Chapter 2
Markov Processes

2.1 Introduction

Markov processes are a special class of stochastic processes. In order to fully understand Markov processes we first need to introduce stochastic processes.

2.2 Stochastic Processes

Consider a sample space \( \mathcal{S} \) and a random variable \( X(\omega) \) where \( \omega \in \mathcal{S} \). A stochastic process may be defined as a collection of points \( X(t, \omega) \), \( t \geq 0 \), indexed over the parameter \( t \), usually representing time but may represent others such as space, etc. Put in a simpler form we can say a stochastic process is a collection of random variables \( X(t) \), \( t \geq 0 \), which is observed at time \( t \). Examples of a stochastic process could be the amount of rainfall \( X(t) \) on day \( t \), the remaining gasoline in the tank of an automobile at any time, etc. Most quantities studied in a queueing system involve a stochastic process. We will introduce the concept in the context of discrete time queues in the next section.

In this section, we consider integer valued stochastic processes in discrete times. Throughout this book we only consider non-negative integer valued processes. Consider a state space \( \mathcal{I} = \{0,1,2,3,\cdots\} \) and a state space \( \mathcal{I}_s \) which is a subset of \( \mathcal{I} \), i.e. \( \mathcal{I}_s \subseteq \mathcal{I} \). Now consider a random variable \( X_n \) which is observable at time \( n = 0,1,2,\cdots \) in discrete time, where \( X_n \in \mathcal{I}_s \).

The collection of these \( X_0, X_1, X_2, \cdots \) form a stochastic process in discrete time; or written as \( \{X_n, n = 0,1,2,\cdots\} \). A discrete parameter stochastic process is simply a family of random variables \( X_n, n = 0,1,2,\cdots \) which evolves with parameter \( n \), and in this case \( n \) could be time or any other quantity, such as an event count. This collection describes how the system evolves with time or event. Our interest is to study the relations of these collections. For example \( X_n \) could be the number of data packets that arrive at a router at time \( n, n = 0,1,2,\cdots \), or the number of packets left
behind in the router by the $n^{th}$ departing packet. Examples of stochastic processes usually of interest to us in queueing are the number of packets in a queue at a particular time or between short intervals of time, the waiting time of packets which arrive in the system at particular times or between intervals of time, the waiting time of the $n^{th}$ packet, etc.

Of interest to us about a stochastic process is usually its probabilistic aspects; the probability that $X_n$ assumes a particular value. Also of interest to us is the behaviour of a collection of a group of the random variables. For example the $Pr\{X_m = a_m, X_{m+1} = a_{m+1}, \ldots, X_{m+y} = a_{m+y}\}$. Finally we may be interested in the moments of the process at different times.

As an example, consider a communication processing system to which messages arrive at time $A_1, A_2, A_3, \ldots$, where the $n^{th}$ message arrives at time $A_n$. Let the $n^{th}$ message require $S_n$ units of time for processing. We assume that the processor can only serve one message at a time in a first come first served order. Let there be an infinite waiting space (buffer) for messages waiting for service. Let $X_t$ be the number of messages in the system at time $t$. $X_t$ is a stochastic process. Further let $D_n$ be the departure time of the $n^{th}$ message from the system after service completion.

In order to understand all the different ways in which this stochastic process can be observed in discrete time, we consider a particular realization of this system. For example, let us consider the process for up to 10 message arrivals for a particular realization as shown in Table (2.1) We can now study this process at any point in time.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$A_n$</th>
<th>$S_n$</th>
<th>$D_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>5</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>19</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>3</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 2.1 Example of a sample path

In discrete time, we observe the system at times $t_1, t_2, t_3, \ldots$, where for simplicity, without loss of generality, we set $t_1 = 0$ to be the time of the first arrival, we have several choices of the values assumed by $t_i$, $i > 1$ in that we can study the system at:

- Equally spaced time epochs such that $t_2 - t_1 = t_3 - t_2 = t_4 - t_3 = \ldots = \tau$. For example, if $\tau = 1$, then $(t_1, t_2, t_3, \ldots) = (0, 1, 2, \ldots)$. In which case, $X_{t_1} = X_0 = 1$, $X_{t_2} = X_1 = 1$, $X_{t_3} = X_2 = 2$, $X_{t_4} = X_3 = 1$, $X_{t_5} = X_4 = 1$, $X_{t_6} = X_5 = 2$, $X_{t_7} = X_6 = 3$, $\ldots$
• Arbitrarily spaced time epochs such that \( t_2 - t_1 = \tau_1, t_3 - t_2 = \tau_2, t_4 - t_3 = \tau_3, \ldots \) with \( \tau_1, \tau_2, \tau_3, \ldots \) not necessarily being equal. For example, if \( \tau_1 = 1, \tau_2 = 2, \tau_3 = 1, \tau_4 = 4, \ldots \) hence \((t_1, t_2, t_3, t_4, t_5, \ldots) = (0, 1, 3, 4, 8, \ldots)\) we have \( X_{t_1} = X_0 = 1, X_{t_2} = X_1 = 1, X_{t_3} = X_3 = 1, X_{t_4} = X_4 = 1, X_{t_5} = X_8 = 2, \ldots \)

• Points of event occurrences. For example if we observe the system at departure times of messages, i.e. \((t_1, t_2, t_3, \ldots) = (D_1, D_2, D_3, \ldots) = (3, 7, 8, \ldots)\). In this case we have \( X_{t_1} = X_3 = 1, X_{t_2} = X_7 = 3, X_{t_3} = X_8 = 2, \ldots \)

Any events that occur between the time epochs of our observations are not of interest to us in discrete time analysis. We are only taking “snap shots” of the system at the specific time epochs.

By this assumption, then any discrete time observation is contained in continuous time observations.

Note that we may also have multivariate stochastic processes. For example, let \( X_n \) be the number of packets in the system at time \( n \) and let \( Y_n \leq X_n \) be how many of the \( X_n \) packets require special service. The stochastic process for this system is \( \{X_n, Y_n, \ n = 0, 1, 2, 3, \ldots\} \), \( X_n \in \mathcal{S} \), \( Y_n \leq X_n \). We will discuss multivariate processes later when we consider Markov chains.

### 2.3 Markov Processes

In 1907, A. A. Markov published a paper [75] in which he defined and investigated the properties of what are now called as Markov processes. A Markov process is a special stochastic process in which the state of the system in the future is independent of the past history of the system but dependent on only the present. The consideration of Markov processes is very important and in fact, is central to the study of queuing systems because one usually attempts to reduce the study of nearly all queueing systems to the study of Markov process.

Consider a stochastic process \( X_n \) in discrete time \( (n = 0, 1, 2, \ldots) \). The set of possible values that \( X_n \) takes on is referred as its state space. \( X_n \) is a Markov process if

\[
Pr\{X_{n+1} \in A_{n+1} | X_n = x_n, X_{n-1} \in A_{n-1}, \ldots, X_0 \in A_0\} = Pr\{X_{n+1} \in A_{n+1} | X_n = x_n\}
\]

holds for each state integral \( n \), each state \( x_n \) and the set of states \( A_{n+1}, A_{n-1}, A_0 \).

If the numbers in the state space of a Markov process are finite or countable, the Markov process is called a Markov chain. In this case, the state space is usually assumed to be the set of integers \( \{0, 1, \ldots\} \). In the case that the numbers in the state space of a Markov process are over a finite or infinite continuous interval (or set of such intervals), the Markov process is referred to as a Markov process. For most times in queueing applications, Markov chains are more common even though continuous space Markov processes, in the form of stochastic fluid models, are getting
popular for applications to queueing systems. We focus mainly on Markov chains in this chapter.

### 2.4 Discrete Time Markov Chains

Consider a discrete time stochastic process \( X_0, X_1, \ldots \), with discrete (i.e. finite or countable) state space \( \mathcal{S} = \{i_0, i_1, i_2, \ldots \} \subseteq \mathcal{I} \). If

\[
\Pr\{X_{n+1} = i_{n+1} | X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0\} = \Pr\{X_{n+1} = i_{n+1} | X_n = i_n\}
\]

holds for any integral \( n \), and the states \( i_{n+1}, i_n, i_{n-1}, \ldots, i_0 \), then \( X_n \) is said to be a discrete time Markov chain (DTMC). If further we have

\[
\Pr\{X_{n+m+1} = j | X_{n+m} = i\} = \Pr\{X_{n+1} = j | X_n = i\}, \forall (i, j) \in \mathcal{I}, \forall (m, n) \geq 0,
\]

we say the Markov chain is time-homogeneous or stationary. Throughout most of this book we shall be dealing with this class of DTMC.

Generally we say

\[
p_{i,j}(n) = P\{X_{n+1} = j | X_n = i\}
\]

is the transition probability from state \( i \) at time \( n \) to state \( j \) at time \( n+1 \). However, for most of our discussions in this book (until later chapters) we focus on the case where the transition is independent of the time, i.e. \( p_{i,j}(n) = p_{i,j}, \forall n \). The matrix \( P = (p_{i,j}), (i, j) \in \mathcal{I} \) is called the transition probability matrix of the Markov chain, in which

\[
p_{i,j} = \Pr\{X_{n+1} = j | X_n = i\}.
\]

The basic properties of the elements of the transition probability matrix of a Markov chain are as follows:

**Properties:**

\[
0 \leq p_{i,j} \leq 1, \quad \forall i, j \in \mathcal{S}
\]

\[
\sum_{j \in \mathcal{S}} p_{ij} = 1
\]

For example consider a DTMC with the state space \( \{1, 2\} \). We show, in Fig (2.1) below, its transition behaviour between time \( n \) and \( n+1 \).
2.4.1 Examples

1. Example 2.1.1: Consider a discrete stochastic process $X_n$ which is the number of data packets in the buffer of a router at time $n$. At time $n$ one packet is removed from the buffer if it is not empty and $A_n$ packets are added at the same time. Let $\{A_n, n = 0, 1, 2, \ldots\}$ be a discrete stochastic process which assumes values in the set $\mathcal{S}$. This DTMC can be written as

$X_{n+1} = (X_n - 1)^+ + A_n, \ n = 0, 1, 2, 3, \ldots$,

where $(z)^+ = \max\{z, 0\}$. For example, let $A_n = A$, $\forall n$ and $a_k = Pr\{A = k\}, k = 0, 1, 2, 3, \ldots$, then

$p_{0,j} = a_j, j = 0, 1, 2, \ldots$,

$p_{i,j} = a_{j-i+1}, 1 \leq i \leq j = 1, 2, 3, \ldots$,

and

$p_{i,j} = 0, j < i$.

The transition matrix $P$ associated with this DTMC can be written as

$$P = \begin{bmatrix}
a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \\
a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \\
a_0 & a_1 & a_2 & a_3 & \cdots \\
a_0 & a_1 & a_2 & \cdots \\
\vdots & \vdots & \vdots & \cdots & \cdots
\end{bmatrix}.$$

2. Example 2.1.2: Use the same example as above but put a limit on how many packets can be in the buffer. If we do not allow the number to exceed $K < \infty$, then this DTMC can be written as

$X_{n+1} = \min\{(X_n - 1)^+ + A_n, K\} \ n = 0, 1, 2, 3, \ldots$. 
For this system we have
\[ p_{0,j} = a_j, \ j = 0, 1, 2, \ldots, K-1, \ p_{0,K} = \tilde{a}_K, \]
\[ p_{i,j} = a_{j-i+1}, 1 \leq i \leq j = 1, 2, 3, \ldots, K-1, \ p_{i,K} = \tilde{a}_{K-i+1}, \]
and
\[ p_{i,j} = 0, j < i, \]
where \( \tilde{a}_w = \sum_{v=w}^{\infty} a_v \).

3. Example 2.1.3: Consider a discrete stochastic process \( Y_n \) which is the number of packets in a buffer at time \( n \). Suppose at time \( n \), one packet is added to contents of the buffer and the minimum of \( Y_n + 1 \) and \( B_n \) items are removed from the buffer, where \( \{B_n, n = 0, 1, 2, \ldots\} \) is a discrete stochastic process which assumes values in the set \( \mathcal{I} \). This DTMC can be written as
\[ Y_{n+1} = (Y_n + 1 - B_n)^+, \ n = 0, 1, 2, 3, \ldots, \]
where \( (z)^+ = \max\{z, 0\} \). For example, let \( B_n = B, \ \forall n, \ b_k = Pr\{B = k\}, k = 0, 1, 2, 3, \ldots, \) and \( \tilde{b}_k = \sum_{v=k+1}^{\infty} b_v \), then
\[ p_{i,0} = \tilde{b}_i, \ i = 0, 1, 2, \ldots, \]
\[ p_{i,j} = b_{i+1-j}, 1 \leq j \leq i = 1, 2, 3, \ldots, \]
and
\[ p_{i,j} = 0, i < j. \]
The transition matrix \( P \) associated with this DTMC can be written as
\[
P = \begin{bmatrix}
\tilde{b}_0 & b_0 & \cdots \\
\tilde{b}_1 & b_1 & b_0 & \cdots \\
\tilde{b}_2 & b_2 & b_1 & b_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}.
\]

4. Example 2.1.4: Consider two types of items; type A and type B. There are \( N \) of each, i.e. a total of \( 2N \) items. Suppose these \( 2N \) items are randomly divided into two sets of equal numbers with \( N \) placed in one urn and the remaining \( N \) placed in the other urn. If there are \( X_n \) of type A items in the first urn at time \( n \) then we have \( N - X_n \) of type B in the first urn. If we now take one item each at random from each urn and exchange their urns, then it is clear that \( X_n, n = 0, 1, 2, 3, \ldots \) is a represented as
\[ X_{n+1} = \begin{cases}
X_n - 1, \text{ with probability } \frac{X_n^2}{N^2} \\
X_n, \text{ with probability } \frac{2X_n(N-X_n)}{N^2} \\
X_n + 1, \text{ with probability } \frac{(N-X_n)^2}{N^2}
\end{cases} \]
This is the well known Bernoulli-Laplace diffusion model. We can write

\[ p_{i,i-1} = \left( \frac{i}{N} \right)^2, \]

\[ p_{i,i} = 2 \frac{i}{N} \left( 1 - \frac{i}{N} \right), \]

\[ p_{i,i+1} = \left( 1 - \frac{i}{N} \right)^2. \]

For the example of when \( N = 4 \) we can write the transition matrix as

\[
P = \begin{bmatrix}
0 & 1 & 1 & 1 \\
\frac{1}{16} & \frac{3}{8} & \frac{9}{16} & 1 \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 1 \\
\frac{9}{16} & \frac{3}{8} & \frac{1}{16} & 0
\end{bmatrix}.
\]

5. Example 2.1.5: Consider a single server queueing system in which arrivals and service of packets are independent. Let \( A_n \) be the interarrival time between the \( n^{th} \) and the \( n-1^{st} \) packet, \( S_n \) the processing time of the \( n^{th} \) packet and \( W_n \) the waiting time of the \( n^{th} \) packet, then we can see that \( W_n, n = 1, 2, 3, \cdots \) is a Markov chain given as

\[ W_{n+1} = (W_n + S_n - A_{n+1})^+. \]

Here the parameter \( n \) is not time but a sequential label of a packet.

6. Example 2.1.6: The Gilbert-Elliot (G-E) channel model is a simple, but effective two-state DTMC. Consider a communication channel which is subject to noise. The G-E model assumes that at all times the channel is in two possible states: state 0 – channel is good, and state 1 – channel is bad. The state of the channel at the next time slot depends only on the state at the current time slot and not on the previous times. The probability of the channel state going from good to bad is \( q \) and probability of it going from bad to good is \( p \). With this system as a DTMC with state space \( \{0, 1\} \), its transition matrix is given as

\[
P = \begin{bmatrix}
1 - q & q \\
p & 1 - p
\end{bmatrix}.
\]

7. Example 2.1.7: The Fritchman channel model is an extension of the G-E model in that it assumes that the channel could be in more than 2 possible states. Let the number of states be \( M \) with \( 2 < M < \infty \), which are numbered \( \{0, 1, 2, \cdots, M-1\} \). We let state 0 correspond to the best channel state and state \( M-1 \) to the worst channel state. The Fritchman model restricts the channel changes from one time to the next time not to jump up or down by more than one state. The probability of going from state \( i \) to \( i+1 \) is given as \( u \) and going from \( i \) to \( i-1 \) is given as \( d \), while that of remaining in \( i \) is given as \( a \). It also assumes that the channel state at the next time depends only on the current state. The transition matrix for this
model is written as

\[
P = \begin{bmatrix}
1 - u & u & d & a & u \\
d & a & u & d & a & u \\
& ... & ... & ... & ... & ... \\
& d & a & u & d & 1 - d
\end{bmatrix}.
\]

### 2.4.2 State of DTMC at arbitrary times

One of our interests in DTMC is the state of the chain at an arbitrary time, i.e. \( Pr\{X_n = i\} \). This we can obtain by using the total probability rule as

\[
Pr\{X_{n+1} = i\} = \sum_{k \in \mathcal{S}} Pr\{X_{n+1} = i | X_n = k\} Pr\{X_n = k\}.
\]

Let \( x^{(n)} = [x_0^{(n)}, x_1^{(n)}, \ldots] \) be the state probability vector describing the state of the system at time \( n \), where \( x_i^{(n)} = \) Probability that the system is in the state \( i \) at time \( n \). Then we can write

\[
x_i^{(n+1)} = \sum_{k \in \mathcal{S}} x_k^{(n)} P_{k,i}
\]

and in matrix form we have

\[
x^{(n+1)} = x^{(n)} P = x^{(0)} P^n.
\]

Consider a DTMC with the state space \( \{1, 2\} \), then the transition probabilities and the state of the system at any arbitrary times are shown in the Fig (2.1).

Another way to view this process is to draw \( N \) nodes directed network as shown in Fig (2.2) below, for the case of \( N = 2 \), and then connect the node pairs \((i, j)\) that have \( p_{ij} > 0 \). If we now consider the case in which the DTMC goes from node \( i \) to node \( j \) with the probability \( p_{ij} \) at each time step, then \( x_i^{(n)} \) is the probability that we find the DTMC at node \( i \) at time \( n \).

We now introduce some examples as follows:

#### 2.4.2.1 Example

Example 2.2.1: Consider a processor which serves messages one at a time in order of arrival. Let there be a finite waiting buffer of size 4 for messages. Assume messages arrive according to the Bernoulli process with parameter \( a \) \((\bar{a} = 1 - a)\) and service is geometric with parameter \( b \) \((\bar{b} = 1 - b)\). A Bernoulli process, to be discussed later, is a stochastic process with only two possible outcomes, viz: success
or failure, with each trial being independent of the previous ones. The parameter \( a \) represents the probability of a success and \( \bar{a} = 1 - a \) represents the probability of a failure. A geometric distribution describes the number of trials between successive successes. These will be discussed in more details in Chapter 3. Let \( X_n \) be the number of messages in the system, including the one being served, at time \( n \). \( X_n \) is a DTMC, with transition matrix \( P \) given as

\[
P = \begin{bmatrix}
\bar{a} & \bar{a}b & a & \bar{a}b & a \\
\bar{a}b & \bar{a}b + ab & \bar{a}b & \bar{a}b + ab & \bar{a}b \\
\bar{a}b & \bar{a}b + ab & \bar{a}b & \bar{a}b & \bar{a}b \\
\bar{a} & \bar{a}b & \bar{a}b & \bar{a}b & \bar{a}b \\
\bar{a}b & \bar{a}b & \bar{a}b & \bar{a}b & \bar{a}b + a
\end{bmatrix}
\]

If \( a = 0.3 \) and \( b = 0.6 \), then we have

\[
P = \begin{bmatrix}
0.70 & 0.30 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.58
\end{bmatrix}
\]

If at time zero the system was empty, i.e. \( \mathbf{x}^{(0)} = [1, 0, 0, 0, 0] \), then \( \mathbf{x}^{(1)} = \mathbf{x}^{(0)}P = [0.70, 0.30, 0.12, 0.0, 0.0] \). Similarly, \( \mathbf{x}^{(2)} = \mathbf{x}^{(1)}P = \mathbf{x}^{(0)}P^2 = [0.616, 0.348, 0.036, 0.0, 0.0] \)
and \( x^{(5)} = x^{(4)} P = x^{(0)} P^5 = [0.5413, 0.3639, 0.0804, 0.0130, 0.0014] \). We also obtain \( x^{(100)} = x^{(99)} P = x^{(0)} P^{100} = [0.5017, 0.3583, 0.1024, 0.0293, 0.0084] \). We notice that \( x^{(101)} = x^{(100)} P = x^{(0)} P^{101} = [0.5017, 0.3583, 0.1024, 0.0293, 0.0084] = x^{(100)} \), in this case. We will discuss this observation later.

Keeping in mind that a Markov chain can also be of infinite state space, consider this same example but where the buffer space is unlimited. In that case, the state size is countably infinite and the transition matrix \( P \) is given as

\[
P = \begin{bmatrix}
\bar{a} & \bar{a} & \bar{a} & \bar{a} & \bar{a} \\
\bar{ab} & \bar{ab} + ab & \bar{ab} & \bar{ab} & \bar{ab} \\
\bar{ab} & \bar{ab} + ab & \bar{ab} & \bar{ab} & \bar{ab} \\
\bar{ab} & \bar{ab} + ab & \bar{ab} & \bar{ab} & \bar{ab} \\
\bar{ab} & \bar{ab} + ab & \bar{ab} & \bar{ab} & \bar{ab} \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\]

We shall discuss the case of infinite space Markov chains later.

### 2.4.2.2 Chapman Kolmogorov Equations

Let \( P^{(n)} \) be the matrix whose \((i,j)\)th entry refers to the probability that the system is in state \( j \) at the \( nth \) transition, given that it was in state \( i \) at the start (i.e. at time zero), then \( P^{(n)} = P^n \), i.e., the matrix \( P^{(n)} \) is the \( nth \) order transition matrix of the system. In fact, let \( p^{(n)}_{ij} \) be the \((i,j)\)th element of \( P^{(n)} \) then we have

\[
p^{(2)}_{ij} = \sum_{v \in \mathcal{I}} p_{iv} p_{vj},
\]

\[
p^{(3)}_{ij} = \sum_{v \in \mathcal{I}} p^{(2)}_{iv} p_{vj} = \sum_{v \in \mathcal{I}} p_{iv} p^{(2)}_{vj}
\]

\[
\vdots = \vdots
\]

\[
p^{(n)}_{ij} = \sum_{v \in \mathcal{I}} p^{(n-1)}_{iv} p_{vj} = \sum_{v \in \mathcal{I}} p_{iv} p^{(n-1)}_{vj}
\]

which when written out in a more general form becomes

\[
p^{(n)}_{ij} = \sum_{v \in \mathcal{I}} p^{(m)}_{iv} p^{(n-m)}_{vj},
\]

and in matrix form becomes

\[
P^{(n)} = P^{(n-m)} P^{(m)}, \quad \text{i.e.,} \quad P^{(n)}_{ij} = \sum_{v \in \mathcal{I}} p^{(m)}_{iv} p^{(n-m)}_{vj}
\]
Since $P^{(2)} = P^2$ from the first of above equation, by a simple induction, we will know $P^{(n)} = P^n$ holds for any non-negative integer $n$. This is the well known Chapman-Kolmogorov equation.

### 2.4.3 Classification of States

In this section, we explain how the states of a Markov chain are classified. Consider a 9 state DTMC shown in Fig(2.3) below. We let an arrow from $i$ to $j$ imply that $pij > 0$, and assume that $p_{ii} > 0$ for all the 9 states. We use this figure to help us in explaining the classes of states of a DTMC.

![Markov Chain Network Example](image)

**Fig. 2.3** Markov Chain Network Example

**Communicating States** - State $j$ is said to be accessible from state $i$ if there exists a finite $n$ such that $p_{i,j}^n > 0$. If two states $i$ and $j$ are accessible to each other, they are said to communicate. Consider the nine state DTMC with state space \{1, 2, 3, 4, 5, 6, 7, 8, 9\} and the following transition matrix
We assume that all the cells with entries have values that are greater than zero. For example, \( p_{1,2} > 0 \), whereas, \( p_{1,3} = 0 \). For example in this DTMC states 1 and 2 do communicate, whereas state 4 is not accessible from state 5 but state 5 is accessible from state 3. This transition matrix can also be shown as a network with each node representing a state and a directed arc joining two nodes signify a non-zero value, i.e. direct access (transition) between the two nodes (states) in one step. Fig(2.3) represents this transition matrix. In general we say state \( j \) is accessible from state \( i \) if a path exists in the network from node \( i \) to node \( j \), and the two states are said to communicate if there are paths from node \( i \) to node \( j \) and vice versa.

Absorbing States - a state \( i \) is said to be an absorbing state if once the system enters state \( i \) it never leaves it. Specifically, state \( i \) is an absorbing state if \( p_{ii} = 1 \). In the example above state 9 is an absorbing state. Also if we set \( p_{6,5} = 0 \), then state 6 becomes an absorbing state and as we can see \( p_{6,6} = 1 \) by virtue of it being the only non-zero element in that row.

Transient States - a state \( i \) is said to be a transient state if there exists a state \( j \) which can be reached in a finite number of steps from state \( i \) but state \( i \) cannot be reached in a finite number of steps from state \( j \). Specifically, state \( i \) is transient if there exists a state \( j \) such that \( p_{ij}^{(n)} > 0 \) for some \( n < \infty \) and \( p_{ji}^{(m)} = 0, \ \forall m < \infty \). In the above example, state 5 can be reached from state 3 in a finite number of transitions but state 3 can not be reached from state 5 in a finite number of transitions, hence state 3 is a transient state.

Recurrent States - a state \( i \) is said to be recurrent if the probability that the system ever returns to state \( i \) is 1. Every state which is not a transient state is a recurrent state. Consider a three state DTMC with states \{0, 1, 2\} and probability transition matrix \( P \) given as

\[
P = \begin{bmatrix}
p_{1,1} & p_{1,2} & p_{1,4} \\
p_{2,1} & p_{2,2} & p_{2,3} \\
p_{3,2} & p_{3,3} & p_{3,4} & p_{3,5} \\
p_{4,1} & p_{4,3} & p_{4,4} & p_{4,5} & p_{4,6} & p_{4,7} & p_{4,9} \\
p_{5,5} & p_{5,6} & p_{5,7} & p_{5,9} \\
p_{6,5} & p_{6,6} \\
p_{7,5} & p_{7,7} & p_{7,8} \\
p_{8,7} & p_{8,8} \\
p_{9,9}
\end{bmatrix}.
\]

All the states here are recurrent. If however, the matrix \( P \) is
2.4 Discrete Time Markov Chains

\[
P = \begin{bmatrix}
0.2 & 0.8 \\
0.1 & 0.3 & 0.6 \\
1
\end{bmatrix},
\]

then states 0 and 1 are transient with state 2 being an absorbing state and technically also being a recurrent state. Later we show that there are two types of recurrent states, viz: positive recurrent for which the mean recurrence time is finite and the null recurrent for which the mean recurrence time is infinite.

Periodic States - a state \( i \) is said to be periodic if the number of steps required to return to it is a multiple of \( k, (k > 1) \). For example, consider a Markov chain with the following transition matrix

\[
P = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix},
\]

This chain is also represented as a network in Fig(2.4). All the three states are periodic with period of \( k = 3 \).

If a state is not periodic, it is said to be aperiodic. In Fig(2.4), to go from any state back to the same state requires hopping over three links, i.e. visiting two other nodes all the time.

2.4.4 Classification of Markov Chains

In this section, we show how Markov chains are classified.

Irreducible Chains - a Markov chain is said to be irreducible if all the states in the chain communicate with each other. As an example consider two Markov chains with five states \( S_1, S_2, S_3, S_4, S_5 \) and the following transition matrices
The DTMC with transition matrix $P_1$ is irreducible, but the one with $P_2$ is reducible into two sub-sets of states: the set of states in $A = \{S_1, S_2\}$ and the set of states in $B = \{S_3, S_4, S_5\}$. The set of states in $A$ do not communicate with the set of states in $B$. Hence, this is a reducible Markov chain. The set of states in $A$ are said to form a closed set and so also are the states in $B$.

Absorbing Chains - a Markov chain is said to be an absorbing chain if any of its states is an absorbing state, i.e. if at least for one of its states $i$ we have $Pr\{X_{n+1} = i | X_n = i\} = p_{i,i} = 1$. As an example consider an absorbing DTMC with states $\{0, 1, 2\}$ in which state 0 is an absorbing state. Fig(2.5) is a an example representation of such a chain.

![absorbing markov chain diagram](image)

State 0 is the absorbing state

*Fig. 2.5* An Absorbing Markov Chain

Recurrent Markov Chain - a Markov chain is said to be recurrent if all the states of the chain are recurrent. It is often important to also specify if a chain is positive or null recurrent type based on if any of the states are positive recurrent or null recurrent. The conditions that govern the differences between these two recurrence
will be presented later in the next subsection. Proving that a Markov chain is positive recurrent is usually involved and problem specific. We will not attempt to give a general method for proving recurrence in this book. However, we will present the method used for proving recurrence for some classes of Markov chains which are commonly encountered in queueing theory.

Ergodic Markov Chain - an irreducible Markov chain is said to be ergodic if all its states are aperiodic and positive recurrent. In addition, if the chain is finite, then aperiodicity is all it needs to be an ergodic chain.

2.5 First Passage Time

In this section we consider first passage time, a very important measure which is used in many situations to characterize a Markov chain. Consider a Markov chain with the state space \( \mathcal{N} = 0, 1, 2, \ldots, N, \mathcal{N} \subseteq \mathcal{I} \) and associated transition matrix \( P \). Given that the chain is in a particular state, say \( i \) at any arbitrary time, we want to find from then on when it reaches a state \( j \) for the first time. Define \( f_{ij}^{(n)} \) as the probability that state \( j \) is visited for the first time at the \( n \)th transition, given that the system was in state \( i \) at the start, i.e. at time 0. When \( j = i \) we have \( f_{ii}^{(n)} \) as the probability that state \( i \) is visited for the first time at the \( n \)th transition, given that the system was in state \( i \) at the start, i.e. at time 0; this is known as first return probability.

Note that \( p_{ij}^{(n)} \) is different from \( f_{ij}^{(n)} \) in that the former gives the probability that the system is in state \( j \) at the \( n \)th transition given that it was in state \( i \) at the start; the system could have entered state \( j \) by the 1st, 2nd, \ldots, or \((n-1)\)th transition and remained in, or went into another state and returned to \( j \) by the \( n \)th transition. Note that \( p_{ij}^{(n)} \geq f_{ij}^{(n)} \), \( \forall n \). Consider a DTMC with states \{1, 2, 3\} we show in Figs (2.6) and (2.7) below, the difference between \( f_{2,3}^{(2)} \) and \( p_{2,3}^{(2)} \).

We see that

\[
P_{2,3}^{(2)} = f_{2,3}^{(2)} + p_{2,3}p_{3,3}.
\]

It is straightforward to see that

\[
p_{ij}^{(n)} = f_{ij}^{(n)} + (1 - \delta_{n1}) \sum_{v=1}^{n-1} f_{ij}^{(v)} p_{jj}^{(n-v)}, \quad n \geq 1,
\]

where \( \delta_{ij} \) is the kronecker delta, i.e. \( \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases} \).

The arguments leading to this expression are as follows: The system is in state \( j \) at the \( n \)th transition, given that it started from state \( i \) at time zero in which case, either

1. the first time that state \( j \) is visited is at the \( n \)th transition - this leads to the first term, or
2. the first time that state \( j \) is visited is at the \( v \)th transition \( (v < n) \) and then for the remaining \( n - v \) transitions the system could visit the \( j \) state as many times as possible (without visiting state \( i \) in between) but must end up in state \( j \) by the last transition.

After rewriting this equation, we obtain

\[
 f_{ij}^{(n)} = p_{ij}^{(n)} - (1 - \delta_{n1}) \sum_{v=1}^{n-1} f_{ij}^{(v)} p_{jj}^{(n-v)}, \ n \geq 1
\]  

(2.6)

Equation (2.6) is applied recursively to calculate first passage time probability. For example, to determine \( f_{ij}^{(T)} \), \( T > 1 \), we initiate the recursion by computing \( p_{ij}^{(n)} \geq f_{ij}^{(n)} , \forall n \). It is straightforward to note that

\[
 f_{ij}^{(1)} = p_{ij}^{(1)}
\]

and then calculate

\[
 f_{ij}^{(2)} = p_{ij}^{(2)} - f_{ij}^{(1)} p_{jj}^{(1)}
\]

\[
 f_{ij}^{(3)} = p_{ij}^{(3)} - \sum_{v=1}^{2} f_{ij}^{(v)} p_{jj}^{(3-v)}
\]
First passage probability $f_{23}^{(2)} = p_{21} p_{13} + p_{22} p_{23}$

**Fig. 2.7** Case of First Passage Probability

$$f_{ij}^{(T)} = p_{ij}^{(T)} - \sum_{v=1}^{T-1} f_{ij}^{(v)} p_{jj}^{(T-v)}$$

Note that we have to calculate $P^n, n = 2, 3, \ldots, T - 1$, during the recursion by this method. However, Neuts [85] has derived a more efficient procedure for computing the first passage probability and is given as:

$$f_{ij}^{(n+1)} = \sum_{v \in \mathcal{N}, v \neq j} p_{iv} f_{vj}^{(n)} - p_{ij} f_{jj}^{(n)}, \quad n \geq 1.$$  \hspace{1cm} (2.7)

The arguments leading to this result are as follows:

- $f_{ij}^{(2)} = \sum_{v \in \mathcal{N}, v \neq j} p_{iv} f_{vj}^{(1)} = \sum_{v \in \mathcal{N}} p_{iv} f_{vj}^{(1)} - p_{ij} f_{jj}^{(1)}$
- $f_{ij}^{(3)} = \sum_{v \in \mathcal{N}, v \neq j} p_{iv} f_{vj}^{(2)} = \sum_{v \in \mathcal{N}} p_{iv} f_{vj}^{(2)} - p_{ij} f_{jj}^{(2)}$
- $\vdots$$\vdots$$\vdots$
- $f_{ij}^{(n+1)} = \sum_{v \in \mathcal{N}, v \neq j} p_{iv} f_{vj}^{(n)} = \sum_{v \in \mathcal{N}} p_{iv} f_{vj}^{(n)} - p_{ij} f_{jj}^{(n)}, \quad n \geq 1.$
If we define the matrix $F^{(n)}$ whose elements are $f_{ij}^{(n)}$, and $F_d^{(n)}$ as the matrix $F$ with only its diagonal elements, i.e. $(F_d^{(n)})_{jj} = f_{jj}^{(n)}$ and $(F_d^{(n)})_{ij} = 0$, $i \neq j$, then the above equation can be written in matrix form as

$$F^{(n+1)} = PF^{(n)} - PF_d^{(n)} = P(F^{(n)} - F_d^{(n)}).$$  \hspace{1cm} (2.8)

This is much easier to work with for computational purposes.

### 2.5.1 Examples

Consider Example 2.2.1 with the state space $\{0, 1, 2, 3, 4\}$ and transition matrix

$$P = \begin{bmatrix}
0.70 & 0.30 & 0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
\end{bmatrix}.$$  

Suppose we currently have 2 messages in the buffer, what is the probability that the buffer becomes full (i.e. buffer will reach state 4) for the first time at the fourth transition? We wish to determine $f_{2,4}^{(4)}$, i.e. the probability that the buffer becomes full for the first time at time 4, given that there were 2 in the system at time 0. Also, given that the buffer is now full, what is the probability that the next time that the buffer is full again is at the fifth transition from now? I let the reader figure this out as an exercise.

What we need to calculate is $F^{(4)}$ and select its $(2,4)$ element. By using the formula above we can actually obtain the whole matrix $F^{(4)}$ recursively as follows:

$$F^{(1)} = P = \begin{bmatrix}
0.70 & 0.30 & 0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
0.42 & 0.46 & 0.12 & 0.42 & 0.58 \\
\end{bmatrix}, \quad F_d^{(1)} = \begin{bmatrix}
0.70 & 0.46 \\
0.46 \\
0.46 \\
0.46 \\
0.58 \\
\end{bmatrix}.$$  

We obtain

$$F^{(2)} = \begin{bmatrix}
0.1260 & 0.2100 & 0.0360 & 0.0 & 0.0 \\
0.1932 & 0.1764 & 0.0552 & 0.0144 & 0.0 \\
0.1764 & 0.1932 & 0.1008 & 0.0552 & 0.0144 \\
0 & 0.1764 & 0.1932 & 0.1008 & 0.0552 \\
0 & 0 & 0.1764 & 0.2436 & 0.0504 \\
\end{bmatrix}.$$  

After the fourth recursion we obtain
\[ F^{(4)} = \begin{bmatrix}
0.0330 & 0.1029 & 0.0414 & 0.0070 & 0.0005 \\
0.0701 & 0.0749 & 0.0362 & 0.0117 & 0.0024 \\
0.1298 & 0.0701 & 0.0302 & 0.0200 & 0.0106 \\
0.1022 & 0.1298 & 0.0726 & 0.0302 & 0.0200 \\
0.0311 & 0.1111 & 0.1526 & 0.0819 & 0.0132 
\end{bmatrix}. \]

Hence we have \( f_{2,4}^{(4)} = 0.0106. \)

### 2.5.2 Some Key Information Provided by First Passage

Define \( f_{ij} = \sum_{n=1}^{\infty} f^{(n)}_{ij} \) as the probability of the system ever passing from state \( i \) to state \( j \). It is clear that

- \( f_{ij} = 1 \) implies that state \( j \) is reachable from state \( i \).
- \( f_{ij} = f_{ji} = 1 \) implies that states \( i \) and \( j \) do communicate
- \( f^{(n)}_{ii} \) is the probability that the first recurrence time for state \( i \) is \( n \). Letting \( f_{ii} = \sum_{n=1}^{\infty} f^{(n)}_{ii} \) then
- \( f_{ii} = 1 \) for all recurrent states
- \( f_{ii} < 1 \) for all transient states.

Sometimes we are interested mainly in the first moment of the passage times. We start by obtaining the probability generating function of the first passage time. Let \( P_{ij}(z) = \sum_{n=1}^{\infty} p^{(n)}_{ij} z^n \) and \( F_{ij}(z) = \sum_{n=1}^{\infty} f^{(n)}_{ij} z^n \), \(|z| < 1\) be the probability generating functions (pgf) of the probability of transition times and that of the first passage times, respectively. Taking the pgf of the first passage equation we obtain:

\[
F_{ij}(z) = P_{ij}(z) - F_{ij}(z) P_{jj}(z) = \frac{P_{ij}(z)}{1 + P_{jj}(z)}.
\]

The \( n \)th factorial moment of the first passage time is then given as \( \frac{d^n F_{ij}(z)}{dz^n} \bigg|_{z=1} \). However, this is not simple to evaluate unless \( P_{ij}(z) \) has a nice and simple structure. The first moment can be obtained in an easier manner as follows.

The mean first passage time from state \( i \) to state \( j \) denoted by \( M_{ij} \) is given as

\[
M_{ij} = \sum_{n=1}^{\infty} n f^{(n)}_{ij}.
\]

Keeping in mind that when the system leaves state \( i \) it may go into state \( j \) directly with probability \( p_{ij} \) in which case it takes one unit of time, or it could go directly to another state \( k (\neq j) \) in which case it takes one unit of time with probability \( p_{ik} \) plus the mean first passage time from state \( k \) to state \( j \). Hence, we can calculate the mean first passage time as follows:
\[ M_{ij} = p_{ij} + \sum_{k \neq j} (1 + M_{kj})p_{ik} = 1 + \sum_{k \neq j} p_{ik}M_{jk}. \]  

(2.9)

Define a matrix \( M = (M_{ij}) \). Let \( M_d \) be a diagonal matrix whose elements are \( M_{ii} \) with \( \mathbf{1} = [1 \ 1 \ \cdots \ 1]^T \) and \( \mathbf{e}_1 = [1 \ 0 \ 0 \ \cdots \ 0]^T \), then the above equation can be written in matrix form as

\[ M = \mathbf{1} \otimes \mathbf{1}^T + \mathbf{P}(M - M_d), \]

where \( \otimes \) is the kronecker product operator, i.e. for two matrices \( A \) of order \( n \times m \) and \( B \) of order \( u \times v \) we have \( C = A \otimes B \) of dimension \( nu \times mv \) with structure

\[
C = \begin{bmatrix}
A_{11}B & A_{12}B & \cdots & A_{1m}B \\
A_{21}B & A_{22}B & \cdots & A_{2m}B \\
\vdots & \vdots & \ddots & \vdots \\
A_{n1}B & A_{n2}B & \cdots & A_{nm}B
\end{bmatrix},
\]

where \( A_{ij}, \ i = 1,2,\cdots,n; \ j = 1,2,\cdots,m \). This equation can in turn be written as a linear equation that can be solved using standard linear algebra techniques as follows.

Let \( M_j = [M_{0,j} \ M_{1,j} \ M_{2,j} \ \cdots \ M_{N,j}]^T \) for a DTMC with \( N+1 \) state space numbered \( \{0,1,2,\cdots,N\} \). Further let \( \mathbf{e}_j \) be the \( (j+1)^{st} \) column of an identity matrix of size \( N+1 \), with \( \mathbf{1} \) being a column vector of ones, then we can write

\[ \mathbf{P}((I - \mathbf{e}_j \mathbf{e}_j^T) - I)M_j = -\mathbf{1}. \]

Further define \( \tilde{P}_j = \mathbf{P}((I - \mathbf{e}_j \mathbf{e}_j^T) - I), \tilde{P} = \begin{bmatrix}
\tilde{P}_0 \\
\tilde{P}_1 \\
\vdots \\
\tilde{P}_N
\end{bmatrix}, \text{ and } \text{Vec}M = \begin{bmatrix}
M_0 \\
M_1 \\
\vdots \\
M_N
\end{bmatrix}, \]

then we have the linear equation

\[ \text{Vec}M = -\tilde{P}^{-1} \mathbf{1}_{(N+1)^2}. \]  

(2.10)

Note that \( \mathbf{1}_S \) represents a column vector of ones of dimension \( S \).

### 2.5.3 Example

Consider a DTMC with the state space \( \{0,1,2\} \) and transition matrix

\[
P = \begin{bmatrix}
0.3 & 0.6 & 0.1 \\
0.2 & 0.5 & 0.3 \\
0.6 & 0.2 & 0.2
\end{bmatrix}.
\]
2.6 Absorbing Markov Chains

We get \( \text{Vec} M = \begin{bmatrix} 3.1471 \\ 3.2353 \\ 2.0588 \\ 1.8000 \\ 2.1400 \\ 2.6000 \\ 4.7826 \\ 3.9130 \\ 4.6522 \end{bmatrix} \), which gives \( M = \begin{bmatrix} 3.1471 & 1.8000 & 4.7826 \\ 3.2353 & 2.1400 & 3.9130 \\ 2.0588 & 2.6000 & 4.6522 \end{bmatrix} \).

2.5.4 Mean first recurrence time

Also, \( M_{ii} \) is known as the mean first recurrence time for state \( i \).

- For every recurrent state \( i \), \( f_{ii} = 1 \),
- if \( M_{ii} < \infty \), we say state \( i \) is positive recurrent.
- if \( M_{ii} = \infty \), we say state \( i \) is null recurrent.

2.6 Absorbing Markov Chains

A Markov chain which has at least one absorbing state is called an absorbing Markov chain. In an absorbing finite Markov chain, every state is either an absorbing state or a transient state. For example, let the states of an absorbing Markov chain be \( \mathcal{N} = \{ S_1, S_2, ..., S_N \} \), with \( |\mathcal{N}| = N \). Let there be \( T \) transient states given as \( \mathcal{T} = \{ S_1, S_2, ..., S_T \} \), with \( |\mathcal{T}| = T \) and \( N - T \) absorbing states in this chain given as \( \mathcal{A} = \{ S_{T+1}, S_{T+2}, ..., S_N \} \), where \( \mathcal{N} = \mathcal{T} \cup \mathcal{A} \) and \( \mathcal{T} \cap \mathcal{A} = \emptyset \). Then the transition matrix can be written as

\[
P = \begin{bmatrix} Q & H \\ 0 & I \end{bmatrix},
\]

where \( Q \) is a square matrix of dimension \( T \), \( I \) is a identity matrix of dimension \( N - T \) and \( H \) is of dimension \( T \times (N - T) \). The matrix \( Q \) represents the transitions within the transient states and the matrix \( H \) represents the transitions from the transient states into the absorbing states. The matrix \( Q \) is substochastic, i.e. the sums of each row is less than or equal to 1, but most important is that at least the sum of one of the rows of \( Q \) is strictly less than 1.

As an example consider a 5 state Markov chain with state space \( \{0,1,2,3,4\} \) of which two states \( \{3,4\} \) are absorbing states. It will have a transition probability...
An absorbing Markov chain with only one absorbing state is very common and very useful in representing some distributions encountered in real life, e.g. the phase type distributions. This will be discussed at length in Chapter 3.

### 2.6.1 Characteristics of an absorbing Markov chain

1. **State of absorption:** It is clear that for an absorbing Markov chain, the system eventually gets absorbed, i.e. the Markov chain terminates eventually. One aspect of interest is usually trying to know into which state the system gets absorbed. Let us define a \( T \times (N - T) \) matrix \( B \) whose elements \( b_{ij} \) represent the probability that the system eventually gets absorbed in state \( j \in \mathcal{A} \), given that the system started in the transient state \( i \in T \). To obtain the equation for \( B \) we first obtain the following. The probability that the system is absorbed into state \( j \) at the \( n \)th transition, given that it started from a transient state \( i \) is given by

\[
\sum_{v} (Q^{n-1})_{iv} H_{vj},
\]

where \((Q^{n-1})_{iv}\) is the \((i, v)\) element of \(Q^{n-1}\) and \(H_{vj}\) is the \((v, j)\) element of \(H\).

The argument leading to this is that the system remains in the transient states for the first \( n - 1 \) transitions and then enters the absorbing state \( j \) at the \( n \)th transition. Hence, \( b_{ij} \) is obtained as the infinite sum of the series given by

\[
b_{ij} = \sum_{n=1}^{\infty} \sum_{v} (Q^{n-1})_{iv} H_{vj}.
\]

This in matrix form is

\[
B = (I + Q + Q^2 + Q^3 + \ldots)H
\]

It is straightforward to show that

\[
B = (I - Q)^{-1}H
\]

provided the inverse exists. This inverse is known to exist if \( Q \) is strictly sub-stochastic and irreducible (see Bhat [21]). In our case, \( Q \) has to be sub-stochastic. The matrix \( Q \) is said to be sub-stochastic if at least for one of the rows \( i \) we have \( \sum_j Q_{ij} < 1 \), \( \forall i \in T \) where \( Q_{ij} \) is the \((i, j)\)th element of the matrix \( Q \). The only other requirement for the inverse to exist is that it is irreducible, which will depend on other properties of the specific Markov chain being dealt with.
If we let \( a_i \) represent a probability of starting from the transient state \( i \in \mathcal{T} \) initially, with vector \( a = [a_1, a_2, ..., a_T] \), then the probability of getting absorbed into the absorbing state \( j' \in \mathcal{A} \) is given by \( b_j' \) which is obtained from the vector
\[
b = [b_1', b_2', ..., b_{(N-T)'}]
\]

\[
b = a(I - Q)^{-1} H
\]  
(2.12)

where \( j' = T + j \).

2. **Time to absorption**: Let us define a \( T \times (N - T) \) matrix \( C^{(n)} \) whose elements \( c^{(n)}_{ij'} \) represent the probability of being absorbed into state \( j' \in \mathcal{A} \) at the \( n \)th transition, given that the system was in state \( i \in \mathcal{T} \) at the start. Then \( c^{(n)}_{ij'} = (Q^{n-1})_{ij'} H_{vj'} \) and in matrix form \( C^{(n)} \) is given as

\[
C^{(n)} = Q^{n-1} H, \quad n \geq 1
\]

If we let the vector \( c^{(n)} = [c^{(n)}_1', c^{(n)}_2', ..., c^{(n)}_{(N-T)'}] \), where \( c^{(n)}_{ij'} \) is the probability that the system gets absorbed into state \( j' \in \mathcal{A} \) at the \( n \)th transition then

\[
c^{(n)} = aQ^{n-1} H, \quad n \geq 1
\]  
(2.13)

Of particular interest is the case of an absorbing Markov chain with only one absorbing state. In that case, \( R \) is a column vector and \( (I - Q)e = H \), where \( I \) is a column vector of 1’s. Hence,

\[
B = I, \quad C^{(n)} = Q^{n-1}(I - Q)I \text{ and } D = (I - Q)^{-1}I
\]

Later we will find that for the case when \( N - T = 1 \), \( c^{(n)}, \ n \geq 1 \), is a proper probability mass function. This \( C^{(n)} \) is popularly known as the phase type distribution and represented by \( (a, Q) \).

3. **Mean time to absorption**: Let us define a \( T \times (N - T) \) matrix \( \hat{D} \) whose elements \( \hat{d}_{ij} \) represent the mean time to absorption into state \( j \in \mathcal{A} \), given that the system started in state \( i \in \mathcal{T} \). Then,

\[
\hat{D} = (I + 2Q + 3Q^2 + 4Q^3 + ...)H.
\]

After some algebraic manipulations we obtain

\[
\hat{D} = (I - Q)^{-2}H.
\]  
(2.14)

Also, if we let the vector \( \hat{d} = [\hat{d}_1', \hat{d}_2', ..., \hat{d}_{(N-T)'}] \), where \( \hat{d}_{ij'} \) is the mean time to absorption to state \( j' \) then

\[
\hat{d} = a(I - Q)^{-2}H
\]  
(2.15)
2.6.1.1 Example:

Consider the 5 state Markov chain with state space \( \{0, 1, 2, 3, 4\} \) of which two states \( \{3, 4\} \) are absorbing states. Let its transition probability matrix be of the form

\[
P = \begin{bmatrix}
0.1 & 0.2 & 0.1 & 0.4 & 0.2 \\
0.3 & 0.1 & 0.5 & 0.1 & 0.0 \\
0.4 & 0.0 & 0.25 & 0.3 & 0.05 \\
0.0 & 0.0 & 0 & 1 & 0 \\
0.0 & 0.0 & 0 & 0 & 1
\end{bmatrix}.
\]

Here we have

\[
I - Q = \begin{bmatrix}
0.9 & -0.2 & -0.1 \\
-0.3 & 0.9 & -0.5 \\
-0.4 & 0.0 & 0.75
\end{bmatrix},
H = \begin{bmatrix}
0.4 & 0.2 \\
0.1 & 0.0 \\
0.3 & 0.05
\end{bmatrix}.
\]

Suppose we have \( a = [0.1 \ 0.6 \ 0.3] \), then we have

\[
b = a(I - Q)^{-1}R = [0.7683 \ 0.2317],
\]

\[
c^{(5)} = aQ^4H = [0.0464 \ 0.0159],
\]

\[
\hat{d} = a(I - Q)^{-2}H = [2.1393 \ 0.7024].
\]

Consider another example with the state space \( \{0, 1, 2, 3\} \) and transition matrix

\[
P = \begin{bmatrix}
0.1 & 0.2 & 0.1 & 0.6 \\
0.3 & 0.1 & 0.5 & 0.1 \\
0.4 & 0.0 & 0.25 & 0.35 \\
0.0 & 0.0 & 0.0 & 1.0
\end{bmatrix}.
\]

Here \( b = 1.0 \). If we start with \( a = [0.1 \ 0.6 \ 0.3] \), then we obtain \( \hat{d} = 2.8417 \) and \( c^{(n)} \) is obtained as

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>\ldots</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c^{(n)} )</td>
<td>0.2250</td>
<td>0.3287</td>
<td>0.1909</td>
<td>0.1044</td>
<td>0.0623</td>
<td>0.0367</td>
<td>0.0215</td>
<td>0.0126</td>
<td>\ldots</td>
</tr>
</tbody>
</table>

2.7 Transient Analysis

If we are interested in transient analysis of a Markov chain, i.e. \( \Pr\{X_n = j|X_0 = i\} \) we may proceed by any of the following three methods (which are algorithmically
only feasible for finite state DTMC), among several other methods. It is known that $x^{(n)} = x^{(n-1)} P = x^{(0)} P^n$.

### 2.7.1 Direct Algebraic Operations

#### 2.7.1.1 Naive repeated application of $P$

In this case we simply start with the known values of $x^{(0)}$ and repeatedly apply the relationship

$$x^{(n)} = x^{(n-1)} P,$$

starting from $n = 1$ until the desired time point. For example for a three state DTMC with transition matrix

$$P = \begin{bmatrix} 1/4 & 3/4 & 0 \\ 0 & 1/2 & 1/2 \\ 1/3 & 0 & 2/3 \end{bmatrix},$$

if we are given $x^{(0)} = [1 \ 0 \ 0]$ and we want to find the probability distribution of the state of the DTMC at the $4^{th}$ time epoch we simply apply the recursion four times as

$$x^{(1)} = x^{(0)} P, \ x^{(2)} = x^{(1)} P, \ x^{(3)} = x^{(2)} P, \ x^{(4)} = x^{(3)} P,$$

and obtain

$$x^{(4)} = [0.2122 \ 0.2695 \ 0.5182].$$

We will notice that as $n$ gets very large then $x^{(n+1)} \approx x^{(n)}$, and in fact the differences get smaller as $n$ increases. This is the limiting behaviour of an ergodic Markov chain and will be presented in detail later.

#### 2.7.1.2 Matrix decomposition approach

A second approach is from observing that the matrix $P$ can be decomposed into the form

$$P = \Lambda D \Lambda^{-1},$$

where $D$ is a diagonal matrix consisting of the eigenvalues of $P$ and $\Lambda$ is a matrix for which each of its row is the eigenvector corresponding to one of the eigenvalues. Hence we have

$$P^n = \Lambda D^n \Lambda^{-1},$$

which leads to

$$x^{(n)} = x^{(0)} \Lambda D^n \Lambda^{-1}. \quad (2.16)$$

For the example given above we have
where the symbol \( i \) refers to complex component of the number. This leads to

\[
x^{(4)} = x^{(0)} \Lambda D^4 \Lambda^{-1} = [0.2122 \ 0.2695 \ 0.5182].
\]

### 2.7.2 Transient Analysis Based on the z-Transform

Consider the relationship \( x^{(n)} = x^{(n-1)} P \) and take the z-transform or the probability generating functions of both sides and let \( X(z) = \sum_{n=0}^{\infty} x^{(n)} z^n, |z| \leq 1 \) we obtain

\[
X(z) - x^{(0)} = zX(z)P \Rightarrow X(z) = x^{(0)} [I - zP]^{-1}.
\]

We know that \( A^{-1} = \frac{1}{\text{det}(A)} \text{Adj}(A) \). While this method is based on a classical theoretical approach, it is not simple to use analytically for a DTMC that has more than 2 states. We use an example to show how this method works. Consider the case of \( P = \begin{bmatrix} 1/3 & 3/4 \\ 1/2 & 1/2 \end{bmatrix} \). For this example, we have \( \text{det}[I - zP]^{-1} = (1-z) \left(1 + \frac{1}{4}z\right) \) and

\[
[I - zP]^{-1} = \frac{1}{1-z} \left(1 + \frac{1}{4}z\right) \begin{bmatrix}
1 - \frac{2}{3}z & -\frac{3}{4}z \\
\frac{2}{3}z & 1 - \frac{3}{4}z
\end{bmatrix}.
\]

Using partial fractions, we have

\[
[I - zP]^{-1} = \frac{1}{1-z} \begin{bmatrix}
\frac{2}{3}z & -\frac{3}{4}z \\
\frac{2}{3}z & \frac{2}{3}z
\end{bmatrix} + \frac{1}{1+\frac{1}{4}z} \begin{bmatrix}
\frac{3}{5} & -\frac{3}{5} \\
-\frac{3}{5} & \frac{2}{5}
\end{bmatrix}.
\]

Keeping in mind that the inverse of the z-transform \( \frac{1}{1-z} \) is the unit step 1 and that of \( \frac{1}{1-az} \) is \( a^n \), then we have

\[
P^n = \begin{bmatrix}
\frac{2}{5}a^n & \frac{3}{5}a^n \\
\frac{3}{5}a^n & \frac{2}{5}a^n
\end{bmatrix} + \left(-\frac{1}{4}\right)^n \begin{bmatrix}
\frac{3}{5} & -\frac{3}{5} \\
-\frac{3}{5} & \frac{2}{5}
\end{bmatrix}.
\]

This is the transient solution to this chain. We notice that as \( n \to \infty \), the second term on the right hand side vanishes and the first term is the limiting or steady state portion of the solution. In general, we can write the closed form of \( X(z) \) as

\[
X(z) = \frac{1}{1-z} x + x^{(0)} T^g(z)
\]
where $T^g(z)$ is the transform part that gives the additional part of the solution to lead to transient solution. When it is removed, we only obtain the steady state solution. For this example

$$T^g(z) = \frac{1}{1 + \frac{1}{4z}} \begin{bmatrix} \frac{3}{5} & -\frac{3}{5} \\ \frac{2}{5} & \frac{2}{5} \end{bmatrix}.$$ 

For details on this method the reader is referred to Howard [56].

### 2.8 Limiting Behaviour of Markov Chains

One very important key behaviour of interest to us with regards to a time-homogeneous DTMC is its limiting behaviour. i.e. after a long period of time. In studying this limiting behaviour of Markov chains, we need to make clear under what conditions we are able to be specific about this behaviour. We identify the difference between the long term behaviour of an ergodic and a non-ergodic DTMC.

#### 2.8.1 Ergodic Chains

Ergodic Markov chains are those chains that are irreducible, aperiodic and positive recurrent. Proving ergodicity could be very involved in some cases. Let $P$ be the transition matrix of an ergodic DTMC. If the probability vector of the initial state of the system is given by $x^{(0)}$, then the probability vector of the state of the system at any time is given by

$$x^{(n)} = x^{(n-1)}P = x^{(0)}P^n$$

As $n \to \infty$ we have $x^{(n)} \to x \implies x^{(n)} \to xP$, which is known as the invariant (stationary) probability vector of the system.

Similarly, we have

$$P^n \to \begin{bmatrix} y \\ y \\ \vdots \\ y \end{bmatrix} \text{ as } n \to \infty$$

where $y$ is known as the limiting or equilibrium or steady state probability vector of the system.

In the case of the ergodic system, we have

$$x = y.$$ 

We return to the Example 2.2.1, where we have
\[
P = \begin{bmatrix}
0.70 & 0.30 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.46 & 0.12 \\
0.42 & 0.58 &
\end{bmatrix}
\]

Here \( x^{(n+1)} = x^{(n)} = x_{n \to \infty} = [0.5017, 0.3583, 0.1024, 0.0293, 0.0084] \). Similarly if we find that \( P^n \to \begin{bmatrix}
y \\
y \\
0 \\
0 \\
y
\end{bmatrix} \) as \( n \to \infty \), where \( y = [0.5017, 0.3583, 0.1024, 0.0293, 0.0084] \).

### 2.8.2 Non-Ergodic Chains

For non-ergodic systems

\( \mathbf{x} \) is not necessarily equal to \( \mathbf{y} \)

even when both do exist.

For example, consider the transition matrix of the non-ergodic system given by

\[
P = \begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

This is a periodic system. For this system, \( \mathbf{x} = [0.5, 0.5] \) and \( \mathbf{y} \) does not exist because the system alternates between the two states. This DTMC is periodic. At any instant in time, \( \mathbf{y} \) could be \([1, 0]\) or \([0, 1]\), i.e. we have

\[
P^n = \begin{cases}
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}, & \text{if } n \text{ is even} \\
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, & \text{if } n \text{ is odd}
\end{cases}
\]

We consider another example of a non-ergodic system by virtue of not being irreducible. For a reducible DTMC described by a transition matrix of the form

\[
P = \begin{bmatrix}
0.5 & 0.5 \\
0.6 & 0.4 \\
0.2 & 0.8 \\
0.9 & 0.1
\end{bmatrix}
\]

For this system, the solution to \( \mathbf{x} = \mathbf{x}P \), \( \mathbf{x}1 = 1 \) does not exist, yet we have

\[
P^n = \begin{bmatrix}
y \\
y \\
z \\
z
\end{bmatrix}, \text{ where}
\]
Finally, we consider another example of a non-ergodic DTMC by virtue of not being positive recurrent. Suppose we have an infinite number of states \( \{0, 1, 2, 3, 4, \ldots\} \) and the transition matrix of this is given as

\[
P = \begin{bmatrix}
0.3 & 0.7 \\
0.1 & 0.1 & 0.8 \\
0.1 & 0.1 & 0.8 \\
0.1 & 0.1 & 0.8 \\
& & \ddots
\end{bmatrix}.
\]

This DTMC will neither have an invariant vector nor a steady state vector.

In the rest of the book we focus mainly on ergodic DTMCs.

### 2.8.3 Mean First Recurrence Time and Steady State Distributions

For an ergodic system we know that \( x_i \) is the probability of being in state \( i \) at any time, after steady state has been reached. But we also know that \( M_{ii} \) is the mean recurrence time for state \( i \). Hence, the probability of being in state \( i \) is given as \( 1/M_{ii} \).

### 2.9 Numerical Computations of the Invariant Vectors

In this section, we distinguish between finite and infinite state Markov chains as we present the methods. Our interest is to compute the invariant probability vector \( x \) rather than the limiting probability vector \( y \). We know and show later that \( x \) exists for every irreducible positive recurrent Markov chain. We therefore focus on such chains only.

We want to determine \( x \) such that

\[
x = xP \quad \text{and} \quad x1 = 1.
\]  

(2.17)

### 2.9.1 Finite State Markov Chains

There are mainly two types of methods used for computing the stationary vector of a DTMC; direct methods and iterative methods. Iterative methods are used more frequently for most DTMC which have a large state space – very common in most practical situations. However, the computational effort for iterative methods are de-
dependent on the size and structure of the DTMC and also the starting solution. Direct methods are suitable for smaller DTMC and the number of steps required to get a solution is known ahead of time.

For the iterative methods we work with the equation in the form

$$\mathbf{x} = \mathbf{x}P, \quad \mathbf{x}1 = 1,$$

whereas for most direct methods we convert the problem to a classical linear algebra problem of the form $AX = b$, by creating a matrix $\tilde{P}$ which is the same as $I - P$, with the last column replaced by a column of ones. Our problem (of $N$ states Markov chain) then becomes

$$e_N^T = \mathbf{x} \tilde{P} \quad \rightarrow \quad \tilde{P}^T \mathbf{x}^T = e_N,$$

which is now in the standard form. We use the notation $e_j$ to represent a column vector which has all zero entries except in the $j^{th}$ column that we have 1 and $e_j^T$ is its transpose.

For example, consider a three state DTMC \{0, 1, 2\} with transition matrix $P$ given as

$$P = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.2 & 0.3 & 0.5 \\ 0.4 & 0.5 & 0.1 \end{bmatrix}.$$ We will use this example in all of our numerical computation methods for invariant vectors.

For the iterative method we work with

$$[x_0 \quad x_1 \quad x_2] = [x_0 \quad x_1 \quad x_2] \begin{bmatrix} .3 & .6 & .1 \\ .2 & .3 & .5 \\ .4 & .5 & .1 \end{bmatrix}, \quad [x_1 \quad x_2 \quad x_3] \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} = 1,$$

whereas for the direct method we work with

$$\begin{bmatrix} 0.7 & -0.2 & -0.4 \\ -0.6 & 0.7 & -0.5 \\ 1.0 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix},$$

where

$$\tilde{P} = \begin{bmatrix} 0.7 & -0.6 & 1.0 \\ -0.2 & 0.7 & 1.0 \\ -0.4 & -0.5 & 1.0 \end{bmatrix} \rightarrow \tilde{P}^T = \begin{bmatrix} 0.7 & -0.2 & -0.4 \\ -0.6 & 0.7 & -0.5 \\ 1 & 1 & 1 \end{bmatrix}.$$ Direct methods do not involve iterations. They simply involve a set of operations which is usually of a known fixed number, depending on the size of the Markov chain. They are efficient for small size Markov chains and usually very time consuming for large Markov chains. One direct method which seems to be able to handle a large size Markov chain is the state reduction method (by Grassmann, Taksar Heyman [46]). Theoretically, one may state that it has no round off errors. The most
popular of the direct techniques is the Gaussian elimination. For more details of this algorithm as applicable to Markov chains, see Stewart [93].

### 2.9.1.1 Direct Methods

In what follows we present some of the well known direct methods. We present only a synoptic kind of view of this approach in this book. First we assume that the equations have been transformed to the standard form

\[ AX = b. \]  

(2.18)

Using the above example we have

\[
P = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.2 & 0.3 & 0.5 \\ 0.4 & 0.5 & 0.1 \end{bmatrix},
\]

which results in

\[
A = \begin{bmatrix} 0.7 & -0.2 & -0.4 \\ -0.6 & 0.7 & -0.5 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad b = \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}.
\]

- **Inverse matrix approach**: This is the simplest of all the direct methods. It simply solves for \( X \) as \( X = A^{-1}b \), assuming that the inverse of \( A \) exists. For most of the finite DTMC we will be discussing in this book, the inverse of \( A \) would exist. From the above example we have

\[
x^T = A^{-1}b = \begin{bmatrix} 0.8955 & -0.1493 & 0.2836 \\ 0.0746 & 0.8209 & 0.4403 \\ -0.9701 & -0.6716 & 0.2761 \end{bmatrix} \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.2836 \\ 0.4403 \\ 0.2761 \end{bmatrix}.
\]

- **Gaussian Elimination**: This is another popular method which is more suitable for moderate size matrices than the inverse approach. The matrix \( A \) is transformed through row operations into an upper triangular matrix \( U \), such that \( Ux^T = b = e_N \) and then the elements of \( x \) are obtained from back substitution as

\[
x_N = 1/u_{N,N}, \quad x_i = -\left( \sum_{k=i+1}^{N} u_{i,k}x_k \right)/u_{i,i}, \quad i = 0, 1, 2, \ldots, N - 1. \]  

(2.19)

For this example, we have

\[
U = \begin{bmatrix} 0.7 & -0.2 & -0.4 \\ 0.5286 & -0.8429 \\ 3.6216 \end{bmatrix}.
\]
Applying the back substitution equation we obtain

$$\mathbf{x}^T = \begin{bmatrix} 0.2836 \\ 0.4403 \\ 0.2761 \end{bmatrix}.$$ 

- **State Reduction Method:** This method is also popularly known as the GTH (Grassmann, Taksar and Heyman [46]) method. It is actually based on Gaussian elimination with state space reduction during each iteration and it uses the form of equation $\mathbf{x} = \mathbf{x}P$, $\mathbf{x}1 = 1$, unlike the other direct methods. Consider the equation $x_j = \sum_{i=0}^{N} x_i p_{ij}$, $j = 0, 1, 2, ..., N$.

The procedure starts by eliminating $x_N$ from the $N$th equation to obtain

$$x_N = \sum_{i=0}^{N-1} x_i p_{iN} / (1 - p_{NN}).$$

Replacing $1 - p_{NN}$ with $\sum_{j=0}^{N-1} p_{N,j}$ and using the property that $x_j = \sum_{i=0}^{N-1} x_i p_{ij}$, $j = 0, 1, 2, ..., N-1$, where

$$p_{i,j}^{(N-1)} = p_{i,j} + p_{i,N} p_{N,j} / (1 - p_{NN}) = p_{i,j} + p_{i,N} p_{N,j} / \sum_{j=0}^{N-1} p_{N,j}.$$ 

This is the principle by which state reduction works.

Generally, $x_n = \sum_{i=0}^{n-1} x_i q_{i,n}$, $n = N, N-1, N-2, ..., 2, 1$ where $q_{i,n} = p_{i,n}^{(n)} / \sum_{j=0}^{n-1} p_{n,j}^{(n)}$.

The algorithm can be stated as follows (Grassmann, Taksar and Heyman [46]) referred to as the GTH:

**GTH Algorithm:**

1. For $n = N, N-1, N-2, ..., 2, 1$, do

   $$q_{i,n} \leftarrow p_{i,n} / \sum_{j=0}^{n-1} p_{n,j}, \quad i = 0, 1, ..., n-1$$

   $$p_{i,j} \leftarrow p_{i,j} + q_{i,n} p_{n,j}, \quad i, j = 0, 1, ..., n-1$$

   $$r_1 \leftarrow 1$$

2. For $j = 2, 3, ..., N$, $r_j \leftarrow \sum_{i=0}^{j-1} r_i q_{i,j}$

3. For $j = 2, 3, ..., N$, $x_j \leftarrow r_j / \sum_{i=0}^{N} r_i$, $j = 0, 1, ..., N$

The advantage with this approach is that it is numerically stable even if the system is ill-conditioned.
2.9.1.2 Iterative Methods:

Iterative methods are usually more efficient and more effective than the direct methods when the size of the matrix is large. The number of iterations required to achieve the desired convergence is not usually known in advance and the resulting distributions have round-off errors associated with them.

Four commonly used iterative methods will be discussed in this section. They are: i) the Power method which is more applicable to transition matrices, ii) the Jacobi method, iii) the Gauss-Seidel method, and iv) the method of successive over-relaxation.

• The Power Method: The power method simply involves starting with a probability vector \( \mathbf{x}^{(0)} \) which satisfies the condition \( \mathbf{x}^{(0)} \mathbf{1} = 1 \) and then applying the relationship \( \mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} \mathbf{P}, \ n \geq 0 \) until \( |\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}|_i < \varepsilon, \ \forall i, \) then we stop the iteration, where \( \varepsilon \) is the convergence criterion. Often the number of iterations required depends on both \( \mathbf{P} \) and the starting vector \( \mathbf{x}^{(0)} \). It is a very naive approach which is only useful for small size problems, and especially if we also need to know the transient behaviour of the DTMC before it reaches steady state. The next three approaches will be discussed here in a modified form for the transition matrix.

First let us write \( \mathbf{Q} = \mathbf{P} - \mathbf{I} \). Our interest is to solve for \( \mathbf{x} = [x_1, x_2, ..., x_n] \), where \( \mathbf{xQ} = \mathbf{0} \) and \( \mathbf{x1} = 1 \). In order to conform to the standard representations of these methods we write the same equation in a transposed form as \( \mathbf{Q}^T \mathbf{x}^T = \mathbf{0} \).

Let us now separate \( \mathbf{Q}^T \) into a diagonal component \( \mathbf{D} \), an upper triangular component \( \mathbf{U} \) and a lower triangular component \( \mathbf{L} \) with \( ij \) elements given as \(-d_{ii}, u_{ij} \) and \( l_{ij} \), respectively. It is assumed that \( d_{ii} > 0 \). Note that the \( \mathbf{D}, \mathbf{L} \) and \( \mathbf{U} \) are not the same as the well known \( \mathbf{LU} \) or \( \mathbf{LDU} \) decompositions. For example given a matrix

\[
\mathbf{Q}^T = \begin{bmatrix}
q_{0,0} & q_{0,1} & q_{0,2} \\
q_{0,1} & q_{1,1} & q_{1,2} \\
q_{0,2} & q_{1,2} & q_{2,2}
\end{bmatrix},
\]

we can write \( \mathbf{Q}^T = \mathbf{L} - \mathbf{D} + \mathbf{U} \), where

\[
\mathbf{L} = \begin{bmatrix}
0 & 0 & 0 \\
q_{0,1} & 0 & 0 \\
q_{0,2} & q_{1,2} & 0
\end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix}
0 & q_{1,0} & q_{2,0} \\
0 & 0 & q_{2,1} \\
0 & 0 & 0
\end{bmatrix}, \\
\mathbf{D} = \begin{bmatrix}
-q_{0,0} & 0 & 0 \\
0 & -q_{1,1} & 0 \\
0 & 0 & -q_{2,2}
\end{bmatrix}.
\]

• The Jacobi Method: Then the Jacobi iteration is given in the scalar form as

\[
x_i^{(k+1)} = \frac{1}{d_{ii}} \left\{ \sum_{j \neq i} (l_{ij} + u_{ij}) x_j^{(k)} \right\}, \quad i = 0, 1, 2, ..., N \quad (2.20)
\]
or in matrix form, it can be written as
\[(\mathbf{x}^{(k+1)})^T = D^{-1}(L + U)(\mathbf{x}^{(k)})^T.\] (2.21)

For example, consider the Markov chain given by
\[
P = \begin{bmatrix}
  p_{0,0} & p_{0,1} & p_{0,2} \\
p_{1,0} & p_{1,1} & p_{1,2} \\
p_{2,0} & p_{2,1} & p_{2,2}
\end{bmatrix} = \begin{bmatrix}
  0.1 & 0.2 & 0.7 \\
  0.3 & 0.1 & 0.6 \\
  0.2 & 0.5 & 0.3
\end{bmatrix}
\]

Suppose we have an intermediate solution \(\mathbf{x}^{(k)}\) for which \(\mathbf{x}^{(k)}\mathbf{1} = 1\), then we have
\[
\begin{align*}
x_0^{(k+1)} &= \frac{1}{1 - p_{0,0}} [x_1^{(k)} p_{1,0} + x_2^{(k)} p_{2,0}] \\
x_1^{(k+1)} &= \frac{1}{1 - p_{1,1}} [x_0^{(k)} p_{0,1} + x_2^{(k)} p_{2,1}] \\
x_2^{(k+1)} &= \frac{1}{1 - p_{2,2}} [x_0^{(k)} p_{0,2} + x_1^{(k)} p_{1,2}]
\end{align*}
\]

- **Gauss-Seidel Method**: The Gauss-Seidel method uses the most up-to-date information on solutions available for computations. It is given by the following iteration in the scalar form:

\[
x_i^{(k+1)} = \frac{1}{d_{ii}} \left\{ \sum_{j=0}^{i-1} l_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{n} u_{ij} x_j^{(k)} \right\} \quad (2.22)
\]

and in matrix form as
\[
(\mathbf{x}^{(k+1)})^T = D^{-1}(L(\mathbf{x}^{(k+1)})^T + U(\mathbf{x}^{(k)})^T) \quad (2.23)
\]

This method seems to be more frequently used than the Jacobi. Using the above example, we have
\[
\begin{align*}
x_0^{(k+1)} &= \frac{1}{1 - p_{0,0}} [x_1^{(k)} p_{1,0} + x_2^{(k)} p_{2,0}] \\
x_1^{(k+1)} &= \frac{1}{1 - p_{1,1}} [x_0^{(k+1)} p_{0,1} + x_2^{(k)} p_{2,1}] \\
x_2^{(k+1)} &= \frac{1}{1 - p_{2,2}} [x_0^{(k+1)} p_{0,2} + x_1^{(k+1)} p_{1,2}]
\end{align*}
\]

- **The Successive Over-relaxation Method**: This method introduces an extra parameter \(\omega\). The method is a generalization of the Gauss-Seidel. The iteration is implemented as follows:
\[ x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega \left\{ \frac{1}{d_{ii}} \left[ \sum_{j=0}^{i-1} l_{ij}x_j^{(k+1)} + \sum_{j=i+1}^{N} u_{ij}x_j^{(k)} \right] \right\}, \quad (2.24) \]

and in matrix form

\[ (x^{(k+1)})^T = (1 - \omega)(x^{(k)})^T + \omega \left\{ D^{-1}(L(x^{(k+1)})^T + U(x^{(k)})^T) \right\}. \quad (2.25) \]

Using the above example, we have

\[ x_0^{(k+1)} = (1 - \omega)x_0^{(k)} + \omega \frac{1}{1-p_{0,0}} [x_1^{(k)} p_{1,0} + x_2^{(k)} p_{2,0}] \]

\[ x_1^{(k+1)} = (1 - \omega)x_1^{(k)} + \omega \frac{1}{1-p_{1,1}} [x_0^{(k)} p_{0,1} + x_2^{(k)} p_{2,1}] \]

\[ x_2^{(k+1)} = (1 - \omega)x_2^{(k)} + \omega \frac{1}{1-p_{2,2}} [x_0^{(k)} p_{0,2} + x_1^{(k)} p_{1,2}] \]

The value of \( \omega \) is usually selected between 0 and 2. When \( \omega < 1 \), we say we have an under-relaxation and when \( \omega > 1 \), we have the Gauss-Seidel iteration. The greatest difficulty with this method is how to select the value of \( \omega \). The Jacobi method is very appropriate for parallelization of computing. Note that diagonal dominance of the matrix \( I - P \) is required for the Jacobi and in fact also for the Gauss-Seidel methods but it is always satisfied for a Markov chain.

### 2.9.2 Bivariate Discrete Time Markov Chains

Consider a bivariate process \( \{(X_n, Y_n), n = 0, 1, 2, \cdots, \} \) that has a Markov structure. We assume that \( X_n \) assumes values in the set \( \mathcal{S} \) and \( Y_n \) assumes values in the set \( \mathcal{M} = \{1, 2, \cdots, M < \infty\} \). An example of this is a system that has two classes of data packets of types 1 and 2, where type 1 has a higher priority than type 2. Suppose we do not allow more than \( M \) of type 2 packets to be in the system at any one time, we can represent \( X_n \) as the number of type 1 and \( Y_n \) as the number of type 2 packets in the system at time \( n \). If we assume that \( \{(X_n, Y_n); n = 0, 1, 2, \cdots\} \) is a DTMC we can write the transition matrix of this Markov chain as

\[ P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \cdots \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots \\ P_{2,0} & P_{2,1} & P_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2.26) \]

where \( P_{i,j}, (i, j) = 0, 1, 2, \cdots \) is a block matrix of order \( M \times M \). The element \((P_{i,j})_{l,k}, (i, j) = 0, 1, 2, \cdots; (l, k) = 1, 2, \cdots, M\), is \( \text{Pr}\{X_{n+1} = j, Y_{n+1} = k | X_n = i, Y_n = l\} \). Usually we call the set \( L_i = \{(i, 1), (i, 2), \cdots, (i, M)\} \) level \( i \). We also say in state \( (i, l) \) that we are in level \( i \) and phase \( l \).
Define $x_{i,k}^{(n)} = Pr\{X_n = i, Y_n = k\}$ and write $x^{(n)} = [x_0^{(n)} \ x_1^{(n)} \ x_2^{(n)} \ \ldots]$ with $x_i^{(n)} = [x_{i,1}^{(n)} x_{i,2}^{(n)} \ \ldots x_{i,M}^{(n)}]$. We have

$$x^{(n+1)} = x^{(n)} P.$$ 

If the DTMC is positive recurrent then we have $x_{i,k}^{(n)} |_{n \to \infty} = x_{i,k}$, hence

$$x = x P, \quad x 1 = 1.$$ 

### 2.9.3 Computing Stationary Distribution for the Finite Bivariate DTMC

The methods presented in Section 2.6.1 are easily extended to the bivariate and all multivariate finite DTMC.

To avoid unnecessary notational complexities we present only the case where $P_{i,j}$ is of dimension $M \times M$, $\forall (i,j)$. We can easily modify the result for cases where the dimension of $P_{i,j}$ depends on $i$ and $j$. Let $1_M$ be an $M$ column vector of ones and $0_M$ an $M \times 1$ column vector of zeros. Assume $X_n$ has a state space given as $0, 1, 2, \ldots, N < \infty$, then we can write the stationary equation as

$$0_M^T = x_0 P_{0,0} - I + x_1 P_{1,0} + x_2 P_{2,0} + \ldots + x_N P_{N,0}$$

$$0_M^T = x_0 P_{0,1} + x_1 (P_{1,1} - I) + x_2 P_{2,1} + \ldots + x_N P_{N,1}$$

$$\vdots$$

$$0_M^T = x_0 P_{0,N-1} + x_1 P_{1,N-1} + \ldots + x_{N-1} (P_{N-1,N-1} - I) + x_N P_{N,N-1}$$

$$1_M^T = x_0 + x_1 + x_2 + \ldots + x_N$$

After transposing we have

$$\begin{bmatrix} P_{0,0} - I & P_{1,0} & P_{2,0} & \ldots & P_{N,0} \\ P_{0,1} & P_{1,1} - I & P_{2,1} & \ldots & P_{N,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{0,N-1} & P_{1,N-1} & P_{2,N-1} & \ldots & P_{N,N-1} \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} 0_M \\ 0_M \\ \vdots \\ 0_M \\ 1_M \end{bmatrix}. \quad (2.27)$$

This is now of the form similar to

$$AX = b$$

and can be solved using the techniques discussed earlier. The techniques are briefly listed here:

- **Direct Methods**
2.9 Numerical Computations of the Invariant Vectors

- **The inverse matrix approach**: This approach can be used for the bivariate DTMC just like in the case of the univariate DTMC. Simply we compute $x$ from

$$X = A^{-1}b. \tag{2.28}$$

- **The block Gaussian elimination method**: The matrix $A$ is transformed through block row operations into an upper triangular block matrix $U$, such that $Ux^T = b$. The triangular matrix consists of block matrices $U_{i,j}$, for which $U_{i,j} = 0, i > j$. The block vectors $(x_i)$ of $x$ are obtained from back substitution as

$$x_i^T = u_{i,i}^{-1}(\sum_{k=i+1}^{N} U_{i,k}x_k^T), i = 0, 1, 2, \cdots, N - 1. \tag{2.29}$$

- **Block state reduction**: This technique and its scalar counterpart are based on censoring of the Markov chain. Consider the DTMC with $N + 1$ levels labelled $\{0, 1, 2, \cdots, N\}$. For simplicity we will call this the block state space $\{0, 1, 2, \cdots, N\}$ and let it be partitioned into two subsets $E$ and $\bar{E}$ such that $E \cup \bar{E} = \{0, 1, 2, \cdots, N\}$ and $E \cap \bar{E} = \emptyset$. For example, we may have $E = \{0, 1, 2, \cdots, M\}$ and $\bar{E} = \{M + 1, M + 2, \cdots, N\}$, $M \leq N$. The transition matrix of the DTMC can be written as

$$P = \begin{bmatrix} H & S \\ T & U \end{bmatrix}.$$  

We can write the stationary vector of this DTMC as $x = [x_e \ x'_e]$ from which we obtain

$$x_e = x_eH + x'_eT, \quad x'_e = x_eS + x'_eU. \tag{2.30}$$

From this we obtain

$$x_e = x_e(H + S(I - U)^{-1}T). \tag{2.31}$$

The matrix $H + S(I - U)^{-1}T$ is a new transition matrix and it is stochastic. Its stationary distribution is the vector $x_e$ which can be obtained as $x_e = x_eP_e$, $x_e1 = 1$ and normalized for it to be stochastic. It is immediately clear that we have reduced or restricted this DTMC to be observed only when it is in the states of $E$. This is a censored DTMC. This idea is used to block-reduce the DTMC and apply the block state reduction to solve for $x$ in the original DTMC.

In block state reduction we first partition the DTMC into two sets with the first set given as $E = \{0, 1, 2, \cdots, N - 1\}$ and the second set $\bar{E} = \{N\}$. We study the states of $E$ only as censored DTMC, and then partition it into $E = \{0, 1, 2, \cdots, N - 2\}$ and $\bar{E} = \{N - 1\}$, and the process continues until we reduce it to just the set $E = \{0\}$. We obtain the stationary distribution of this reduced DTMC. We then start to expand it to states $E = \{0, 1\}$ and continue to expand until we reach the original set of states $\{0, 1, 2, \cdots, N\}$. 

We demonstrate how this procedure works through a small example. Consider a bivariate DTMC that has four levels in its states given by \{0, 1, 2, 3\} and the phases are of finite dimension \(M\). Its transition matrix \(P\) has block elements \(P_{i,j}\) of order \(M \times M\). If we partition the state space into \(E = \{0, 1, 2\}\) and \(\bar{E} = \{3\}\), then

\[
P_e = \begin{bmatrix}
P_{00} & P_{01} & P_{02} \\
P_{10} & P_{11} & P_{12} \\
P_{20} & P_{21} & P_{22}
\end{bmatrix} + \begin{bmatrix}
P_{03} \\
P_{13} \\
P_{23}
\end{bmatrix}(I - P_{33})^{-1}\begin{bmatrix}
P_{30} & P_{31} & P_{32}
\end{bmatrix}.
\]

When this new set \(\{0, 1, 2\}\) is further partitioned into \(E = \{0, 1\}\) and \(\bar{E} = \{2\}\) we get another transition matrix \(P_e\). Another step further leads to a partitioning into \(E = \{0\}\) and \(\bar{E} = \{1\}\). Writing \(\bar{x} = \bar{x}P\), we can write the last block equation of this stationary distribution as

\[
\bar{x}_3 = \bar{x}_0P_{0,3} + \bar{x}_1P_{1,3} + \bar{x}_2P_{2,3} + \bar{x}_3P_{3,3}.
\]

From this we have

\[
\bar{x}_3 = (\bar{x}_0P_{0,3} + \bar{x}_1P_{1,3} + \bar{x}_2P_{2,3})(I - P_{33})^{-1}.
\]

The block equation before the last can be written as

\[
\bar{x}_2 = \bar{x}_0P_{0,2} + \bar{x}_1P_{1,2} + \bar{x}_2P_{2,2} + \bar{x}_3P_{3,2},
\]

which can be also be written as

\[
\bar{x}_2 = \bar{x}_0(P_{0,2} + P_{0,3}K_{3,2}) + \bar{x}_1(P_{1,2} + P_{1,3}K_{3,2}) + \bar{x}_2(P_{2,2} + P_{2,3}K_{3,2}),
\]

where \(K_{3,2} = (I - P_{3,3})^{-1}P_{3,2}\). This in turn leads to

\[
\bar{x}_2 = (\bar{x}_0(P_{0,2} + P_{0,3}K_{3,2}) + \bar{x}_1(P_{1,2} + P_{1,3}K_{3,2}))(I - (P_{2,2} + P_{2,3}K_{3,2}))^{-1}.
\]

This process is continued further to reduce the block equation and obtain \(\bar{x}_1\) in terms of \(\bar{x}_0\) and then solve for \(\bar{x}_0\) after one more reduction. This process is then reversed to expand into obtaining \(\bar{x}_1\) from \(\bar{x}_0\), followed by obtaining \(\bar{x}_2\) using \(\bar{x}_0\) and \(\bar{x}_1\) and finally \(\bar{x}_3\) using \(\bar{x}_0\), \(\bar{x}_1\) and \(\bar{x}_2\). The final result is then normalized. A detailed description of this algorithm together with the steps can be found in [46].

- **Iterative Methods**: The same sets of methods described earlier can be used for bivariate DTMC, i.e. the power method, Jacobi, Gauss-Seidel and the Successive over relaxation. The power method approach is straightforward and will not be elaborated on here.

First let us write \(Q = P - I\). In that case, our interest is to solve for \(x = [x_0\ x_1\ x_2\ \cdots\ x_n]\), where \(xQ = 0\) and \(x\ I = 1\). In order to conform to the stan-
2.9 Numerical Computations of the Invariant Vectors

Let us now separate $Q^T$ into a diagonal component $D$, an upper triangular component $U$ and a lower triangular component $L$ with $i j$ block elements given as $-D_{i j}, U_{i j}$ and $L_{i j}$, respectively. It is assumed that the inverse of $D_{i j}$ exists. Note that the $D$, $L$ and $U$ are not the same as the well known $LU$ or $LDU$ decompositions.

- **The Jacobi Method**: Then the Jacobi iteration is given in the scalar form as
  \[
  (x^{(k+1)})^T = D^{-1}(L + U)(x^{(k)})^T \tag{2.32}
  \]

- **Gauss-Seidel Method**: The Gauss-Seidel method uses the most up-to-date information on solutions available for computations. It is given by the following iteration
  \[
  (x^{(k+1)})^T = (D)^{-1}(L(x^{(k+1)})^T + U(x^{(k)})^T). \tag{2.33}
  \]

- **The Successive Over-relaxation Method**: This method introduces an extra parameter $\omega$. The method is a generalization of the Gauss-Seidel. The iteration is implemented as follows:
  \[
  (x^{(k+1)})^T = (1 - \omega)(x^{(k)})^T + \omega\{D^{-1}(L(x^{(k+1)})^T + U(x^{(k)})^T)\}. \tag{2.34}
  \]

### 2.9.4 Special Finite State DTMC

For special types of finite state DTMC which are of the block tridiagonal structures there are specially effective algorithms for obtaining the stationary vectors. Such a DTMC has a matrix of the form

\[
P = \begin{bmatrix}
A_0 & U_0 \\
D_1 & A_1 & U_1 \\
& \ddots & \ddots & \ddots \\
& & D_N & A_N
\end{bmatrix},
\tag{2.35}
\]

where the entries $A_i, U_i, D_i, i = 0, 1, \cdots, N$ could be block matrices of finite dimensions. Our interest is to solve for the vector $x = [x_0, x_1, \cdots, x_N]$ where

\[
x = xP, \quad x1 = 1,
\]

and $x_k, k = 0, 1, 2, \cdots, N$, is also a vector.

Three methods are presented here due to Gaver, Jacobs and Latouche [45], Grassmann, Taksar and Heyman [46], and Ye and Li [102]. All the three methods are based on direct methods relating to the Gaussian elimination.

- **Level Reduction**: This method is due to Gaver, Jacobs and Latouche [45]. It is based on Gaussian elimination and closely related to the block state reduction
which is due to Grassmann, Taksar and Heyman [46]. They are all based on
censored Markov chain idea in that a set of states is restricted – thereby the
name level reduction. We consider the level reduced DTMC associated with this
Markov chain at the $k^{th}$ stage, we can write

$$P_k = \begin{bmatrix}
C_k & U_k \\
D_{k+1} & A_{k+1} & U_{k+1} \\
\vdots & \vdots & \vdots \\
D_N & A_N
\end{bmatrix}, \quad 0 \leq k \leq N - 1,$$

(2.36)

where

$$C_0 = A_0, \quad C_k = A_k + D_k(I - C_{k-1})^{-1}U_{k-1}, \quad 1 \leq k \leq N.$$

The solution is now given as

$$\mathbf{x}_N = \mathbf{x}_NC_N,$$

(2.37)

$$\mathbf{x}_k = \mathbf{x}_{k+1}D_{k+1}(I - C_k)^{-1}, \quad 0 \leq k \leq N - 1,$$

(2.38)

and

$$\sum_{k=0}^{N} \mathbf{x}_k \mathbf{1} = 1.$$

The first equation is solved using one of the methods proposed earlier, if the
blocks of matrices are of dimension more than one.

- **Block State Reduction**: This method which is due to Grassmann, Taksar and
Heyman [46], is the reverse of the first method. We censor the Markov chain
from the bottom as follows

$$P_{N-k} = \begin{bmatrix}
A_0 & U_0 \\
D_1 & A_1 & U_1 \\
\vdots & \vdots & \vdots \\
D_{N-k} & E_{N-k}
\end{bmatrix}, \quad 0 \leq k \leq N - 1,$$

(2.39)

where

$$E_N = A_N, \quad E_{N-k} = A_{N-k} + U_{N-k}(I - A_{N-k+1})^{-1}D_{N-k+1}, \quad 1 \leq k \leq N.$$

The solution is now given as

$$\mathbf{x}_0 = \mathbf{x}_0E_0,$$

(2.40)

$$\mathbf{x}_{N-k} = \mathbf{x}_{N-k-1}U_{N-k-1}(I - E_{N-k})^{-1}, \quad 1 \leq k \leq N,$$

(2.41)

and

$$\sum_{k=0}^{N} \mathbf{x}_k \mathbf{1} = 1.$$
• **Folding Algorithm**: This method is related to the first two in that they all group the DTMC into two classes. The folding algorithm groups them into odd and even whereas the other two group the DTMC into one and others. It then rearranges the DTMC as \(\{0, 2, 4, 6, \ldots, N, 1, 3, 5, \ldots, N-1\}\), if \(N\) is even, as an example. The transition matrix is then partitioned accordingly leaving it in a good structure. For the details about this algorithm you are referred to Ye and Li [102].

### 2.9.5 Infinite State Markov Chains

Consider a DTMC \(\{X_n, n = 0, 1, 2, \cdots\}\), \(X_n = 0, 1, 2, \cdots\). Since \(X_n \in \mathcal{S}\) we say its an infinite DTMC. The finite case is when \(X_n \in \{0, 1, 2, \cdots, N < \infty\}\). If the infinite DTMC is positive recurrent we know that it has a stationary distribution \(\pi\) such that

\[
\pi = \pi P, \quad \pi 1 = 1.
\]

Even if we know that it has a stationary distribution we still face a challenge on how to compute this distribution since all the methods presented for finite DTMC can no longer be used because the state space is infinite.

One may suggest that we truncate the infinite DTMC at an appropriate state \(N < \infty\) to a finite DTMC and apply the earlier techniques. A truncation method could involve selecting the smallest \(N\) such that \(1 - \sum_{j=0}^{N} P_{ij} < \varepsilon\), \(\forall i\), where \(\varepsilon\) is a very small value, say \(10^{-12}\), for example. Such truncations could lead to problems at times. However for some special infinite DTMCs there are well established methods for determining the stationary distribution without truncation. In addition there are methods for establishing positive recurrency for such DTMCs. We will be presenting those classes of infinite DTMCs.

First we discuss the general case of infinite DTMC and present what is known about them. We then present the three classes of infinite DTMCs, called the GI/M/1 types, M/G/1 types and QBD types, which are numerically tractable and for which the results are well known. Since the results to be developed here are valid for both univariate and multivariate DTMCs, with only the first variable allowed to be unbounded, we will present results for the bivariate case.

### 2.9.6 Some Results for Infinite State Markov Chains with Repeating Structure

In this section, we introduce some associated measures and results related to Markov chains with infinite state block-structured transition matrices. Results presented here are for discrete time Markov chains. However, corresponding results for continuous time Markov chains can be obtained in parallel.
Let \( \{Z_n = (X_n, Y_n); n = 0, 1, 2, \ldots \} \), with \( X_n = 0, 1, 2, 3, \ldots; Y_n = 1, 2, \ldots, K_{X_n} < \infty \), be the Markov chain, whose transition matrix \( P \) is expressed in block matrix form:

\[
P = \begin{bmatrix}
P_{0,0} & P_{0,1} & P_{0,2} & \cdots & \cdots \\
P_{1,0} & P_{1,1} & P_{1,2} & \cdots & \cdots \\
P_{2,0} & P_{2,1} & P_{2,2} & \cdots & \cdots \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
\end{bmatrix},
\]  

(2.42)

where \( P_{i,j} \) is a matrix of size \( K_i \times K_j \) with both \( K_i < \infty \) and \( K_j < \infty \). In general, \( P \) is allowed to be substochastic. Let the state space be \( S \) and partitioned accordingly

\[
S = \bigcup_{i=0}^{\infty} L_i,
\]  

(2.43)

with

\[
L_i = \{(i, 1), (i, 2), \ldots, (i, K_i)\}.
\]  

(2.44)

In state \((i, r)\), \( i \) is called the level variable and \( r \) the state variable. We also use the notation

\[
L_{\leq i} = \bigcup_{k=0}^{i} L_k.
\]  

(2.45)

Partitioning the transition matrix \( P \) into blocks is not only done because it is convenient for comparison with results in the literature, but also because it is necessary when the Markov chain exhibits some kind of block structure. For the above Markov chain, we define matrices \( R_{i,j} \) for \( i < j \) and \( G_{i,j} \) for \( i > j \) as follows. \( R_{i,j} \) is a matrix of size \( K_i \times K_j \) whose \((r, s)\)th entry is the expected number of visits to state \((j, s)\) before hitting any state in \( L_{\leq (j-1)} \), given that the process starts in state \((i, r)\). \( G_{i,j} \) is a matrix of size \( K_i \times K_j \) whose \((r, s)\)th entry is the probability of hitting state \((j, s)\) when the process enters \( L_{\leq (i-1)} \) for the first time, given that the process starts in state \((i, r)\). We call matrices \( R_{i,j} \) and \( G_{i,j} \) respectively, the matrices of expected number of visits to higher levels before returning to lower levels and the matrices of the first passage probabilities to lower levels.

The significance of \( R_{i,j} \) and \( G_{i,j} \) in studying Markov chains, especially for ergodic Markov chains, has been pointed out in many research papers, including Zhao, Li and Alfa [107], Grassmann and Heyman [47], Grassmann and Heyman [48], and Heyman [55]). In these papers, under the ergodic condition, stationary distribution vectors, the mean first passage times and the fundamental matrix are discussed in terms of these matrices. For some recent results, see Zhao, Li and Alfa [108], Zhao, Li and Braun [107].

For special type of Markov chains in block form that have the property of repeating rows (or columns), which means that the transition probability matrix partitioned as in [108] has the following form:
2. Let $P$ be stochastic.

- Let $R = \sum_{i=1}^{\infty} R_i$. The Markov chain $P$ in (2.46) is positive recurrent if and only if $R < \infty$ and $\lim_{k \to \infty} R_k = 0$.
- Let $G = \sum_{i=1}^{\infty} G_i$. If $\sum_{k=1}^{\infty} A_k$ is stochastic, then the Markov chain $P$ in (2.46) is recurrent if and only if $G$ is stochastic.

2. Let $P$ be stochastic. If $\sum_{k=1}^{\infty} kP_{0,k}e < \infty$ and if $A = \sum_{k=1}^{\infty} A_k$ is stochastic and irreducible with the stationary distribution $\pi$, then

- $P$ is positive recurrent if and only if $\pi R < \pi$;
- $P$ is null recurrent if and only if $\pi R = \pi$ and $Ge = e$; and
- $P$ is transient if and only if $Ge < e$.

3. If $\sum_{k=1}^{\infty} kP_{0,k}e < \infty$ then $P$ is positive recurrent if and only if $sp(R) < 1$;

- $P$ is transient if and only if $sp(G) < 1$; and
- $P$ is null recurrent if and only if $sp(R) = sp(G) = 1$.

These are some of the general results. In what follows we focus on specific cases of infinite DTMCs.

### 2.9.7 Matrix-Analytical Method for Infinite State Markov Chains

The matrix-analytical method (MAM) is most suited to three classes of Markov chains, viz: i) those with the GI/M/1 structure ii) those with the M/G/1 structure, and those with the QBD structure which actually embodies the combined properties of both the GI/M1 and M/G/1 types structures. Before we go into the details, let us define the state space which is partially general for our discussions.

Consider a bivariate Markov chain $\{X_n, Y_n; n \geq 0\}$ such that $P_{i,j;v,k} = \Pr\{X_{n+1} = i, Y_{n+1} = j | X_n = v, Y_n = k\}$; $i, v \geq 0, 1 \leq j, k \leq M$ and $i$ and $v$ are referred to as the levels $i$ and phase $v$, $(>0)$, respectively. Further define $P^{(m)}_{i,j;v,k} = \Pr\{X_{n+m} = i, Y_{n+m} = j | X_n = v, Y_n = k\}$ such that $X_{n+w} \neq i, w = \ldots$
1, 2, ⋯, m. This probability \( P_{i,j,v,k}^{(m)} \) is the taboo probability that given the DTMC starts from state \((i,j)\) it visits state \((v,k)\) at time \(m\) without visiting level \(i\) during that period.

2.9.7.1 The GI/M/1 Type

Let the transition matrix \( P \) be given in the block partitioned form as follows:

\[
P = \begin{bmatrix}
  B_{00} & B_{01} & & \\
  B_{10} & B_{11} & A_0 & \\
  B_{20} & B_{21} & A_1 & A_0 & \\
  & & & & \\
  B_{30} & B_{31} & A_2 & A_1 & A_0 & \\
  & & & & & & & \\
  & & & & & & & & \\
  & & & & & & & & & \cdots \cdots \cdots \cdots \cdots \\
\end{bmatrix}
\]  

(2.47)

The blocks \( A_{i-j+1} \) refer to transitions from level \(i\) to \(j\) for \(i \geq 1, j \geq 2, i \geq j - 1\). The blocks \( B_{ij} \) refer to transitions in the boundary areas from level \(i\) to level \(j\) for \(i \geq 0, j = 0, 1\). This transition matrix is of the GI/M/1 type and very well treated in Neuts [80]. It is called the GI/M/1 type because it has the same structure as the transition matrix of the GI/M/1 queue embedded at arrival points. It is skip-free to the right. This system will be discussed at later stages. It suffices at this point just to accept the name assigned to this DTMC.

We consider only the case for which the matrix \( P \) is irreducible and positive recurrent. Neuts [80] shows that for the matrix \( P \) to be positive recurrent, we need the condition that

\[
\pi \nu > 1
\]

(2.48)

where \( \pi = \pi A, \pi 1 = 1, A = \sum_{i=0}^{\infty} A_i \) and \( \nu = \sum_{k=0}^{\infty} k A_i 1 \).

We define \( \mathbf{x} = [x_0, x_1, x_2, \ldots] \) as the invariant probability vector associated with \( P \), then for \( \mathbf{x} = \mathbf{x} P \), we have

\[
x_0 = \sum_{i=0}^{\infty} x_i B_{i0},
\]

(2.49)

\[
x_1 = \sum_{i=0}^{\infty} x_i B_{i1},
\]

(2.50)

\[
x_j = \sum_{i=0}^{\infty} x_{j+v-1} A_v, \quad j \geq 2
\]

(2.51)

Let us define a matrix \( R \) whose elements \( R_{i,j} \) are given as

\[
R_{j,w} = \sum_{n=0}^{\infty} i P_{i,j,i+1,w}^{(n)}, \quad i \geq 0, 1 \leq j \leq m, 1 \leq w \leq m.
\]
The element $R_{j,w}$ is the expected number of visits to state $(i+1,w)$ before returning to level $i$, given that the DTMC started in state $(i,j)$. If the matrix $P$ satisfies (2.48) then there exists a non-negative matrix $R$ which has all its eigenvalues in the unit disk for which the probabilities $x_{i+1}$ can be obtained recursively as

$$x_{i+1} = x_i R, \quad x_{i+1} = x_1 R^i, \quad i \geq 1. \quad (2.53)$$

This is the matrix analogue of the scalar-geometric solution of the GI/M/1 queue and that is why it is called the matrix-geometric solution. For details and rigorous proof of this result see Neuts [80]. We present a skeletal proof only in this book. Following Neuts [80] closely, the derivation of this result can be summarized as follows:

$$P_{i+1,j,i+1,j}^{(n)} = iP_{i+1,j,i+1,j}^{(n)} + \sum_{v=1}^{M} \sum_{r=0}^{n} P_{i+1,j,i+1,v}^{(r)} \times P_{i+1,j,i+1,j}^{(n-r)}, \quad \forall n \geq 1. \quad (2.54)$$

As shown in Neuts [80], if we sum this equation from $n = 1$ to $n = N$ and divide by $N$ and then let $N \to \infty$, we have

$$\frac{1}{N} \sum_{n=1}^{N} P_{i+1,j,i+1,j}^{(n)} \bigg|_{N \to \infty} \to x_{i+1,j}$$

$$\frac{1}{N} \sum_{n=1}^{N} P_{i+1,j,i+1,j}^{(n)} \bigg|_{N \to \infty} \to 0$$

$$\frac{1}{N} \sum_{n=1}^{N} \sum_{r=0}^{n} P_{i+1,j,i+1,v}^{(r)} \bigg|_{N \to \infty} \to x_{i,v}$$

and

$$\sum_{n=1}^{N} P_{i+1,j,i+1,j}^{(n)} \bigg|_{N \to \infty} \to R_{v,j}$$

Hence,

$$\sum_{v=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \sum_{r=0}^{n} P_{i+1,j,i+1,v}^{(r)} \bigg|_{N \to \infty} \to \sum_{v=1}^{M} x_{i,v} R_{v,j}$$

Neuts [80] also showed that

$$(R^q)_{j,w} = R_{j,w}^{(q)} = \sum_{n=0}^{\infty} P_{i,j,i+q,w}^{(n)} \quad (2.55)$$

and we use this result in what follows.

The matrix $R$ is the minimum non-negative solution to the matrix polynomial equation
\[ R = \sum_{k=0}^{\infty} R^k A_k. \]  

(2.56)

The arguments leading to this are as follows. Consider the equation for $x_j$ given as

\[ x_j = \sum_{v=0}^{\infty} x_{j+v-1} A_v, \quad j \geq 2 \]

Now if we replace $x_{j+v}$ with $x_{j-1} R^{v+1}$, $v > 0$, $j > 2$, we obtain

\[ x_{j-1} R = \sum_{k=0}^{\infty} x_{j-1} R^k A_k, \quad j \geq 2 \]  

(2.57)

After some algebraic manipulations, we obtain Equation (2.56).

The matrix $R$ is known as the rate matrix. The elements $R_{ij}$ of the matrix $R$ is the expected number of visits into state $(k+1, j)$, starting from state $(k, i)$, until the first return to state $(k, \cdot)$, $k > 1$. The results quoted in (2.54 and 2.55) are based on taboo probabilities and a detailed presentation of the derivations can be found in Neuts [80].

The boundary probabilities $(x_0, x_1)$ are determined as

\[ x_0 = \sum_{i=0}^{\infty} x_i B_{i0} = x_0 B_{00} + \sum_{i=1}^{\infty} x_i R^{i-1} B_{i0} \]  

(2.58)

and

\[ x_1 = \sum_{i=0}^{\infty} x_i B_{i1} = x_0 B_{01} + \sum_{i=1}^{\infty} x_i R^{i-1} B_{i1} \]  

(2.59)

When rearranged in matrix form, we obtain

\[ (x_0 \quad x_1) = (x_0, x_1) B[R] \]  

(2.60)

where

\[ B[R] = \left[ \begin{array}{cc} B_{00} & B_{01} \\ \sum_{n=1}^{\infty} R^{n-1} B_{n0} & \sum_{n=1}^{\infty} R^{n-1} B_{n1} \end{array} \right] \]  

(2.61)

and then $(x_0, x_1)$ is normalized by

\[ x_0 \mathbf{1} + x_1 (I - R)^{-1} \mathbf{1} = 1 \]  

(2.62)

The argument behind this is that

\[ x_0 \mathbf{1} + x_1 \mathbf{1} + x_2 \mathbf{1} + \ldots = x_0 \mathbf{1} + x_1 (I + R + R^2 + \ldots) \mathbf{1} = 1 \]

Because of the geometric nature of Equation (2.53), this method is popularly known as the matrix - geometric method.
2.9.7.2 Key Measures

The key measures usually of interest in this DTMC are

- The marginal distribution of the first variable (level) is given as
  \[ y_k = \mathbf{x}_k \mathbf{1} = \mathbf{x}_1 R^{k-1} \mathbf{1}, \quad k \geq 1. \]  
  (2.63)

- The first moment is given as
  \[ E[X] = \mathbf{x}_1 (I - R)^{-2} \mathbf{1}. \]  
  (2.64)

- Tail behaviour, i.e. \( p_k = Pr\{X \geq k\} = \sum_{v=k}^{\infty} \mathbf{x}_v \mathbf{1} = \mathbf{x}_1 R^{k-1} [I - R]^{-1} \mathbf{1}. \) Let \( \eta = sp(R) \), i.e. the spectral radius of \( R \) (i.e. its maximum absolute eigenvalue), and let \( \mathbf{v} \) and \( \mathbf{u} \) be the corresponding left and right eigenvectors which are normalized such that \( \mathbf{u} \mathbf{1} = 1 \) and \( \mathbf{u} \mathbf{v} = 1 \). It is known that \( R^k = \eta^k \mathbf{v} \mathbf{u} + o(\eta^k), \quad k \to \infty. \) Hence \( \mathbf{x}_j = \mathbf{x}_1 R^{j-1} \mathbf{1} = \kappa \eta^{j-1} + o(\eta^j), \quad j \to \infty, \) where \( \kappa = \mathbf{x}_1 \mathbf{v}. \) Hence we have
  \[ p_k = \kappa \eta^{k-1} (1 - \eta)^{-1} + o(\eta^k), \quad k \to \infty. \]  
  (2.65)

The key ingredient to the matrix-geometric method is the matrix \( R \) and how to compute it efficiently becomes very important especially when it has a huge dimension and its spectral radius is close to 1.

2.9.7.3 Computing matrix \( R \)

There are several methods for computing the matrix \( R \) and they are all iterative. Letting \( R(v) \) be the computed value of matrix \( R \) at the \( v^{th} \) iteration, the computation is stopped once \( |R(v+1) - R(v)|_{i,j} < \varepsilon \), where \( \varepsilon \) is a small convergent criterion with values of about \( 10^{-12} \). The simplest but not necessarily most efficient methods are the linear ones, given as:

- Given that
  \[ R = \sum_{k=0}^{\infty} R^k A_k \]
  we can write
  \[ R(v+1) = \sum_{k=0}^{\infty} R^k(v) A_k, \]  
  (2.66)
  with \( R(0) := A_0. \)

- Given that
  \[ R = \left( \sum_{k=0, k \neq 1}^{\infty} R^k A_k \right) (I - A_1)^{-1} \]
  we can write
\[ R(v+1) = \left( \sum_{k=0,k\neq 1}^{\infty} R^k(v+1)A_k \right)(I-A_1)^{-1}, \quad (2.67) \]

with \( R(0) := A_0 \).

- Given that
  \[ R = A_0[I - \sum_{k=1}^{\infty} R^{k-1}A_k]^{-1} \]
  
we can write
  \[ R(v+1) = A_0[I - \sum_{k=1}^{\infty} R^{k-1}(v)A_k]^{-1} \quad (2.68) \]

with \( R(0) := A_0 \).

More efficient methods are the cyclic reduction method by Bini and Meini [22] and the method using the non-linear programming approach by Alfa, et al. [15]. These methods are more efficient than the linear approach however, they cannot take advantage of the structures of the matrices \( A_k \), \( k = 0, 1, 2, \ldots \), when they can be exploited.

- **NON-LINEAR PROGRAMMING APPROACH FOR R:** By definition, \( R \) is the minimal non-negative solution to the matrix polynomial equation \( R = \sum_{k=0}^{\infty} R^k A_k \). Consider the matrix \( A^*(z) = \sum_{k=0}^{\infty} A_k z^k \), \( |z| \leq 1 \), and let \( \chi^*(z) \) be the eigenvalue of \( A^*(z) \). It is known from Neuts [80] that the Perron eigenvalue of \( R \) is the smallest solution to \( z = \chi^*(z) \) and that \( u \) is its left eigenvector. Let \( X \) be any non-negative square matrix of order \( m \) and a function \( f(X) = \sum_{k=0}^{\infty} X^k A_k \). For two square matrices \( Y \) and \( Z \) of the same dimensions, let \( Y \circ Z \) denote their elementwise product. Now we define a function

\[ H(X) = \sum_{(i,j)=1}^{m} \left( [f(X)]_{i,j} - X_{i,j} \right)^2 = \mathbf{1}^T ((f(X) - X) \circ (f(X) - X)) \mathbf{1}. \quad (2.69) \]

It was shown in Alfa et al. [15] that if the matrix \( P \) is positive recurrent then the matrix \( R \) is the unique optimal solution to the non-linear programming problem

\[ \text{minimize} \quad H(X) \quad (2.70) \]

subject to

\[ uX = \eta u, \quad (2.71) \]

\[ X \geq 0. \quad (2.72) \]

Two optimization algorithms were presented by Alfa, et al. [15] for solving this problem.

- **THE INVARIANT SUBSPACE METHOD FOR R:** Akar and Sohraby [1] developed this method for \( R \) matrix. The method requires that \( A^*(z) = \sum_{k=0}^{\infty} z^k A_k \) be rational. However, whenever we have \( A_k = 0 \), \( K < \infty \), this condition is usually met. For most practical situations \( A_k = 0 \), \( K < \infty \), hence this method is appropriate in such situations.
2.9.7.4 Some Special Structures of the Matrix $R$ often Encountered

The matrix-geometric method is a very convenient method to work with, and also because it has a probabilistic interpretation that makes it more meaningful with certain appeal to application oriented users. Its weakness however, is that as the size of the matrix $R$ gets very large, the computational aspects become cumbersome. In that respect, it is usually wise to search for special structures of matrix $R$ that can be exploited. Some of the very useful special structures are as follows:

- If $A_0$ is of rank one, i.e. $A_0 = \omega \beta$, then

$$R = \omega \xi = A_0 [I - \sum_{v=1}^{\infty} \eta^{-1}A_v]^{-1},$$  

(2.73)

where $\eta = \xi \omega$ is the maximal eigenvalue of the matrix $R$.

The argument leading to this are based on the iteration

$$R(0) := 0,$$

$$R(k + 1) := A_0 (I - A_1)^{-1} + \sum_{v=2}^{\infty} R^v(k) A_v (I - A_1)^{-1}, \quad k \geq 0,$$

and letting $R(k) = \omega \xi(k)$ we have

$$\xi(1) = \beta(I - A_1)^{-1},$$

$$\xi(k) = \beta(I - A_1)^{-1} + \sum_{v=2}^{\infty} (\xi(k - 1) \omega)^{-1} \xi(k - 1) A_v (I - A_1)^{-1}, \quad k \geq 0.$$  

This leads to the explicit result given above.

- For every row of $A_0$ that is all zeros, the corresponding rows of $R$ are also zeros.
- When the matrices $A_i$, $i \geq 0$, are of the sparse block types, it is advantageous to write out the equations of $R$ in smaller blocks in order to reduce the computations.
- Other special structures can be found in Chakravarthy and Alfa [29], Alfa and Chakravarthy [7] and Alfa, Dolhun and Chakravarthy [8].

2.9.7.5 The M/G/1 Type:

Let the transition matrix $P$ be given in the block partitioned form as follows:

$$P = \begin{bmatrix}
B_0 & B_1 & B_2 & B_3 & \cdots \\
A_0 & A_1 & A_2 & A_3 & \cdots \\
A_0 & A_1 & A_2 & \cdots \\
& A_0 & A_1 & \cdots \\
& & & \vdots & \ddots
\end{bmatrix}.$$  

(2.74)
The blocks $A_{j-i+1}$ refer to transitions from level $i$ to level $j$ for $j \geq i - 1$. The blocks $B_j$ refer to transitions at the boundary area from level 0 to level $j$. It is skip-free to the left.

The transition matrix is of the M/G/1 type and very well treated in Neuts [81].

Let $A = \sum_{k=0}^{\infty} A_k$ and let $\nu = \sum_{k=1}^{\infty} kA_k I$ be finite. If $A$ is irreducible, then there exists an invariant vector $\pi$ such that $\pi = \pi A$, and $\pi I = 1$. The Markov chain $P$ is irreducible if and only if $\rho = \pi \nu < 1$.

If the Markov chain $P$ is irreducible, then there exists a stochastic matrix $G$ which is the minimal non-negative solution to the matrix polynomial equation

$$G = \sum_{k=0}^{\infty} A_k G^k$$

The elements $G_{ij}$ of this matrix refers to the probability that the system will eventually visit state $(v+1, j)$ given that it started from state $(v, i)$, $(v > 1)$. The arguments that lead to Equation (2.75) will be presented in a skeletal form and the rigorous proof can be found in Neuts [81].

It is clear that starting from level $i+1$ the DTMC can reach level $i$ in one step according to the matrix $A_0$, or in two steps according to $A_1 G$, i.e. it stays in level $i+1$ in the first step and then goes to level $i$ in the second step; or in three steps according to $A_2 G^2$, i.e. it goes up to level $i+2$ in the first step, comes down two consecutive times in the second and third steps according to $G \times G = G^2$; and this continues. Hence

$$G = A_0 + A_1 G + A_2 G^2 + \cdots = \sum_{v=1}^{\infty} A_v G^v.$$ 

It was shown in Neuts [81] in section 2.2.1 that

- If $P$ is irreducible, $G$ has no zero rows and it has eigenvalue $\eta$ of maximum modulus.
- If $P$ is recurrent $G$ is stochastic.

### 2.9.7.6 Stationary distribution

In order to obtain the stationary distribution for the M/G/1 system we proceed as follows.

Let $g = gG$, where $gI = 1$. Further define $M$ such that

$$M = G + \sum_{v=1}^{\infty} A_v \sum_{k=0}^{v-1} G^k M G^{v-k-1}$$

and also define $u$ such that

$$u = (I - G + 1g)[I - A - (1 - \beta)g]^{-1}1$$
Further define $K$ such that
\[ K = B_0 + \sum_{v=1}^{\infty} B_v G^v \] (2.78)
and let $\kappa = \kappa K$, where $\kappa 1 = 1$. The $(i, j)$ elements of the matrix $K$ refer to the probability that the system will ever return to state $(0, j)$ given that is started from state $(0, i)$. Note that we say “return” because we are considering the same level 0 in both cases, so we mean a return to level 0.

If we now define the vector $x = [x_0, x_1, x_2, \ldots]$ then we can obtain $x_0$ as
\[ x_0 = (\kappa \tilde{\kappa})^{-1} \kappa \] (2.79)
where
\[ \tilde{\kappa} = 1 + \sum_{v=1}^{\infty} B_v \sum_{k=0}^{v-1} G^k u \]

Once $x_0$ is obtained the remaining $x_n, n > 1$, are obtained as follows (see Ramaswami [89]):
\[ x_n = \begin{bmatrix} x_0 & B_n + (1 - \delta_{n,1}) \sum_{j=1}^{n-1} x_j A_{n-j+1} \end{bmatrix} (I - A_1), \quad n \geq 1 \] (2.80)
where
\[ B_v = \sum_{i=v}^{\infty} B_i G^{i-v} \text{ and } A_v = \sum_{i=v}^{\infty} A_i G^{i-v}, \quad v \geq 0 \]
The key ingredient to using this method is the matrix $G$. Just like its GI/M/1 counterpart the effort involved in its computation could be enormous if its size is huge and traffic intensity is close to 1.

2.9.7.7 Computing Matrix $G$

There are several methods for computing the matrix $G$ and they are all iterative. Letting $G(v)$ be the computed value of matrix $G$ at the $v$th iteration, the computation is stopped once $|G(v+1) - G(v)|_{i,j} < \varepsilon$, where $\varepsilon$ is a small convergent criterion with values of about $10^{-12}$. The simplest but not necessarily most efficient methods are the linear ones, given as:

- Given that
\[ G = \sum_{k=0}^{\infty} A_k G^k \]
we can write
\[ G(v+1) = \sum_{k=0}^{\infty} A_k G^k(v)A_k, \] (2.81)
with $G(0) := 0$. It has been shown that starting with a $G(0) = I$ sometimes works better.

- Given that
  \[
  G = (I - A_1)^{-1} \left( \sum_{k=0, k \neq 1}^{\infty} A_k G^k \right)
  \]
  we can write
  \[
  G(v + 1) = (I - A_1)^{-1} \left( \sum_{k=0, k \neq 1}^{\infty} A_k G^k (v + 1) \right),
  \tag{2.82}
  \]
  with $G(0) := 0$.

- Given that
  \[
  G = [I - \sum_{k=1}^{\infty} A_k G^{k-1}]^{-1} A_0
  \]
  we can write
  \[
  G(v + 1) = [I - \sum_{k=1}^{\infty} A_k G^{k-1}(v)]^{-1} A_0
  \tag{2.83}
  \]
  with $G(0) := 0$.

More efficient methods are the cyclic reduction method by Bini and Meini [22], subspace method by Akar and Soharby [1] and the method using the non-linear programming approach by Alfa, Sengupta and Takine [16]. These methods are more efficient than the linear approach however, they cannot take advantage of the structures of the matrices $A_k$, $k = 0, 1, 2, \ldots$, when they can be exploited.

- **Cyclic Reduction for $G$**: The cyclic reduction (CR) method developed by Bini and Meini [22] capitalizes on the structure of the matrix $P$ in trying to compute $G$. Note that the matrix equation for $G$, i.e. $G = \sum_{k=0}^{\infty} A_k G^k$ can be written as $G^v = \sum_{k=0}^{\infty} A_k G^{k+v}$, $v \geq 0$, or in matrix form as

  \[
  \begin{bmatrix}
  I - A_1 & -A_2 & -A_3 & \cdots \\
  -A_0 & I - A_1 & -A_2 & \cdots \\
  -A_0 & -A_0 & I - A_1 & \cdots \\
  \vdots & \vdots & \vdots & \ddots 
  \end{bmatrix}
  \begin{bmatrix}
  G \\
  G^2 \\
  G^3 \\
  \vdots
  \end{bmatrix}
  =
  \begin{bmatrix}
  A_0 \\
  0 \\
  0 \\
  \vdots
  \end{bmatrix}
  \tag{2.84}
  \]

By using the odd-even permutation idea, used by Ye and Li [102] for the folding algorithm together with an idea similar to the Logarithmic Reduction for QBDs (to be presented later), on the block matrices we can write this matrix equation as

\[
\begin{bmatrix}
I - U_{1,1} & -U_{1,2} \\
-U_{2,1} & I - U_{2,2}
\end{bmatrix}
\begin{bmatrix}
V_0 \\
V_1
\end{bmatrix}
= \begin{bmatrix}
0 \\
B
\end{bmatrix},
\]

where

\[
U_{1,1} = U_{2,2} = \begin{bmatrix}
I - A_1 & -A_3 & \cdots \\
I - A_1 & \cdots \\
\vdots & \ddots
\end{bmatrix},
\]
\[ U_{1,2} = \begin{bmatrix} -A_0 & -A_2 & \cdots \\ -A_2 & -A_4 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \]

\[ U_{2,1} = \begin{bmatrix} -A_2 & -A_4 & \cdots \\ -A_0 & -A_2 & \cdots \\ 0 & \cdots & \ddots \end{bmatrix}, \quad B = \begin{bmatrix} A_0 \\ 0 \\ \vdots \end{bmatrix}, \]

\[ V_0 = G^{2k}, \quad V_1 = G^{2k+1}, \quad k = 1, 2, \ldots. \]

By applying standard block gaussian elimination to the above equation we obtain

\[ [I - U_{2,2} - U_{2,1} (I - U_{1,1})^{-1} U_{1,2}] V_1 = B. \quad (2.85) \]

By noticing that this equation (2.85) is also of the form

\[ \begin{bmatrix} I & -A_1^{(1)} & -A_2^{(1)} & -A_3^{(1)} & \cdots \\ -A_0^{(1)} & I & -A_2^{(1)} & -A_3^{(1)} & \cdots \\ -A_0^{(1)} & -A_1^{(1)} & I & \cdots & \vdots \\ -A_0^{(1)} & \cdots & -A_0^{(1)} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & I \\ \end{bmatrix} \begin{bmatrix} G \\ G^3 \\ G^5 \\ G^7 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \]

Bini and Meini [22] showed that after the \( n \)th repeated application of this operation we have

\[ \begin{bmatrix} I - U_{1,1}^{(n)} & -U_{1,2}^{(n)} \\ -U_{2,1}^{(n)} & I - U_{2,2}^{(n)} \end{bmatrix} \begin{bmatrix} V_0^{(n)} \\ V_1^{(n)} \end{bmatrix} = \begin{bmatrix} 0 \\ B \end{bmatrix}, \]

where

\[ U_{1,1}^{(n)} = U_{2,2}^{(n)} = \begin{bmatrix} I - A_1^{(n)} & -A_3^{(n)} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \]

\[ U_{1,2}^{(n)} = \begin{bmatrix} -A_0^{(n)} & -A_2^{(n)} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \]

\[ U_{2,1}^{(n)} = \begin{bmatrix} -A_2^{(n)} & -A_4^{(n)} & \cdots \\ -A_0^{(n)} & -A_2^{(n)} & \cdots \\ 0 & \cdots & \ddots \end{bmatrix}, \quad B = \begin{bmatrix} A_0 \\ 0 \\ \vdots \end{bmatrix}, \]

\[ V_0 = G^{2k.2^n}, \quad V_1 = G^{2k.2^n+1}, \quad k = 1, 2, \ldots, n \geq 0. \]

In the end they show that the matrix equation is also of the form
\[ [I - U^{(n)}_{2,2} - U^{(n)}_{2,1} (I - U^{(n)}_{1,1})^{-1} U^{(n)}_{1,2}]_1 = B, \]

which results in

\[ G = (I - A^{(n)}_1)^{-1} (A_0 + \sum_{k=1}^{\infty} A^{(n)}_k G^{2k+1}). \quad (2.86) \]

They further showed that as \( n \to \infty \) the second term in the brackets tends to zero, and \((I - A^{(n)}_1)^{-1}\) exists, hence

\[ G = (I - A^{(\infty)}_1)^{-1} A_0. \quad (2.87) \]

Hence \( G \) can be obtained from this expression. Of course the operation can only be applied in a finite number of times. So it is a matter of selecting how many operations to carry out given the tolerance desired.

- **THE INVARIANT SUBSPACE METHOD FOR \( G \):** Akar and Sohraby [1] developed this method for \( G \) matrix. The method requires that \( A^*(z) = \sum_{k=0}^{\infty} z^k A_k \) be rational. However, whenever we have \( A_k = 0, \ K < \infty \), this condition is usually met. For most practical situations \( A_k = 0, \ K < \infty \), hence this method is appropriate in such situations.

- **NON-LINEAR PROGRAMMING APPROACH FOR \( G \):** The non-linear programming for the matrix \( G \) is similar in concept to that used for the matrix \( R \), with some minor differences. Consider an \( m \times m \) matrix of non-negative real values such that \( g(X) = \sum_{k=0}^{\infty} A_k X^k \), where \( g : \mathbb{R}^m \to \mathbb{R}^m \). Further define a function \( \phi(X) \) such that \( \phi : \mathbb{R}^m \to \mathbb{R}^1 \) and it is order preserving, i.e. \( X < Y \) implies \( \phi(X) < \phi(Y) \) where both \( (X, Y) \in \mathbb{R}^m \). A good example is \( \phi(X) = 1^T X 1 \). It was shown by Alfa, et al. [16] that the matrix \( G \) is the unique and optimal solution to the non-linear programming problem

\[
\begin{align*}
\text{minimize} & \quad \phi(X) \quad (2.88) \\
\text{subject to} & \quad g(X) - X \leq 0, \quad (2.89) \\
& \quad X \geq 0. \quad (2.90)
\end{align*}
\]

Several algorithms were proposed by Alfa, et al. [16] for solving this problem.

### 2.9.7.8 Some Special Structures of the Matrix \( G \) often Encountered

The matrix-geometric method is a very convenient method to work with, and also because it has a probabilistic interpretation that makes it more meaningful with certain appeal to application oriented users. Its weakness is that as the size of the matrix \( G \) gets very large, the computational aspects become cumbersome. In that respect, it is usually wise to search for special structures of matrix \( G \) that can be exploited. Some of the very useful special structures are as follows:

- If \( A_0 \) is of rank one, i.e. \( A_0 = v \alpha \), with \( \alpha 1 = 1 \), then
2.9 Numerical Computations of the Invariant Vectors 65

\[ G = 1\alpha, \]  

(2.91)

if the Markov chain with transition matrix \( P \) is positive recurrent. This is obtained through the following arguments. Note that

\[ G(1) := A_0, \]

\[ G(k + 1) := \sum_{\ell=0}^{\infty} A_\ell G(k)^{\ell}, \quad k \geq 1. \]

If we write \( G(k) = \gamma(k)\alpha, \quad k \geq 1 \), where

\[ \gamma(1) = v, \]

\[ \gamma(k + 1) = \sum_{\ell=0}^{\infty} A_\ell (\alpha \gamma(k - 1))^{\ell-1} \gamma(k - 1). \]

Hence we have \( G = \gamma\alpha \), and since the Markov chain is positive recurrent we have \( G1 = 1 \), implying that \( \gamma = 1 \) and hence

\[ G = 1\alpha. \]

- For every column of \( A_0 \) that is all zeros, the corresponding columns of \( G \) are also all zeros.
- When the matrices \( A_i, i \geq 0 \), are of the sparse block types, it is advantageous to write out the equations of \( G \) in smaller blocks in order to reduce the computations.
- Other special structures can be found in Chakravarthy and Alfa [29], Alfa and Chakravarthy [7] and Alfa, Dolhun and Chakravarthy [8].

2.9.7.9 QBD

The QBD (Quasi-Birth-and-Death) DTMC is the most commonly encountered DTMC in discrete time queues. It embodies the properties of both the GI/M/1 and M/G/1 types of DTMCs. This infinite DTMC will be dealt with in more details because of its importance in queueing theory.

Let the transition matrix \( P \) be given in the block partitioned form as follows:

\[
P = \begin{bmatrix}
  B & C \\
  E A_1 A_0 \\
  A_2 A_1 A_0 \\
  A_2 A_1 A_0 \\
  \cdots & \cdots & \cdots 
\end{bmatrix}.
\]  

(2.92)
The QBD only goes a maximum of one level up or down. It is skip-free to the left and to the right. These are very useful properties which are fully exploited in the theory of QBDs. For detailed treatment of this see Latouche and Ramaswami [67].

We assume that the matrices $A_k$, $k = 0, 1, 2$ are of dimension $n \times n$ and matrix $B$ is of dimension $m \times m$, hence $C$ and $E$ are of dimensions $m \times n$ and $n \times m$, respectively. The key matrices that form the ingredients for analyzing a QBD are the $R$ and $G$ matrices, which are given as

$$R = A_0 + RA_1 + R^2 A_2$$  \hspace{1cm} (2.93)

and

$$G = A_2 + A_1 G + A_0 G^2.$$  \hspace{1cm} (2.94)

$R$ and $G$ are the minimal non-negative solutions to the first and second equation, respectively. Another matrix that is of importance is the matrix $U$, which records the probability of visiting level $i$ before level $i-1$, given the DTMC started from level $i$. The matrix $U$ is given as

$$U = A_1 + A_0 (I - U)^{-1} A_2.$$  \hspace{1cm} (2.95)

There are known relationships between the three matrices $R$, $G$ and $U$, and are given as follows

$$R = A_0 (I - U)^{-1},$$  \hspace{1cm} (2.96)

$$G = (I - U)^{-1} A_2,$$  \hspace{1cm} (2.97)

$$U = A_1 + A_0 G,$$  \hspace{1cm} (2.98)

$$U = A_1 + RA_2.$$  \hspace{1cm} (2.99)

Using these known relationships we have

$$R = A_0 (I - A_1 - A_0 G)^{-1},$$  \hspace{1cm} (2.100)

$$G = (I - A_1 - RA_2)^{-1} A_2.$$  \hspace{1cm} (2.101)

These last two equations are very useful for obtaining $R$ from $G$ and vice-versa, especially when it is easier to compute one of them when our interest is in the other.

In what follows we present a skeletal development of the relationship between the vector $x_{i+1}$ and $x_i$ through the matrix $R$ using a different approach from the one used for the GI/M/1 system. The method is based on censoring used differently by Latouche and Ramaswami [67] and Grassmann and Heyman [48]. Consider the matrix $P$ above. It can be partitioned as follows

$$P = \begin{bmatrix} T_L & T_R \\ B_L & B_R \end{bmatrix}.$$  \hspace{1cm} (2.102)

Suppose we select $n$ from $0, 1, 2, \cdots$ as the point of partitioning. Then the vector $x$ is also partitioned into $x = [x_T \ x_B]$, with
\[ \mathbf{x}_T = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_B = [\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \ldots]. \]

The result is that we have
\[ T_R = \begin{bmatrix}
0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & \cdots \\
A_0 & 0 & 0 & \cdots
\end{bmatrix}, \quad (2.103) \]
and
\[ B_R = \begin{bmatrix}
A_1 & A_0 \\
A_2 & A_1 & A_0 \\
& & \ddots & \ddots & \ddots
\end{bmatrix}. \quad (2.104) \]

Keeping in mind that \( \mathbf{x} = \mathbf{x}P \), we can write
\[ \mathbf{x}_T = \mathbf{x}_T T_L + \mathbf{x}_B B_L \]
\[ \mathbf{x}_B = \mathbf{x}_T T_R + \mathbf{x}_B B_R. \]
Equation (2.106) can be written as
\[ \mathbf{x}_B = \mathbf{x}_T T_R (I - B_R)^{-1}, \]
which can be further written as
\[ [\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \ldots] = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n] T_R (I - B_R)^{-1}. \]

By the structure of \( T_R \) we know that the structure of \( T_R (I - B_R)^{-1} \) is as follows
\[ T_R (I - B_R)^{-1} = \begin{bmatrix}
0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & \cdots \\
A_0 V_{11} & A_0 V_{12} & A_0 V_{13} & \cdots
\end{bmatrix}, \quad (2.108) \]
where \( V_{ij} \) are the block elements of \( (I - B_R)^{-1} \). It is clear that
\[ \mathbf{x}_{n+1} = \mathbf{x}_n A_0 V_{11} = \mathbf{x}_n R, \]
where \( R \) is by definition equivalent to \( A_0 V_{11} \). This is another way to see the matrix geometric result for the QBD. See Latouche and Ramaswami [67] for details.

Similar to the GI/M/1 type, if the QBD DTMC is positive recurrent then the matrix \( R \) has a spectral radius less than 1 and also the matrix \( G \) is stochastic just like the case for the M/G/1 type. Once we know \( R \) we can use the result \( \mathbf{x}_{i+1} = \mathbf{x}_i R, \ i \geq 1 \). However, we still need to compute \( R \) and determine the boundary behaviour of the matrix \( P \). First we discuss the boundary behaviour.
The censored stochastic transition matrix representing the boundary is \( B[R] \) and can be written as
\[
B[R] = \begin{bmatrix} B & C \\ E & A_1 + RA_2 \end{bmatrix}.
\]

Using this matrix we obtain
\[
\mathbf{x}_1 = \mathbf{x}_1 [E(I - B)^{-1}C + A_1 + RA_2],
\]
which after getting solved for \( \mathbf{x}_1 \) we can then solve for \( \mathbf{x}_0 \) as \( \mathbf{x}_0 = \mathbf{x}_1 E(I - B)^{-1} \), and then normalized through
\[
\mathbf{x}_0 1 + \mathbf{x}_1(I - R)^{-1} 1 = 1.
\]

2.9.7.10 Computing the matrices \( R \) and \( G \)

Solving for \( R \) and \( G \) can be carried out using the techniques for the GI/M/1 type and M/G/1 type respectively. However, very efficient methods have been developed for the QBD, such as

- **THE LOGARITHMIC REDUCTION (LR) METHOD**: This is a method developed by Latouche and Ramaswami [65] for QBDs and is quadratically convergent. The Cyclic Reduction method discussed earlier is similar in principle to the Logarithmic Reduction method. The LR method is vastly used for analyzing QBDs. Readers are referred to the original paper [65] and the book [67], both by Latouche and Ramaswami, for detail coverage of the method. In this section we present the results and give an outline of the algorithm. Essentially the algorithm works on computing the matrix \( G \) by kind of restricting the DTMC to only time epochs when it changes main levels, i.e. it goes up or down one, hence the transitions without a level change are censored out. Once the matrix \( G \) is obtained, the matrix \( R \) can be easily calculated from the relationship given earlier between \( R \) and \( G \).

Consider the transition matrix \( P \) and write the \( G \) matrix equation given as
\[
G = A_2 + A_1 G + A_0 G^2.
\]
This can be written as \((I - A_1)G = A_2 + A_0 G^2\) or better still as
\[
G = (I - A_1)^{-1} A_2 + (I - A_1)^{-1} A_0 G^2 = L + HG^2,
\]
with
\[
L = (I - A_1)^{-1} A_2, \quad \text{and} \quad H = (I - A_1)^{-1} A_0.
\]

If we restrict the process to only up and down movements while not considering when there is no movement, then the matrix \( L \) captures its down movement while the matrix \( H \) captures its up movements. Note that \( G \) and \( G^2 \) record first passage probabilities across one and two levels, respectively. So if we repeat this process
we can study first passage probabilities across 4, 8, \cdots levels. So let us proceed and define

\[ A_k^{(0)} = A_k, \quad k = 0, 1, 2 \]

\[ H^{(0)} = (I - A_1^{(0)})^{-1} A_0^{(0)}, \quad L^{(0)} = (I - A_1^{(0)})^{-1} A_2^{(0)}, \]

\[ U^{(k)} = H^{(k)} L^{(k)} + L^{(k)} H^{(k)}, \quad k \geq 0, \]

\[ H^{(k+1)} = (I - U^{(k)})^{-1} (H^{(k)})^2, \quad L^{(k+1)} = (I - U^{(k)})^{-1} (L^{(k)})^2, \quad k \geq 0. \]

After applying this process repeatedly Latouche and Ramaswami [65] showed that

\[ G = \sum_{k=0}^{\infty} \left( \prod_{i=0}^{k-1} H^{(i)} \right) L^{(k)}. \quad (2.112) \]

Since we can not compute the sum series infinitely, the sum is truncated at some point depending on the predefined tolerance.

- **THE CYCLIC REDUCTION METHOD:** The cyclic reduction method developed by Bini and Meini [22] for the M/G/1 is another appropriate algorithm for the QBD in the same sense as the LR. We compute the matrix \( G \) and use that to compute the matrix \( R \). The algorithm will not be repeated here since it has been presented under the M/G/1 techniques.

- **THE INVARIANT SUBSPACE METHOD:** Akar and Sohraby [1] developed this method for \( G \) and \( R \) matrices, and since for the QBDs \( A(z) \) are rational this method is appropriate for such analysis.

### 2.9.7.11 Some Special Structures of the Matrix \( R \) and the matrix \( G \)

If \( A_2 \) or \( A_0 \) is of rank one, then the matrix \( R \) can be obtained explicitly as the inverse of another matrix, and \( G \) matrix can also be obtained as product of a column vector and a row vector, as follows.

- if \( A_2 = v.\alpha \), then \( R = A_0 [I - A_1 - A_0 G]^{-1} \), where \( G = 1.\alpha \),
- if \( A_0 = \omega.\beta \), then \( R = A_0 [I - A_1 - \eta A_2]^{-1} \), where \( \xi = \beta [I - A_1 - \eta A_2]^{-1} \), \( \xi \omega = \eta, \xi A_2 1 = 1 \) and the matrix \( R \) satisfies \( R^i = \eta^{i-1} R, i \geq 1. \)
- All the special structures identified for the M/G/1 and GI/M/1 types are also observed for the QBDs.

### 2.9.8 Other special QBDs of interest

#### 2.9.8.1 Level-dependent QBDs:

A level dependent QBD has a transition matrix \( P \) given in the block partitioned form as follows:
We assume that the matrices \( A_{k,k} \), \( k = 0, 1, 2 \) are of dimension \( m_k \times m_k \) and the dimensions of the matrices \( A_{k,k+1} \) are \( m_k \times m_{k+1} \) and of \( A_{k+1,k} \) are \( m_{k+1} \times m_k \). Because the transition matrix is level dependent, the matrices \( R \) and \( G \) are now level dependent. We have \( R_k \) which records the rate of visiting level \( k \) before coming back to level \( k-1 \), given it started from level \( k-1 \), and \( G_k \) records the probability of first visit from level \( k \) to level \( k-1 \). The matrices are obtained as

\[
R_k = A_{k-1,k} + R_k A_{k,k} + R_k R_{k+1} A_{k+1,k},
\]

and

\[
G_k = A_{k,k-1} + A_{k,k} G_k + A_{k,k+1} G_{k+1} G_k.
\]

For this level dependent QBD we have matrix-product solution given as

\[
\varphi_{k+1} = \varphi_k R_k, \quad k \geq 0.
\]

The condition for this QBD to be positive recurrent is given in Latouche and Ramaswami [67] as follows:

**Condition:** That

\[
\varphi_0 = \varphi_0 (A_{0,0} + A_{0,1} G_1),
\]

with

\[
\varphi_0 \sum_{n=0}^{\infty} \prod_{k=1}^{n} R_k 1 = 1.
\]

The level dependent QBD is usually difficult to implement in an algorithm form in general. However, special cases can be dealt with using specific features of the DTMC. For more details on level dependent QBDs and algorithms see Latouche and Ramaswami [67]. Some of the special cases will be presented later in the queueing section of this book, especially when we present some classes of vacation models.

### 2.9.8.2 Tree structured QBDS

Consider a DTMC \( \{(X_n, Y_n), n \geq 0\} \) in which \( X_n \) assumes values of the nodes of a \( d \)-ary tree, and \( Y_n \) as the auxiliary variables such as phases. As an example a 2-ary has for level 1 the set \{1,2\}, for level 2 the set \{11,12,21,22\} for level 3 the set \{111,112,121,122,211,212,221,222\} and so on. If we now allow this DTMC to go up and down not more than one level, we then have a tree structured QBD. Details of such DTMC can be found in Yeung and Alfa [103]. We can also have
the GI/M/1 type and M/G/1 type of tree structured DTMC. For further reference see Yeung and Sengupta [104] and Takine, Sengupta and Yeung [95], respectively.

A $d$-ary tree is a tree which has $d$ children at each node, and the root of the tree is labelled $\{0\}$. The rest of the nodes are labelled as strings. Each node, which represents a level say $i$ in a DTMC, has a string of length $i$ with each element of the string consisting of a value in the set $\{1, 2, \cdots, d\}$. We use $+$ sign in this section to represent concatenation on the right. Consider the case of $d = 2$ and for example if we let $J = 1121$ be a node and $k = 2$ then $J + k = 11212$. We adopt the convention of using upper case letters for strings and lower case letters for integers. Because we are dealing with a QBD case we have that the DTMC, when in node $J + k$ could only move in one step to node $J$ or node $J + k + s$ or remain in node $J + k$, where $k, s = 1, 2, \cdots, d$. The last come first served single server queues have been analysed using this tree structured QBD.

For this type of QBD most of the results for the standard QBD still follow with only minor modifications to the notations. Consider the chain in a node $J + k$ and phase $i$ at time $n$, i.e. the DTMC is in state $(J + k, i)$, where $i$ is an auxiliary variable $i = 1, 2, \cdots, m$. At time $n + 1$ the DTMC could be at state:

- $(J, j)$ with probability $d^{i,j}_k$, $k = 1, 2, \cdots, d$
- $(J + s, j)$ with probability $a^{i,j}_{k,s}$, $k, s = 1, 2, \cdots, d$
- $(J + ks, j)$ with probability $u^{i,j}_{s,k}$, $k, s = 1, 2, \cdots, d$

Let $G_k$ be an $m \times m$ matrix which records the probability that the DTMC goes from node $J + k$ to node $J$ for the first time. This is really equivalent to the $G$ matrix in the standard QBD and to the $G_k$ matrix in the level dependent QBD. If we define $m \times m$ matrices $D_k$, $A_{k,s}$, and $U_s$ as the matrices with $(i, j)$ elements $d^{i,j}_k$, $a^{i,j}_{k,s}$ and $u^{i,j}_{s,k}$, then we have

$$G_k = D_k + \sum_{s=1}^{d} A_{k,s} G_s + \sum_{s=1}^{d} U_s G_s G_k, \quad k = 1, 2, \cdots, d. \quad (2.119)$$

Note that

$$ (D_k + \sum_{s=1}^{d} A_{k,s} + \sum_{s=1}^{d} U_s) \mathbf{1} = \mathbf{1}. \quad (2.120) $$

Similarly, if we define $R_k$ as the matrix that records the expected number of visits to node $(J + k)$ before visiting node $J$, given that it started from node $J$, then we have

$$R_k = U_k + \sum_{s=1}^{d} R_s A_{s,k} + \sum_{s=1}^{d} R_k R_s D_s, \quad k = 1, 2, \cdots, d. \quad (2.121)$$

Because the summations operations in the equations for both the $G_k$ and $R_k$ matrices are finite, we can simply apply the linear approach for the standard $R$ and $G$ matrices to compute these $G_k$ and $R_k$ matrices.

Provided the system is stable we can apply the linear algorithm as follows:

- Set
\[ G_k(0) := D_k, \quad R_k(0) := U_k, \quad k = 1, 2, \ldots, d \]

- Compute

\[ G_k(n+1) := D_k + \sum_{s=1}^{d} A_{k,s} G_s(n) + \sum_{s=1}^{d} U_s G_s(n) G_k(n), \quad k = 1, 2, \ldots, d, \]

\[ R_k(n+1) := U_k + \sum_{s=1}^{d} R_s(n) A_{s,k} + \sum_{s=1}^{d} R_k(n) R_s(n) D_s, \quad k = 1, 2, \ldots, d. \]

- Stop if

\[ (G_k(n+1) - G_k(n))_{i,j} < \varepsilon, \forall i, j, k, \]

and

\[ (R_k(n+1) - R_k(n))_{i,j} < \varepsilon, \forall i, j, k, \]

where \( \varepsilon \) is a preselected tolerance value.

### 2.9.9 Re-blocking of transition matrices

For most practical situations encountered we find that the resulting transition matrices do not have exactly any of the three structures presented. However, in most cases the structures are close to one of the three and by re-blocking the transition matrix we can achieve one of them.

#### 2.9.9.1 The non-skip-free M/G/1 type

As an example, we can end up with a non-skip-free M/G/1 type DTMC with a transition matrix of the following form

\[
P = \begin{bmatrix}
C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} & C_{0,4} & \cdots \\
C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
C_{k-1,0} & C_{k-1,1} & C_{k-1,2} & C_{k-1,3} & C_{k-1,4} & \cdots \\
A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\
A_0 & A_1 & A_2 & A_3 & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}.
\]  

(2.122)

This can be re-blocked further into \( k \times k \) superblocks so that we have
2.9 Numerical Computations of the Invariant Vectors

\[ P = \begin{bmatrix}
C_0 & C_1 & C_2 & C_3 & \cdots \\
H_0 & H_1 & H_2 & H_3 & \cdots \\
H_0 & H_1 & H_2 & \cdots \\
\vdots & \vdots & \ddots & \ddots \\
\end{bmatrix}, \quad (2.123) \]

where

\[ C_j = \begin{bmatrix}
C_{0,jk} & \cdots & C_{0,(j+1)k-1} \\
\vdots & \ddots & \vdots \\
C_{k-1,jk} & \cdots & C_{k-1,(j+1)k-1} \\
\end{bmatrix}, \quad j = 0, 1, 2, \cdots \\
\]

\[ H_j = \begin{bmatrix}
A_{jk} & \cdots & A_{(j+1)k-1} \\
\vdots & \ddots & \vdots \\
A_{(j-1)k} & \cdots & A_{jk} \\
\end{bmatrix}, \quad j = 1, 2, \cdots, \]

and

\[ H_0 = \begin{bmatrix}
A_0 & A_1 & \cdots & A_{k-1} \\
A_0 & \cdots & A_{k-2} & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
A_0 & \cdots & A_{k-2} & A_0 \\
\end{bmatrix}. \]

This non-skip-free M/G/1 type DTMC has now been converted to an M/G/1 type which is skip-free to the left. Standard M/G/1 results and techniques can now be applied to analyze this DTMC. We point out that there is a class of algorithms specifically designed for this class of problems by Gail, Hantler and Taylor [43].

Suppose we have a situation where \( C_j = 0 \) and \( H_j+w = 0, \forall j > M, \ 0 \leq w < \infty \), then we can further reblock this transition matrix to the QBD type as follows. We display the case of \( w = 0 \) only. Let

\[ B = C_0, \ C = [C_1, C_2, \cdots, C_M], \ E = \begin{bmatrix}
H_0 \\
0 \\
\vdots \\
0 \\
\end{bmatrix}, \]

\[ U = \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
H_M & 0 & 0 & \cdots & 0 \\
H_{M-1} & H_M & \cdots & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
H_1 & H_2 & \cdots & H_M & 0 \\
\end{bmatrix}, \ D = (e_1 \otimes e_M^T) \otimes H_0, \]
\[ A = \begin{bmatrix} H_1 & H_2 & H_3 & \cdots & H_M \\ H_0 & H_1 & H_2 & \cdots & H_{M-1} \\ H_0 & H_1 & \cdots & \cdots & H_{M-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ H_0 & H_1 & \cdots & \cdots & H_0 \end{bmatrix}, \]

where \( e_j \) is a column vector of zeros with one in location \( j \) and \( e_j^T \) is its transpose, then we can write the transition matrix as

\[ P = \begin{bmatrix} B & C \\ E & A U \\ D & A U \\ D & A U \end{bmatrix}, \]

which is of the QBD type. In this case we can apply the results of QBD to analyze the system.

2.9.9.2 The non-skip-free GI/M/1 type

Similar to the M/G/1 type DTMC we can have a GI/M/1 non-skip-free type of DTMC with transition matrix of the form

\[ P = \begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,k-1} & A_0 \\ B_{1,0} & B_{1,1} & \cdots & B_{1,k-1} & A_1 & A_0 \\ B_{2,0} & B_{2,1} & \cdots & B_{2,k-1} & A_2 & A_1 & A_0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (2.124) \]

This can also be re-blocked further into \( k \times k \) superblocks to obtain

\[ P = \begin{bmatrix} B_0 & F_0 \\ B_1 & F_1 & F_0 \\ B_2 & F_2 & F_1 & F_0 \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2.125) \]

where

\[ B_j = \begin{bmatrix} B_{jk,0} & \cdots & B_{jk,k-1} \\ \vdots & \ddots & \vdots \\ B_{(j+1)k-1,0} & \cdots & B_{(j+1)k-1,k-1} \end{bmatrix}, \quad j = 0, 1, 2, \cdots, \]

and

\[ H_j = \begin{bmatrix} A_{jk} & \cdots & A_{(j-1)k} \\ \vdots & \ddots & \vdots \\ A_{(j+1)k-1} & \cdots & A_{jk} \end{bmatrix}, \quad j = 1, 2, \cdots, \]
\[ H_0 = \begin{bmatrix} A_0 \\ A_1 & A_0 \\ \vdots & \vdots & \ddots \\ A_{k-1} & A_{k-2} & \cdots & A_0 \end{bmatrix} . \]

This non-skip-free DTMC has now been converted to a GI/M/1 type which is skip-free to the right. Standard GI/M/1 results can now be applied to analyze this DTMC. Once again we point out that there is a class of algorithms specifically designed for this class of problems by Gail, Hantler and Taylor [43].

Suppose we have a situation where \( B_j = 0 \) and \( F_{j+\nu} = 0, \forall j > N, 0 \leq \nu < \infty \), then we can further reblock this transition matrix to the QBD type as follows. We display the case of \( \nu = 0 \) only. Let

\[
B = B_0, \quad C = [F_0, 0, 0, \ldots, 0], \quad E = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix},
\]

\[
A = \begin{bmatrix} F_1 & F_0 \\ F_2 & F_1 & F_0 \\ \vdots & \vdots & \vdots & \ddots \\ F_N & F_{N-1} & F_{N-2} & \cdots & F_1 \end{bmatrix},
\]

\[
D = \begin{bmatrix} 0 & F_N & F_{N-1} & \cdots & F_2 \\ F_N & \cdots & F_3 \\ \vdots & \vdots \\ F_N \\ 0 \end{bmatrix}, \quad U = (e_1 \otimes e_1^T) \otimes F_1,
\]

then we can write the transition matrix as

\[
P = \begin{bmatrix} B & C \\ E & A & U \\ D & A & U \\ D & A & U \end{bmatrix},
\]

which is of the QBD type. In this case we can apply the results of QBD to analyze the system.

In general a banded transition matrix can be reduced to a QBD in structure.
2.9.9.3 Time-inhomogeneous Discrete Time Markov Chains

Up till now we have focussed mainly on Markov chains that are independent of time, i.e. where \( Pr\{X_{n+1} = j | X_n = i\} = p_{i,j} \). Here we deal with cases where \( Pr\{X_{n+1} = j | X_n = i\} = p_{i,j}^{(n)} \), i.e. the transition probability depends on the time of the transition. We present the case of time-inhomogeneous QBDs here. The case of the GI/M/1 and M/G/1 types have been discussed in details by Alfa and Margolius [2].

For the time-inhomogeneous QBDs we let the bivariate process be \( \{X_n, J_n\}, n \geq 0 \) with \( X_n \geq 0, 1 \leq J_n \leq M < \infty \). We can write the associated transition matrix as

\[
P^{(n)} = \begin{bmatrix}
A^{(n)}_{0,0} & A^{(n)}_{0,1} & A^{(n)}_{0,2} & \cdots & A^{(n)}_{0,J_n} \\
A^{(n)}_{1,0} & A^{(n)}_{1,1} & A^{(n)}_{1,2} & \cdots & A^{(n)}_{1,J_n} \\
A^{(n)}_{2,0} & A^{(n)}_{2,1} & A^{(n)}_{2,2} & \cdots & A^{(n)}_{2,J_n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
A^{(n)}_{J_n,0} & A^{(n)}_{J_n,1} & A^{(n)}_{J_n,2} & \cdots & A^{(n)}_{J_n,J_n}
\end{bmatrix},
\]

(2.126)

where the matrices \( A^{(n)}_{i,j} \) records the transition from time \( n \) to time \( n+1 \) with \( X_n \) changing from \( i \) to \( j \). Hence \( (A^{(n)}_{i,j})_{\ell,k} = Pr\{X_{n+1} = j, J_{n+1} = k | X_n = i, J_n = \ell\} \). If we now define \( (A^{(n,m)}_{i,j})_{\ell,k} = Pr\{X_{n+1} = j, J_{m+1} = k | X_n = i, J_n = \ell\}, m \geq n \). Note that we have \( A^{(n,n)} = A^{(n)} \). Let the corresponding transition matrix be \( P^{(n,m)} \). For simplicity we only display the case of \( m = n+1 \). We can write

\[
P^{(n,n+1)} = \begin{bmatrix}
A^{(n,n+1)}_{0,0} & A^{(n,n+1)}_{0,1} & A^{(n,n+1)}_{0,2} & \cdots & A^{(n,n+1)}_{0,J_n} \\
A^{(n,n+1)}_{1,0} & A^{(n,n+1)}_{1,1} & A^{(n,n+1)}_{1,2} & \cdots & A^{(n,n+1)}_{1,J_n} \\
A^{(n,n+1)}_{2,0} & A^{(n,n+1)}_{2,1} & A^{(n,n+1)}_{2,2} & \cdots & A^{(n,n+1)}_{2,J_n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
A^{(n,n+1)}_{J_n,0} & A^{(n,n+1)}_{J_n,1} & A^{(n,n+1)}_{J_n,2} & \cdots & A^{(n,n+1)}_{J_n,J_n}
\end{bmatrix},
\]

(2.127)

where \( A^{(n,n+1)}_{i,j} = \sum_{v=i-1}^{i+1} A^{(n)}_{i,v} A^{(n)}_{v,j} \), keeping in mind that \( A^{(m)}_{\ell,k} = 0, \forall |\ell - k| > 1, m \geq 0 \). Also note that \( A^{(n,n+1)}_{i,j} = 0, \forall |i - j| > 2 \). For the general case of \( P^{(n,m)} \), the block matrices \( A^{(n,m)}_{i,j} \) can be obtained through recursion as

\[
A^{(n,m)}_{i,j} = \sum_{v=i-m-1}^{i+1} A^{(n,m)}_{i,v} A^{(m)}_{v,j},
\]

(2.128)

keeping in mind that some of the block matrices inside the summation will have zero values only and \( A^{(n,m)}_{i,j} = 0, \forall |i - j| > m - n + 1 \).

If we know the state of the system at time \( n \) we can easily determine its state at time \( m+1 \). For example, let \( \mathbf{x}_n = [x_{0}^{(n)}, x_{1}^{(n)}, \cdots] \) and \( \mathbf{x}_i^{(n)} = [x_{i,1}^{(n)}, x_{i,2}^{(n)}, \cdots, x_{i,M}^{(n)}] \), then we have

\[
\mathbf{x}_i^{(m+1)} = \mathbf{x}_i^{(n)} P^{(n,m)}, 0 \leq n \leq m.
\]

(2.129)
Usually of interest to us in this system is the case that we call “periodic”, i.e.
where we have for some integer \( \tau \geq 1 \) and thus \( P^{(n)} = P^{(n+\tau)} \). In this case we have
\[
P^{(n+K\tau,n+(K+1)\tau-1)} = P^{(n,n+\tau-1)}, \quad \forall K > 0.
\] (2.130)
Let \( \mathcal{P}^{(n)} = P^{(n,n+\tau-1)} \), and \( \mathbf{x}^{(n+(K+1)\tau)} = [\mathbf{x}^{(n+K\tau)}, \mathbf{x}^{(n+K\tau)+1}, \ldots, \mathbf{x}^{(n+(K+1)\tau-1)}] \),
then we have
\[
\mathbf{x}^{(n+(K+1)\tau)} = \mathbf{x}^{(n+K\tau)} \mathcal{P}^{(n)}.
\] (2.131)
Under some stability conditions (which have to be determined for each case) we have \( \mathbf{x}^{(n+(K+1)\tau)}|_{K \to \infty} = \mathbf{x}^{(n)} \). Hence we have
\[
\mathbf{x}^{(n)} = \mathbf{x}^{(n)} \mathcal{P}^{(n)}.
\]

### 2.9.9.4 Time-inhomogeneous and spatially-homogeneous QBD

Suppose we have a case where
\[
P^{(n)} = \begin{bmatrix}
B^{(n)} & C^{(n)}
E^{(n)} & A_1^{(n)} & A_0^{(n)}
A_2^{(n)} & A_1^{(n)} & A_0^{(n)}
& & \\
& & \\
& & \\
& & \\
& & \\
& & \\
& & \\
& & \\
\end{bmatrix},
\] (2.132)
i.e. the DTMC is time-inhomogeneous but spatially-homogeneous. In this case the
matrix \( \mathcal{P}^{(n)} \) can be re-blocked. For example, if \( \tau = 2 \) then
\[
\mathcal{P}^{(n)} = P^{(n,n+1)} = 
\begin{bmatrix}
A_{0,0}^{(n,n+1)} & A_{0,1}^{(n,n+1)} & A_{0,2}^{(n,n+1)}
A_{1,0}^{(n,n+1)} & A_{1,1}^{(n,n+1)} & A_{1,2}^{(n,n+1)} & A_{1,3}^{(n,n+1)}
A_{2,0}^{(n,n+1)} & A_{2,1}^{(n,n+1)} & A_{2,2}^{(n,n+1)} & A_{2,3}^{(n,n+1)} & A_{2,4}^{(n,n+1)}
A_{3,0}^{(n,n+1)} & A_{3,1}^{(n,n+1)} & A_{3,2}^{(n,n+1)} & A_{3,3}^{(n,n+1)} & A_{3,4}^{(n,n+1)} & A_{3,5}^{(n,n+1)}
& & & & & \\
& & & & & \\
& & & & & \\
& & & & & \\
& & & & & \\
\end{bmatrix},
\] (2.133)
which results in
\[
\mathcal{P}^{(n)} = 
\begin{bmatrix}
B_{0,0}^{(n,n+1)} & B_{0,1}^{(n,n+1)} & C_{0,2}^{(n,n+1)}
B_{1,0}^{(n,n+1)} & B_{1,1}^{(n,n+1)} & A_{1}^{(n,n+1)} & A_{0}^{(n,n+1)}
B_{2,0}^{(n,n+1)} & A_{3}^{(n,n+1)} & A_{2}^{(n,n+1)} & A_{1}^{(n,n+1)} & A_{0}^{(n,n+1)}
A_{4}^{(n,n+1)} & A_{3}^{(n,n+1)} & A_{2}^{(n,n+1)} & A_{1}^{(n,n+1)} & A_{0}^{(n,n+1)}
& & & & & \\
& & & & & \\
& & & & & \\
& & & & & \\
& & & & & \\
\end{bmatrix}.
\] (2.134)
This can be reblocked to have
\[ \mathcal{P}(n) = \begin{bmatrix} \mathcal{B}(n) & \mathcal{C}(n) \\ \mathcal{A}^{(n)} & \mathcal{A}^{(n)} \\ \mathcal{A}^{(n)} & \mathcal{A}^{(n)} \\ \cdots & \cdots & \cdots \end{bmatrix}, \]

where for example

\[ \mathcal{A}^{(n)} = \begin{bmatrix} A_{2n+1} & A_{2n+2} \\ A_{2n+1} & A_{2n+2} \end{bmatrix}. \]

For the general case of \( 1 \leq \tau < \infty \) we can write and reblock \( \mathcal{P}(n) \) into a QBD type transition matrix. Given that \( \mathcal{A}^{(n)} = \sum_{k=0}^{\infty} \mathcal{A}^{(k)} \) and let \( \pi(n) \) be the stationary distribution associated with it, i.e.

\[ \pi(n) = \pi(n) \mathcal{A}^{(n)}, \quad \pi(n) \mathbf{1} = 1. \]  \hfill (2.135)

The stability conditions for standard QBDs apply, i.e. the system is stable iff

\[ \pi(n)(\mathcal{A}^{(1)} + 2 \mathcal{A}^{(2)}) \mathbf{1} > 1. \]  \hfill (2.136)

For a stable system we compute the stationary distribution of the transition matrix as

\[ \mathcal{x}^{(n+j+1)} = \mathcal{x}^{(n+j)} R^{(n)}, \]  \hfill (2.137)

where \( R^{(n)} \) is the minimal non-negative solution to the matrix equation

\[ R^{(n)} = \mathcal{A}^{(n)} + R^{(n)} \mathcal{A}^{(n)} + (R^{(n)})^2 \mathcal{A}^{(n)}. \]  \hfill (2.138)

In summary, standard matrix-analytic results can be used for the case of time-inhomogeneous DTMCs that are periodic.

### 2.10 Software Tools for Matrix-Analytic Methods

There are now software tools available to the public for analysing the M/G/1, GI/M/1 and QBD systems. The website is:

http://win.ua.ac.be/~vanhoudt/

The packages use efficient algorithms for the three systems and do provide supporting documents and papers for them also.
Queueing Theory for Telecommunications
Discrete Time Modelling of a Single Node System
Alfa, A.S.
2010, XIV, 238 p., Hardcover