4
Continuous Time Markov Chains and Queues

Figure 4.1: The wait is on because queues are everywhere. But the seventeenth century English poet, John Milton, lends us hope. He writes: *They also serve who only stand and wait.*

Queues feature in our daily lives like never before. From the checkout counter in the community grocery store to customer support over the phone, queues are theatres of great social and engineering drama. Entire business operations of many leading companies are geared towards providing hassle free customer support and experience - timely and effective resolution of client queries about services on a regular basis. Alternatively, it could be effective traffic management and resource optimization for a multiplex cinema operator involved in ticket sales. Sometimes it may not involve humans at all like in the case of a database query to a computer server for a specific information that may be routed through a job queue. How a queue moves in time and how services are offered over epochs determine how businesses will be able to make profit or how efficiently computer servers will execute tasks. All these have a huge technological and economical impact. No wonder we have seen huge investments by concerned stakeholders to upgrade and upscale hardware and software infrastructure to re-engineer queues towards greater system efficiency and profitability. The mathematical technology of queues is crafted out of models that investigate and replicate stochastic behavior of engineering systems. This is the subject of our study in this chapter.

4.1 Chapter objectives

The chapter objectives are listed as follows.

1. Students will study and apply continuous time Markov chains to solve engineering problems.

2. Students will model and analyze queues using probability distributions.

3. Students will learn the inter-relationship between the probability transition matrix and the stochastic generator matrix.

4. Students will learn to derive Kolmogorov's backward and forward equations, use the principle of detailed balance, and find stationary distributions of stochastic processes using a stochastic generator matrix.

5. Students will deduce birth and death stochastic processes and analyze their equilibrium and/or asymptotic behavior.

6. Students will solve an engineering problem using a classical queuing model.

4.2 Chapter project: Queues and crowd management at COVID test centers

4.2.1 Prologue: Crowd management at COVID test kiosks

Crowd management at test centers is a major concern for health care administrators. This issue has been amplified during the COVID pandemic when transmission rates have been very high at times and huge numbers of people have been infected on a regular basis, especially during peak periods of the disease. Administrators at test centers are faced with a dual challenge - (i) they have to contain the number of people presenting themselves at the test kiosks to a manageable number to minimize the risk of disease transmission from over-crowding, and (ii) manage the exploding cost of clinical care and rapidly rising expenditure of procuring test kits and setting up clean test kiosks that have been sanitized. The latter point along with the fact that availability of employees to conduct tests during a raging pandemic comes with a premium - demand intelligent engineering and management of hospital and test facilities.

The sick who have symptoms or those who have tested positive using home kits or quick Rapid Antigen Tests (RATs) may present themselves to test kiosks set up at hospital premises in order to either confirm the results of their preliminary screening tests (that may not have been very accurate) or to evaluate if their disease state may require them to be admitted in a hospital for more dedicated care and treatment. The test kiosks at hospitals have access to more accurate and fast testing equipments to evaluate the severity and/or progress of the disease and will provide better decision points for doctors if a tested patient must be admitted for clinical care.

In this project, we will source real data from one such hospital facility in New York - located at the premises of The Brooklyn City hospital. The data used in the case study includes day-wise and week-wise information over the period January 1 through December 31, 2021. This data includes the number of people who presented themselves at the hospital test center after a preliminary positive result of tests conducted at their homes. This data has been sourced from the official website whose link is provided here. For convenience we have organized the data in a tabular matrix as provided at the end of this chapter in Table 4.1. The tabular data can also be found in a spreadsheet named COVID-data.csv.

4.3 Introduction: a gentle initiation to queues

We will discuss a simple n-server queuing system as a motivation to study continuous stochastic processes within the framework of continuous Markov chains. This discussion will depend on Poisson arrivals of clients. Therefore, it is prudent to revisit the definition of a Poisson process.

Poisson process: Consider an arrival process that counts the number of arrivals over time. $N(t),\ t \geq 0$ is a Poisson process with rate $\lambda > 0$ that counts the number of arrivals if the following statements hold true.

1. $N(0) = 0$.

2. $N(t)$ has independent increments, i.e. the numbers of arrivals in two non-overlapping intervals are stochastically independent.

3. $N(\delta t) \sim Poisson(\lambda \delta t)$, where $N(\delta t)$ refers to the number of arrivals in duration δt.

A simple calculation (**cf.** sec. 5.3.6) shows that the inter-arrival times between Poisson arrivals follow an exponential distribution with the same rate parameter. This will be an important consideration in many of our discussions in this chapter. Let us return to our example of the n-server queue.

Figure 4.2: A queue in a bank with multiple tellers.

Example: n-server FIFO queue

Consider a scenario where we have n servers which provide service in such a way that the service times of each server is an i.i.d. random variable that is distributed according to the exponential law with rate parameter μ. Clients enter a queue according to a Poisson process with rate λ. The service principle follows a *first in first out* (FIFO) law applied to the queue. Further, any arrival that finds all servers busy, leaves without service. The latter condition (hereafter referred to `cond++`) is different from the one shown in Figure 4.2 but we will consider it here in our calculations as an additional constraint. We ask the following question. If an arrival finds that all servers are busy, then what is the expected number of busy servers observed by the next arriving client? For convenience, we will refer to this client as client #13-A.

<ins>Solution</ins>: Before we proceed with the calculations, let us pause and reflect on the situation a little. When client #13-A arrives, it could find that all servers are busy if no service was completed in the intervening time between its arrival and the exit of the

previous customer who departed the queue on finding all servers to be busy. Alternatively, client #13-A could find any of $0 \leq m \leq (n-1)$ servers to be busy depending on the number of services that were completed in the duration between its arrival and the exit of the previous client who left out of frustration. And since there is no guarantee when client #13-A actually enters the queue and the fact that the service times of any of the servers are not uniform, this is truly a stochastic process.
To proceed with our calculations, let us consider T_k to be the expected number of busy servers found by client #13-A if there are currently k busy servers. Here the word currently refers to the epoch of the exit of the frustrated client. We want to estimate T_n. It turns out that the definition of T_k above is equivalent to the following: T_k is the expected number of busy servers found by client #13-A in a k-server system when there are currently k busy servers. The underlined phrase is true because the fact that there will be at least $(n - k)$ idle servers can be ignored because of the memoryless (Markovian) property of exponential service times and exponential interarrival times. The boundary condition $T_0 = 0$ is self evident and needs no further explanation. Our next objective will be to find T_1 again for a 1-server system. In such a case, either 1 or 0 servers can be found to be busy by client #13-A. Therefore,

$$\begin{aligned} T_1 &= (1) \times \texttt{Prob}(\text{client \#13-A finds one server busy}) \\ &\qquad\qquad +(0) \times \texttt{Prob}(\text{client \#13-A finds zero servers busy}) \\ &= (1)\underset{\nearrow}{\frac{\lambda}{\lambda+\mu}} + (0)\frac{\mu}{\lambda+\mu} \end{aligned}$$

cf. equation 2.49

$$= \frac{\lambda}{\lambda+\mu}. \qquad (4.1)$$

For the general case, with k busy servers, we will obtain T_k by conditioning upon what happens next. If we label an event of a new arrival or a completion of a service to an alarm clock going off then with k busy servers, we will need k numbers of $exp(\mu)$ alarm clocks and 1 number of $exp(\lambda)$ alarm clock. We will use the following two partitioning events to condition our calculation of T_k using the law of total expectation.

1. Event $\mathfrak{E}_1$: A service completion happens first during the intervening time before client #13-A arrives.

2. Event $\mathfrak{E}_2$: An arrival happens first (the arrival of client #13-A) before any server becomes available.

Additionally, if $\tau_i,\ i = 1, 2, ..., k$ denotes time to complete a service by server i, then the time till the next service completion after the exit of the frustrated client is distributed as $min(\tau_1, \tau_2, ..., \tau_k) \sim exp(k\mu)$. Therefore, $\texttt{Prob}(\text{ event } \mathfrak{E}_1) = \frac{k\mu}{\lambda+k\mu}$ and

$\texttt{Prob}(\text{ event } \mathfrak{E}_2) = \frac{\lambda}{\lambda+k\mu}$. Now we will use the law of total expectation.

$$\begin{aligned}
T_k &= E(\#\text{busy servers}\big|\mathfrak{E}_1 \,\&\text{ currently } k \text{ busy servers})\texttt{Prob}(\mathfrak{E}_1 \mid \text{currently } k \text{ busy servers}) \\
&= \quad E(\#\text{busy servers}\big|\mathfrak{E}_2 \,\&\text{ currently } k \text{ busy servers})\texttt{Prob}(\mathfrak{E}_2 \mid \text{currently } k \text{ busy servers}) \\
&= T_{k-1}\frac{k\mu}{\lambda+k\mu} + \underset{\nearrow}{k}\frac{\lambda}{\lambda+k\mu}. \qquad (4.2)
\end{aligned}$$

due to condition `cond++`

Here we have used the phrase #busy servers as a short-hand for *'number of busy servers'*. Now, we may use the above recurrence relation and find $T_2, T_3, ...$ and we list a few of them below.

$$\begin{aligned}
T_2 &= T_1\frac{2\mu}{2\mu+\lambda} + \frac{2\lambda}{2\mu+\lambda} \\
&= \frac{\lambda}{\lambda+\mu}\frac{2\mu}{2\mu+\lambda} + \frac{2\lambda}{2\mu+\lambda}.
\end{aligned}$$

$$\begin{aligned}
T_3 &= T_2\frac{3\mu}{3\mu+\lambda} + \frac{3\lambda}{3\mu+\lambda} \\
&= \frac{\lambda}{\lambda+\mu}\frac{2\mu}{2\mu+\lambda}\frac{3\mu}{3\mu+\lambda} \\
&\quad + \frac{2\lambda}{2\mu+\lambda}\frac{3\mu}{3\mu+\lambda} + \frac{3\lambda}{3\mu+\lambda}.
\end{aligned}$$

And in general,

$$T_n = \frac{n\lambda}{n\mu+\lambda} + \sum_{i=1}^{n-1}\frac{i\lambda}{i\mu+\lambda}\prod_{j=i+1}^{n}\frac{j\mu}{j\mu+\lambda}. \qquad (4.3)$$

The above example is in fact an illustration of a continuous time stochastic process and is associated with a continuous time Markov chain with rate parameters $q_{i,i+1} = \lambda_i = \lambda$ and $q_{i,i-1} = \mu_i = \mu$ and a state space $\mathcal{S} = \{0, 1, 2, ..., n\}$ that denotes the number of busy servers (or number of people in the system). These notations and the concept of a continuous time Markov chain (CTMC) will be the subject of our study in the following sections.

Example: Stochastic arrivals follow a Poisson distribution

Consider identical and independent arrivals at a fixed rate λ on a linear time axis beginning with the epoch $t = t_0 = 0$ from the state $\mathfrak{E}_0$. $\mathfrak{E}_k$ denotes the event of the k^{th} arrival at epoch $t = t_k$. Here the subscript refers to the number of arrivals up to that instant. This stochastic process defines a jump transition from $\mathfrak{E}_j$ to $\mathfrak{E}_{j+1}$ between two successive arrivals. Whatever the state $\mathfrak{E}_j$ at a certain epoch $t_j \le t < t_{j+1}$, the probability of a jump (an arrival) between epochs t and $(t+h)$ (for small $h > 0$) is $\lambda h + o(h)$ whereas the probability of more than one jump (arrival) is $o(h)$. Define a random variable $Z(t)$ that counts the number of arrivals in an arbitrary interval of time of length t.

Figure 4.3: An artist's rendition of queues at the checkout counters in a shopping mall. At peak hour and during festive season, the mall manager will be interested to know the average number of people in the queue at any given point of time in order to marshal his employee resources judiciously.

Show that

$$p_n(t) = P\big(Z(t) = n\big) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{4.4}$$

Solution: We will work with $n \geq 1$. Consider the event at epoch $t + h$ that is designated by the state $\mathfrak{E}_n$. The probability of this event is $p_n(t + h)$ as per the definition above. This event can happen in three mutually exclusive manners as enumerated below.

1. Event $\mathcal{E}_1$: The system was in state $\mathfrak{E}_n$ at epoch t and no arrival happened between t and $t+h$. The probability of this event $P(\mathcal{E}_1) = p_n(t)p_0(h) = p_n(t)\Big\{1 - \lambda h\Big\} + o(h)$.
2. Event $\mathcal{E}_2$: The system was in state $\mathfrak{E}_{n-1}$ at epoch t and exactly one arrival happened between t and $t+h$. $P(\mathcal{E}_2) = p_{n-1}(t)p_1(h) = p_{n-1}(t)\Big\{\lambda h\Big\} + o(h)$.
3. Event $\mathcal{E}_3$: The number of arrivals between epochs t and $t + h$ is more than one and the probability of such an event $P(\mathcal{E}_3) = o(h)$ as defined in the problem statement.

Since the events $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ are mutually exclusive, the probabilities simply add up and we have the following result.

$$p_n(t+h) = p_n(t)\Big\{1 - \lambda h\Big\} + p_{n-1}(t)\Big\{\lambda h\Big\} + o(h), \tag{4.5}$$

which may be rewritten as

$$\frac{p_n(t+h)-p_n(t)}{h} = -\lambda p_n(t) + \lambda p_{n-1}(t) + \frac{o(h)}{h}. \tag{4.6}$$

Upon evaluating the above in the limit $h \rightarrow 0$, we have the following recurrence relation.

$$p_n'(t) = -\lambda p_n(t) + \lambda p_{n-1}(t), \quad n \geq 1. \tag{4.7}$$

When $n = 0$, the only possible route at our disposal is through the event $\mathcal{E}_1$ which leads to $p_0'(t) = -\lambda p_0(t)$. The boundary condition for $n = 0$ is $p_0(0) = 1$ whence we get $p_0(t) = e^{-\lambda t}$. Using this result in the recurrence relation 4.7 and solving for $p_1(t)$ we obtain $p_1(t) = \lambda t e^{-\lambda t}$ that is in agreement with equation 4.4. Proceeding in this recursive manner, we can deduce the generic relation 4.4 for $p_n(t)$.

Note: The little-o notation used in the above example is defined as follows: $\lim_{h\rightarrow 0} \frac{o(h)}{h} = 0$.[1]

4.4 *Continuous Time Markov Chains (CTMC)*

We will begin this section by revisiting the Markov property in the context of a continuous time stochastic process. We will then deduce an equation whose solution generates the stationary distribution of the stochastic process. This result will enable us to investigate asymptotic behavior of queuing systems with wide economic implications for businesses to conduct their operations efficiently by allocating optimal resources.

4.4.1 *Markov property for continuous time processes*

Consider a continuous time stochastic process $\{X(t)\}_{t\geq 0}$ which takes on discrete values (states) from the state space $\mathcal{S}$. The Markov property[2] for continuous time stochastic processes can be stated as follows.

$$P\big(X(t) = j \big| X(s) = i, X(t_{n-1}) = i_{n-1}, ..., X(t_1) = i_1\big) = P\big(X(t) = j \big| X(s) = i\big); \tag{4.8}$$

where $0 \leq t_1 \leq t_2 \leq \cdots \leq t_{n-1} \leq s \leq t$ and $i_1, i_2, ..., i_{n-1}, i, j \in \mathcal{S}$ are the $(n+1)$ states in the state space $\mathcal{S}$ for all $n \geq 1,\ n \in \mathbb{I}^+$.

4.4.2 *Definition: Continuous time Markov chains*

A continuous time stochastic process $\{X(t)\}_{t\geq 0}$ is called a continuous time Markov chain (CTMC) if it obeys the Markov property 4.8. Additionally, a CTMC may be time-homogeneous (or stationary) if it satisfies the condition discussed below.

4.4.3 *Time-homogeneity of Markov chains*

We say that a continuous time Markov chain is *time-homogeneous* if for any $s \leq t$ and any states $i, j \in \mathcal{S}$, the following is true.

$$\begin{aligned} p_{i,j}(t-s) \equiv P\big(X(t) = j \big| X(s) = i\big) &= P\big(X(t-s) = j \big| X(0) = i\big) \\ &= P\big(X(t_1) = j \big| X(s + t_1 - t) = i\big). \end{aligned} \tag{4.9}$$

[1] Equivalently, we say $f(n) = o(g(n))$ if $\lim_{n\rightarrow\infty} \frac{f(n)}{g(n)} = 0$.

[2] Markov property is also colloquially known as the memoryless property.

The important thing to note in the above statement is that the conditional probabilities do not depend on any particular epoch but only on the interval $(t-s)$. Such a time-homogeneous process is also a *stationary* process.[3] It is essential to emphasize that not all CTMC are time-homogeneous (or stationary) but in this chapter we will be mostly concerned with time-homogeneous stochastic processes.

4.4.4 *Holding time of a Markov chain*

When the stochastic process enters state i, the time it spends in that state before it leaves state i is called the *holding time* $\mathcal{T}_i$. if the arrival rate of this process is λ_i, then $\mathcal{T}_i \sim exp(\lambda_i)$ as was the case in the introductory example of this chapter.

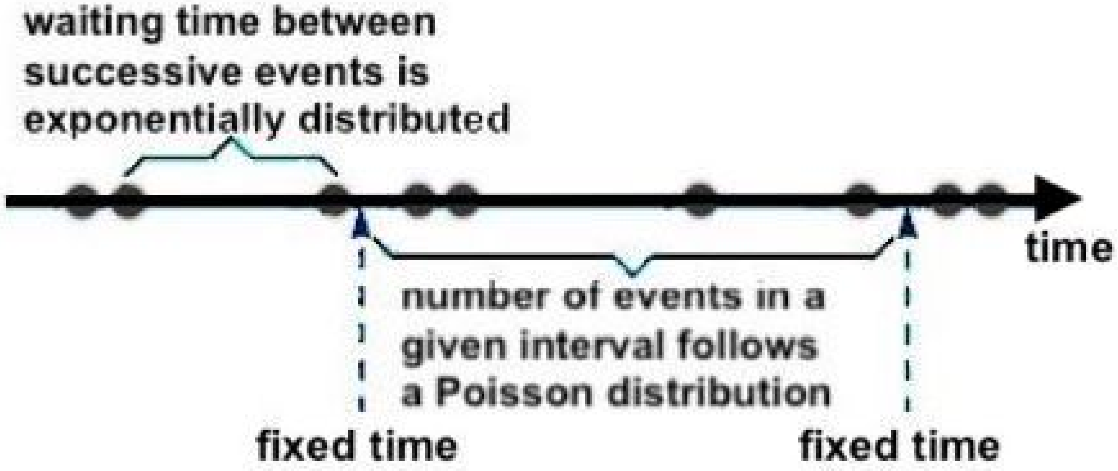

waiting time between successive events is exponentially distributed
time
number of events in a given interval follows a Poisson distribution
fixed time
fixed time

Figure 4.4: Probability distributions associated with a Poisson point process.

Recall that for a discrete stochastic process such as the discrete time Markov chain, the probability of transition from state i to state j, denoted by p_{ij}, along with the initial set of probability distribution, completely determines the state of the system at all latter times. For a CTMC, the rate at which events occur not only characterizes the epochs of state transitions[4] but also determines the long-run distribution of states of the system. We will devote the next few sections to illustrate these facts in a rigorous manner.

In order to deduce a model of the generator of this continuous time Markov chain, we will have to begin by defining the rates as follows. Let $q_{i,j}$ is the rate at which the system goes from state i to state j. This is akin to the rates at which the exponential alarm clocks of section 4.3 go off. These rates are functions of time as are the probability transitions, i.e. $p_{i,j} \equiv p_{i,j}(t)$ and $q_{i,j} \equiv q_{i,j}(t)$. Since the $p_{i,j}$s are probabilities, it should make sense to define them as follows.

$$p_{i,j} := \frac{q_{i,j}}{\sum_{j\in\mathcal{S}} q_{i,j}}. \tag{4.10}$$

If we denote $v_i = \sum_{j\in\mathcal{S}} q_{i,j} < \infty$, then the rates may be defined as follows.

$$q_{i,j} := v_i p_{i,j}. \tag{4.11}$$

By definition we have $q_{i,i} = 0$. Further, if $v_i = 0$, then state i is an absorbing state. The entries $p_{i,j}$ constitute the *stochastic matrix* $\mathbb{P} \equiv \mathbb{P}(t)$. The entries $q_{i,j}$ constitute the *generator matrix*

[3] Intuitively, this means that whenever the process enters state i, the way it evolves stochastically from that epoch is the same as if the process started in state i at time 0.

[4] This is true in the sense of a probability distribution.

for reasons that will become clear during the ensuing discussion. It may be noted that v_is and $p_{i,j}$s can be computed from the $q_{i,j}$s. Further, because $q_{i,j}$s are essentially rates, they are positive quantities.

4.4.5 *Chapman-Kolmogorov equation for CTMC*

Analogous to the discrete case (**cf.** equation 3.6), the Chapman-Kolmogorov equation for CTMC is presented below.

$$p_{i,j}(t+s) = \sum_{k \in \mathcal{S}} p_{i,k}(t) p_{k,j}(s) = \sum_{k \in \mathcal{S}} p_{i,k}(s) p_{k,j}(t) \tag{4.12}$$

is the $(i,j)^{th}$ entry of the matrix $\mathbb{P}(t+s) = \mathbb{P}(t)\mathbb{P}(s)$. The Chapman-Kolmogorov equation will be used to derive the most important result of this chapter known as the Kolmogorov backward and forward equations. This is explained in the following paragraphs.

4.4.6 *Kolmogorov equations and the generator matrix*

Let us reconsider some of the ideas we used in solving the last example problem of section 4.3. In that case, since we had an arrival process, and we assumed that we will allow for only one arrival at a time for all practical purposes, the transitions between the states happened in increments of one. However, we will relax this restriction; in the case of a general CTMC (which may not be an arrival process only but may capture something far broader in scope) we will allow for successive transitions between states i and j which need not only differ by one. In the set-up of our previous example, j was necessarily $i+1$. However, the transitioning events are still the same in the infinitesimally small duration h, namely, $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ with the same probabilities. The only difference in the present case is that $\lambda = \lambda_i = v_i = \sum_{j \in \mathcal{S}} q_{i,j}$ as this is the most generic form of the rate of transition emanating from state i. One may pose a very pertinent question here - *why is the rate of transition from state i given as a sum over $q_{i,j}$s?* In order to understand this formulation, we invite the reader to think through the motivating example of this chapter on the n-server queuing model. In that case, we had k servers, so the time until one of those servers completed service was distributed as $exp(k\mu)$. In other words, the servers became available at the rate of sum of k μs, which is analogous to the rate $\sum_{j \in \mathcal{S}} q_{i,j}$, the sum over all the rates of transitions from state i.

In order to derive a differential equation for the probability transitions, we will begin with the Chapman-Kolmogorov relation 4.12.

$$\begin{aligned} p_{i,j}(t+h) &= \sum_{k \in \mathcal{S}} p_{i,k}(h) p_{k,j}(t) \\ &= p_{i,j}(t)\{1 - v_i h + o(h)\} + \sum_{k \neq i} p_{i,k}(h) p_{k,j}(t) \\ &= p_{i,j}(t)\{1 - v_i h + o(h)\} + \sum_{k \neq i} p_{k,j}(t)\{p_{i,k} v_i h + o(h)\}. \end{aligned} \tag{4.13}$$

The first term on the r.h.s. of the above equation stems from an event of the type $\mathcal{E}_1$ (**cf.** last example of sec. 4.3), whereas the second term arises from events of the types $\mathcal{E}_2$ and $\mathcal{E}_3$. The computation of $p_{i,k}(h)$ is performed by applying the law of total probability and upon recalling.

$$\begin{aligned} p_{i,k}(h) &= P\big(X(h)=k\big|X(0)=i,\Delta\mathcal{E}_2\big)P(\Delta\mathcal{E}_2) \\ &\quad +P\big(X(h)=k\big|X(0)=i,\Delta\mathcal{E}_3\big)P(\Delta\mathcal{E}_3) \\ &= p_{i,k}\big(v_i h\big)+p^2_{i,k}\big(o(h)\big) \\ &= p_{i,k}\big(v_i h\big)+o(h). \end{aligned} \tag{4.14}$$

Here $\Delta\mathcal{E}_i,\ i=2,3$ is the event of one jump transition (for $i=2$) and more than one jump transitions (for $i=3$) when the system migrates from state i to k as is explained in the last example of section 4.3. Re-arranging the terms in equation 4.13 and dividing all the terms by h followed by evaluating the terms in the limit $h\to 0$, we get

$$p'_{i,j}(t)=\sum_{k\neq i} q_{i,k}p_{k,j}(t)-v_i p_{i,j}(t), \tag{4.15}$$

Here we have used the definition of the transition rates $q_{i,k}=v_i p_{i,k}$. In matrix form, equation 4.15 can be expressed as follows.

$$\mathbb{P}'(t)=\mathbb{G}\mathbb{P}(t), \tag{4.16}$$

or equivalently,

$$\Big(\mathbb{P}'(t)\Big)_{i,j}=\Big(\mathbb{G}\mathbb{P}(t)\Big)_{i,j}. \tag{4.17}$$

Equations 4.15-4.17 are called the *Kolmogorov backward equations*. The infinitesimal generator matrix $\mathbb{G}\equiv\big(g_{i,j}\big)$ is summarized below.

$$\begin{aligned} g_{i,j} &= q_{i,j}=v_i p_{i,j}, \quad \forall i\neq j, && (4.18)\\ g_{i,i} &= -v_i, && (4.19) \end{aligned}$$

with boundary conditions $\mathbb{P}(0)=\mathbb{I}$, where $\mathbb{I}$ is the identity matrix.

Likewise, we can derive the *Kolmogorov forward equations* which we simply state below in matrix form.

$$\mathbb{P}'(t)=\mathbb{P}(t)\mathbb{G}. \tag{4.20}$$

The solutions to the Kolmogorov equations 4.15-4.20 with the prescribed boundary conditions can be computed very easily in the form of matrix exponentials.

$$\mathbb{P}(t)=e^{t\mathbb{G}}:=\mathbb{I}+t\mathbb{G}+\frac{(t\mathbb{G})^2}{2!}+\cdots. \tag{4.21}$$

In the case of a finite state space system, the solution 4.21 can be well approximated by truncating the terms of the infinite sum in 4.21. The solution $\mathbb{P}(t)=e^{t\mathbb{G}}$ underscores the importance of the generator matrix $\mathbb{G}$ vis-à-vis a continuous time Markov chain because it completely generates the solution $\mathbb{P}(t)$. Consequently, $\mathbb{G}$ plays an important role in obtaining the stationary distribution of a Markov chain. This implies that in the case of CTMC (unlike in the case of DTMC), the rates of state transitions completely determine the solutions of the system. Further, the CTMC with the generator (or rate) matrix $\mathbb{G}$ (or equivalently $\big(g_{i,j}\big)$) bears with it an *embedded* DTMC with transition probabilities prescribed by the matrix $\big(p_{i,j}\big)$.

4.4.7 *Stationary distribution of CTMC*

Consider a CTMC $\{X(t)\}_{t\geq 0}$ with finite state space $\mathcal{S}$,[5] generator $\mathbb{G}$, and matrix of probability transition functions $\mathbb{P}(t)$. The $|\mathcal{S}|$-dimensional row vector $\boldsymbol{\pi} \equiv (\pi_i)_{i\in\mathcal{S}}$ with $\pi_i \geq 0$, $\forall i$ and with the constraint $\sum\limits_{i\in\mathcal{S}} \pi_i = 1$ is a stationary distribution of the Markov chain if

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbb{P}(t), \quad \forall t \geq 0. \tag{4.22}$$

Example: $\boldsymbol{\pi}\mathbb{G} = \mathbf{0}$ describes the stationary solution of CTMC

Use the generator matrix $\mathbb{G}$ to deduce a condition for stationarity of a CTMC.

Solution: If $\boldsymbol{\pi}$ is a stationary distribution of a Markov chain, then the following are true.

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbb{P}(t),\ \forall t \geq 0,$$

$$\underset{\nearrow}{=} \boldsymbol{\pi} \sum_{n=0}^{\infty} \frac{(t\mathbb{G})^n}{n!}.$$

cf. equation 4.21 whence $\mathbb{P} = e^{t\mathbb{G}} = \cdots$

$$\underset{\nearrow}{\Longleftrightarrow} \quad \boldsymbol{\pi} - \boldsymbol{\pi} = \sum_{n=1}^{\infty} \frac{t^n}{n!}\boldsymbol{\pi}\mathbb{G}^n$$

this symbol stands for *if and only if*

$$\Longleftrightarrow \quad 0 \underset{\nearrow}{=} \boldsymbol{\pi}\mathbb{G}^n,\ \forall n \geq 1,$$

sum of positive terms is zero $\Longleftrightarrow$ summand is zero.

$$\Longleftrightarrow \quad \boxed{\boldsymbol{\pi}\mathbb{G} = \mathbf{0}}. \tag{4.23}$$

Using equation 4.23 to find the stationary distribution of states is computationally more convenient than using equation 4.22. In the former case, one simply has to solve a system of $|\mathcal{S}|$ linear equations. Of course the solution must be compliant with the fundamental axiom of probability $\sum\limits_{i\in\mathcal{S}} \pi_i = 1$.

For a CTMC $\{X(t)\}_{t\geq 0}$, the stationary probability distribution is also the limiting probability ($t \to \infty$), i.e.

$$\lim_{t\to\infty} p_{i,j}(t) = \lim_{t\to\infty} P\big(X(t) = j \big| X(0) = i\big) \equiv \pi_j. \tag{4.24}$$

4.4.8 *Global balance equations $\boldsymbol{\pi}\mathbb{G} = \mathbf{0}$*

Let us investigate the equation 4.23 further. In component form, equation 4.23 is $\sum\limits_{i\in\mathcal{S}} \pi_i g_{i,j} = \pi_j g_{j,j} + \sum\limits_{i\neq j} \pi_i g_{i,j} = 0$. These $|\mathcal{S}|$ equations can be written more explicitly by referring to the equations 4.18-4.19.

$$\pi_j v_j = \sum_{i\neq j} \pi_i q_{i,j}, \tag{4.25}$$

[5] Here we will assume that the cardinality of the set $\mathcal{S}$ is $|\mathcal{S}|$.

where $v_j = \sum\limits_{i \in \mathcal{S}} q_{j,i}$. Each of the terms in the above equation can be interpreted in the following manner.

1. $\underline{\pi_j}$: This term refers to the long run proportion of time the system stays in state j.
2. $\underline{v_j = \sum\limits_{i \in \mathcal{S}} q_{j,i}}$: This term estimates the rate of departure from state j when the system is in state j.
3. $\underline{\pi_j v_j}$: This term computes the long run rate of leaving state j.
4. $\underline{\pi_i q_{i,j}}$: This term calculates the long run rate of going from state i to state j.
5. $\underline{\sum\limits_{i \neq j} \pi_i q_{i,j}}$: This term estimates the long run rate of going to state j starting not from state j.

Therefore, equation 4.25 is a statement of balance between flux out of state j and flux into state j, i.e. it is a statement of dynamic equilibrium. This is why the equation 4.23 is known as the *global balance equation* or simply the *balance equation*.

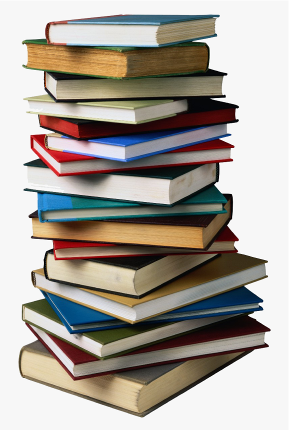


Figure 4.5: A huge pile of books that can stand firm against a gust of breeze. Not only are individual books in balance with their neighbors but the whole pile is in balance as a whole.

Example: State of a machine prototype

A new prototype washing machine using a revolutionary technology is under development. During the development cycle, it passes through one of three distinct states, viz., normal, test, and repair modes. The rate diagram of the states depicting the entries of the generator matrix is shown in the figure in the margin. Find the stationary distribution of states.

Solution: The generator matrix can be deduced from the rate diagram as follows $\mathbb{G} = \begin{pmatrix} -1 & 1 & 0 \\ 2.5 & -5 & 2.5 \\ 3 & 0 & -3 \end{pmatrix}$. In order to find the stationary distribution of states, we must solve $\boldsymbol{\pi}\mathbb{G} = \mathbf{0}$ and $\pi_1 + \pi_2 + \pi_3 = 1$. The former simplifies to $\pi_2 = \frac{3\pi_3}{2.5}$ and $\pi_1 = 5\pi_2$. When we use this simplification in the latter equation, we obtain $\pi_3 = \frac{5}{41}$. Substituting this in the previously deduced results, we obtain $\pi_2 = \frac{6}{41}$ and $\pi_1 = \frac{30}{41}$. Thus the stationary distribution of states of the prototypical washing machine is $\left(\frac{30}{41} \; \frac{6}{41} \; \frac{5}{41}\right)$ corresponding to the normal, test, and repair modes.

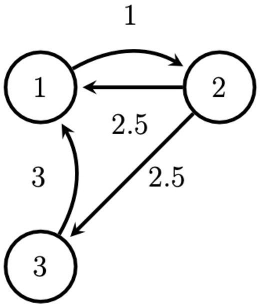


Figure 4.6: Schematic representation of the generator (rate) matrix.

4.4.9 *Detailed (or local) balance*

For a CTMC with a generator matrix $\mathbb{G} \equiv \left(q_{i,j}\right)$, if we can find a distribution of states $\boldsymbol{\pi}$ such that for every pair of states i and j, the following relation holds

$$\pi_i q_{i,j} = \pi_j q_{j,i}, \tag{4.26}$$

then it can be easily shown, by summing over all the states in $\mathcal{S}$, that $\boldsymbol{\pi}$ is a stationary distribution and satisfies the global balance equation 4.25. Let us sum over all the states i to obtain $\sum\limits_{i \in \mathcal{S}} \pi_i q_{i,j} = \sum\limits_{i \in \mathcal{S}} \pi_j q_{j,i} = \pi_j \sum\limits_{i \in \mathcal{S}} q_{j,i} = \pi_j v_j$, which is the equation 4.25.[6] The detailed

[6] The equivalent detailed balance for DTMC is $\pi_i p_{i,j} = \pi_j p_{j,i}, \ \forall i, j \in \mathcal{S}$.

balance may not always hold even when the global balance holds. The converse is always true as shown above.

When does detailed balance not hold for sure? A quick way to check this is to inspect the rate or the generator matrix $\mathbb{G}$ and see if there are any rates $q_{i,j}$ and $q_{j,i}$ such that $q_{i,j} > 0$ and $q_{j,i} = 0$ (or conversely $q_{i,j} = 0$ and $q_{j,i} > 0$). If any of these pathological cases exist for any pair i, j, then the local analysis using the detailed balance condition to estimate the stationary distribution will be futile.

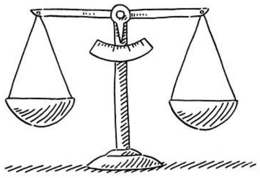


Figure 4.7: Everything is in perfect balance.

Theorem: A CTMC is reversible if and only if the detailed balance condition holds for every pair (i, j) in $\mathcal{S}$.

4.4.10 *Application of detailed balance condition*

For the following two well-known continuous stochastic processes, the detailed balance does hold and is useful to find the stationary probability distribution of states. They are

1. birth-death processes,
2. $M/M/1$ queues (Poisson arrivals and exponential service rate by a one-server system).

We discuss these two stochastic processes in detail in the subsequent sections of this chapter.

4.5 *Birth and death processes*

A birth and death process is a homogeneous stochastic Markov process where the transitions between two successive epochs[7] constitute either a jump by one (i.e. state i goes to state $i+1$) or a drop by one (i.e. state i goes to $i-1$). In other words, only *nearest-neighbor* transitions are permissible. Since this stochastic process is a continuous time process, the transitions between states happen at a prescribed rate.

$$q_{i,i+1} = \lambda_i > 0, \quad i \geq 0, \quad \text{(birth rates); and} \tag{4.27}$$

$$q_{i,i-1} = \mu_i > 0, \quad i \geq 1, \quad \text{(death rates).} \tag{4.28}$$

The state space $\mathcal{S}$ is over the space of all whole numbers.

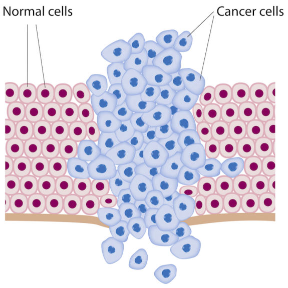


Figure 4.8: Carcinogenic growth of cells and apoptosis can be studied using birth and death models.

4.5.1 *Example: Stationary distribution of a birth and death stochastic process*

Deduce the stationary distribution of the birth-death process defined in equations 4.27 and 4.28.

Solution: Since births and deaths are well-defined and non-trivial at every point on the state space, it is easy to validate by inspection that the local analysis of the detailed balance will be useful to find the stationary distribution of states. Therefore, we begin by writing the detailed balance equations at every point of state transition as

[7] Here an epoch is a time instant when an event (or a transition of state) occurs such as an arrival or a death, etc.

follows.

$$
\begin{aligned}
\pi_i q_{i,i+1} &= \pi_{i+1} q_{i+1,i} \\
\implies \pi_i \lambda_i &= \pi_{i+1} \mu_{i+1}, \ \forall i \geq 0; \\
\implies \pi_{i+1} &= \frac{\lambda_i}{\mu_{i+1}} \pi_i \\
&\underset{\substack{\nearrow \\ \text{recursively}}}{=} \frac{\lambda_i \lambda_{i-1}}{\mu_{i+1} \mu_i} \pi_{i-1} \\
&\ \ \vdots \\
&\underset{\substack{\nearrow \\ \text{continuing recursively}}}{=} \frac{\lambda_i \cdots \lambda_0}{\mu_{i+1} \cdots \mu_1} \pi_0. \qquad (4.29)
\end{aligned}
$$

Additionally, the π_is must satisfy the axioms of probability; specifically, the relation $\sum\limits_{i \in \mathcal{S}} \pi_i = 1$ must hold. Re-indexing $i = j - 1$ in equation 4.29 and using the aforementioned normalization of the probabilities, we have

$$
\begin{aligned}
\pi_0 + \sum_{j=1}^{\infty} \pi_j &= \pi_0 + \sum_{j=1}^{\infty} \frac{\lambda_{j-1} \cdots \lambda_0}{\mu_j \cdots \mu_1} \pi_0 = 1 \\
&\implies \pi_0 \left\{ 1 + \sum_{j=1}^{\infty} \frac{\lambda_{j-1} \cdots \lambda_0}{\mu_j \cdots \mu_1} \right\} = 1 \\
&\implies \pi_0 = \frac{1}{1 + \sum\limits_{j=1}^{\infty} \frac{\lambda_{j-1} \cdots \lambda_0}{\mu_j \cdots \mu_1}}. \qquad (4.30)
\end{aligned}
$$

We must ensure that the infinite power series in the denominator of 4.30 is finite.

4.5.2 *Example: servicing of machines by a repairman*

Let us consider a factory that houses and operates m machines. The machines are also serviced by a repairman in case any of them break down and need repair. The repairman services one machine at a time on a first-come-first-serve basis, so when more than one machine is dysfunctional, then the idle machines go in a service queue. Each of these machines can go from a working state to a service state (due to a breakdown/failure) within a randomly distributed time $\mathcal{T}_f \sim exp(\lambda)$ (measured from any given time). Further, the service time $\mathcal{T}_s$ of each machine is distributed according to a rate μ exponential distribution, $\mathcal{T}_s \sim exp(\mu)$. Answer the following questions.

1. What is the stationary distribution of states of working machines?
2. What is the expected number of machines in the waiting line?

Figure 4.9: A repairman servicing a machine in a factory line.

Solution: Let us define the state of the system S_k when k of the m machines are idle due to a failure. This is a birth and death process because only one of the two transitions is possible: (i) $S_k \rightarrow S_{k+1}$ when a new failure happens, and (ii) $S_k \rightarrow S_{k-1}$ when a repair is successfully completed and a machine is brought back to function. We will begin by writing the detailed balance relations:

$$\begin{aligned} \pi_i q_{i,j} &= \pi_j q_{j,i}, \\ \pi_t q_{i,i+1} &= \pi_{i+1} q_{i+1,i}, \\ \pi_i (m-i)\lambda &= \pi_{i+1}\mu. \end{aligned} \tag{4.31}$$

The jump rate $q_{i,i+1}$ is $(m - i)\lambda$ can happen owing to any of the $(m - i)$ working machines becoming dysfunctional but the death rate is constant $q_{i+1,i} = \mu$ because the repairman works on only one machine at a time until it is restored. Setting $j = i + 1$, equation 4.31 can be re-written as

$$\pi_{j-1} = \frac{\mu}{\lambda}\frac{1}{m-j+1}\pi_j. \tag{4.32}$$

In order to deduce a recursive relation for the stationary distribution of working machines, we set $j = m$ in equation 4.32

$$\pi_{m-1} = \frac{\mu}{\lambda}\frac{1}{1!}\pi_m, \tag{4.33}$$

$$\pi_{m-2} = \left(\frac{\mu}{\lambda}\right)^2 \frac{1}{1 \times 2}\pi_m, \tag{4.34}$$

$$\cdot = \tag{4.35}$$

$$\cdot = \tag{4.36}$$

$$\cdot = \tag{4.37}$$

$$\pi_{m-k} = \left(\frac{\mu}{\lambda}\right)^k \frac{1}{k!}\pi_m. \tag{4.38}$$

Next we will use one of the axioms of probability

$$\sum_{k=0}^{m} \pi_{m-k} = 1$$

to find $\pi_m = \frac{1}{1+\sum\limits_{k=1}^{m}\frac{1}{k!}\left(\frac{\mu}{\lambda}\right)^k}$. The stationary distribution of working machines is listed below.

$$\pi_{m-k} = \left(\frac{\mu}{\lambda}\right)^k \frac{1}{k!} \frac{1}{1+\sum\limits_{k=1}^{m}\frac{1}{k!}\left(\frac{\mu}{\lambda}\right)^k}. \tag{4.39}$$

This is known as the famous *Erlang's loss formula*.

1. Special cases emerge depending on different values of k; eg., $k = m$ in equation 4.39 gives us π_0, which can be interpreted as the long-run probability that the repairman is idle (all machines working).

2. The expected number of machines in the service queue (awaiting repair) is given by

$$\begin{aligned} M_q &= \sum_{k=0}^{m-1} k\pi_{k+1} \\ &= \sum_{k=1}^{m} k\pi_k - (1-\pi_0). \end{aligned} \tag{4.40}$$

Summing equation 4.31 over $i = 0$ through m yields $m\lambda - \lambda \sum_{k=1}^{m} k\pi_k = \mu(1-\pi_0)$, which in conjunction with equation 4.40 gives us the expected number of machines in the service queue.

$$M_q = m - \frac{\lambda+\mu}{\lambda}(1-\pi_0). \tag{4.41}$$

We leave the readers here with an exercise to work out a model and its solution akin to the example 4.5.2 above but with $r < m$ repairmen who can work concurrently to repair the idle machines.

4.6 Queues (Erlang-T models)

All it provided was hope for people to cling to and a reason to stay in the queue.[8] At some point in our lives, a good majority of us have gone through this emotion while waiting in a queue either at a train ticket counter or voting booth or perhaps any of the many queues we may have inhabited, albeit for a fleeting moment when compared to the vastness of our lived experience. A common thread of thoughts that we share in such moments is *how long do I have to stay in the queue?* or *how many people on an average are in the queue at a certain time?* These questions and their answers not only have an ontological basis but also a very practical material value. Systems and businesses operate around finding optimal answers to such questions. We will formally study single and multi-server queues in this section and also in the chapter project.

4.6.1 $M/M/n$ queue and Kendall's notation

The server-queue models that will be discussed in this chapter have Poisson arrivals.[9] We will use Kendall's notation here $M/M/n$, where the first M from the left stands for the memoryless property of the exponentially distributed inter-arrival time with rate parameter λ, the second M stands for the memoryless property of the exponentially distributed service times with rate parameter μ, and n refers to the number of servers. Let $X(t)$ be the random variable that denotes the number of customers in the system at time t.

We will begin our analysis when $n = 1$. We summarize below the main attributes of the $M/M/1$ system.

1. It is a single server system.

2. Customers enter the system and arrive in a queue if the server is busy. If the server is available, then they proceed straight to service.

3. Arriving clients are served by the server on a first-come-first-serve basis.

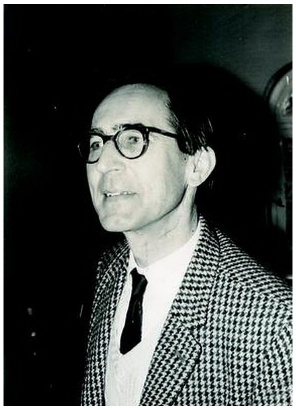


Figure 4.10: British statistician David George Kendall known for his pioneering work in queuing theory (*courtesy: Wikimedia Commons*).

[8] Excerpt from **The Queue** by Basma Abdel Aziz.

[9] cf. PASTA (Poisson Arrival See Time Averages) - here the state of the queue-server system is invariant in distribution to the location of the observer (the observer can be within the system/can be arriving in the queue or the observer can be fully outside the system).

4. Upon completion of service, the clients depart the system.

The $M/M/1$ model is a birth and death system with state space $\mathcal{S} = \left\{0, 1, 2, ...\right\}$. The birth and death rates are $\lambda_i = \lambda,\ \forall i \geq 0$ and $\mu_i = \mu,\ \forall i \geq 1$. Recall from the birth and death model of the previous section, i.e. equations 4.29 and 4.30, the necessary condition for a stationary distribution is $1 + \sum\limits_{i=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^i < \infty$ or $\sum\limits_{i=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^i < \infty$. This geometric series converges if $\frac{\lambda}{\mu} < 1$. This means that for a stationary distribution of states to exist (steady state), $\lambda < \mu$. Thus the stability criterion for the $M/M/1$ queue is

$$\boxed{\text{arrival rate } \lambda < \text{service rate } \mu}. \tag{4.42}$$

4.6.2 *Example: Stationary distribution of a stable $M/M/1$ queue*

Find the stationary distribution of states for an $M/M/1$ queue with arrival and service rates λ and μ respectively. Assume that $\lambda < \mu$.

<u>Solution</u>: Based on equation 4.30, we deduce $\pi_0 = \frac{1}{\sum\limits_{i=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^i} = \left(\frac{1}{1-\frac{\lambda}{\mu}}\right)^{-1} = 1 - \frac{\lambda}{\mu}$. Here we have used the result of the sum of an infinite geometric series.
Likewise, using equation 4.29, we have

$$\begin{aligned} \pi_i &= \frac{\lambda_{i-1} \cdots \lambda_0}{\mu_i \cdots \mu_1} \pi_0 \\ &= \left(\frac{\lambda}{\mu}\right)^i \pi_0 \\ &= \left(\frac{\lambda}{\mu}\right)^i \left(1 - \frac{\lambda}{\mu}\right), \quad \forall i \geq 1 \\ \text{Therefore } \pi_i &\sim \text{geom}_0\left(1 - \frac{\lambda}{\mu}\right). \end{aligned} \tag{4.43}$$

Thus π_i has the distribution of a geometric random variable that counts the number of failures[10] before first success with probability $p = 1 - \frac{\lambda}{\mu}$.

[10] Here failures must be counted as number of people ahead of the currently arriving person who will be served before the currently arriving person is served.

4.6.3 *Little's law*

In a general queuing system, let us say the n^{th} arrival spends W_n units of time in the system (time spent in the queue + service time). W_n is known as the *sojourn time* of the n^{th} arrival. The average sojourn time is $E(W) \approx \lim_{n\to\infty} \frac{1}{n} \sum\limits_{i=1}^{n} W_i$.[11] The average rate of arrivals is $\lambda = \lim_{t\to\infty} \frac{N(t)}{t}$ where $N(t)$ denotes the number of arrivals by time t. Further, we define by $L(t)$, the total number of clients in the system (clients in queue + clients being served) at time t and we define by $\overline{L} = \lim_{t\to\infty} \frac{1}{t} \int_0^{\infty} L(\tau) d\tau$, the (temporal) average number of clients in the

[11] $E(W)$ refers to the average sojourn time among a large number of clients who entered (and exited) the system.

system.[12] Then the Little's law states that the following relation holds.

$$\overline{L} = \lambda E(W). \tag{4.44}$$

$\overline{L}$ is thus the average size of the system (in terms of the number of clients in the queue and the number of clients being served).

4.6.4 *Mean sojourn time $E(W)$ and mean queue size $E(\pi_Q)$ of an $M/M/1$ queue*

Let us once again consider the $M/M/1$ queue as above. We now know that the stationary distribution of such a queue is a geometric distribution with parameter $p = 1 - \frac{\lambda}{\mu}$, $\pi \equiv \pi_i \sim \text{geom}_0(1-\rho)$ where $\rho = \frac{\lambda}{\mu}$. $E(\pi) = \frac{1-p}{p} = \frac{\rho}{1-\rho}$ is the expected number of clients in the stationary $M/M/1$ queue. If W_n is the sojourn time of client n, then by applying Little's law 4.44 and invoking ergodicity, we get

$$E(\pi) = \lambda E(W) = \frac{\rho}{1-\rho}, \tag{4.45}$$

where W is the average time spent by a client in the system. $W = \mathcal{T}_Q + \mathcal{T}_S$ where $\mathcal{T}_Q$ is the random time spent by a client in the queue (not including time spent during service) and $\mathcal{T}_S$ is the random time spent by the client while being served. From equation 4.45, we obtain the mean sojourn time

$$\boxed{E(W) = \frac{1}{\lambda}\frac{\rho}{1-\rho} = \frac{1}{\mu(1-\rho)}}. \tag{4.46}$$

Taking the expectation of $W = \mathcal{T}_Q + \mathcal{T}_S$ and using $E(\mathcal{T}_S) = \frac{1}{\mu}$, we obtain the average time spent by a client in the queue (discounting the time of service) as follows.

$$\boxed{E(\mathcal{T}_Q) = E(W) - E(\mathcal{T}_S) = \cdots = \frac{\rho}{\mu(1-\rho)}}. \tag{4.47}$$

By applying the Little's law exclusively to the queue, we get the average number of people in the queue (queue size) in the long run as follows.

$$\boxed{E(\pi_Q) = \lambda E(\mathcal{T}_Q) = \cdots = \frac{\rho^2}{1-\rho}}. \tag{4.48}$$

Further, since $\pi = \pi_Q + \pi_S$, we obtain the average number of clients being served in the long run as follows.

$$\boxed{E(\pi_S) = E(\pi) - E(\pi_Q) = \cdots = \rho = \frac{\lambda}{\mu}}. \tag{4.49}$$

Similar results can be deduced for a generic case such as that of an $M/M/n$ queue or an $M/G/n$ queue that may not have an exponential service time. We will see an application of the former in the chapter project and similar examples in the exercise problems of this chapter.

[12] The temporal average is also well approximated by an ensemble average (expected value) because a ***stationary*** **CTMC is always** ***ergodic***. Conversely, if the ergodicity condition is satisfied by a CTMC, then it surely has a stationary solution.

Example: Delay in a communication channel

A communication channel has a capacity of 1000 bits per second. This channel is used to carry 8 bit characters. The channel can handle a total call volume of 6000 characters per minute. Estimate the average numbers of characters waiting to be transmitted and the mean sojourn time of transmission.

Solution: The arrival rate $\lambda = \frac{6000}{60} = 100$ characters/sec. The service rate $\mu = \frac{1000}{8} = 125$ characters/sec. The utilization factor is $\rho = \frac{\lambda}{\mu} = 0.8$. $E(\pi_Q) = \frac{\rho^2}{1-\rho} = 3.2$ is the average number of characters in the queue. The mean sojourn time is $E(W) = \frac{E(\pi)}{\lambda} = \frac{\frac{\rho}{1-\rho}}{100} = \frac{\frac{0.8}{0.2}}{100} = \frac{4}{100} = 0.04$ seconds = 40 milliseconds.

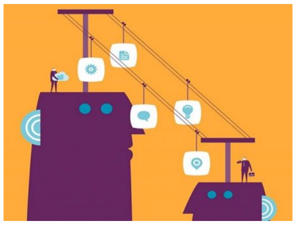


Figure 4.11: Modern technology and business operations rely heavily on efficient communication channels.

4.7 Chapter project: Queues and crowd management at COVID test centers

4.7.1 Interlude: The mathematical technology of analyzing and managing queues

As people start showing up at the test kiosk according to a Poisson process, the counters at the kiosks start testing the patients and the queue starts growing. We will use the $M/M/n$ queuing model to answer the following questions. The arrival rate λ must be estimated from the data provided in Table 4.1. The service time at the test-counters can be assumed to follow an exponential distribution with rate $\mu = 12$ per hour.

1. **Give a condition in terms of the system utilization factor ρ so that the queue is stable (i.e. the queue does not grow in an unbounded manner).**
2. **Based on the data provided in the table, estimate the arrival rate λ.[13]**
3. **What is the minimum number of test counters required to satisfy the stability condition of question 1?**
4. **Deduce the expression for the steady-state probability distribution π_k for the number of patients k in the system upon a random arrival. Consider both the cases when $k \leq n$ and $k > n$.**
5. **What is the probability π_0 that all test counters are idle upon a random arrival? In other words, what is the probability that no patient is currently being tested?**
6. **Compute the probability that all test counters are busy upon a random arrival.**
7. **What is the average number of patients in the system at any given instant? Use this result and the Little's theorem to find the mean sojourn time of a patient.**

4.7.2 $M/G/n$ queues and the Pollaczek-Khinchin formula

Once again we will restrict our analysis to $n = 1$ in this section. More importantly, we will relax the memoryless constraint of service times and consider a general probability distribution for the service times with a mean service time of $\frac{1}{\mu}, \ \mu > 0$. The arrivals continue

[13] In this section, consider a *serpentine* queue, i.e. a single queue. In the final section of the chapter project, we will ask the readers to compare this type of queue with a multiple queue version of the system.

to follow a Poisson distribution with rate $\lambda > 0$. The utilization factor $\rho = \frac{\lambda}{\mu} < 1$ ensures that we have a stable queue with a stationary distribution of states (number of clients in the system). We will not derive the formula for the stationary distribution but simply compute the mean sojourn time and the average size of the queue at steady state.

At the outset, let us define the relevant random variables.

1. $\mathcal{T}_{S_i}$: service time experienced by client i,
2. N_q: number of clients in the queue when client $i = (N_q + 1)$ arrives,
3. $\mathcal{T}_q$: the time spent by the i^{th} client in the queue,
4. R_i: residual service time of the client in service at the instant when client i arrives,[14]
5. W_i: sojourn time of a client i in the system,
6. B_i: event when the i^{th} client arrives and finds the server busy and N_q clients ahead of it in the queue, and
7. I_i: event when the i^{th} client arrives and finds the server available available for service (idle).

Figure 4.12: Austrian-French mathematician Felix Pollaczek known for his work on queuing models and probability theory (*courtesy: Wikimedia commons*).

The probability that the i^{th} client finds the server busy upon arrival is $\rho = \lambda E(\mathcal{T}_S) = \frac{\lambda}{\mu} < 1$ for a stable queue as stated earlier. Since we are primarily interested in analyzing the stationary state of the system, we will drop the subscript i from the variables. The sojourn time at steady state may be expressed as follows.

$$\begin{aligned} W &= \mathcal{T}_q + \mathcal{T}_S, & (4.50)\\ E(W) &= E(\mathcal{T}_q) + \frac{1}{\mu}. & (4.51) \end{aligned}$$

Note that we have not included R in W because it is already absorbed in the term $\mathcal{T}_q$. We will use the law of total expectation to compute

$$E(\mathcal{T}_q) = \rho E(\mathcal{T}_q | B) + (1-\rho) E(\mathcal{T}_q | I) = \rho E(\mathcal{T}_q | B),$$

which results from the fact that $\mathcal{T}_q \equiv 0$ when the event I happens. In order to compute the average time spent in the stationary queue by a client, we use the following result.

$$\begin{aligned} E(\mathcal{T}_q | B) &= E(R + \mathcal{T}_{S_1} + \mathcal{T}_{S_2} + \cdots + \mathcal{T}_{S_{N_q}} | B)\\ &= E(R|B) + E(\mathcal{T}_{S_1} + \mathcal{T}_{S_2} + \cdots \mathcal{T}_{S_{N_q}} | B)\\ &= E(R|B) + E(\mathcal{T}_S|B) E(N_q|B) \end{aligned}$$

cf. exercise on random sums of ch. 1

$$= E(R|B) + \frac{\lambda}{\mu\rho} E(\mathcal{T}_q). \qquad (4.52)$$

Here $E(N_q|B)$ is evaluated by the law of total expectation $E(N_q) = E(N_q|B)P(B) + E(N_q|I)P(I)$ whence the second term on the r.h.s. is zero and $E(N_q|B) = \mu E(\mathcal{T}_q)$ using Little's law $E(N_q) = \lambda E(\mathcal{T}_q)$. Here we have used $\rho = P(B)$. Thus we have

$$E(\mathcal{T}_q) = \frac{\rho E(R|B)}{1-\rho}. \qquad (4.53)$$

[14] $R_i \equiv 0$ when client i enters an empty system.

We will now invoke a useful theorem without providing a proof and use it to find the average time spent by a client in the stationary queue. Interested readers may refer to this article[15]. The mean residual service time $E(R|B)$ is given by $\frac{1}{2}\frac{E(\mathcal{T}_S^2)}{E(\mathcal{T}_S)} = \frac{\mu}{2}E(\mathcal{T}_S^2)$. Using this result, we find

$$\boxed{E(\mathcal{T}_q) = \frac{\lambda E\left(\mathcal{T}_S^2\right)}{2(1-\rho)}}. \tag{4.54}$$

The mean sojourn time is given by

$$\boxed{E(W) = \frac{\lambda E\left(\mathcal{T}_S^2\right)}{2(1-\rho)} + \frac{1}{\mu}}. \tag{4.55}$$

Equations 4.54 and 4.55 are known as the ***Pollaczek-Khinchin formula***. The average number of the clients in the system $\overline{N}$ can be obtained by applying Little's law.

$$\boxed{\overline{N} = \lambda E(W) = \rho + \frac{\lambda^2 E\left(\mathcal{T}_S^2\right)}{2(1-\rho)}}. \tag{4.56}$$

Example: An $M/G/1$ queue in action

Consider a single server $M/G/1$ queue at steady state. Arrivals happen according to a Poisson distribution with rate $\lambda = 6$ per hour. The service time is uniformly distributed between 7 and 9 minutes. Answer the following questions.

1. What is the average number of clients in the queue?
2. What is the average waiting time in the queue?
3. What is the average number of clients in the system?
4. What proportion of time is the server idle?

Solution: We will fix the time units to minutes for consistency. Therefore, $\lambda = \frac{1}{10}$ per minute. $E(\mathcal{T}_S) = \frac{7+9}{2}$. Therefore the service rate is $\mu = \frac{1}{8}$ per minute. $Var(\mathcal{T}_S) = \sigma_S^2 = \frac{(9-7)^2}{12} = \frac{1}{3}$.

1. Average number of clients in the queue $\overline{N}_q = \lambda E(\mathcal{T}_q) = \frac{\lambda^2 E(\mathcal{T}_S^2)}{2(1-\rho)} = \frac{\lambda^2\sigma_S^2+\lambda^2\frac{1}{\mu^2}}{2(1-\rho)} = \frac{\lambda^2\sigma_S^2+\rho^2}{2(1-\rho)} \approx 1.61$.
2. Average waiting time in the queue $E(\mathcal{T}_q) = \frac{\overline{N_q}}{\lambda} \approx 16.1$ minutes.
3. Average sojourn time in the system is given by the *Pollaczek-Khinchin* equation 4.55. $E(W) \approx 24.1$ minutes. Now using Little's law $\overline{N} = \lambda E(W) = 2.41$ are the average number of clients in the system.
4. Proportion of the time the server stays idle is equal to $1-\rho = 1-\frac{\lambda}{\mu} = 1-0.8 = 0.2$.

[15] W. C. Chan, T. C. Lu, and R. J. Chen. "Pollaczek-Khinchin formula for the M/G/I queue in discrete time with vacations". In: *IEE Proc.-Comput. Digit Tech.* 144.4 (1997), pp. 222–226

There are many different queuing models commensurate with a variety of scenarios. Interested readers may want to check out the textbooks mentioned in the chapter bibliography for a detailed treatment on queuing models.

4.8 Chapter project: Queues and crowd management at COVID test centers

4.8.1 Epilogue: Simulating the queue using Matlab's Simulink

We will simulate the behavior of the queue at the test kiosks using the Simulink feature of Matlab. Using these simulations we will be able to compare the behavior of two different types of queues, viz., the *serpentine queue* and the *multi-line queue*.[16] The Simulink environment is available within the SimEvents module of Matlab. In order to access these built-in Simulink queuing models please ensure that the Simulink toolbox is installed in your computer.

Description of the Simulink queuing model:

The Simulink model is available with two parallel versions of a simple model of n service counters: (i) one that uses n separate queues, and (ii) one with a single *serpentine queue* that provides service to all patients. Patients arrive at random with exponentially distributed inter-arrival times and they are simulated using entities in SimEvents. The exponential service times are also simulated using the relevant SimEvents sub-module. The average arrival time is set to a default value $\frac{1}{\lambda} = 2$ hours and the average service time is set to a default value of $\frac{1}{\mu} = 1$ hour. For the multi-line queue, each patient is cloned after a generation so that the different line configurations can be exercised identically. The Matlab-generated schemata is shown in Figure 4.13

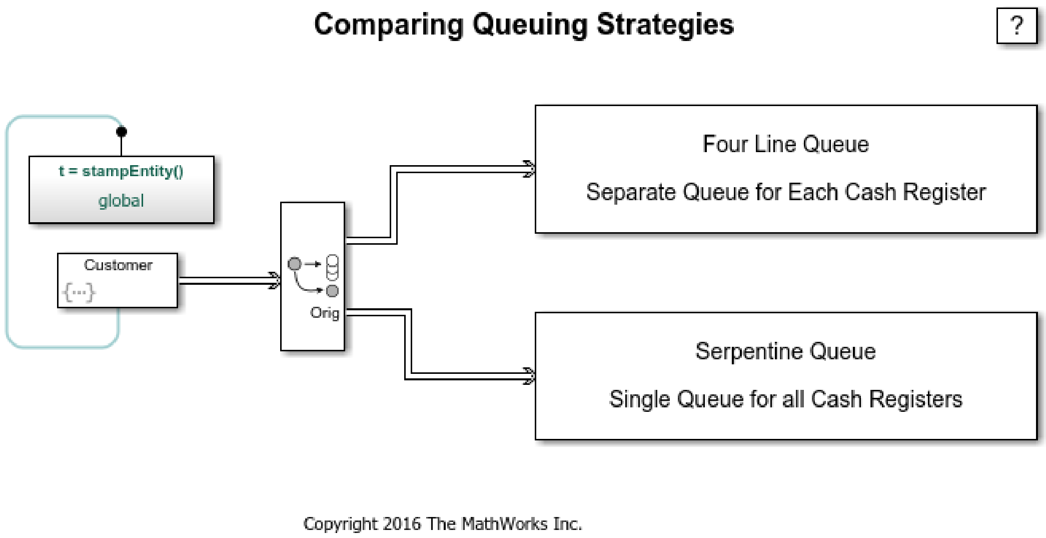

Comparing Queuing Strategies
?
t = stampEntity()
global
Customer
{...}
Orig
Four Line Queue
Separate Queue for Each Cash Register
Serpentine Queue
Single Queue for all Cash Registers
Copyright 2016 The MathWorks Inc.

Figure 4.13: Queuing model of the SimEvent module of Simulink

[16] Eg., the multi-line queue of a 3 server system will have three different queues leading to each of the servers. A randomly arriving client will choose the shortest available queue.

Multi-line queues

In this model, multi-line queues feed the n service counters. Arriving clients (patients) are routed to the shortest queue. Each queue then feeds the service representing a check-out together. This server holds the patient for the amount of time that was set up during generation. Figure (4.14) represents the sub-model of a multi-line queue that is embedded into the main simulation model.

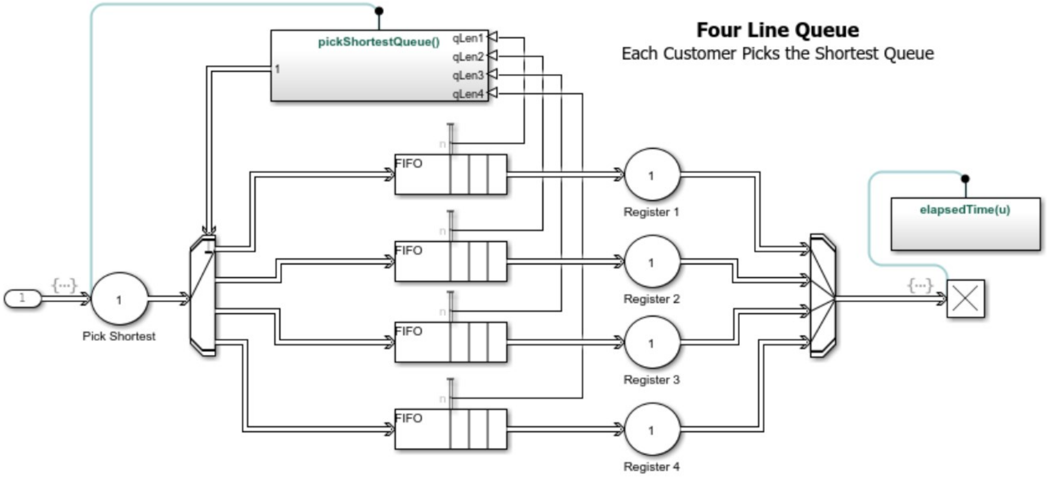


Figure 4.14: Multi-line queuing model. The example shown here corresponds to $n = 4$.

Serpentine Queue

In this model, a single queue is involved that feeds all n servers via a switch that routes customers to a free service counter when one becomes available. Figure (4.15) represents this model.

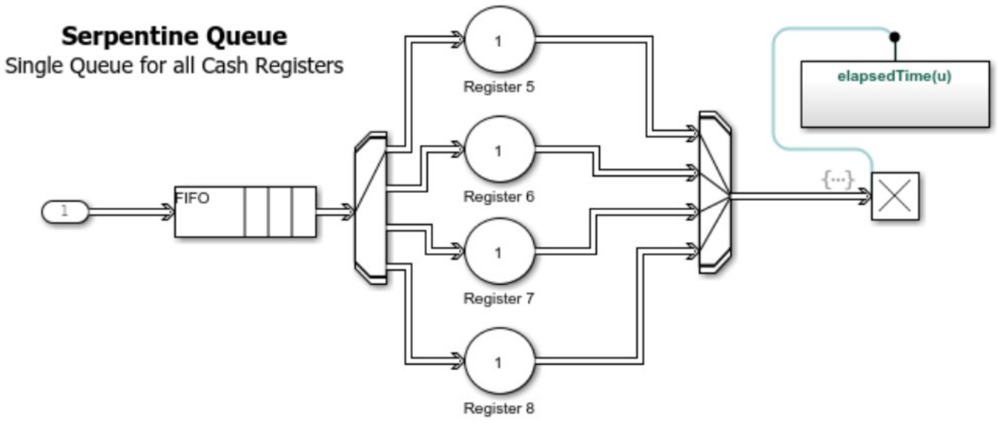


Figure 4.15: Serpentine queuing model

Output of Simulink

The output of this simulation is presented in the form of plots as shown in figure (4.16) which displays the average sojourn time of a patient for both cases, i.e. the multi-line queuing model, and the serpentine queuing model.

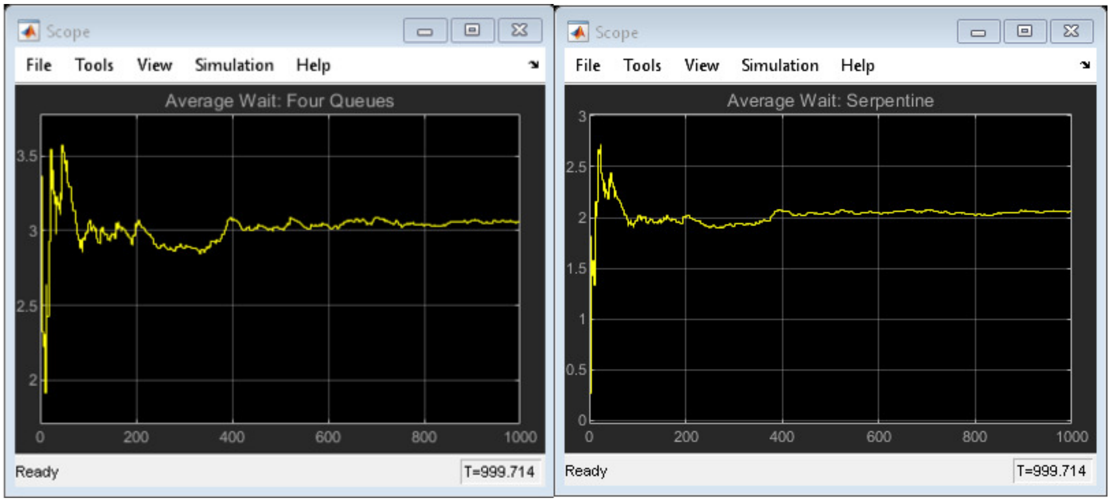


Figure 4.16: Average sojourn time for the multi-line queue (left), and the serpentine queue (right).

Steps to simulate the queue using Simulink

We now provide the detailed steps to run the simulations using Simulink.

Step 1: Execute the command `openExample('simevents/QueuingStrategiesExample')` in the MATLAB "Command window". This should prompt a new window which contains the model as shown in Figure (4.17):

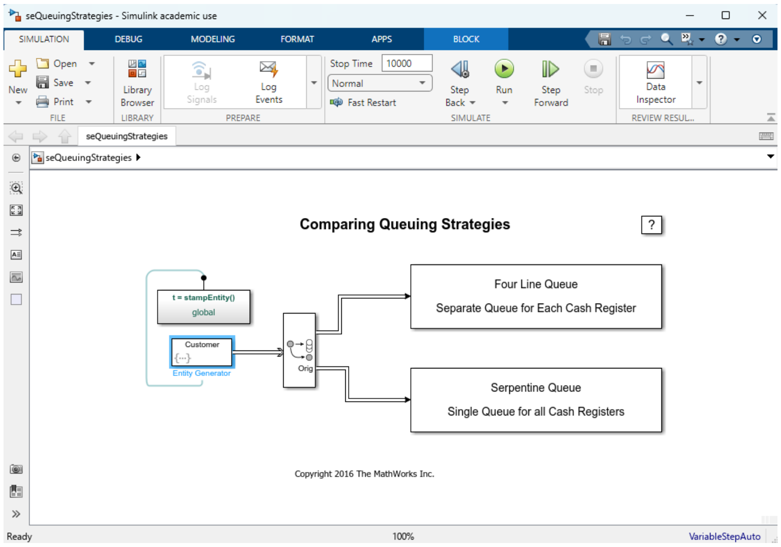
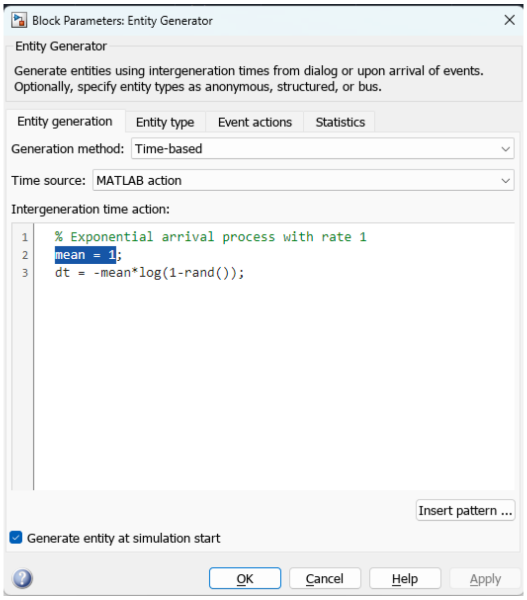


Figure 4.17: Simulink model

Step 2: Now in order to change the parameters (arrival rate and service rate) as required, double-click on the block model "Customer (Entity Generator)" shown in figure 4.17. A dialogue box named "Block Parameter: Entity Generator" (cf. Figure 4.18) will appear.



Block Parameters: Entity Generator

Entity Generator

Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Generation method: Time-based

Time source: MATLAB action

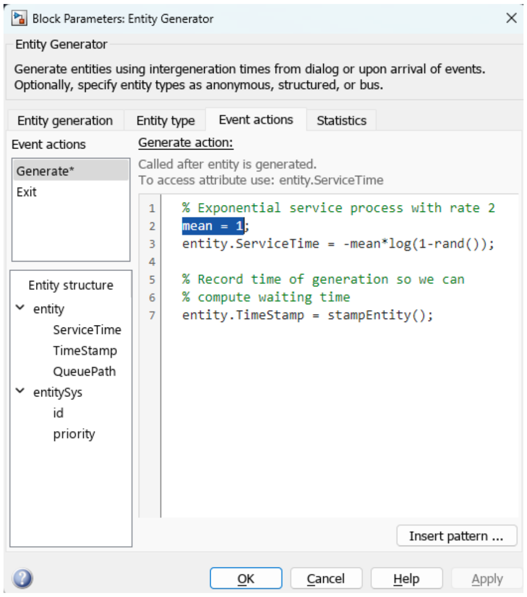
Intergeneration time action:

```
% Exponential arrival process with rate 1
mean = 1;
dt = -mean*log(1-rand());
```

☑ Generate entity at simulation start

Figure 4.18: "Entity generation" section

Step 3: Change the value of the mean in "Entity generation" (see figure 4.18) and "Event actions" (see Figure 4.19) as required. Finally, click on "Apply" followed by "OK".



Block Parameters: Entity Generator

Entity Generator

Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Generate action:

Called after entity is generated.
To access attribute use: entity.ServiceTime

```
% Exponential service process with rate 2
mean = 1;
entity.ServiceTime = -mean*log(1-rand());

% Record time of generation so we can
% compute waiting time
entity.TimeStamp = stampEntity();
```

Figure 4.19: "Event actions" section

Step 4: Now "RUN" the Simulink model (as seen in Figure 4.17). This will produce the results in the form of plots (see Figure 4.20)

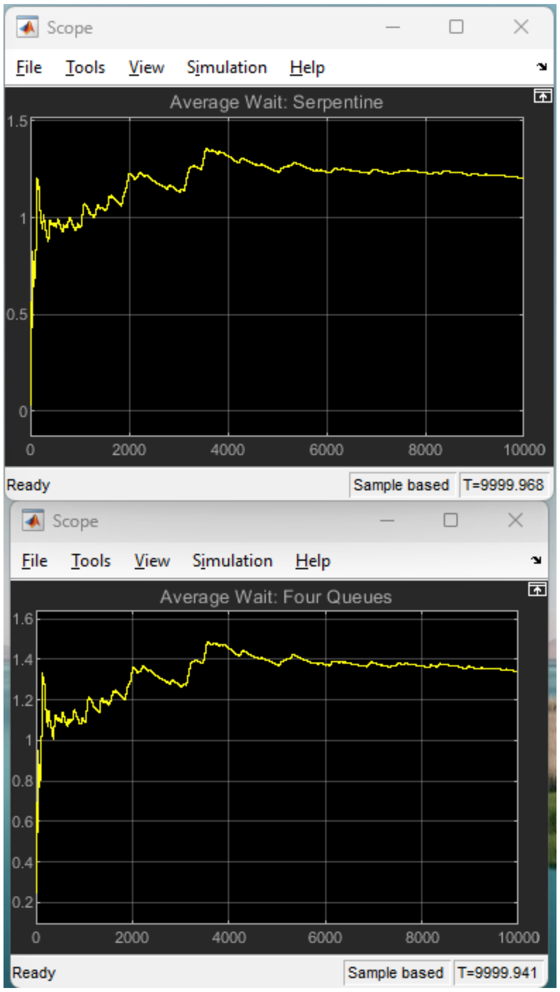


Figure 4.20: "Scope window" displaying the results for sojourn time.

Consider the answer for the minimum number of servers in question 3 to be $\hat{n}$. Now use the Simulink model to simulate the $M/M/\hat{n}$ with the estimated values of λ and μ. Run the simulation for the $M/M/\hat{n}$ model for both cases: multi-line and serpentine) in order to compute the average waiting time in the system. Discard the statistics corresponding to initial transients and then collect the steady-state values. The steady state of the system can be identified by inspection. The documentation of the Simulink model can be accessed through `https://in.mathworks.com/help/simevents/ug/comparing-queuing-strategies.html`. FOllow the steps suggested above to perform the simulation and answer the following questions.

1. **Simulate the model using the estimated value of λ in question 2 and the given value of μ.**

2. **Verify the simulated value of the average sojourn time of the patient in the system with the theoretically calculated value in question 7.**

3. **What do you observe from the simulation for the multi-line queue and the serpentine queue? Which queue type do you think is more preferable for both patients and health administrators? Explain your answer.**

4.9 Selected bibliography

1. *An Introduction to Probability Theory and Its Applications: Vol. 2* by *William Feller*, John Wiley & Sons, Inc. (second edition), 1971.

2. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications* by *Kishor Shridharbhai Trivedi*, Wiley (second edition), 2001.

3. *Stochastic Models: Analysis and Applications* by *B. R. Bhat*, New Age international Publishers (second edition), 2019.

4. *Markov Chains: Models, Algorithms, and Applications* by *Wai-Ki Ching, Michael K. Ng*, Springer (first edition), 2006.

5. *Essentials of Stochastic Processes* by *Richard Durrett*, Springer (third edition), 2016.

6. *An Introduction to Queuing Systems* by *Sanjay K. Bose*, Springer Science + Business Media (first edition), 2002.

4.10 Exercise problems

1. (***Traffic in a call center booth***) Assume that on an average a call to a customer care number lasts for three minutes. If the customer support number is busy, callers are placed on hold and pushed to a queue. The hold can last for a maximum of three minutes after which the caller is automatically dropped from the queue and is advised to call again later. What is the maximum call rate that can be supported by one customer care number?

2. ($M/M/k$ ***queues***) Consider a queuing system with Poisson arrival of jobs at rate λ and exponential service rate μ at each of $k \geq 1$ counters. Deduce an expression for the average number of jobs in the system? How many servers are busy on an average? Further, derive an expression for the probability that a randomly arriving job enters a queue (and does not directly go to service).

3. ($M/M/1$ ***queue - chances of finding a fixed queue size***) Consider an $M/M/1$ queue in steady state, what are the chances that a new arrival will find n clients in the system?

4. (***Parking lots***) Consider a parking lot with N vehicle slots. Incoming traffic follow a Poisson process wit rate λ. When the lot is full, traffic is not allowed in the lot and is instead diverted to a different lot. Each vehicle that parks in the lot stays there for a random time that is distributed according to an exponential distribution with rate μ. Construct a model for computing the probability of finding exactly $n \leq N$ spaces occupied.

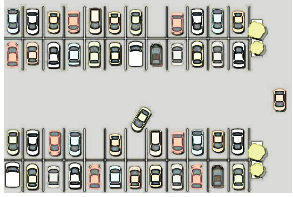


Figure 4.21: Distribution of cars in a parking lot.

5. ($M/M/1$ ***queues - optimizing service rate to maximize profit***) A single server queuing system provides service at rate μ where the service time is exponentially distributed. The cost of providing service is estimated as 10μ per hour. Additionally, a gross profit of 10 INR is made for every client served by the system. If the system has a capacity $N = 10$ (i.e. no more than 10 clients can be allowed in the system at any given point of time), estimate the rate μ that maximizes the total profit.

6. (***Lazy service***) Consider an $M/M/1$ queue which operates like a classical one except when the queue is empty. In this exceptional case, normal service is resumed only when at least 10 customers are again present in the queue. Answer the following questions.

(a) What is the probability that the system is empty?

(b) What is the probability distribution for the number of customers in the system?

(c) What is the mean number of customers in the system?

(d) What is the average time to service resumption from the instant the queue is empty?

7. (***Impatient client***) A bank has a single server. Clients arrive at the bank gate according to a rate $\lambda = 2$ Poisson process. The client enters the bank only if the server is free else he walks away in search of a new bank. Further, the service time of the server is distributed according to a distribution G with mean $\mu_G = 3$, then compute the following.

(a) What is the rate at which clients enter the bank?

(b) What fraction of arriving clients actually enter the bank?

8. (***Impatient client with a fickle mind***) Answer question 7 with the caveat that the arriving client enters the bank either when the server is free or with a probability $p = \frac{1}{2}$ when the server is busy.

9. (***Renewal process***) $\{N(t)\}_{t\geq 0}$ is a *counting process* marking the arrival of $N(t)$ point-events up to time t and let $\mathcal{T}_1, \mathcal{T}_2, ...$ denote the inter-arrival times between successive arrivals and are i.i.d. random variables. Then $N(t)$ is called a *renewal process*. Thus, a renewal process is a counting process such that at the instant of an event occurrence a renewal is said to have happened. This is so because the inter-arrival times have the same distribution irrespective of the epoch of the previous event. The renewal epochs are denoted by R_i where $R_0 = 0$ and $R_n = \sum_{i=1}^{n} \mathcal{T}_i, \;\; n \geq 1$. Further, let $\mu = E(X_n), \; n \geq 1$. Prove that with probability equal to one, we have the following result.

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{\mu}. \tag{4.57}$$

10. (***Changing diapers***) Alex must take care of his 4 month old baby and part of that job entails changing the baby's diapers as soon as it is soiled. If a diaper lasts for a uniformly distributed time between 1 hour and 2 hours before it is soiled, then at what rate does Alex have to change his baby's diapers in the long run? Further, if Alex does not have a back-up supply of diapers handy and must have to fetch a new one from the storage which takes him any where between 15 and 30 minutes (again uniformly distributed), then what is the average rate at which Alex changes the baby's diapers?

Figure 4.22: How often does Alex have to change his baby's diapers?

□

WEEK\DAYS	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	**Sunday**	**Total**
WEEK-01	-	-	-	-	415	1107	1065	2587
WEEK-02	1894	1759	1605	1640	1455	1009	1007	10369
WEEK-03	1755	1615	1534	1557	1402	1073	904	9840
WEEK-04	1462	1475	1352	1405	1324	1012	908	8938
WEEK-05	1763	1420	1504	1432	1099	875	883	8976
WEEK-06	344	873	1505	1388	1245	900	567	6822
WEEK-07	1345	1090	1061	931	934	671	585	6617
WEEK-08	860	1110	1006	591	855	681	792	5895
WEEK-09	1117	1073	1023	1027	943	625	734	6542
WEEK-10	1174	936	956	934	888	594	595	6077
WEEK-11	1107	1086	958	964	937	627	644	6323
WEEK-12	1063	986	941	857	902	660	710	6119
WEEK-13	1215	1027	1048	1147	999	632	574	6642
WEEK-14	1043	938	959	861	847	638	497	5783
WEEK-15	1039	933	868	813	803	564	462	5482
WEEK-16	853	775	722	640	595	411	357	4353
WEEK-17	641	518	513	489	438	308	270	3177
WEEK-18	498	391	418	392	322	190	220	2431
WEEK-19	316	297	243	268	225	121	106	1576
WEEK-20	207	216	196	147	145	94	99	1104
WEEK-21	151	138	129	134	115	81	63	811
WEEK-22	105	102	81	62	67	32	34	483
WEEK-23	35	59	70	62	66	44	35	371
WEEK-24	64	55	58	46	44	28	35	330
WEEK-25	54	37	46	52	59	29	33	310
WEEK-26	40	50	48	51	61	37	28	315
WEEK-27	49	53	75	63	60	41	32	373
WEEK-28	62	125	147	110	111	63	79	697
WEEK-29	176	152	185	171	183	109	141	1117
WEEK-30	276	233	309	326	319	188	214	1865
WEEK-31	423	406	384	444	427	263	294	2641
WEEK-32	518	489	522	517	520	373	321	3260
WEEK-33	626	536	546	578	499	348	422	3555
WEEK-34	628	595	555	543	505	338	276	3440
WEEK-35	642	613	570	525	464	296	343	3453
WEEK-36	583	531	487	478	429	255	334	3097
WEEK-37	274	434	466	630	504	335	504	3147
WEEK-38	633	609	550	451	559	357	464	3623
WEEK-39	545	397	509	513	452	309	333	3058
WEEK-40	518	355	369	455	408	280	458	2843
WEEK-41	512	507	487	406	451	250	308	2921
WEEK-42	363	464	409	426	322	194	260	2438
WEEK-43	364	314	294	341	261	162	198	1934
WEEK-44	363	238	288	311	230	158	199	1787
WEEK-45	314	302	300	349	290	198	205	1958
WEEK-46	401	358	363	347	372	241	258	2340
WEEK-47	428	430	432	428	353	279	293	2643
WEEK-48	517	482	460	189	364	302	318	2632
WEEK-49	661	577	700	629	579	382	463	3991
WEEK-50	849	895	982	912	983	619	870	6110
WEEK-51	2390	3421	4123	4514	4275	3243	3437	25403
WEEK-52	8933	9446	9452	9606	5572	2344	7215	52568
WEEK-53	15029	14203	14298	13483	8744	-	-	65757
Grand Total	57222	56124	57106	55635	45421	24970	30446	**326924**

Table 4.1: Number of infected persons who presented themselves at the COVID test kiosks