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

Emmy Voita and Sammie Vaicius

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

Ricardo Citro

10 April 2022

**System performance context description:**

*Part 1:*

This project allows us to use the Lorenz System to model the chaotic behavior in memory. The system processes are swapped in and out between the disk and memory, when they are moved and removed from memory, holes are created. When the memory starts to fill up, these gaps / holes are then filled because the system divides files to fit in the gaps. In this case we are using the Lorenz System to simulate and visualize the butterfly effect on code.

*Part 2:*

In part 2 we are observing the butterfly effect of the lorenz equation. Chaotic systems are highly sensitive to initial conditions and that sensitivity is best known as the butterfly effect. A queueing model represents a system’s physical configuration and deals with the mathematical modeling / analysis of systems that provide service to random demands. Queueing theory is a helpful analytic modeling technique used for computer systems performance analysis.

**Specific problem solved:**

*Part 1:*

The Lorenz System can be used to model chaotic behavior in the save and delete processes in memory. We are able to manipulate the r value to demonstrate changes to the system. When we increase r, the number of cycles increases. This results in the number of processes increasing, bringing the system closer to the threshold of chaos as r increases.

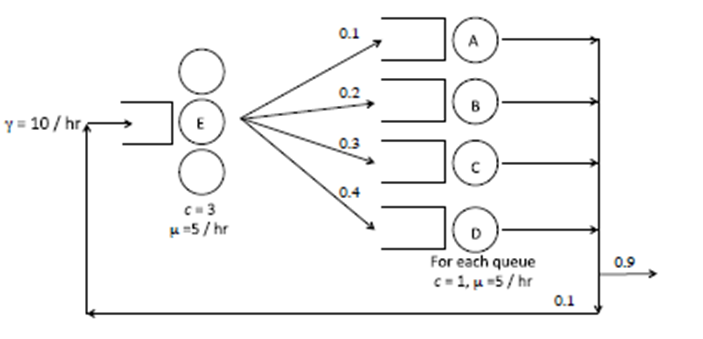
*Part 2:*

1. A) Determine the average number of customers at each station (A, B C, D, E)

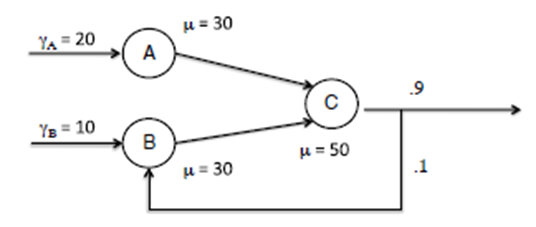
B) Determine the average time spent in the system

\* Note γ-mean arrival rate; μ-average service rate;c-number of servers.

\* 0.1,0.2,0,3,and 0.4 are transition probabilities and they sum to 1.



1. A. Determine the average number of customers at each station  
   B. Determine the average time in the system for an arbitrary customer  
   C. Determine the average time in the system for a customer who enters through Node A



1. Table below displays arrival times and service durations for customers in a FCFS single queue. From this data compute L(q) and L(q)^A [respectively known as the the time average number in queue and the average number in queue as seen by arriving customers].

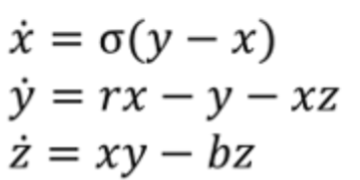
\* For L(q) use a time horizon of [0,15.27] where 15.27 is the time that the last customer exits the system. Assume the system is empty at t =0. Calculate by hand for each inter-arrival time and write a python code and generate 5 plots

1. Customer arrival time as a function of service start time
2. Customer arrival time as a function of exit time
3. Customer arrival time as a function of time in queue
4. Customer arrival time as a function of the number of customers in the system
5. Customer arrival time as a function of number of customers in queue

**The mathematical approach for solving it:**

*Part 1:*

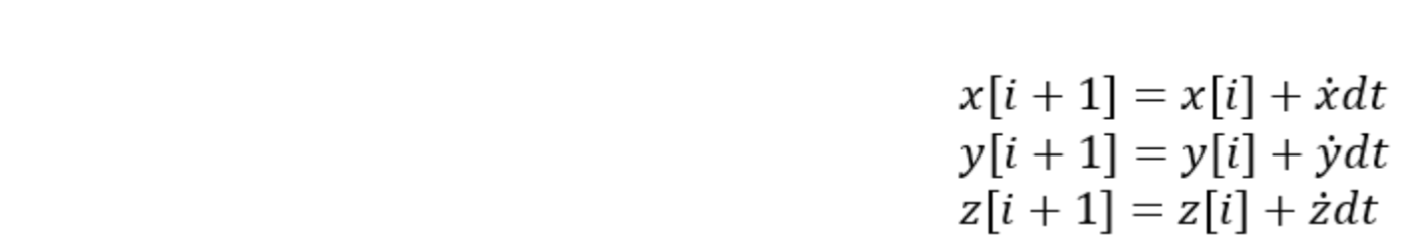
Defining a Lorenz System using three ODEs as discussed in class using:



With this set of equations, we can define x prime, y prime, and z prime values for each iteration needed.

σ = Prandtl number ; ρ = Rayleigh number ; β = certain physical dimensions of the layer itself.

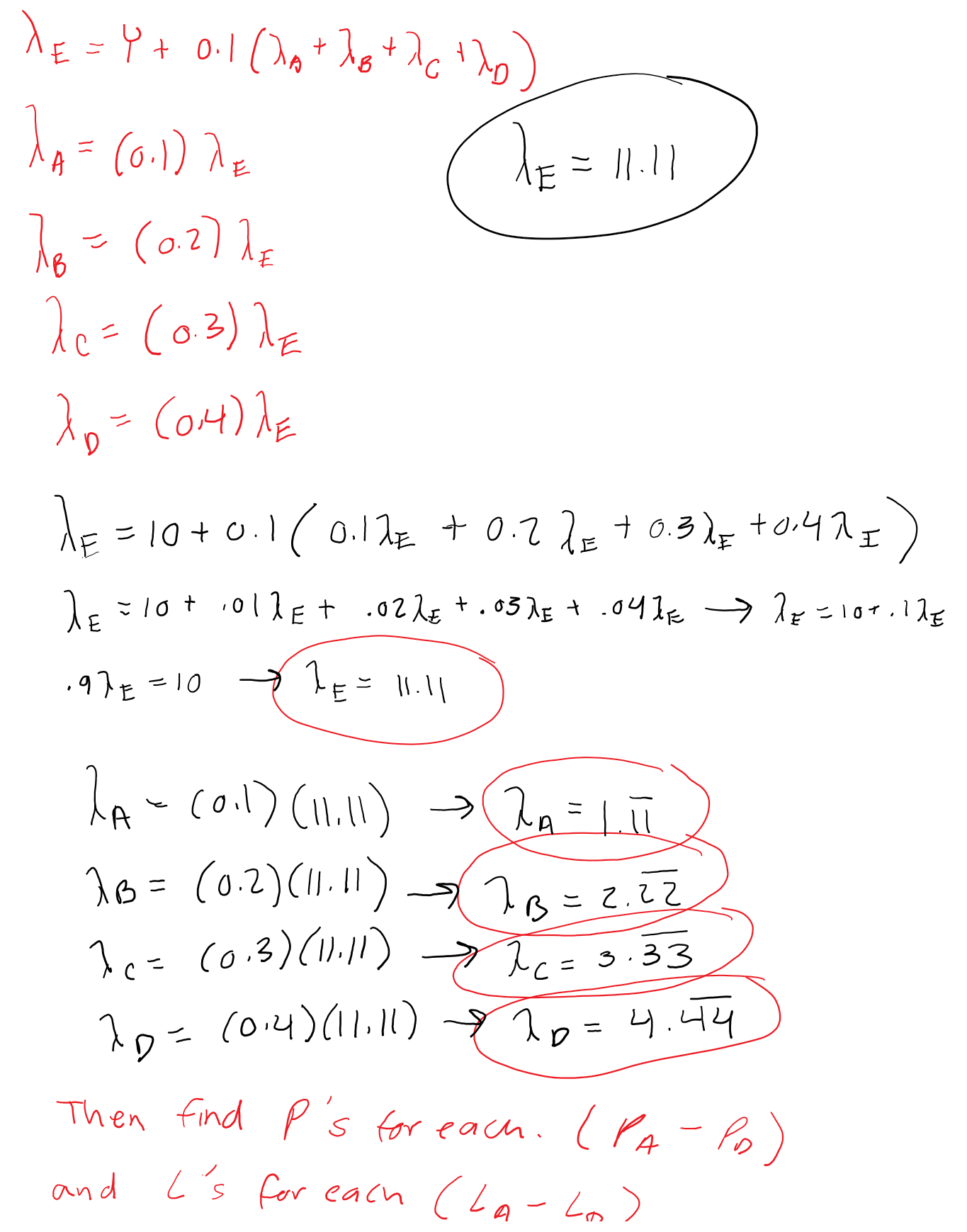
When we call on the previous set of equations and define our variables, we can estimate the next values. We are able to do this using a determined step size and using the following system:

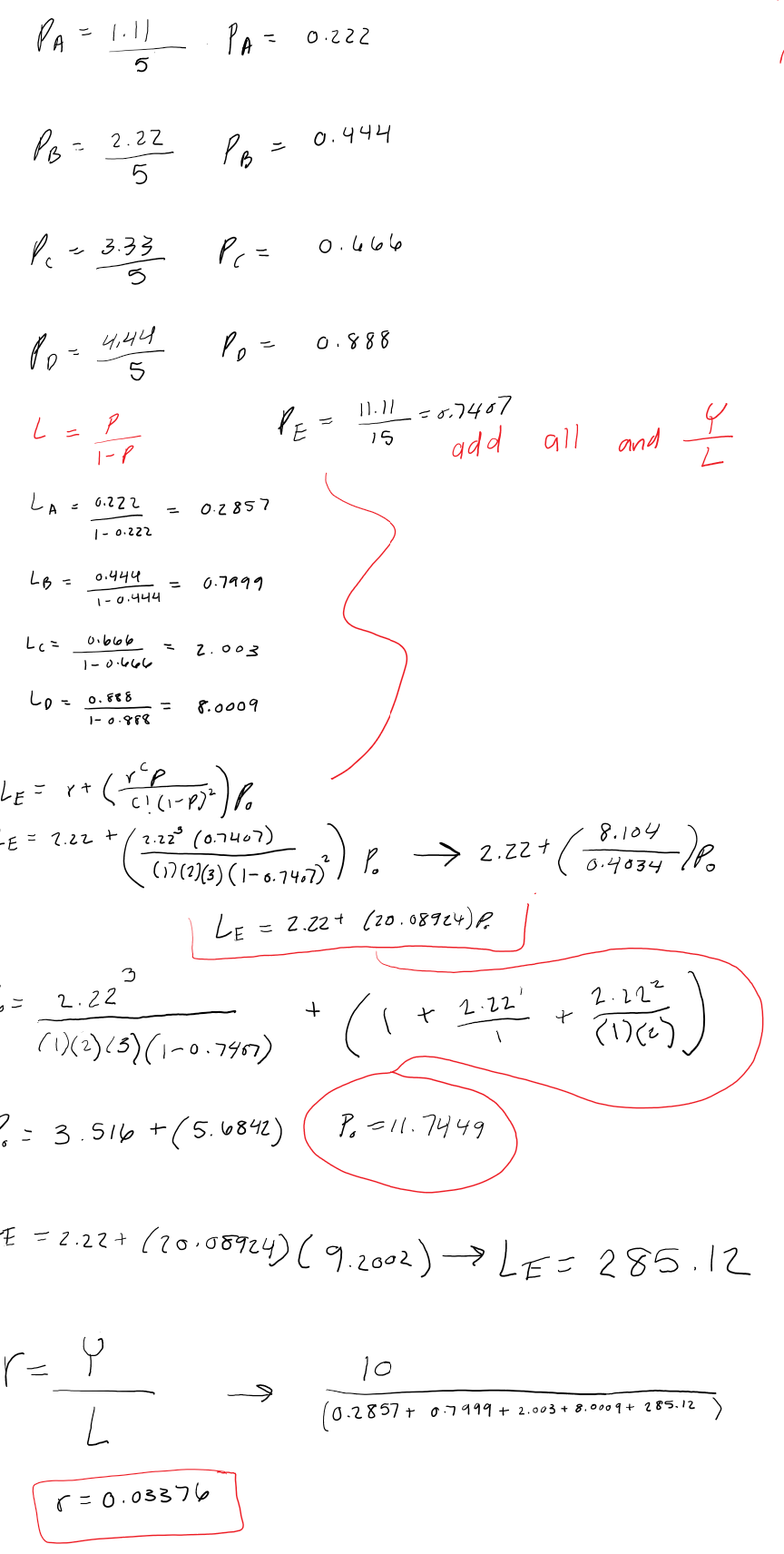


For each of these, we are able to use our respective x prime, y prime, and z prime values calculated previously to estimate the next x, y, and z values. These could then be added to an array for each to create a set of coordinates. We are able to loop through these steps of calculations any desired number of times to help portray the calculations to display on a graph.

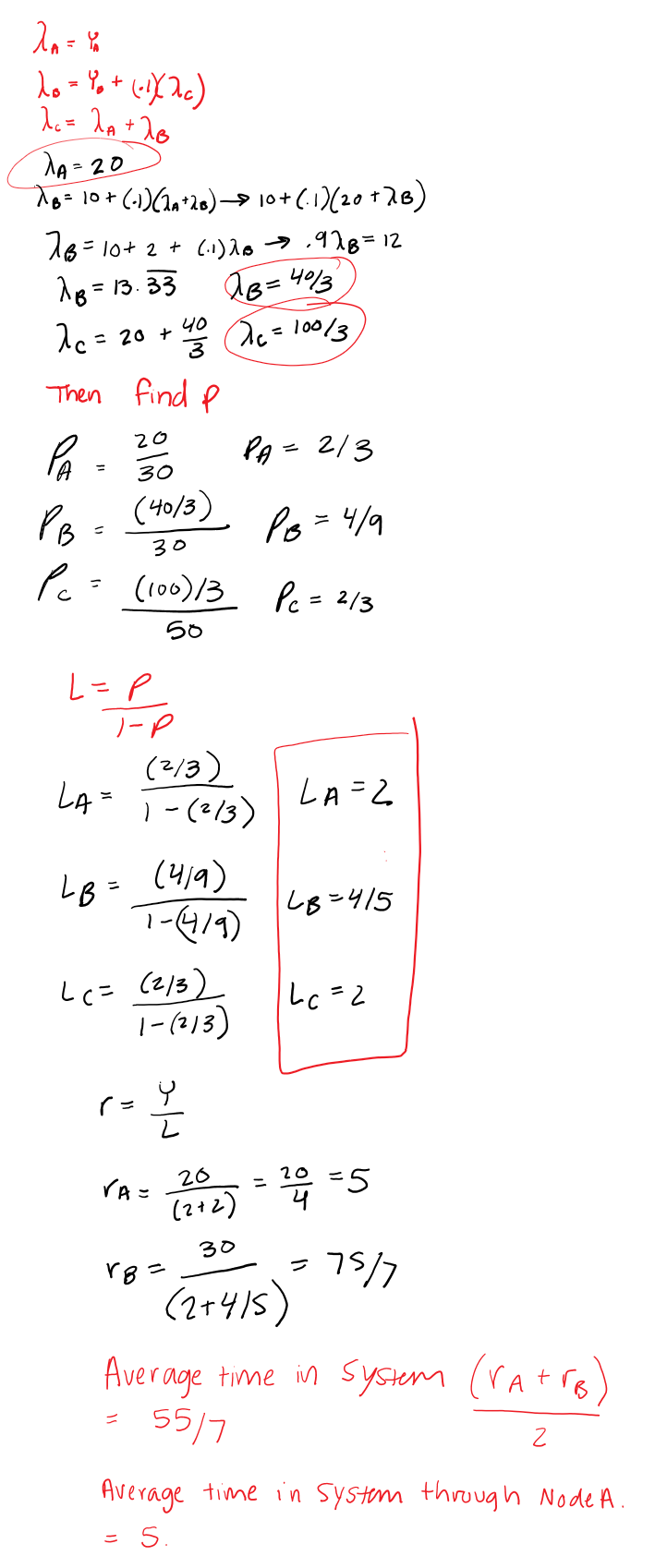
*Part 2:*



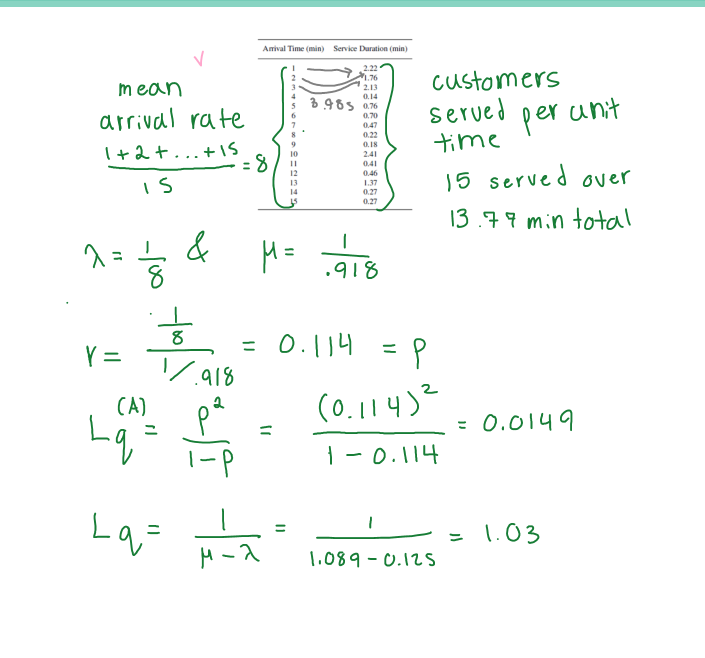


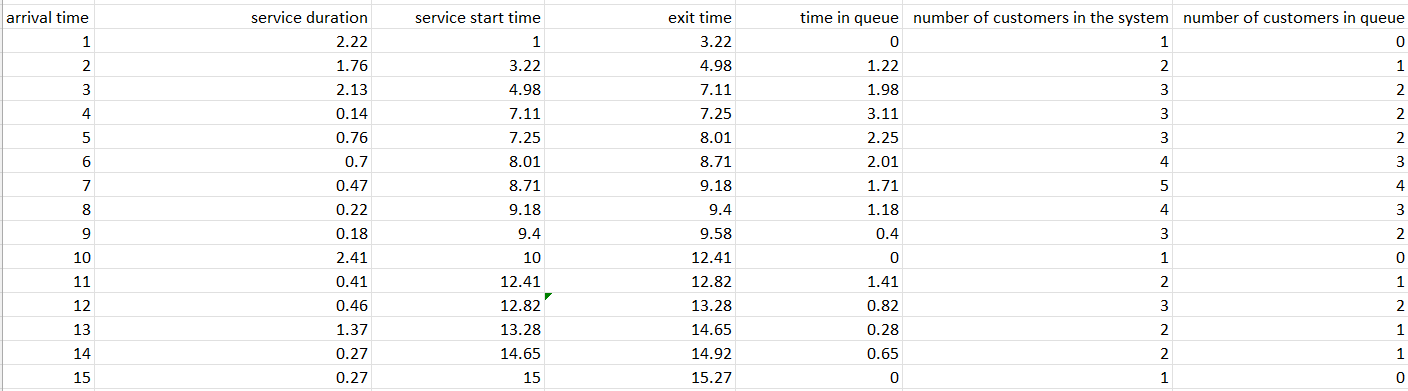


1. The average number of customers at each station A = 0.2857 , B= 0.7999, C=2.003, D = 8.009, E= 285.12
2. Average time spent in the system = 0.03376



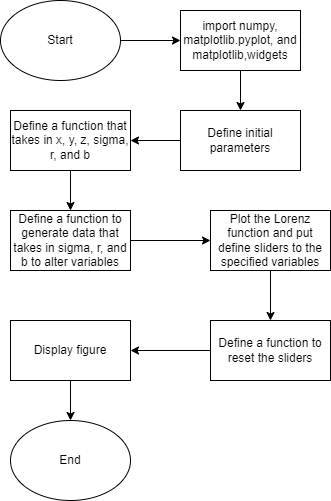
1. Average number of customers at each station: A = 2, B = 4/5 , C = 2.
2. Average time in the System = 55/7
3. Average time in System through Node A = 5



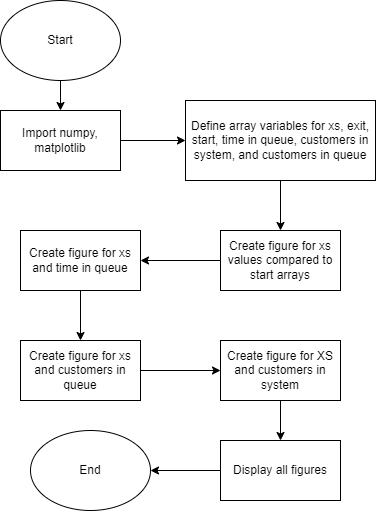
**

**The approach for implementation in code (e.g., algorithm, flowchart):**

*Part 1:*

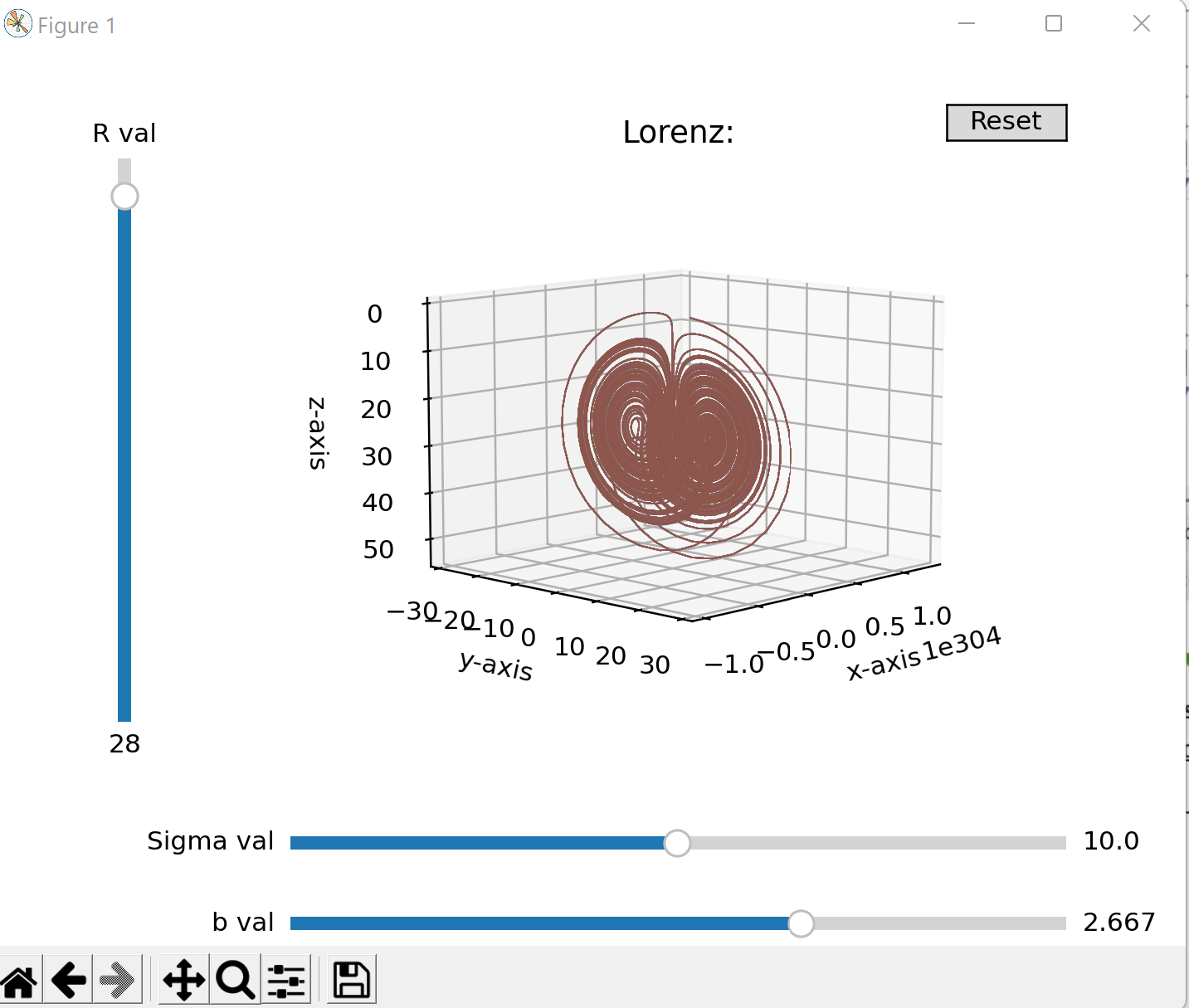
**

*Part 2:*

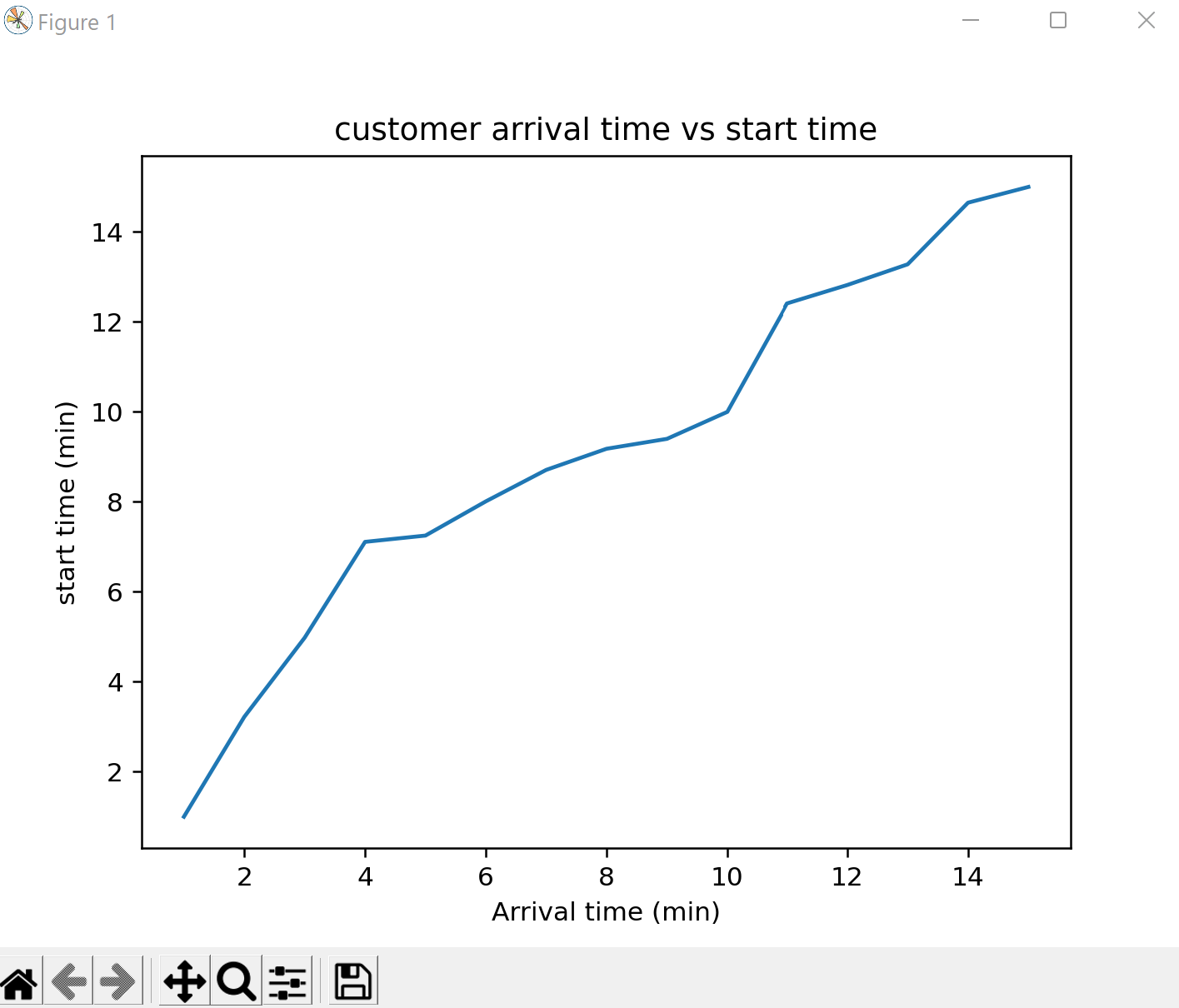
**

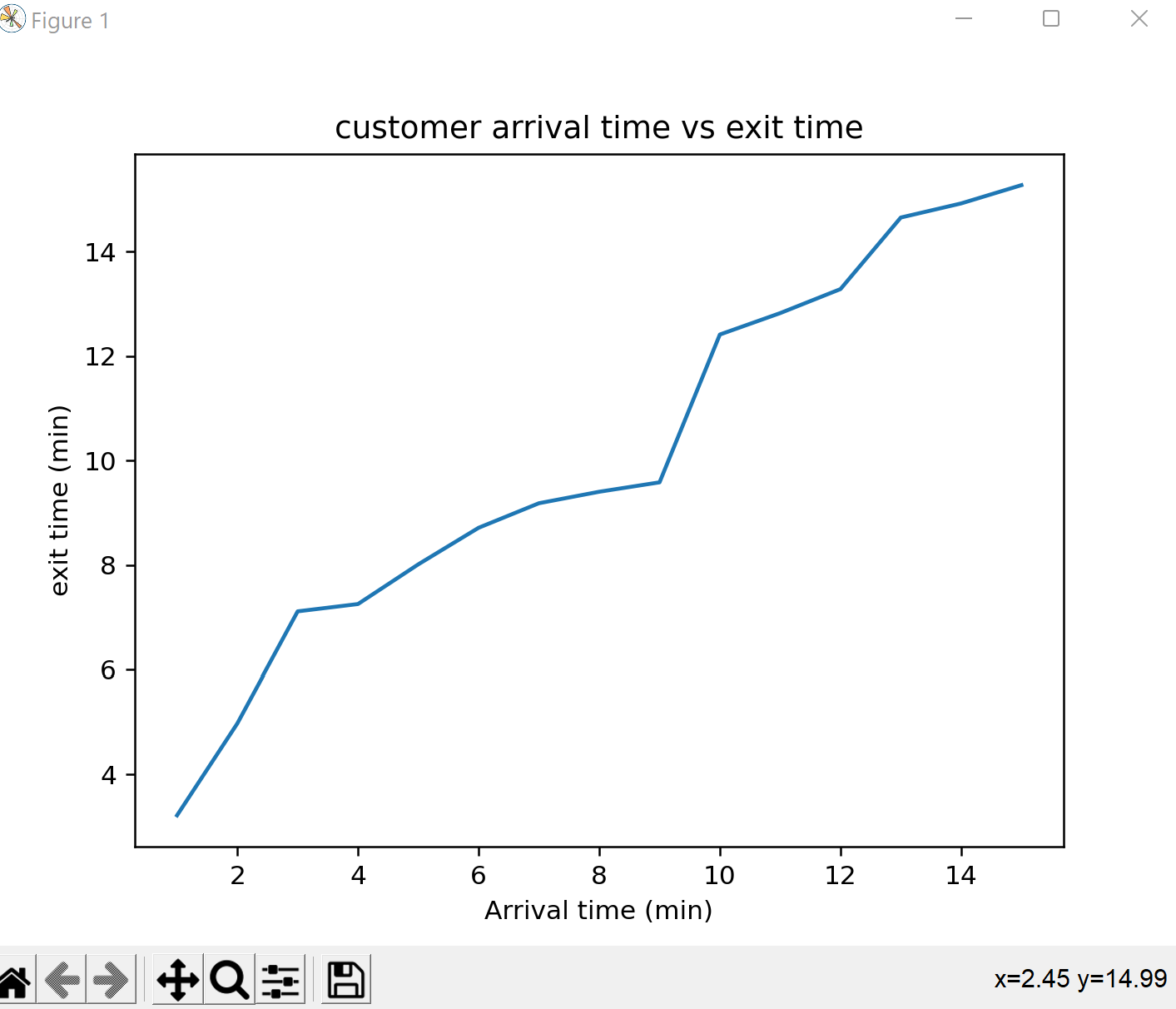
**Screenshots depicting key phases in the program execution:**

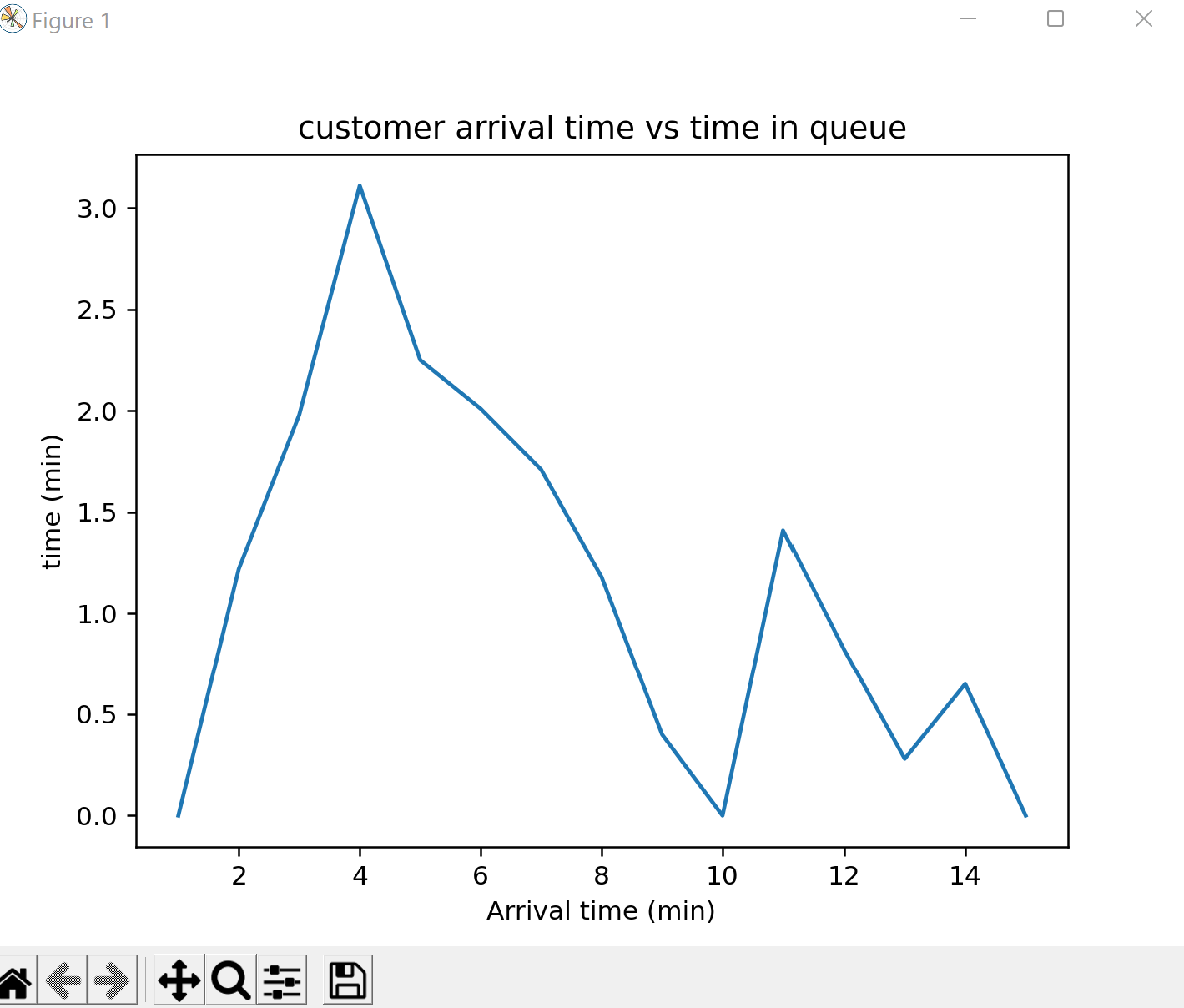
*Part 1:*

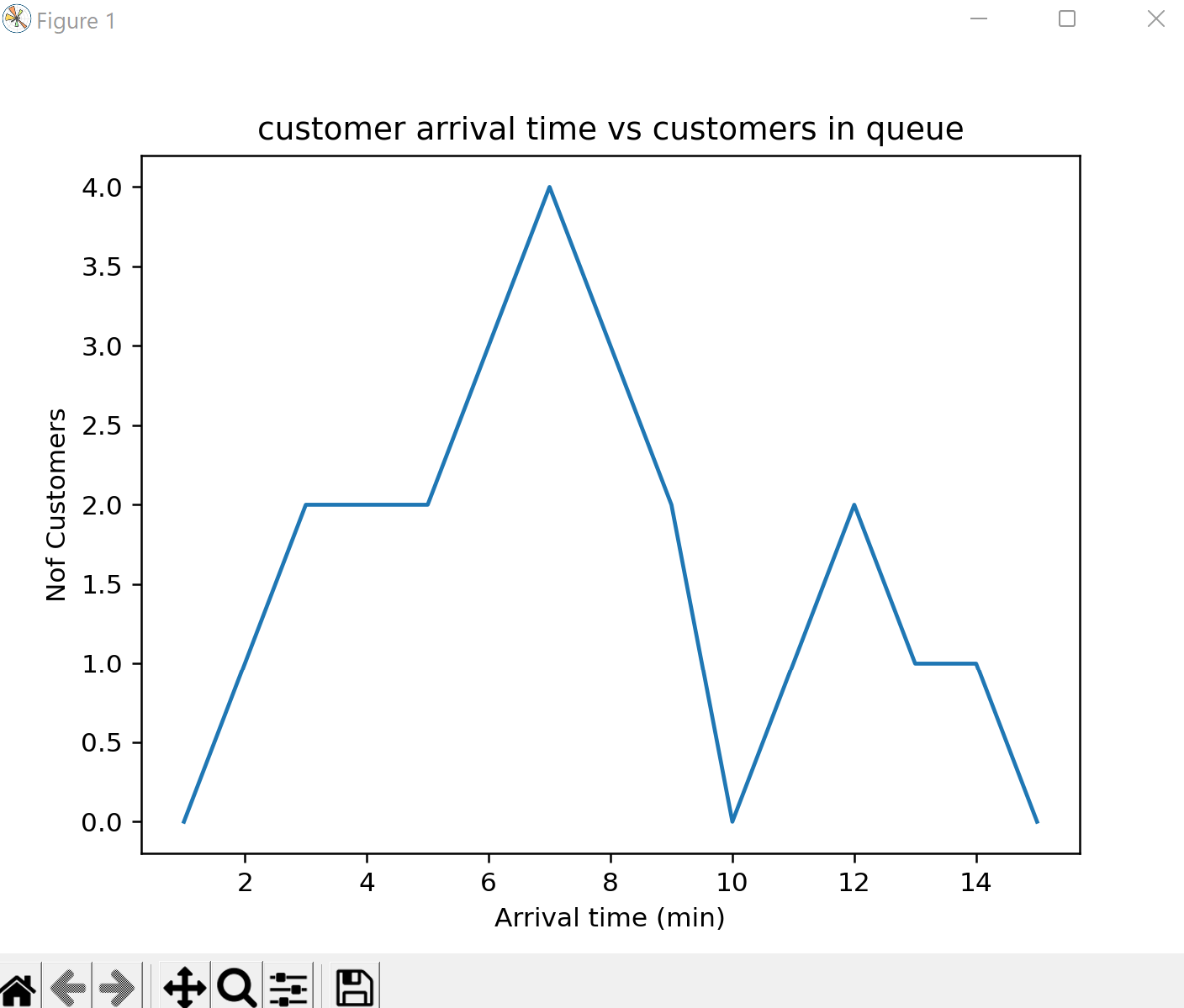
**

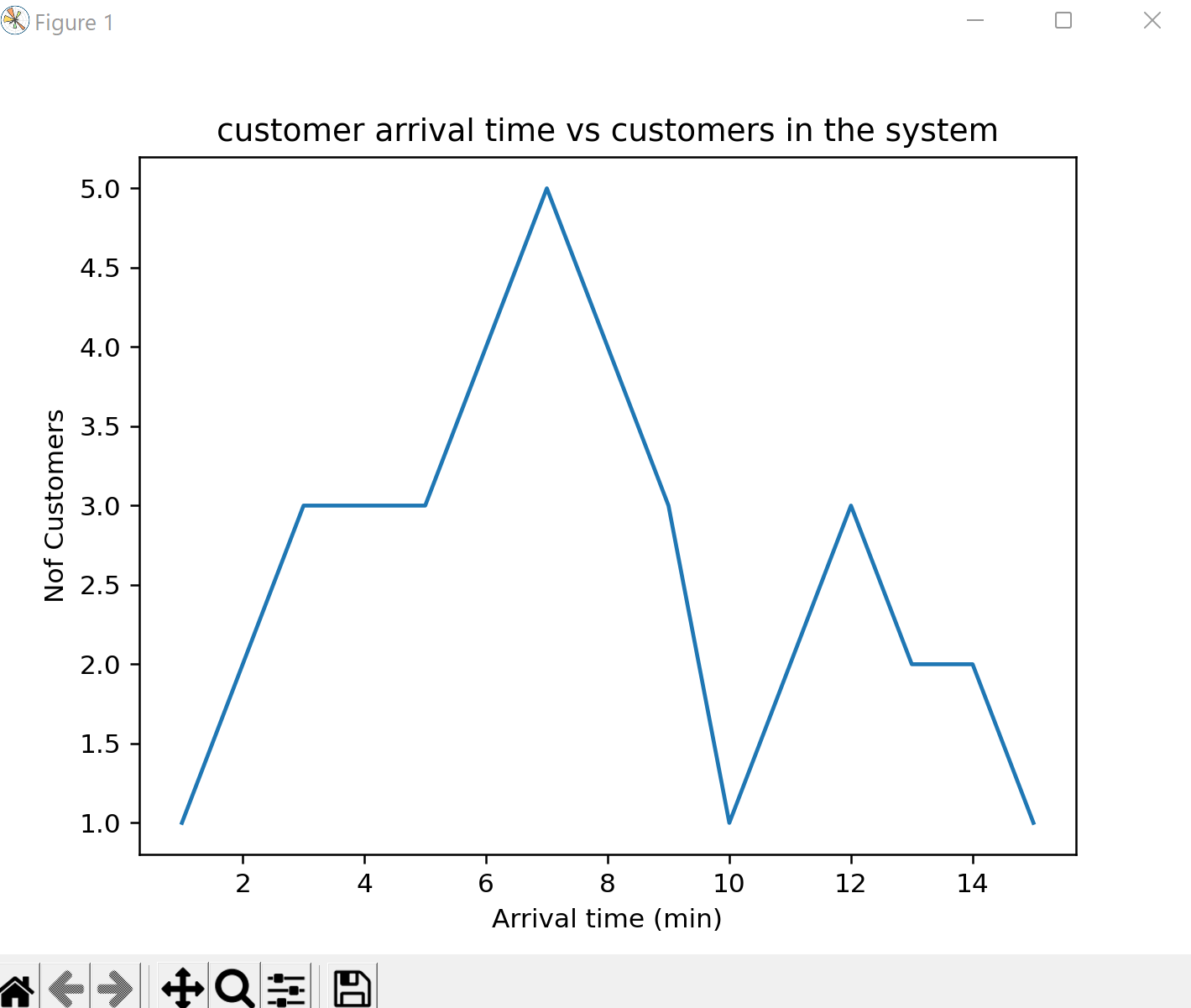
*Part 2:*

****

****

****

****

****

**Analysis:**

*Part 1:*

As stated above, the parameters σ, *ρ,* and  *β* represent the Prandtl number, Rayleigh number, and the certain physical dimensions of the layer respectively. The sliders allow us to observe the changes in the threshold of chaos with the changes in the variables.

*Part 2:*

1. The average number of customers at each station was A = 0.2857 , B= 0.7999, C=2.003, D = 8.009, E= 285.12 and the Average time spent in the system was 0.03376. In the diagram of Part 2 Question 1, we can see that 0.4 is channeled to D, which is why D results in having the highest average number of customers at each station. On the other hand, A had the smallest decimal resulting in the least amount of customers at each station. Because there are 4 channels available, the average time spent in the system is relatively low. If there were less channels the average time spent in the system would be greater.

2. The results showed that the average number of customers at each station was: A = 2, B = 4/5 , and C = 2. The Average time spent in the system was 55/7. Finally the Average time in the system through node A was 5. The average time in the system through Node A can handle a greater input and can handle the information without looping back through. Whereas Node B takes in less input and requires looping back into Node B after Node C to compute. This is why the overall average of the system is greater than just the average of Node A.

3. The Lq and Lq with respect to an incoming customer were determined to be 1.03 and 0.0149 respectively. These results are to be expected given that the average service duration was just slightly below the rate of incoming arrivals. With respect to the graphs created, all five could be used to better analyze the system. For example, the graph of *arrival time vs customers in the system* could be used to determine when the system is congested, and this information could then be used to make modifications to the system.

**References for theory and code sources:**

Citro, Ricardo. “Queueing Theory”. GCU. <https://padlet.com/ricardo_citro/3u72gdpkp3lo>