­­­­

**Project 7 – Code Errors and the Butterfly Effect**

Kyungchan Im

Grand Canyon University

Course number: CST - 305

Professor Ricardo Citro

Due Apri 23, 2023

**Project 7 – Code Errors and the Butterfly Effect**

**PART 1: Define Lorenz System using Three ODEs**

**Description**

Write a computer program that implements the model and solves it (i.e., the Lorenz attractor) and plot an animated graph that visualizes it. Explain the meaning of the three parameters: σ, ρ, β. The program should be interactive, allowing the user to change the parameters of the equations and visualize the effect of the change.

**Implementation**

[ Baseline of the Lorenz Equation ]

In this project, we will use the parameters 𝜎 = 10 (𝑃𝑟𝑎𝑛𝑑𝑡𝑙 𝑛𝑢𝑚𝑏𝑒𝑟), 𝑏 = 8/3 (𝑅𝑎𝑦𝑙𝑒𝑖𝑔ℎ 𝑛𝑢𝑚𝑏𝑒𝑟), 𝑎𝑛𝑑 𝑐ℎ𝑎𝑛𝑔𝑒 𝑡ℎ𝑒 𝑣𝑎𝑙𝑢𝑒𝑠 𝑓𝑜𝑟 𝑟. Furthermore, Prandtl number and Rayleigh number in this project defines the physical size.

[ Modeling the Lorenz Equation ]

When modeling file systems using the Lorenz System, the initial values for 𝑥, 𝑦, and 𝑧 were chosen to depict file size, while 𝜎 and 𝑏 were left unchanged. The variable 𝑟 was manipulated due to its effect on stability and fixed points. In the Lorenz System, a sink origin exists when 𝑟 is greater than 1. As 𝑟 increases, the number of cycles also increases, leading to a more chaotic system.

For this project, I set the x, y, and z value for = 11.8 KBs, = 4.4 KBs, = 2.4KBs. Those represents the average size of the number of image files with different file extensions. x for JPG, y for the PNG, and z for the GIF. Therefore, 𝑟 values of 8, 15, and 28 were chosen for the project to demonstrate different levels of neutral, in-between, and chaotic level.

######################################################################################

# import the packages for drawing a lorenz graph

import numpy as np

import matplotlib.pyplot as plt

######################################################################################

# initial values are assigned s = 10, b = 8/3

def lorenz(xyz, \*, s=10, r=28, b=(8/3)):

"""

Parameters

----------

xyz : array-like, shape (3,)

Point of interest in three-dimensional space.

s, r, b : float

Parameters defining the Lorenz attractor.

Returns

-------

xyz\_dot : array, shape (3,)

Values of the Lorenz attractor's partial derivatives at \*xyz\*.

"""

x, y, z = xyz

x\_dot = s\*(y - x)

y\_dot = r\*x - y - x\*z

z\_dot = x\*y - b\*z

return np.array([x\_dot, y\_dot, z\_dot])

# setting the number of steps and increment of t

dt = 0.01

num\_steps = 10000 (range of t times)

xyzs = np.empty((num\_steps + 1, 3)) # Need one more for the initial values

xyzs[0] = (11.8, 4.4, 2.4) # Set initial values for x y z

[ Visualization ]

1. Neutral (Periodic) State

Depends on the r value, we can define the Lorenz of neutral (periodic) state. When r value is 8 on this model, we can see the neutral state.

Chart

Description automatically generatedChart

Description automatically generated

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

1. In-Between State

Depends on the r value, we can define the Lorenz of In-between state. When r value is 15 on this model, we can see the In-between state.

Chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart

Description automatically generatedChart, histogram

Description automatically generated

1. Chaotic State

Depends on the r value, we can define the Lorenz of Chaotic state. When r value is 28 on this model, we can see the chaotic state.

Chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

[ Explanation ]

The Lorenz system is a mathematical model that is used to simulate atmospheric convection. In this system, the initial values of x, y, and z were chosen to represent file size, with the average size of image files being used as the values. The constants 𝜎 and 𝑏 were kept unchanged since they are fixed values in the system. Therefore, only the r value was allowed to change under the given conditions. The r value represents the rate of change in the model. When r decreases, the number of cycles in the system also decreases, according to the formula. Conversely, increasing r leads to an increase in the number of cycles. This logic works because r is the only variable that determines the conditions of the cycles, as 𝜎 and 𝑏 are constant values. Given 𝑟 > 1, the Lorenz System contains a sink origin. • As 𝑟 increases, the number of cycles also increases (Tantzen & Sauter, 2015). Due to this nature, 𝑟 values were chosen as follows: 8, 15, 28. 8 was chosen because it is unchanged by the sink origin property so it represents the neutral state; thus, it will present itself uniquely. 15 was chosen because it was still relatively small but would have enough cycles to create visually remarkable results. Lastly, 28 was chosen because of its historical usage in the Lorenz System.

**PART 2: Butterfly Effect**

Section 1)

The Table below lists the arrival times and service durations for customers in a FCFS single server queue. From this data, compute:

Table

Description automatically generated

𝐿𝑞 (the time average number in queue)

𝐿𝑞(𝐴) (the average number in queue as seen by arriving customers).

For 𝐿𝑞, use a time horizon of [0,15.27], where 15.27 is the time that the last customer exits the system. Assume the system is empty at t = 0.

Calculate by hand for each inter-arrival time and write a Python code and generate 5 plots, that is:

1. the customer arrival time as a function of service start time

* In order to calculate the service start time, it is important to define the initial or first value of the service start time. As a given information, the time horizon is from a range 0 to 15.27. When we assume the system is empty at t=0, then the first service start time can start after the arrival time which is 1. The rest of service start time can be filled with adding a service duration time for each interval.
* Service Start Time = Previous Service Start Time + Service Duration

1. the customer arrival time as a function of exit time

* The last exit time is 15.27 and from there, you can calculate by subtracting service duration, then it will be going to be the previous exit time.
* Exit Time = Start from the last interval
* = Exit time – Service Duration

1. the customer arrival time as a function of time in queue

* Time in queue can be calculated by the service start subtract by the arrival time. The ready queue state represents the time that is holding the next customer.
* Time in Queue = Service Start Time - Arrival Time

1. the customer arrival time as a function of the number of customers in system

* Number of previous customers whose exit time is greater than current customer’s arrival time.

1. the customer arrival time as a function of number of customers in queue.

* The Number of customers in queue = Number in System - 1

[ Data Table]

Table

Description automatically generated

𝐿𝑞 (the time average number in queue) = 1.1347

𝐿𝑞(𝐴) (the average number in queue as seen by arriving customers) = 0.8

Chart, line chart

Description automatically generated

Section 2)

On a network gateway, measurements show that the packets arrive at a mean rate of 125 packets per second (𝑝𝑝𝑠) and the gateway takes about 2 milliseconds to forward them. Using an M/M/1 model, analyze the gateway. What is the probability of buffer overflow if the gateway had only 12 buffers? How many buffers do we need to keep packet loss below one packet per million?

Once we find the arrival rate and service rate, we can compute the gate utilization as below:

The probability of packets in the gateway is also defined as

The mean number of packets in the gateway defined as

The mean time spent in the gateway is defined as

The probability of buffer overflow is defined as

To count the number of buffers that keep packet loss below one packet per million, we can set the number of packets lower or equal to one packet per million.

, Once we take log for both side, we can find the n

Therefore, the system needs about 10 buffers to keep packets loss.

Section 3)

Given an M/M/1 system (with λ < μ), suppose that we increase the arrival rate λ and the service rate μ by a factor of k each. How are the following affected?

1. Utilization, ρ?

Utilization in M/M/1 system defines as . If we increase the arrival rate and service rate by a factor k, then it remains the same still.

Chart, line chart

Description automatically generated

The slope for the utilization is positive, over time t, therefore the graph shows positive slope for both Utilization and Factored Utilization.

1. Throughput, X?

Throughput is defined as an arrival rate in jobs per unit time, which is λ, therefore if we increase the rate by a factor of k, then throughput is also increased by a factor of k.

Chart, line chart

Description automatically generated

As time increases, the factored throughput increases in K times. Since Throughput is a calculation of lambda itself, if we add a k factor, then it is k times lambda.

1. Mean number in the system, E[N]?

The mean number in the system equation is defined as . If we increase the factor of arrival rate and the service rate, it doesn’t affect the system. We have proved that the Utilization p doesn’t get affected by the increasing factor of k.

Chart, line chart

Description automatically generated

As calculation above, if we multiply k factor on both lambda and mu, then it cancels out for both nominator and denominator. Therefore, the equation gets any effect by the factor k.

1. Mean time in system, E[T]?

In the M/M/1 model of queue theory, the mean time in system (E[T]) can be calculated using Little's Law, which states that the long-term average number of customers in a system (L) is equal to the long-term average arrival rate (λ) multiplied by the long-term average time.

Since the M/M/1 model assumes a single server and exponential interarrival and service times, we can use the following formulas for λ and the service rate (μ):

We can then use the following formula to calculate E[T]:

E[T] = 1 / (μ - λ) where μ > λ Note that E[T] includes both waiting time and service time, so it represents the total time a customer spends in the system.

Once we factor the values by k, then it will be

Therefore, the mean time in system increase by factor.

Graphical user interface, application

Description automatically generated

Section 4)

Given an M/M/1 server, what is the maximum allowable arrival rate of jobs if the mean job size (service demand) is 3 minutes and the mean waiting time (E[TQ]) must be kept under 6 minutes?

* Max arrival rate can be calculated by dividing the mean job size by (mean waiting time + mean job size). The max arrival rate is the rate of the maximum approach of the rate; therefore, it can be calculated as follow:

Therefore, the max arrival rate is 0.222 jobs per minute to keep waiting time under 6 minutes.

Section 5)

Diagram

Description automatically generated

According to the descriptions given, the model turns out above. The queueing model for a Single Bus Tightly Coupled Multiprocessor (SBTCMP) architecture consists of multiple Processing Elements (PEs), each with a task pool, a CPU, a Bus Interface Unit (BIU) for shared bus access, and associated queues. Each CPU and BIU has mean service rates 𝜇(𝑖,1) and 𝜇(𝑖,2), respectively, and tasks have a mean sleep time of 𝜇(𝑖,0)−1 in the task pool. The CPU and BIU operate independently, and all BIUs are combined into an "equivalent BIU". Branching probabilities 𝑝(𝑖,1), 𝑝(𝑖,2), and 𝑝(𝑖,3) dictate task behavior, with 𝑝(𝑖,3) representing the probability of a task joining the CPU queue after using the BIU. Interrupts occur at a mean rate of 𝜇(𝑖,3), and the model assumes a fixed workload.

**References**

Bronson, R. & Costa, G. (2014). Schaum's outline of differential equations (4th ed.). McGraw-Hill: New York, NY. ISBN-13: 9780071822862

*Lorenz attractor#*. Lorenz attractor - Matplotlib 3.7.1 documentation. (n.d.). Retrieved March 22, 2023, from https://matplotlib.org/stable/gallery/mplot3d/lorenz\_attractor.html

Tantzen J & Sauter E, (2015, July, 22). Discussion of the Lorenz Equations. Leibniz Universit¨at Hannover, from https://www.itp.uni-hannover.de/fileadmin/itp/user/ag\_flohr/lectures/proseminar/ss15/Vortrag11.pdf

Source: GCU Library; IEEE TRANSACTIONS ON COMPUTERS, VOL. 38, NO. 3, MARCH 1989 (Diagram: Queueing model for the multiprocessor.)