**Analytical Modelling**

Table of Contents

[Continuous Time Markov Chains 2](#_Toc87178022)

[Analytical Models 3](#_Toc87178023)

[Single-Server Queuing System 4](#_Toc87178024)

[Problem Description 4](#_Toc87178025)

[State Transition Diagram 5](#_Toc87178026)

[Steady-State System 5](#_Toc87178027)

[State Transitions 6](#_Toc87178028)

[Probability Model 7](#_Toc87178029)

[Special Solution 10](#_Toc87178030)

[Expected Number of Customers 11](#_Toc87178031)

[Time-Average System Time 12](#_Toc87178032)

## Continuous Time Markov Chains

Once we are done with our simulation and have found some output data, we still need to ensure that the data we got is an accurate representation of the real system. We will do this by comparing the **simulation results** to the **analytical results**. All of the things we did using a simulation system can also be done analytically. This is where **Continuous Time Markov Chains** come in.

In the **Single-Server Queueing System**, the state variables we considered where the **sever status** and the **queue length** . Alternatively, we could combine these two into a single variable, the **number of customers** in the system, .



The **state** of the system can be described by the state transition diagram above. At every state, an **arrival event** could occur with the rate , which will cause the system to move to the next state. At every state except state , a **departure** event could occur with the rate , which will cause the system to move to the previous state.

This diagram is an example of a Continuous Time Markov Chain.

The single server queuing system can also be represented as an system, where the indicates that there is a single sever, the first indicates the arrival distribution and the second indicates the service distribution. The reason we use s is because both distributions are **exponential** or **memoryless**. Such a system can be analytically solved.

The values that we calculated so far have been **averages**. Averages can be calculated analytically only if the system is **stable**, i.e. the states do not fluctuate too often. This can be achieved if the rate at which the system enters a state is equal to the rate at which it leaves a state.

## Analytical Models

There are cases where a **closed form solution** for a given problem does not exist, meaning it cannot be solved analytically. There are also cases where the mathematics behind an analytical solution becomes extremely **complex**. Because of this, it is not generally preferable to try to find an analytical solution. However, as mentioned before, an analytical solution can help **verify** a simulated solution, so if at all possible, we should try to find the analytical solution as well.

## Single-Server Queuing System

We will initially be looking into the analytical solution for a **Single-Server Queuing System**. There is a basic tool to solve this, but we will not be using that for now. Instead, we will be using basic mathematics.

A major component used in solving a Single-Server Queueing System analytically is the **Continuous-Time Markov Chain**, but it is quite complex, so we will do our best to avoid it for now.

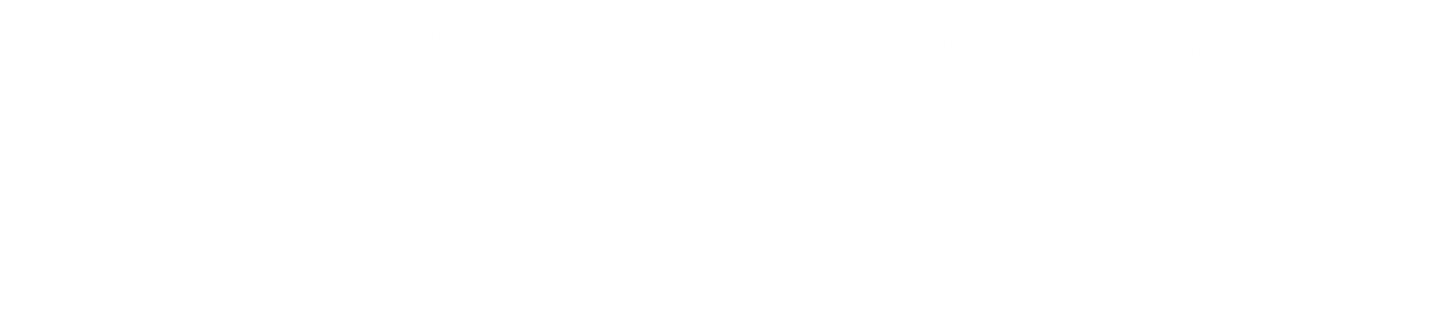
### Problem Description

In a Single-Server Queueing System, we have customers arriving at a rate to receive a service. The rate at which the service is provided is . There is only one server, so if the server is busy, the customers join a queue. The queue is assumed to have no maximum length.

Both the arrival and the service rates are Poisson distributions. The inter-arrival times and the inter-departure times, also called the service times, are Exponentially distributed. This makes both the arrival and the departure processes memoryless, so the Single-Server Queueing System can be described as .

### State Transition Diagram

We will only concern ourselves with a single state variable, the **number of customers** in the system, . Because of this, the state transition diagram will look like this:



The system goes from the th state to the th state at the rate of and goes from the th state to the th state at the rate of . There is no maximum state, since there is no maximum queue length.

### Steady-State System

The specific scenario we will be looking into is called a **steady-state scenario**. This means that there are a certain number of states between which the system is moving at a steady rate. This is different however, from the scenario where the system is always at one state, which would make it **deterministic**.

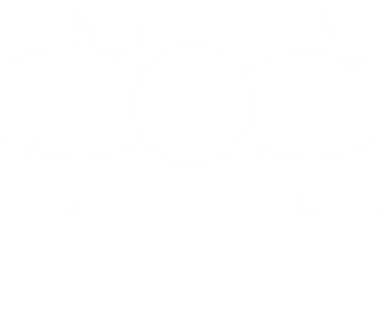
A steady-state scenario allows us to calculate **average values** for parameters. For example, we could calculate the average queue length, the average system time, etc.

Since the state of the system itself tells us the **number of people** in the system, if we can find the probability distribution of system states, we can find the **expected value** for the number of people in the system at any given time, .

For example, for the th state, we can find , the steady state probability that the system is in state . Usually, this is just the fraction of time for which the system is in that state. Then, finding is as simple as .

### State Transitions

We have mentioned the system being assumed to be a steady-state system. This means that the rate at which the system **enters** a particular state and the rate at which it **leaves** that state are equal.



For a given state , where , the system can **enter the state** from the **previous state** at a rate or from the **next state** at the rate . The system can also **leave the state** and go into the **previous state** at the rate and go into the **next state** at the rate . Thus,

However, these values are in units of time, e.g. is the arrival rate when the system is in state and it has a value of per unit time. We need these values to be related to the **total system time**. In order to do this, we multiple the values by **.** Thus, the rate at which the system leaves state and goes to state is and the rate at which the system leaves state and joins state is .

From this, we can say that in general, the equations for each state become:

State

State

State

State ,

### Probability Model

The next step is to find the **probability model** for . To do this, we need to calculate the unknown quantities in the equations we just made for each state. These equations form a **system of linear equations**, so we can solve them to find the value of for every possible .

Normally, for a large system of linear equations like this, we would try **Gaussian Elimination**. In this system however, we have some unique properties. One quantity on the **left-hand side** of an equation appears on the **right-hand side** of the **next equation** and one quantity on the **right-hand side** of an equation appears on the **left-hand side** of the **next equation**. For example, on the left of the first equation appears on the right of the second equation. And on the right of the first equation appears on the left of the second equation.

This observation means if we **add the th equation** with the **th equation**, two terms will get cancelled out. This causes the set of equations to become like this:

State

State

State

State

Solving this set of equations is very simple.

We further know that the **sum of all probabilities** must always be . Hence,

From this equation, we can find .

And once we have this, we can find any .

An issue with this solution is that the term can become **infinite**, in which case we will not have a solution at all. Thus, for there to be a solution at all, this term has to be less than infinity. This is a necessary and sufficient condition.

### Special Solution

If all the values of are **equal** and all the values of are equal the equations become much prettier.

The term is just a **geometric series**, which means to have a solution, must always be true, i.e. . Remember how if customers are arriving faster than we can serve them, then the system becomes impractical.

In queueing systems, the value is often denoted as and is called the **traffic intensity**.

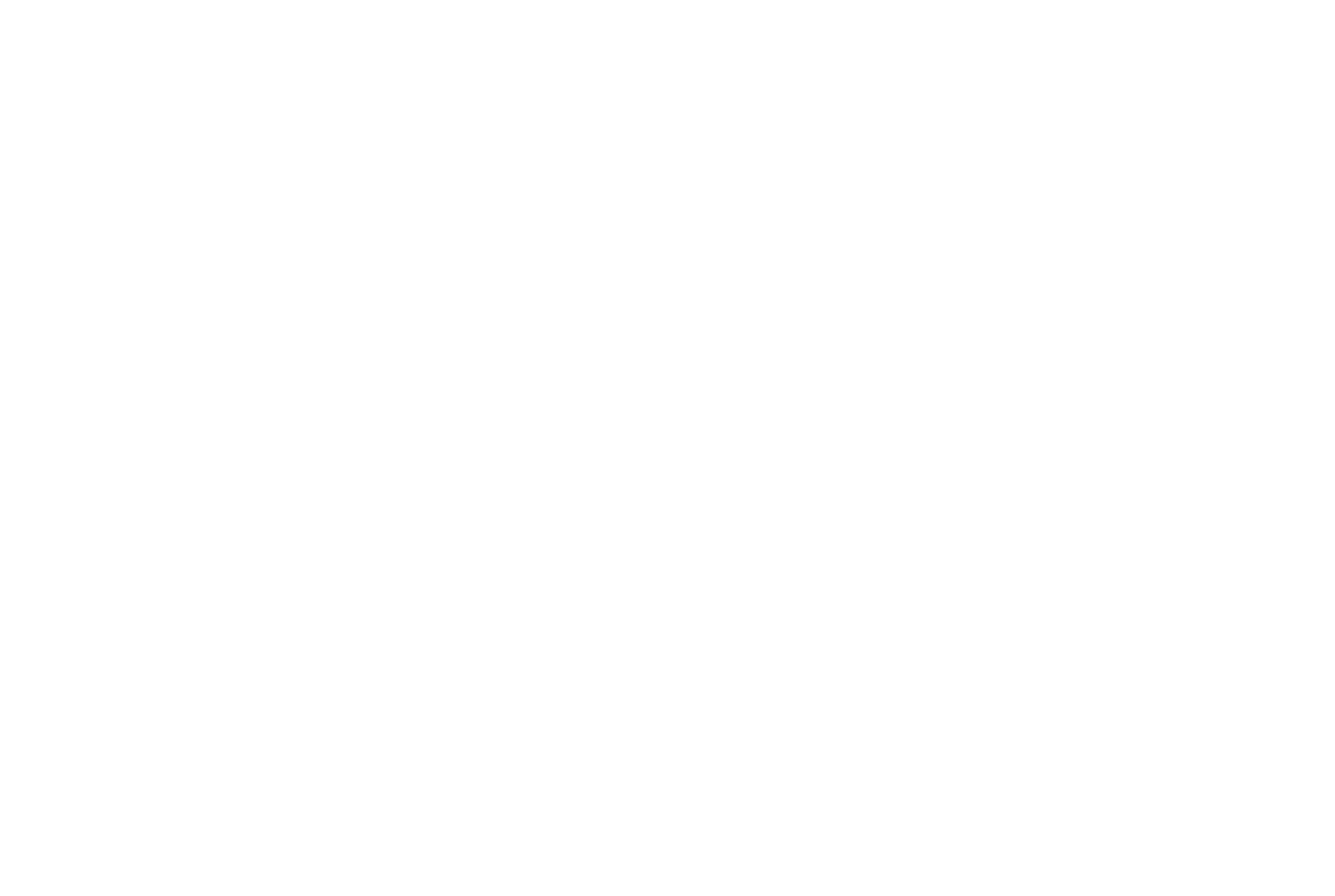
Since this is a geometric series, we can find that

By varying , we can find the probability models for different configurations.

### Expected Number of Customers

Since we have the distribution now, we can find the **job-average** **number of customers** in the system.

The graph for this equation looks like this:



### Time-Average System Time

Once we have the job-average number of customers in the system, we can find the **time-average system time**.

Comparing the graphs, we are getting for these equations to the graphs for the results of the simulation, we can validate our results.