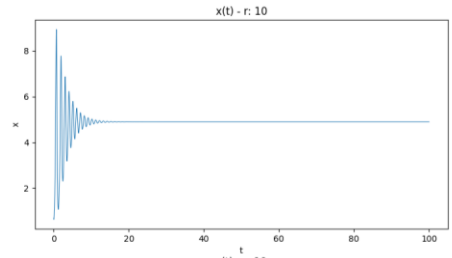

Lorenz
X Axis
Y Axis
Z Axis

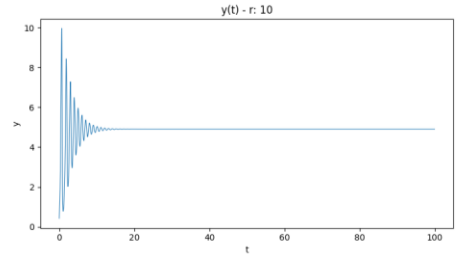

x(t) - r: 1
x
t

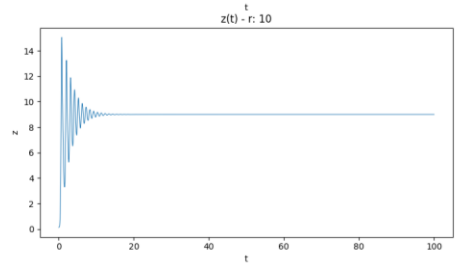

y(t) - r: 1
y
t

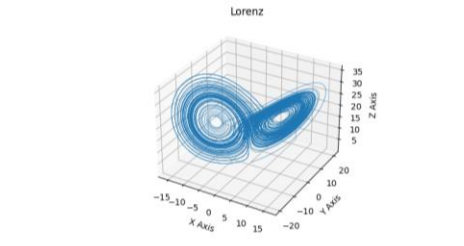

z(t) - r: 1
z
t

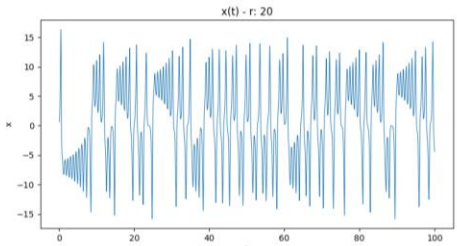

Lorenz
X Axis
Y Axis
Z Axis

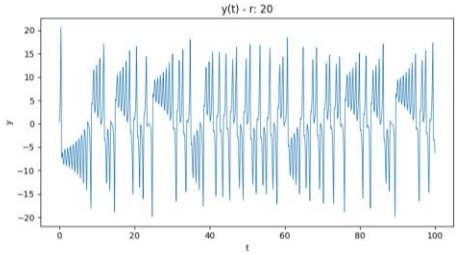

x(t) - r: 10
x
t

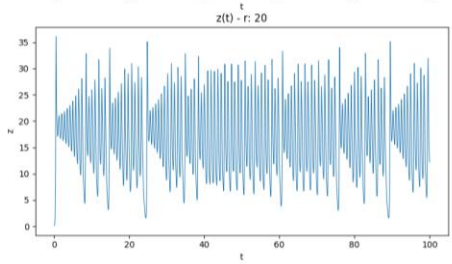

y(t) - r: 10
y
t


z(t) - r: 10
z
t


Lorenz
X Axis
Y Axis
Z Axis


x(t) - r: 20
x
t


y(t) - r: 20
y
t


z(t) - r: 20
z
t

Lorenz Graph Analysis:

There are three R values used in this analysis: R = 1, R = 10, and R = 20. Increasing R causes the following 3D graph evolution:

- A curved line
- A line orbiting around one point
- A line orbiting around two points

As you can see on these hectic graphs with high R values, the system approaches this critical point which is analogous to a file system that is no longer functional. The graphs for each individual variable show how the file sizes will change as the R variable changes with time.