Markov Chains: A Graph Theoretical Approach

by

Sinclair Antony Marcon

Dissertation submitted in fulfilment for the Degree
of
Master of Science
in
Mathematics

Faculty of Science
UNIVERSITY OF JOHANNESBURG

Supervisor: Prof. E. Jonck
Co–supervisor: Dr. M. J. Dorfling

2012
AFFIDAVIT: MASTER’S AND DOCTORAL STUDENTS
TO WHOM IT MAY CONCERN

This serves to confirm that I, _____________________________________________________

Full Name(s) and Surname

ID Number____________________________________________________________________

Student number__________________________________________________ enrolled for the

Qualification__________________________________________________________________

Faculty of Science

Herewith declare that my academic work is in line with the Plagiarism Policy of the University of
Johannesburg which I am familiar.

I further declare that the work presented in the________________________________________

__________________________________(minor dissertation/dissertation/thesis) is authentic and
original unless clearly indicated otherwise and in such instances full reference to the source is
acknowledged and I do not pretend to receive any credit for such acknowledged quotations, and
that there is no copyright infringement in my work. I declare that no unethical research practices
were used or material gained through dishonesty. I understand that plagiarism is a serious
offence and that should I contravene the Plagiarism Policy notwithstanding signing this affidavit, I
may be found guilty of a serious criminal offence (perjury) that would amongst other
consequences compel the UJ to inform all other tertiary institutions of the offence and to issue a

Signed at Johannesburg, ________________________________________on this _______ of

____________________2013

Signature__________________________________ Print name_________________________

STAMP COMMISSIONER OF OATHS
Affidavit certified by a Commissioner of Oaths
This affidavit conforms with the requirements of the JUSTICES OF THE PEACE AND COMMISSIONERS OF OATHS
ACT 16 OF 1963 and the applicable Regulations published in the GG GNR 1258 of 21 July 1972; GN 903 of 10 July
Acknowledgements

I would like to thank the NRF (National Research Foundation) for their financial assistance for this dissertation.

Furthermore, I would like to express my gratitude towards my mother, Zelda Marcon, for her constant support and encouragement throughout my studies.
Abstract

In chapter 1, we give the reader some background concerning digraphs that are used in the discussion of Markov chains; namely, their Markov digraphs. Warshall’s Algorithm for reachability is also introduced as this is used to define terms such as transient states and irreducibility.

Some initial theory and definitions concerning Markov chains and their corresponding Markov digraphs are given in chapter 2. Furthermore, we discuss $l$-step transitions (walks of length $l$) for homogeneous and inhomogeneous Markov chains and other generalizations.

In chapter 3, we define terms such as communication, intercommunication, recurrence and transience. We also prove some results regarding the irreducibility of some Markov chains through the use of the reachability matrix. Furthermore, periodicity and aperiodicity are also investigated and the existence of walks of any length greater than some specified integer $N$ is also considered. A discussion on random walks on an undirected torus is also contained in this chapter.

In chapter 4, we explore stationary distributions and what it means for a Markov chain to be reversible. Furthermore, the hitting time and the mean hitting time in a Markov digraph are also defined and the proof of the theorems regarding them are done. The demonstrations of the theorems concerning the existence and uniqueness of stationary distributions and the Markov Chain Convergence Theorem are carried out. Later in this chapter, we define the Markov digraph of undirected graphs, which are Markov chains as well. The existing theory is then applied to these.

In chapter 5, we explore and see how to simulate Markov chains on a computer by using Markov Chain Monte Carlo Methods. We also show how these apply to random $q$-colourings of undirected graphs.

Finally, in chapter 6, we consider a physical application of these Graph Theoretical concepts—the simulation of the Ising model. Initially, the relevant concepts of the Potts model are given and then the Gibbs sampler algorithm in chapter 5 is modified and used to simulate the Ising model. A relation between the chromatic polynomial and the partition function of the Potts model is also demonstrated.
## Contents

Acknowledgements ii  
Abstract iii  
List of Figures vi  

1 Introduction 1  
  1.1 Basic definitions and background 1  
  1.2 Subdigraphs, Deletion and Contraction 5  

2 Markov Chains: Transition Matrices and Initial distributions 21  
  2.1 A simple idea 21  
  2.2 $l$-step transition probabilities 25  
  2.3 Inhomogeneous Markov Chains 32  
  2.4 Other Generalizations for Markov Chains 34  
  2.5 Some other interesting results 35  

3 Classification of Vertices and Types of Markov Chains 42  
  3.1 Vertex/State Classification 42  
  3.2 Reducibility and Irreducibility of Markov Chains 45  
  3.3 Periodicity and Aperiodicity of States and Markov Chains 49  

4 Stationary Distributions and Reversible Markov Chains 57  
  4.1 Stationary Distributions 57  
  4.2 Reversible Markov Chains 76  
  4.3 Random Walks on Undirected Graphs 79  
    4.3.1 Some Definitions 80  

5 Computer Simulation of Markov Chains 91  
  5.1 Computer Simulation of Markov Chains 91  
  5.2 Markov Chain Monte Carlo (MCMC) Methods 101  
    5.2.1 Gibbs Sampler 103  
    5.2.2 Random $q$-colourings 113  
  5.3 The Propp–Wilson and the Sandwiching Algorithms 120  
    5.3.1 The Propp–Wilson Algorithm 120  
    5.3.2 Sandwiching 126
6 The Ising Model and the Tutte Polynomial 132
   6.1 A few definitions .................................................. 132
   6.2 The Ising Model and its Simulations .............................. 134
      6.2.1 The Ising Model .............................................. 134
      6.2.2 Mean Field Theory ......................................... 135
      6.2.3 Criticality ............................................... 139
      6.2.4 Modified Gibbs Sampler .................................. 140
      6.2.5 Some Interesting Results ................................ 145
   6.3 The partition function of the Potts model ....................... 150

Concluding Remarks 163

A Additional Information 164
   A.1 Proofs in Chapter 1 ............................................ 164
      A.1.1 Other Theorems concerning Digraphs ...................... 165
   A.2 Chapter 2 .................................................... 165
   A.3 Chapter 6 .................................................... 167
   A.4 Code ........................................................ 169
      A.4.1 $a \times a$ Grids ......................................... 169
      A.4.2 Implementation of the Systematic Sweep Gibbs Sampler 174

Bibliography 176
List of Figures

1.1 Arcs $a_1$ and $a_2$ are clearly not equal. ............................................. 1
1.2 Digraph $D$ with its vertex and arc sets written below. .......................... 2
1.3 Digraph $D$ and with its neighbourhoods written below. ......................... 5
1.4 Digraph $D$ and with its vertex and arc sets written below. ........................ 6
1.5 The subdigraph $D'$ of $D$ that is induced by the above set $U$. ..................... 6
1.6 The subdigraph $D''$ of $D$ that is induced by the arc subset $A'$. ................. 7
1.7 The subdigraph $D - v_4$. .............................................................. 7
1.8 The subdigraph $D - e_4$. .............................................................. 8
1.9 The directed pseudograph, $D$. ......................................................... 8
1.10 The vertex–induced directed subpseudograph $D_1$. ............................... 9
1.11 The arc-induced directed subpseudograph $D_2$. ................................... 9
1.12 The directed subpseudograph $D_3$ that was formed by deleting $v_3$ from $D$. .... 10
1.13 The directed subpseudograph, $D_4$ that was formed by deleting $e_7$ from $D$. .... 10
1.14 The undirected graph $G$ before any edge contraction. ........................... 11
1.15 The undirected graph $G/e_3$. ........................................................ 11
1.16 The digraph $D$. .............................................................................. 12
1.17 The digraph $D/e_6$. ........................................................................ 12
1.18 The digraph $D$ with its adjacency matrix $A$ written below. ..................... 13
1.19 The digraph $D$. .............................................................................. 14
1.20 A $C_5$ cycle in $D$. ........................................................................... 15
1.21 A directed pseudograph $D$ with its adjacency matrix below. .................... 17
1.22 A digraph $D$ and its underlying graph $G$. .......................................... 20

2.1 The Markov digraph $D$ of the above Markov chain. ............................... 22
2.2 The Markov digraph $D$ from example 2.1.1. ........................................... 25
2.3 The directed tree for the conditional probability, $p_{11} = P(s_1(1) | s_1(0)) = 0.75$. ... 26
2.4 The directed tree for the conditional probability, $P(s_1(2) | s_1(0)) = 0.6875$. ........ 26
2.5 The Markov digraph $D$ as a result of $P$. ............................................ 38
2.6 The Markov digraph $D_1$ as a result of $[Q]$. ......................................... 41

3.1 The Markov digraph $D$ with $V(D) = \{1, 2, \ldots, n\}$ and recurrence. ....... 44
3.2 The Markov digraph $C_n$ which is a directed $n$-cycle. ......................... 44
3.3 The Markov digraph $D$ and reducibility. ............................................ 46
3.4 The directed subgraph of $D$, $D_1$. ..................................................... 46
3.5 The Markov digraph $D$. ...................................................................... 47
3.6 The Markov digraph $D$. ...................................................................... 48
3.7 The Markov digraph $D$ with its reachability matrix below. ..................... 49
3.8 The Markov digraph of the defined transition mechanism. ..................... 52
4.1 An undirected graph \( G \) ......................................................... 80  
4.2 The Markov digraph \( D \) of the undirected graph \( G \) ......................................................... 81  
4.3 An undirected connected non–bipartite graph \( G \) ......................................................... 89  
4.4 The Markov digraph \( D \) of the connected non–bipartite graph \( G \) ......................................................... 89  

5.1 The input colour configuration for the function \( q\text{coloring} \) ......................................................... 118  
5.2 A possible \( q \)-colouring of graph \( G \) after 2000 steps ......................................................... 119  
5.3 The Markov digraph \( G_1 \) ......................................................... 124  
5.4 A \( 3 \times 3 \) grid ......................................................... 125  
5.5 The Markov digraph \( D \) ......................................................... 125  
5.6 The Markov chain of the defined birth–death process ......................................................... 127  
5.7 The Markov digraph \( G_1 \) ......................................................... 130  

6.1 A graphical scheme of working out the Tutte polynomial of \( G \) ......................................................... 150  
6.2 Computing the Tutte polynomial of \( G_1 \) ......................................................... 151  
6.3 Computing the Tutte polynomial of \( G_2 \) ......................................................... 151  
6.4 Computing the Tutte polynomial of \( G_3 \) ......................................................... 152  

A.1 State space, \( \Omega \) ......................................................... 166  
A.2 A partitioning of the state space, \( \Omega \) ......................................................... 166  
A.3 A graphical representation of event \( A \) in the state space, \( \Omega \) ......................................................... 166  
A.4 A graphical representation of an \( a \times a \) grid ......................................................... 170  
A.5 A graphical depiction of case a) ......................................................... 170  
A.6 A \( 3 \times 3 \) grid ......................................................... 170  
A.7 A random feasible “2–colouring” of the previous graph ......................................................... 173  
A.8 A random initial \( q \)-colouring of the \( 3 \times 3 \) grid ......................................................... 173  
A.9 The last colour configuration on the graph after 10000 steps of \( q\text{coloring} \) have been applied ......................................................... 174
Chapter 1

Introduction

1.1 Basic definitions and background

Before beginning the discussion on Markov chains and Markov digraphs, we need to familiarize ourselves with some concepts from Graph Theory. For this purpose we will consult the following references: [5], [6], [8], [11] and [18].

Firstly, a digraph $D$ is a finite non-empty set of objects called vertices together with a set of ordered pairs of vertices called arcs. The latter set may be empty and then we have a null graph. The above can also simply be stated as a usual graph in which the edges have direction. This is why we place an arrowhead on an edge in the appropriate direction when we draw a digraph.

Exactly what is meant by direction/orientation?
Consider the following simple argument (see Figure 1.1). Let there be some ordered pair of vertices, say $a_1 = (u, v)$ or simply $uv$. This means that we have an arc moving from $u$ to $v$.

Now consider the arc $a_2 = vu$. Clearly, we have that this arc is directed from $v$ to $u$.

Graphically, the above can be depicted as follows:

![Figure 1.1: Arcs $a_1$ and $a_2$ are clearly not equal.](image)

Let the vertex set of $D$ be denoted by $V(D)$. Similarly, we may denote the arc set by $A(D)$ or $E(D)$. (Lets see what this means graphically.)

Consider the following example.

Example 1.1.1.
The order of a digraph $D$ is the cardinality of its vertex set and is denoted by $n(D)$ or simply $n$; $n_D$ is also used when dealing with many digraphs at the same time. Similarly, as in the case of undirected graphs, the size of a digraph $D$, denoted by $m(D)$ or $m$, is the cardinality of its arc set. Therefore, if we consider example 1.1.1 again,

\[
|V(D)| = 4 \quad \Rightarrow \quad n = 4
\]

\[
|E(D)| = 4 \quad \Rightarrow \quad m = 4
\]

If $a = (u, v) = uv$ is an arc of a digraph $D$, then $a$ is said to join $u$ to $v$. Since the direction of $a$ is from $u$ to $v$, we define $a$ to be incident from $u$ and incident to $v$. Consequently, $u$ is adjacent to $v$, whilst $v$ is adjacent from $u$ (as a result of there being a direction).

From example 1.1.1 and the above definitions, $v_1$ is adjacent to $v_2$; however, $v_2$ is not adjacent to $v_1$: $v_2$ is adjacent from $v_1$. We also have that $v_2$ and $v_4$ are adjacent to and adjacent from the other.

Considering the above definitions, and with a bit of reasoning, two vertices $u$ and $v$ are nonadjacent if neither $u$ nor $v$ are adjacent to or from each other.

Recall that for an undirected graph $G$, the number of edges incident to a vertex $v$ gives us the degree of $v$—this is usually denoted by $deg v$ or more specifically as $deg_G v$. In the case of directed graphs, there are two other types of degrees—indegrees and outdegrees.

- The outdegree of a vertex $v$ of a digraph $D$ is the number of vertices of $D$ that are adjacent from $v$. It may be denoted as $od v$ or $deg^+ v$ or more specifically, $d^+_D v$. We will use $d^+_D v$. 

Figure 1.2: Digraph $D$ with its vertex and arc sets written below.

\begin{align*}
V(D) &= \{v_1, v_2, v_3, v_4\} \\
E(D) &= \{(v_1, v_2), (v_2, v_4), (v_4, v_2), (v_3, v_2)\} \\
&= \{e_1, e_2, e_3, e_4\}
\end{align*}
• The *indegree* of a vertex \( v \in V(D) \) is the number of vertices adjacent to \( v \). It can be denoted by \( \text{id}v \) or \( d^-v \) or more specifically, \( d^-_Dv \). We will use \( d^-_Dv \).

The *degree* of a vertex \( v \in V(D) \) is the sum of its indegrees and outdegrees—that is,

\[
\text{deg}_Dv = d^+_Dv + d^-_Dv
\]

Going back to example 1.1.1, we now write down the various degrees of two of the vertices—\( \{v_2, v_3\} \).

\[
\begin{align*}
d^+_Dv_2 &= 1 & d^-_Dv_2 &= 3 & \Rightarrow \text{deg}_Dv_2 = d^+_Dv_2 + d^-_Dv_2 = 1 + 3 = 4 \\
d^+_Dv_3 &= 1 & d^-_Dv_3 &= 0 & \Rightarrow \text{deg}_Dv_3 &= 1
\end{align*}
\]

It is very important to note that we are referring to digraphs. In digraphs, we have no loops nor multiple arcs.

Just as we had a *minimum degree* (\( \delta \)) and a *maximum degree* (\( \Delta \)) in an undirected graph, we can also define the following:

• The *minimum outdegree* of \( D \) is

\[
\delta^+ = \delta^+(D) = \min \{d^+_Dv : v \in V(D)\}
\]

and the *maximum outdegree* of \( D \) is

\[
\Delta^+ = \Delta^+(D) = \max \{d^+_Dv : v \in V(D)\}
\]

• Similarly, the *minimum indegree* and the *maximum indegree* are defined respectively as follows

\[
\begin{align*}
\delta^- &= \delta^-(D) = \min \{d^-_Dv : v \in V(D)\} \\
\Delta^- &= \Delta^-(D) = \max \{d^-_Dv : v \in V(D)\}
\end{align*}
\]

• We also have the *minimum semi-degree* of \( D \), \( \delta^0 = \delta^0(D) = \min \{\delta^+(D), \delta^-(D)\} \) and the *maximum semi-degree* of \( D \), \( \Delta^0(D) = \max \{\Delta^+(D), \Delta^-(D)\} \).

A digraph \( D \) is said to be *regular* if \( \delta^0(D) = \Delta^0(D) \) (another term that can also be used is \( \delta^0(D) \)-regular).

Consider the following theorem that gives us a relation between the outdegrees and the indegrees of the vertices of a digraph \( D \). This theorem will be used in a later chapter. The proof of this theorem can be found in Appendix A.

**Theorem 1.1.2. The First Theorem of Digraph Theory**

If \( D \) is a digraph of order \( n \) and size \( m \) with the vertex set \( V(D) = \{v_1, v_2, v_3, \ldots, v_n\} \), then

\[
\sum_{i=1}^{n} d^+ v_i = \sum_{i=1}^{n} d^- v_i = m \tag{1.1.1}
\]

3
Since we do not deal with multiple directed and undirected edges in this dissertation, we only need to concern ourselves with one other type of arc called a loop.

For an arc $uv$, the first vertex $u$ is also known as its tail and $v$ is known as the arc $uv$’s head. A loop is an arc whose head and tail coincide.

A digraph in which parallel arcs and loops are allowed, is defined to be a directed pseudograph. A directed pseudograph in which no loops occur is known as a directed multigraph.

Consider two digraphs $D_1 = (V_1 = V(D_1), E_1 = E(D_1))$ and $D_2 = (V_2 = V(D_2), E_2 = E(D_2))$. If there exists an isomorphism $\phi$, i.e. a one-to-one mapping from $V(D_1)$ onto $V(D_2)$ such that $uv \in E(D_1)$ if and only if $\phi(u) \phi(v) \in E(D_2)$, then $D_1$ is isomorphic to $D_2$. ($\phi$ preserves adjacency and incidence.)

Let $\cong$ denote the relation “isomorphic to”. This relation is an equivalence relation.

Recall the definition of an equivalence class from Abstract Algebra:

An equivalence relation on a set $S$ is a set $R$ of ordered pairs of elements of $S$ such that

a) $(a,a) \in R$ for all $a \in S$—(reflexivity)

b) $(a,b) \in R$ implies $(b,a) \in R$—(symmetry)

c) $(a,b) \in R$ and $(b,c) \in R$ implies that $(a,c) \in R$—(transitivity)

As a consequence of $\cong$ being an equivalence relation, we find two different equivalence classes: one is being isomorphic and the other is being nonisomorphic.

We will now discuss the different types of neighbourhoods that can occur in a digraph $D$. For a vertex $v$ of a digraph $D = (V,E)$, we now define the following:

- out-neighbourhood,
- in-neighbourhood,
- neighbourhood of $v$,
- out-neighbours, in-neighbours and neighbours of $v$.

The out-neighbourhood of $v$ can be defined as

$$N^+_D(v) = \{u \in V - \{v\} : vu \in E(D)\}.$$  \hspace{1cm} (1.1.2)

Therefore, the out-neighbourhood of $v$ is the set of vertices that are the heads of the arcs that have $v$ as their tail. The vertices in $N^+_D(v)$ are known as the out-neighbours of $v$.

Similarly, the in-neighbourhood of $v$ can be defined as

$$N^-_D(v) = \{u \in V - \{v\} : uv \in E(D)\}.$$ \hspace{1cm} (1.1.3)

Therefore, the in-neighbourhood of $v$ is the set of vertices in the digraph $D$ that are the tails of the arcs that have $v$ as their head. The vertices in $N^-_D(v)$ are known as the in-neighbours of $v$.

The union of the sets in (1.1.2) and (1.1.3) gives us the neighbourhood of $V$. The elements in this set are referred to as the neighbours of $v$. In set notation, the neighbourhood of $v$ is defined as

$$N_D(v) = N^+_D(v) \cup N^-_D(v).$$
Consider the following example:

**Example 1.1.3.** Suppose that we would like to find the neighbourhoods of vertex $v_2$ in the accompanying digraph $D$.

By definition, we have

$$N^+_D(v_2) = \{v_1, v_3, v_5\}$$

$$N^-_D(v_2) = \{v_3, v_4\}$$

Taking the union of the above two sets, we get

$$N_D(v_2) = \{v_1, v_3, v_4, v_5\}$$

---

### 1.2 Subdigraphs, Deletion and Contraction

A digraph $D$ in which all its vertices and arcs are labelled (given specific names in $V(D)$ and $E(D)$, respectively) is referred to as a *labelled digraph*. A labelled digraph $H$ is a *subdigraph* of a labelled digraph $D$ if $V(H) \subseteq V(D)$, $E(H) \subseteq E(D)$ and every arc in $E(H)$ has both end vertices in $V(H)$.

An unlabelled digraph $H$, is a *subdigraph* of an unlabelled digraph $D$, if $H$ and $D$ can be labelled such that we have the above case again. The notation that is used to indicate that $H$ is a subdigraph of $D$ is:

$$H \subseteq D$$

Furthermore, if $V(H) = V(D)$, then $H$ is a *spanning subdigraph* of $D$.

A subset $V(H) \subseteq V(D)$ together with all the arcs that are formed by these vertices and that exist in $E(D)$, is known as a *vertex induced subdigraph* or an *induced subdigraph*.

For a subset $A' \subseteq E(D)$, the subdigraph that is *arc-induced* by $A'$ is the digraph $D\langle A'\rangle = (V', A')$, where $V'$ is the subset of the vertices in $V(D)$ that are incident with the arcs in $A'$. 
A vertex-deleted subdigraph of a digraph $D$, is obtained by deleting/removing one or more vertices from $V(D)$ and all the arcs in $E(D)$ that are incident with the removed vertex/vertices.

Arc-deleted subdigraphs of a digraph $D$ are obtained by simply removing arcs from $D$. Consequently, if $H'$ is an arc-deleted subdigraph, then $V(H') = V(D)$ and $E(H') \subset E(D)$.

We now consider examples in which we discuss the above definitions.

**Example 1.2.1.**

![Figure 1.4: Digraph $D$ and with its vertex and arc sets written below.](image)

$V(D) = \{v_1, v_2, v_3, v_4\}$

$E(D) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

i) The subdigraph induced by $U = \{v_1, v_2, v_3\}$ is given below.

![Figure 1.5: The subdigraph $D'$ of $D$ that is induced by the above set $U$.](image)

$V(D') = \{v_1, v_2, v_3\} \subset V(D)$ and $E(D') = \{e_1, e_4\} \subset E(D)$. Hence, $D' \subseteq D$.

ii) An example of an arc-induced subdigraph.

Consider the subdigraph induced by $A' = \{e_1, e_2, e_3\}$. 

Clearly, $V(D') = \{v_1, v_2, v_3\} \subset V(D)$ and $E(D') = \{e_1, e_4\} \subset E(D)$. Hence, $D' \subseteq D$. 

We have $V(D'') = \{v_1, v_2, v_3, v_4\} \subseteq V(D)$ and $A' = E(D'') = \{e_1, e_2, e_5\} \subseteq E(D)$. Hence, $D'' \subseteq D$.

iii) If we were to delete the vertex $v_4$ from $D$, we would have the following digraph.

We have $V(D - v_4) = \{v_1, v_2, v_3\} \subset V(D)$ and $E(D - v_4) = \{e_1, e_4\} \subset E(D)$. Clearly, $D - v_4 \subseteq D$.

iv) Now, if we were to delete the arc $e_4$ from $D$, we would have the following:
Figure 1.8: The subdigraph $D - e_4$.

We have

$$V(D - e_4) = \{v_1, v_2, v_3, v_4\} \subseteq V(D)$$

and

$$E(D - e_4) = \{e_1, e_2, e_3, e_5, e_6\} \subset E(D).$$

Hence, $D - e_4 \subseteq D$.

Similarly, for directed pseudographs/multigraphs, we can define the concepts: **vertex-deleted**, **arc-deleted**, **induced** and **arc-induced directed subpseudographs/submultigraphs**. The definitions of these are similar to those for subdigraphs, except that the graph of which they are subsets is a directed pseudograph/multigraph.

Next, we discuss an example of a directed pseudograph.

**Example 1.2.2.** Consider the directed pseudograph $D$ drawn below.

Figure 1.9: The directed pseudograph, $D$. 
\[ V(D) = \{v_1, v_2, v_3, v_4, v_5\} \] and \[ E(D) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} \].

The following graphs are directed subpseudographs of \( D \):

i) \( D_1 \) is induced by the vertex set \( V' = \{v_1, v_3, v_4\} \).

![Figure 1.10: The vertex-induced directed subpseudograph \( D_1 \).](image)

ii) The following digraph \( D_2 \) is an arc-induced directed subpseudograph of \( D \) induced by the subset \( \{e_1, e_3, e_5, e_6\} \).

![Figure 1.11: The arc-induced directed subpseudograph \( D_2 \).](image)

iii) If we were to delete \( v_3 \) from \( D \), we would get the graph \( D_3 = D - v_3 \).
iv) If we were to delete the arc $e_7$ from $D$, we would get the following directed subpseudograph $D_4$.

**Example 1.2.3.** Consider the following depiction of the graph $G = (V(G), E(G))$, where $V(G) = \{v_1, v_2, v_3, v_4\}$ and $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. 

In an undirected graph $G$, the contraction of an edge $e$ with endpoints $u$ and $v$ is the replacement of $u$, $v$ and $e$ with a single vertex, say $v'$, whose incident edges are all the edges other than $e$ that were incident with $u$ and $v$. Let $G/e$ denote the contraction of the edge $e$ in $G$.

The next example deals with the above.
Based on the above definition, if we let our new vertex be $v'$, then $G/e$ is given by the following depiction:

\[ |V(G/e)| = 3 \quad \text{and} \quad |E(G/e)| = 5 \]

Similarly, in a digraph $D$, the contraction of an arc $e$ with the endpoints $u$ and $v$ is the replacement of $u$, $v$ and $e$ with a single vertex, say $v'$, whose incident arcs are all the arcs other than $e$ that were incident with $u$ and $v$. Let $D/e$ denote the arc contraction of the arc $e$ in the digraph $D$. Note that the above extends to directed pseudographs and directed multigraphs.

Consider the following example.

**Example 1.2.4.** Consider the following depiction of the digraph $D = (V(D), E(D))$, where $V(D) = \{v_1, v_2, v_3, v_4, v_5\}$ and $E(D) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. 

11
Next we define the adjacency matrix of a digraph. Note that this concept extends to directed pseudographs.

Let $D = (V, E)$ be a digraph, where $V = \{v_1, v_2, \ldots, v_n\}$.

The adjacency matrix of a digraph $D$ is an $n \times n$ matrix, say $A$, where for the element $a_{ij}$ in $A$, the entry in the $i$-th row and $j$-th column is defined by

$$
a_{ij} = \begin{cases} 
1 & \text{if } v_iv_j \in E \\
0 & \text{if } v_iv_j \notin E
\end{cases}
$$
Later on, the above matrix will be used to get the reachability matrix of a digraph.

As an example, we have the following digraph $D$ and its corresponding adjacency matrix $A$.

**Example 1.2.5.**

![Digraph D and its adjacency matrix A](image)

Figure 1.18: The digraph $D$ with its adjacency matrix $A$ written below.

$$A = \begin{bmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}$$

Note that $n = 4$ and $A$ is a $4 \times 4$ matrix.

Before we consider the definition of the reachability matrix, we need to familiarize ourselves with the concepts of paths, walks and cycles of a digraph $D$.

Let $D = (V,E)$ be a directed pseudograph for the discussion on the following definitions: walks, paths and cycles.

A walk $W$ in $D$ is an alternating sequence of vertices and arcs in $D$. We write $W = x_1 a_1 x_2 a_2 \ldots a_{k-1} x_k$ such that $x_i$ is the tail of arc $a_i$ and $x_{i+1}$ is the head, where $i = 1, 2, \ldots, k - 1$.

This walk $W$ in the above definition is a $(k - 1)$-step walk as there are $k - 1$ arcs between $x_1$ and $x_k$ in the sequence. Note that $x_i$ has been used to indicate the consecutive vertices on the walk $W$ and $a_i$ has been used to indicate the consecutive arcs on the walk $W$. There may be repetitions of vertices and arcs of $D = (V,E)$. Vertices in $V$ and arcs in $E$ were renamed to avoid confusion.

The first vertex in a (directed) walk is the initial vertex and the last vertex in a (directed) walk is the terminal vertex. The length of a (directed) walk is the number of arcs that are present in the walk.

A (directed) walk $W$ is closed if the initial vertex $x_1$ in the sequence of $W$ coincides with the terminal vertex $x_k$ in $W$, i.e. $x_1 = x_k$. A (directed) walk $W$ is open if the initial vertex $x_1$ and the terminal vertex $x_k$ in $W$ do not coincide.

A (directed) trail of $D$ is a (directed) walk in which all arcs are distinct.

A (directed) path of a directed pseudograph $D$ is a (directed) walk of $D$ in which no vertices or arcs are repeated.
A \textit{(directed) cycle} of a directed pseudograph $D$ is a closed directed path in $D$.

When we speak of a \textit{(directed)} $u$-$v$ walk/path, then we are referring to a directed walk/path with the initial vertex $u$ and the terminal vertex $v$.

The \textit{length} of a \textit{(directed)} path, \textit{(directed)} cycle or \textit{(directed)} trail in $D$ is also defined as the number of arcs present. A \textit{(directed)} cycle of length $k$ is referred to as a \textit{(directed)} $k$-\textit{cycle}.

\textbf{Example 1.2.6.}

\begin{figure}[h]
\centering
\scalebox{0.5}{
\begin{tikzpicture}
  \node (v1) at (0,0) {$v_1$};
  \node (v2) at (-1,1) {$v_2$};
  \node (v3) at (1,1) {$v_3$};
  \node (v4) at (0,2) {$v_4$};

  \draw [-stealth] (v1) -- (v2) node[midway, above] {$e_1$};
  \draw [-stealth] (v1) -- (v3) node[midway, above] {$e_2$};
  \draw [-stealth] (v1) -- (v4) node[midway, above] {$e_3$};
  \draw [-stealth] (v2) -- (v3) node[midway, above] {$e_4$};
  \draw [-stealth] (v2) -- (v4) node[midway, above] {$e_5$};
  \draw [-stealth] (v3) -- (v4) node[midway, above] {$e_6$};
\end{tikzpicture}
}
\caption{The digraph $D$.}
\end{figure}

Since the listing of the vertices in a path implicitly implies direction, we will henceforth leave out the arcs.

<table>
<thead>
<tr>
<th>Possible (directed) $v_2$-$v_1$ walks are:</th>
<th>Possible (directed) $v_2$-$v_1$ trails</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_1 : v_2 v_1$</td>
<td>$T_1 : v_2 v_1$</td>
</tr>
<tr>
<td>$W_2 : v_2 v_4 v_1$</td>
<td>$T_2 : v_2 v_4 v_1$</td>
</tr>
<tr>
<td>$W_3 : v_2 v_4 v_1 v_5 v_3 v_1$</td>
<td>$T_3 : v_2 v_4 v_1 v_5 v_3 v_1$</td>
</tr>
<tr>
<td>$W_4 : v_2 v_4 v_3 v_1$</td>
<td></td>
</tr>
<tr>
<td>$W_5 : v_2 v_1 v_5 v_3 v_1$</td>
<td></td>
</tr>
<tr>
<td>$W_6 : v_2 v_1 v_5 v_3 v_2 v_1$</td>
<td></td>
</tr>
</tbody>
</table>

The cycle $C_5 : v_4 v_1 v_5 v_3 v_2 v_4$ is highlighted on the digraph below:
Looking at the above, we can see that a directed path is contained in a directed cycle. One such path is

\[ P_4 : v_4 v_1 v_5 v_3 v_2. \]

Some \( v_2-v_1 \) paths are listed in the table below.

<table>
<thead>
<tr>
<th>( v_2-v_1 ) paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 : v_2 v_1 )</td>
</tr>
<tr>
<td>( P_2 : v_2 v_4 v_1 )</td>
</tr>
</tbody>
</table>

As a matter of interest, the adjacency matrix of \( D \) is given by

\[
A(D) = A = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
\end{bmatrix}
\]

We now present the following two theorems on the existence of a directed path in a digraph—these proofs can be found in [6]. Note that these theorems extend to directed pseudographs as any directed pseudograph, say \( D \), contains a digraph; we simply remove any loops and parallel arcs from \( D \) to get an “underlying” digraph.

**Theorem 1.2.7.** Let \( D \) be a digraph and \( u \) and \( v \) be two distinct vertices such that \( u, v \in V(D) \). If \( D \) has a closed (directed) \( u-u \) walk, then \( W \) contains a cycle \( C \) through \( u \) such that \( E(C) \subseteq E(W) \).

**Theorem 1.2.8.** Let \( D \) be a digraph and \( u \) and \( v \) be two distinct vertices such that \( u, v \in V(D) \). If \( D \) has a (directed) \( u-v \) walk, then \( D \) contains a (directed) \( u-v \) path.

We also take note of the next two definitions.

The length of the shortest (directed) path between two vertices \( u \) and \( v \) in a digraph \( D \) is known as the *distance* between \( u \) and \( v \). We write \( d(u, v) \).
A digraph $D$ is *acyclic* if it does not contain any cycles.

We now consider the concepts of reachable and reachability. A vertex $v$ is *reachable* from a vertex $u$ in a digraph/directed pseudograph $D$, if $D$ has a $u$-$v$ walk. Another way of stating the above is to say that $v$ can be *reached* from $u$ if $D$ has a $u$-$v$ walk.

Thus we can define the reachability matrix of a digraph/directed pseudograph $D$.

The *reachability matrix* of a digraph (or a directed pseudograph) $D$ is an $n \times n$ matrix $R(D)$ where $r_{ij}$, the entry in the $i$th row and the $j$th column of $R(D)$, is defined by

$$ r_{ij} = \begin{cases} 1 & \text{if } j \text{ is reachable from } i \\ 0 & \text{if } j \text{ is not reachable from } i \end{cases} $$

We present two ways of determining the reachability of a digraph/directed pseudograph $D$. They are:

- the Breadth-first search (BFS) algorithm and
- Warshall’s algorithm.

In BFS, we use the definition of distance in a digraph/directed pseudograph.

**Breadth-First Search (BFS) Algorithm for Digraphs**

The idea of BFS is to maintain a set $R$ of vertices that have been reached from an initial vertex $u$, but these vertices have not been searched for. Similarly, one must also maintain (update) a set $S$ of vertices that have been searched for.

The set $R$ is managed as a queue; i.e. a First-In-First-Out list. FIFO means that the first elements are the first to be processed and the first to be returned in a list. Therefore, considering a digraph, the first vertices found are the first to be explored.

**Input:** A digraph $D$ and an initial (starting) vertex $u$.

**Output:** A set of vertices that are reachable from the initial vertex $u$.

**Initialization:** $R = \{u\}$, $S = \phi$ (the empty set) and $d(u, v) = 0$.

**Iteration:**

(i) For $R \neq \phi$, search from the first vertex $v$ of $R$.

(ii) If $N(v) \notin S \cup R$, then add $N(v)$ to the back of the list $R$ and associate $N(v)$ with the distance $d(u, v) + 1$.

(iii) Remove $v$ from $R$ and place it in $S$ as it has already been searched for.

The largest distance from a vertex $u$ is the *eccentricity* of $u$ This is denoted by $\varepsilon(u)$. The largest eccentricity of the digraph $D$ is called the *diameter* of $D$, denoted $\text{diam } D$. Informally, by using the BFS algorithm, we are looking for $\text{diam } D$.

Consider the following example on how to determine the reachability in a directed pseudograph $D$. 

---

16
Example 1.2.9.

![Figure 1.21: A directed pseudograph $D$ with its adjacency matrix below.](image)

$$A(D) = A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Let us tabulate the list of adjacent vertices.

<table>
<thead>
<tr>
<th>Adjacent from:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- Say that 1 is the initial vertex: Therefore, $S = \emptyset$ and $d(1, y) = d = 0$.
  Firstly, we search for all arcs $1y \in A$, where $y$ is a vertex in $D$. From $A$, 3 and 4 are reachable from 1. (These vertices have a distance of 1 from vertex 1—$d = 1$.) Hence, we can list 3 and 4 in $S$. Therefore, the $R$ in the algorithm would now contain the neighbours of vertices 3 and 4. That is, $R = \{1, 3\}$.
  We note that 1 is not in the list $S$ yet as there is the possibility that it may not be reachable from itself.
  Now consider all the vertices adjacent to 3—these are the vertices that have a distance $d = 2$ from vertex 1. The search yields the result of 3. Therefore, the list $S$ does not change and $R = \{1\}$.
  Consider vertex 4. We now want to search for the vertices that have a distance $d = 2$ from vertex 1. From $A(D)$, we have that 1 is reachable from 4. Hence, we can include 1 in our list, which is now $S = \{3, 4, 1\}$. Furthermore, $R = \emptyset$. Therefore, we set $R_1 = S = \{3, 4, 1\}$, where $R_1$ is the set of vertices that are reachable from 1.

- Considering vertex 2, we find that the vertices 1, 3 and 4 are reachable from 2. Therefore, $R_2 = \{1, 3, 4\}$.

- Vertex 3 is reachable only from itself. Therefore, $R_3 = \{3\}$.

- Now consider vertex 4.
  Since 1 is adjacent from 4, it is reachable from 4 and hence, the first member of the list. From $A(D)$, 1 is adjacent to 3 and 4 and therefore, $R_4 = \{1, 3, 4\}$.

17
A more systemic approach to reachability will now be considered, which is called Warshall’s Algorithm.

Firstly, we need to define Boolean addition in terms of vectors. Let \( \overrightarrow{u} \) and \( \overrightarrow{v} \) be \( 1 \times n \) vectors in \( \mathbb{R}^n \), where each entry is either 0 or 1. Define the following addition between vectors:

\[
\overrightarrow{u} + \overrightarrow{v} = [u_1 \ u_2 \ldots \ u_n] + [v_1 \ v_2 \ldots \ v_n] = [u_1 + v_1 \ u_2 + v_2 \ldots \ u_n + v_n]
\]

We add the components as follows for \( i = 1, 2, \ldots, n \):

\[
u_i + v_i = \begin{cases} 
0 + 1 = 1 & \text{if } u_i = 0 \text{ and } v_i = 1 \\
1 + 0 = 1 & \text{if } u_i = 1 \text{ and } v_i = 0 \\
0 + 0 = 0 & \text{if } u_i = 0 \text{ and } v_i = 0 \\
1 + 1 = 1 & \text{if } u_i = 1 \text{ and } v_i = 1 
\end{cases}
\]

We use the above vector addition for the row operations in the construction of our reachability matrix in Warshall’s Algorithm.

Continuing, we let \( D \) be a digraph and \( V(D) = \{1, 2, \ldots, n\} \). Let \( Q \) be an \( n \times n \) matrix with the entries

\[
q_{ij} = \begin{cases} 
1 & \text{if } j \text{ is reachable from } i \\
0 & \text{if it is not known whether } j \text{ is reachable from } i 
\end{cases}
\]

for \( i, j = 1, 2, \ldots, n \).

As in the previous example, \( A(D) \) would give the initial set of vertices that are reachable from each other. Therefore, let \( Q_0 = A(D) \).

Take a vertex \( j \) and keep it fixed.

i) Consider row \( j \) of \( Q \). If \( k \) is reachable from \( j \), then \( q_{jk} = 1 \). Otherwise, \( q_{jk} = 0 \).
ii) Considering the \( j \text{th} \) column of \( Q \). If \( j \) is reachable from \( i \), then \( q_{ij} = 1 \). Otherwise, \( q_{ij} = 0 \).
iii) From i) and ii), if \( j \) is reachable from \( i \) and \( k \) is reachable from \( j \), then \( k \) is reachable from \( i \)—since we have a \( i-j \) walk followed by a \( j-k \) walk which is a \( i-k \) walk. Therefore, replace the entry \( q_{ij} \) by 1 in \( Q \) if \( q_{ij} = 0 \).

In step iii), we are performing Boolean addition on the rows of \( Q \). We add \( Row_j(Q) = R_j \) to \( Row_i(Q) = R_i \) when the entry \( q_{ij} = 1 \).

Warshall’s Algorithm:

Let \( D = (V, E) \) be a digraph, where \( V(D) = \{1, 2, \ldots, n\} \), and \( A(D) \) be the adjacency matrix of \( D \). Let \( R_i \) denote row \( i \) and \( Col_j \) denote column \( j \).

i) Set \( Q_0 = A(D) \)—initialization.
ii) For \( i = 1 \) to \( n \) do the following:

Add \( R_i \) to every row of \( Q_{i-1} \) in which 1 is the entry in the \( i \text{th} \) column of \( Q_{i-1} \).
iii) Set \( R = Q_n \).

In example 1.2.9, we have the adjacency matrix

\[
Q_0 = A(D) = A = \begin{bmatrix}
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\]

ii) From Warshall’s Algorithm, column 1 has 1’s in rows 2 and 4; therefore, add \( R_1 \) to \( R_2 \) and \( R_4 \).

\[
R_2 \leftarrow R_2 + R_1, \quad R_4 \leftarrow R_4 + R_1 
\]

\( Q_1 = \begin{bmatrix}
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\]

No row of \( Q_1 \) has 1 in column 2. Hence, \( Q_2 = Q_1 \).

For \( Q_3 \), rows 1, 2 and 4 of \( Q_2 \) have 1 in their 3\textsuperscript{rd} columns, therefore

\[
R_1 \leftarrow R_1 + R_3, \quad R_2 \leftarrow R_2 + R_3, \quad R_4 \leftarrow R_4 + R_3 
\]

\( Q_3 = \begin{bmatrix}
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\]

For \( Q_4 \), rows 1 and 2 of \( Q_3 \) have 1 in their 4\textsuperscript{th} columns. Thus

\[
R_1 \leftarrow R_1 + R_4, \quad R_2 \leftarrow R_2 + R_4 
\]

\( Q_4 = \begin{bmatrix}
1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\]

Hence, \( R = Q_4 \). This shows that

- \( \{1, 3, 4\} \) are reachable from 1,
- \( \{1, 3, 4\} \) are reachable from 2,
- 3 is reachable from 3 and
- \( \{1, 3, 4\} \) are reachable from 4.

The above confirms our results from earlier.

Another concept that we need to define is a \textit{weighted arc} in a directed pseudograph. The term \textit{weighted} is the term used when we have a mapping \( c \) from the arc set to the set of real numbers, i.e.

\[
c : E(D) \rightarrow \mathbb{R}
\]

Depending on the type of weight on an arc, positive real numbers are in the direction of the arc, whereas negative real numbers indicate that the direction of the arc must be reversed. A value of zero indicates an arc that does not exist and so it is not drawn in a directed pseudograph. For our purposes, we will be considering arc weights with positive real numbers that are less than or equal to one.
We denote a \textit{weighted directed pseudograph} \( D \) by

\[ D = (V(D), E(D), c) \]

We are preparing ourselves step by step to deal with a particular weighted directed pseudograph; namely, \textit{Markov digraphs/chains}.

Another important aspect needed is that of \textit{connected digraphs}.

A digraph \( D \) is \textit{strong} or \textit{strongly-connected} if for every pair of vertices \( u, v \in V(D) \), there is a directed \( u-v \) path and a directed \( v-u \) path. This is also referred to as being \textit{disconnected}.

The \textit{underlying graph} of a digraph \( D \) is the graph \( G \) obtained by treating the arcs of \( D \) as unordered pairs. Hence, \( G \) is a simple graph where the vertex set remains unchanged and the arc set is now referred to as the edge set.

**Example 1.2.10.**

![Figure 1.22: A digraph \( D \) and its underlying graph \( G \).](image)

The sets associated with the above graphs are listed below:

\[
\begin{align*}
V(D) &= \{v_1, v_2, v_3\} & E(D) &= \{v_1v_2, v_3v_2, v_3v_1\} \\
V(G) &= \{v_1, v_2, v_3\} & E(G) &= \{v_1v_2, v_3v_2, v_3v_1\}
\end{align*}
\]

A digraph \( D \) is \textit{weakly-connected} if its underlying graph \( G \) is connected. A nontrivial graph \( G \) is \textit{connected} if there exists a path between every two distinct vertices in \( G \).

Other important concepts are \textit{vertex disjoint} or \textit{disjoint} and \textit{arc disjoint} in a digraph \( D \).

Let \( D \) be a digraph and \( Q \) and \( S \) be directed subgraphs of \( D \). Two directed subgraphs of a digraph \( D \) are said to be \textit{vertex disjoint} if the intersection of their vertex sets is empty—\( V(Q) \cap V(S) = \emptyset \).

Similarly, two directed subgraphs, \( Q \) and \( S \) of digraph \( D \) are \textit{arc disjoint} if \( E(Q) \cap E(S) = \emptyset \).

Let \( W_1 : u, x_1, x_2, \ldots, x_k, v \) and \( W_2 : u, y_1, y_2, \ldots, y_l, v \) be two directed \( u-v \) walks in a digraph \( D \). If \( W_1 \cap \{y_1, y_2, \ldots, y_l\} = \emptyset \) and \( \{x_1, x_2, \ldots, x_k\} \cap W_2 = \emptyset \), \( W_1 \) and \( W_2 \) are \textit{internally disjoint}.

These concepts will be useful in our discussion of irreducible Markov chains.
Chapter 2

Markov Chains: Transition Matrices and Initial distributions

2.1 A simple idea

Markov processes are concerned with the probability of transitioning between two states \( s_i \) to \( s_j \). The state space of a Markov chain is a mutually exclusive and collectively exhaustive set of states that our process could transition to at discrete time intervals. Markov chains in general terms are digraphs where the states are represented by vertices and the probabilities of transitioning between vertices are given by weighted arcs—the weights of the arcs are the probabilities of transitioning in the direction of the arc to another adjacent vertex. Since successor vertices/states are dependent only on their current vertex/vertices, the Markov process is said to be memoryless.

In the discussion of Markov chains/digraphs, the following statements may be used interchangeably:

- Markov chain — Markov digraph
- State — vertex.

Note: Some other necessary definitions and notations will be defined when they are needed. The following references will be consulted for this purpose: [13], [19], [28]—used for some definitions from probability theory—and [29].

Since probabilities are involved in the discussion of Markov chains, it is important for us to familiarize ourselves with basic probability theory. Chapter 1 in [19] gives a good introduction about these concepts. However, these concepts, definitions, etc. will be discussed in appropriate sections, examples, etc.

Example 2.1.1. Consider the following example in which the graphical representation for a Markov chain is given. Some notations and concepts are also defined. Afterwards, more general explanations are given.

Say we have two distinct states, \( s_1 \) and \( s_2 \). Furthermore, it is stated that

i) the probability of moving to \( s_2 \) given that the process is currently at \( s_1 \) is 0.25. This can be written as

\[
P[\text{moving to } s_2 \text{ at time } = 1 \mid \text{starting at } s_1 \text{ at time } = 0] = 0.25
\]
The following is equivalent to the above:

\[ P[s_2(1) \mid s_1(0)] = 0.25 \]

\( s_2(1) \): \( s_2(1) \) means that we will be at vertex \( s_2 \) in 1 step (a walk of length 1).

\( s_1(0) \): The 0 indicates the vertex/state in which the Markov process has begun.

The 1 and the 0 in brackets may also be interpreted as the number of trials.

ii) The probability of staying in state \( s_2 \) is 0.5. Using the above notation, we write

\[ P[s_2(1) \mid s_2(0)] = 0.5 \]

Since a Markov chain is an example of a stochastic process, the sum of all weights on the arcs of which a state \( s_i \) is a tail, must add to 1.

Therefore, we still have to consider the following:

i) \( P[s_1(1) \mid s_1(0)] \)

ii) \( P[s_1(1) \mid s_2(0)] \)

To see what the probability is to stay in \( s_1 \), we consider \( s_1 \) as the tail. We have

\[
\begin{align*}
P[s_1(1) \mid s_1(0)] &= 1 - P[s_2(1) \mid s_1(0)] \\
&= 1 - 0.25 \\
&= 0.75
\end{align*}
\]

To see what the probability is to begin at \( s_2 \) and to move to \( s_1 \), we take \( s_2 \) as the tail of the arcs. We have

\[
\begin{align*}
P[s_1(1) \mid s_2(0)] &= 1 - P[s_2(1) \mid s_2(0)] \\
&= 1 - 0.5 \\
&= 0.5
\end{align*}
\]

Graphically, the above is given by:

\[ \begin{aligned}
D : \\
\begin{array}{ccc}
s_1 & 0.25 & s_2 \\
0.75 & & 0.5 \\
 & 0.5 & \\
\end{array}
\end{aligned} \]

Figure 2.1: The Markov digraph \( D \) of the above Markov chain.

\[ |V(D)| = n(D) = n = 2 \text{ and } |E(D)| = m(D) = m = 4 \text{ as} \]

\[ V(D) = \{s_1, s_2\} \text{ and } E(D) = \{s_1s_2, s_2s_1, s_1s_1, s_2s_2\} \]

Clearly, this is a directed pseudograph that looks like a chain. Hence the name Markov chain.

We have the adjacency matrix of \( D \), given by

\[ A(D) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \]

22
Now consider the weighted adjacency matrix
\[ w(A(D)) = \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix}. \]

The rows of this matrix give us our initial vertices/states or tails and the columns give us the terminal vertices/states or heads. The entries in the above matrix are the weights of the arcs of \( D \). This is known as the transition probability matrix of the Markov chain \( D \). A more general definition will be given momentarily. Furthermore, the sum of each row adds to one.

A practical example of a Markov chain is the Ehrenfest model of diffusion, which we will now discuss.

**Example 2.1.2.** In the Ehrenfest model of diffusion, we have two containers, say \( A \) and \( B \) that are placed adjacent to each other. Furthermore, we have \( m \in \mathbb{N} \) gas molecules that can pass between these containers through a small aperture joining them.

For this example, we assume that we are working with discrete time intervals. At each epoch of time, we select one molecule randomly (uniformly) from the \( m \) molecules and pass it through the aperture. We define our vertices in our Markov digraph to be the number of molecules in container \( A \)—therefore, we have \( m + 1 \) vertices. Let \( X_k \) be the number of molecules in container \( A \) after \( k \) units of time.

Say that there are \( i \) molecules in container \( A \). If a molecule from \( A \) is randomly selected to move to \( B \), then we have \( i - 1 \) molecules in \( A \). Since all molecules in \( A \) are as likely as each other to be selected, we have
\[
p_{i(i-1)} = P(X_1 = i - 1 \mid X_0 = i) \\
= P(s_{i-1}(1) \mid s_i(0)) \\
= \frac{i}{m} \quad \text{for} \quad 0 \leq i \leq m \tag{2.1.1}
\]

Similarly, all \((m - i)\) molecules in \( B \) are equally likely to be selected to move to \( A \). Therefore, we have the following transition probability for this event.
\[
p_{i(i+1)} = P(X_1 = i + 1 \mid X_0 = i) \\
= P(s_{i+1}(1) \mid s_i(0)) \\
= \frac{m - i}{m} \tag{2.1.2}
\]

The remaining transition probabilities are as follows.

- \( p_{ii} = 0 \) for \( 0 \leq i \leq m \) since a molecule always moves between the two containers; we cannot stay at the same state.
- \( p_{i(i-j)} = 0 \) where \( 2 \leq j \leq i; \ j \in \mathbb{N} \). This is as a result of selecting one molecule at each discrete time interval—we do not select 2 or more molecules from \( A \) to move through the aperture to the other container.
- \( p_{i(i+j)} = 0 \) where \( 2 \leq j \leq (m - i); \ j \in \mathbb{N} \). This is as a result of selecting one molecule at each discrete time interval—we do not select 2 or more molecules from \( B \) to move through the aperture to \( A \).
Clearly, all transition probabilities are based on the current number of molecules in $A$; hence, this process is only concerned with the current state. Therefore, the above is an example of a Markov chain.

When considering the Markov chain for $k$ steps given some initial state, we have the following

$$P(X_{k+1} = s_j \mid X_0 = s_{i_0}, X_1 = s_{i_1}, \ldots, X_k = s_{i_k})$$

The above is read as: “The probability of being in state $s_j$ in $k + 1$ steps given that the process was in $s_{i_0}$ at time 0, then transitioned to $s_{i_1}$ in 1 step, then the process carried on through a sequence of states until state $s_{i_k}$ in $k$ steps.”

Another way of writing the above is

$$P(s_j(k + 1) \mid s_{i_0}(0), s_{i_1}(1), \ldots, s_{i_k}(k))$$

We now come to the defining property of a finite Markov chain; namely, the Markov property.

The Markov Property (The Markov Condition)

Let $a, b, c, l, k, z, j, n \in \mathbb{N}_0$. The Markov condition states

$$P(s_j(k) \mid s_a(0), s_b(1), \ldots, s_l(k-2), s_z(k-1)) = P(s_j(k) \mid s_z(k-1))$$

The above definition means that the present state of the process (at $k - 1$) specifies all historical information relevant to the future behaviour of a Markov process.

We now redefine the following to agree with the transition probability matrix of a Markov chain $D$. Let $p_{ij} = P(s_j(k) \mid s_i(k-1))$ for $1 \leq i, j \leq n$, where $n = |V(D)|$. Therefore, $p_{ij}$ is the transition probability and is the entry in the $i^{th}$ row and $j^{th}$ column of a transition matrix $P$. Since there are $n$ vertices in a Markov digraph $D$, $P$ is an $n \times n$ matrix. Furthermore, there are no negative entries in $P$ as $0 \leq p_{ij} \leq 1$; $1 \leq i, j \leq n$.

$$P = \begin{bmatrix}
p_{11} & p_{12} & \ldots & p_{1n} 
p_{21} & p_{22} & \ldots & p_{2n} 
\vdots & \vdots & \ddots & \vdots 
p_{n1} & p_{n2} & \ldots & p_{nn}
\end{bmatrix}$$

We remind the reader that the sum of each row is equal to 1, i.e.

$$\sum_{j=1}^{n} p_{ij} = 1 \quad \text{for } i = 1, 2, \ldots, n$$

(2.1.3)

In terms of the probability function, (2.1.3) would appear as follows:

$$\sum_{j=1}^{n} P(s_j(k+1) \mid s_i(k)) = 1 \quad i = 1, 2, \ldots, n \text{ and } k \in \mathbb{N}_0$$

(2.1.4)
As a consequence of the Markov condition, the following is equivalent to (2.1.4).

\[ \sum_{j=1}^{n} P(s_j(1) \mid s_i(0)) = 1 \]

Therefore, \( p_{ij} \) is independent of \( k \) (the number of steps). This is known as time homogeneity.

We now consolidate the above to give the following definition.

Let \( D \) be a Markov digraph with \( n \) vertices/states. Let \( P \) be an \( n \times n \) matrix with elements \( \{ p_{ij} : i, j = 1, 2, \ldots, n \} \). A random process \((X_0, X_1, \ldots)\) with finite state space/vertex set \( V(D) = S = \{ s_1, s_2, \ldots, s_n \} \) is said to be a homogeneous Markov chain with the transition matrix \( P \), if for all \( k \), all \( i, j \in \{ 1, 2, \ldots, n \} \), and all \( i_0, \ldots, i_{k-1} \in \{ 1, 2, \ldots, n \} \), we have

\[
P(s_j(k+1) \mid s_{i_0}(0), s_{i_1}(1), \ldots, s_{i_{k-1}}(k-1), s_i(k)) = P(s_j(k+1) \mid s_i(k)) = p_{ij}
\]

### 2.2 \( l \)-step transition probabilities

We start this section by defining the \( l \)-step transition probability, \( p_{ij}(l) \). This is the conditional probability that the random process will be in state \( s_j \) after exactly \( l \) trials/steps (a walk of length \( l \)) from the current/present state/vertex \( s_i \). We write,

\[
p_{ij}(l) = P(s_j(k+l) \mid s_i(k))
\]

From the above, the previous section was based on the special case when \( l = 1 \).

We revisit example 2.1.1.

**Example 2.2.1.**

![Diagram](image)

**Figure 2.2: The Markov digraph \( D \) from example 2.1.1.**

Consider \( p_{11} = P(s_1(1) \mid s_1(0)) = 0.75. \) We can represent this conditional probability with directed trees.
Figure 2.3: The directed tree for the conditional probability, \( p_{11} = P(s_1(1) \mid s_1(0)) = 0.75 \).

We determine \( p_{11}(2) \).

\[
p_{11}(2) = P(s_1(2) \mid s_1(0)) \\
= P(s_1(2) \mid s_2(1)) P(s_2(1) \mid s_1(0)) + P(s_1(2) \mid s_1(1)) P(s_1(1) \mid s_1(0)) \\
= (0.5)(0.25) + (0.75)(0.75) \\
= 0.6875
\]

Let us compare this with the directed branching trees. Since we start with \( s_1 \) at \( t = 0 \), \( s_1(0) \) is our initial vertex and since we end with \( s_1 \) at \( t = 2 \), \( s_1(2) \) is a terminal vertex. The graphical representation is

Figure 2.4: The directed tree for the conditional probability, \( P(s_1(2) \mid s_1(0)) = 0.6875 \).

There are two walks from \( s_1 \) back to \( s_1 \) in 2 steps. These are \( s_1(0)s_1(1)s_1(2) \) or simply \( W_1 : s_1s_1s_1 \) and \( s_1(0)s_2(1)s_1(2) \) or simply \( W_2 : s_1s_2s_1 \).

Now consider the adjacency matrix of \( D \) in Figure 2.2 as well as its transition matrix \( P \), which is the weighted adjacency matrix of \( D \).

\[
A = A(D) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\
A^2 = AA = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}
\]

\[
P = \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix}
\]
Let $a_{ij}(l)$ be the entry in row $i$ and column $j$ of the matrix $A^l$. In particular, $a_{11}(2) = 2$, which is the number of walks from $s_1$ to $s_1$ and is also the number of branches and terms that start with $s_1$ and end at $s_1$ in 2 steps.

More interestingly,

$$P^2 = PP = \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.6875 & 0.3125 \\ 0.625 & 0.375 \end{bmatrix}$$

Let $p_{ij}(l)$ be the entry in row $i$ and column $j$ of the matrix $P^l$. Now consider the entry $p_{11}(2)$. We have $p_{11}(2) = 0.6875$, which is our transition probability from $s_1$ to $s_1$ in 2 steps.

Furthermore, the reachability matrix, which was defined in the introductory chapter, can be used to see whether there is at least one walk between any two vertices $s_i$ and $s_j$. If there is no walk, then $p_{ij}$ would be 0 and no calculations really need to be done. Therefore, it is a good idea to determine the reachability matrix $R(D)$ after having drawn the Markov digraph and written down the transition matrix and adjacency matrix.

In this example, $R = A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and consequently, all vertices are reachable from each other. [This concept will be visited later in this text.]

Clearly, the number of branches in the probability tree is the number of walks.

**Theorem 2.2.2.** If $A$ is the adjacency matrix of a directed pseudograph $D$ with $V(D) = \{s_1, s_2, \ldots, s_n\}$, then the entry in the $i^{th}$ row and $j^{th}$ column of $A^l$, $l \geq 1$, is the number of $s_i$–$s_j$ walks of length $l$ in $D$.

**Proof.** The proof is by induction on $l$.

**Base Case:** $l = 1$

The adjacency matrix $A$ indicates the set of all adjacent vertices $s_i$ and $s_j$. There exist $s_i$–$s_j$ walks of length 1 if and only if $s_isj \in E(D)$. Therefore, $A$ gives the number of $s_i$–$s_j$ walks of length 1.

**Inductive Step:** Assume that $A^{l-1} = [a_{ij}(l-1)]$ and that $a_{ij}(l-1)$ gives the number of $s_i$–$s_j$ walks of length $l - 1$.

From matrix multiplication, we have $A^l = A^{l-1}A$, where

$$a_{ij}(l) = \sum_{w=1}^{n} a_{iw}(l-1)a_{wj}; \quad 1 \leq i, j \leq n$$

Every $s_i$–$s_j$ walk of length $l$ has a $s_i$–$s_\tau$ walk of length $l - 1$, where $s_\tau$ is adjacent to $s_j$, followed by an arc $s_\tau s_j$ and then the vertex $s_j$.

From the inductive hypothesis, we have $a_{i\tau}(l - 1) s_i$–$s_\tau$ walks of length $l - 1$. By adding the edge $s_\tau s_j$ to each $s_i$–$s_\tau$ walk, we get $a_{ij}(l)$ $s_i$–$s_j$ walks of length $l$.

Now consider a very useful result from probability theory; namely, “The Law of Total Probability”. For the proof of this law, refer to [28] pg 18.
Let $\Omega$ be the entire state space. Let $B_1, B_2, \ldots, B_k$ be such that $\cup_{i=1}^{k} B_i = \Omega$ and $B_i \cap B_j = \emptyset$ for $i \neq j$. Then for any event $A$, we have

$$P(A) = \sum_{1 \leq i \leq k} P(A \mid B_i) P(B_i)$$

where $P(B_i) > 0$.

It is important to note that $P(A \mid B_i)$ is not defined when $P(B_i) = 0$. In general for the sake of simplicity, people often write the above as

$$P(A) = \sum_{i=1}^{k} P(A \mid B_i) P(B_i)$$

and assume that the terms for which $P(B_i) = 0$ are dropped. In light of the above, we will adopt the general method used by people.

Continuing with our discussion, since the transition matrix $P$ is the weighted adjacency matrix, we now look at a result that is similar to the above for Markov chains, which are directed weighted pseudographs. The above law will be used to prove this result.

**Theorem 2.2.3.** If $P$ is the transition matrix of a Markov digraph $D$, with $V(D) = \{s_1, s_2, \ldots, s_n\}$ where state $s_i$ is the vertex $s_i$ in $D$ for $i = 1, 2, \ldots, n$, then the $(i,j)$th entry in $P^l; l \geq 1$, is the transition probability from the state $s_i$ to $s_j$ for a walk of length $l$ in $D$. Moreover, the sum of each row in $P^l; l \in \mathbb{N}$ is 1.

**Proof.** There are two things that need to be proven, namely:

i) The entries in $P^l$ are the transition probabilities for a walk of length $l$ between vertices $s_i$ and $s_j$.

ii) The stochastic property of $P^l$ holds—i.e. the sum of each row is 1.

**Proof of (i):** Induction on $l$

$l = 1$

The transition probability for a $s_i$–$s_j$ walk of length 1 is the probability of moving from vertex $s_i$ to vertex $s_j$ in 1 step. This is the $(i,j)^{th}$ entry in $P$. The above is true for all $1 \leq i, j \leq n$.

$l \neq 1$ (Inductive hypothesis)

Assume that the theorem is true for $l-1$. That is, the transition probabilities to move from vertex $s_i$ to $s_j$ for all $1 \leq i, j \leq n$ in $(l-1)$ steps are given by the entries in $P^{l-1}$.

Note: The transition probability to move/walk from $s_i$ to $s_j$ in $(l-1)$ steps will be denoted by $p_{ij}(k-1); \forall i, j$.

Consider

$$P^l = P^{l-1} P \quad (2.2.1)$$

To find the transition probability from $s_i$ to $s_j$, we have to consider all the states $s_w \mid 1 \leq w \leq n; w \in \mathbb{N}$ that can go to $s_j$ in 1 step—that is, there is a weighted arc from $s_w$ to $s_j$. 

28
From the inductive hypothesis, the transition probability from \( s_i \) to \( s_w \) in a walk of length \((l - 1)\) is given by the \((i,w)\)th entry in \( P^{l-1} \). By using the law of total probability, we get

\[
p_{ij}(l) = P(s_j(l) \mid s_i(0))
= \sum_{w=1}^{n} P(s_w(l-1) \mid s_i(0)) P(s_j(l) \mid s_w(l-1), s_i(0)) \quad \text{(from the law of total probability)}
= \sum_{w=1}^{n} P(s_w(l-1) \mid s_i(0)) P(s_j(l) \mid s_w(l-1)) \quad \text{(from the Markov property)}
= \sum_{w=1}^{n} P(s_w(l-1) \mid s_i(0)) P(s_j(1) \mid s_w(0)) \quad \text{(from time homogeneity)}
= \sum_{w=1}^{n} p_{iw}(l-1)p_{wj}
\]

Furthermore, the entries of \( P^l \) can be calculated as follows:

\[
p_{ij}(l) = \begin{bmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{nj} \end{bmatrix} = \sum_{w=1}^{n} p_{iw}(l-1)p_{wj}
\]

Since the above arguments are the same, it follows from the principle of Mathematical Induction that \( P^l \) gives the transition probabilities between any two states \( s_i \) and \( s_j \) in \( l \) steps for all \( l \geq 1 \).

Note: There is no problem when considering all the entries \( p_{wj} \) \((w = 1, 2, \ldots, n)\) as \( p_{wj} = 0 \) would imply no transition for a specific \( w \).

Proof of (ii):
We need to verify that

\[
\sum_{j=1}^{n} p_{ij}(l) = 1 \quad \text{for each row } i, \ 1 \leq i \leq n; \ l \geq 1, \ l \in \mathbb{N}
\]

For \( P \): \( \sum_{j=1}^{n} p_{ij} = 1 \) for \( 1 \leq i \leq n, \ i \in \mathbb{N} \).

For the inductive hypothesis, assume that \( \sum_{j=1}^{n} p_{ij}(l-1) = 1 \) for \( P^{l-1} \).

Now consider \( P^l \).
\[
\sum_{j=1}^{n} p_{ij}(l) \\
= \sum_{j=1}^{n} \sum_{w=1}^{n} p_{iw}(l-1)p_{wj} \\
= \sum_{w=1}^{n} \sum_{j=1}^{n} p_{iw}(l-1)p_{wj} \text{ (since } j \text{ is independent of } w, \text{ we can swap} \\
\text{the summations around)} \\
= \sum_{w=1}^{n} p_{iw}(l-1)\sum_{j=1}^{n} p_{wj}
\]

Since \( \sum_{j=1}^{n} p_{wj} = 1 \) and \( \sum_{w=1}^{n} p_{iw}(l-1) = 1 \) from the inductive hypothesis, it follows that

\[
\sum_{j=1}^{n} p_{ij}(l) = \sum_{w=1}^{n} p_{iw}(l-1)(1) \\
= 1
\]

Next we consider the initial distribution, which is a row vector that shows the initial state of the Markov chain. Let the initial distribution be denoted by

\[
\vec{\mu}(0) = \begin{bmatrix} \mu_1^{(0)} & \mu_2^{(0)} & \cdots & \mu_n^{(0)} \end{bmatrix} = \begin{bmatrix} P(s_1(0)) & P(s_2(0)) & \cdots & P(s_n(0)) \end{bmatrix}
\]

where \( n \) is the number of vertices in the Markov digraph \( D \). Since \( \vec{\mu}(0) \) is a probability distribution, \( \sum_{j=1}^{n} \mu_j^{(0)} = 1 \).

Consider example 2.2.1, where we calculated \( p_{11} \). Since we were starting at \( s_1 \), we are certain that the Markov process is in this state \( s_1 \) at \( t = 0 \). Therefore, \( P(s_1(0)) = 1 \) and our initial distribution is

\[
\vec{\mu}(0) = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

Now consider

\[
\vec{\mu}(1) = \vec{\mu}(0)P \quad \text{where} \quad P = \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix}
\]

\[
= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix}
\]

\[
= \begin{bmatrix} 0.75 & 0.25 \end{bmatrix}
\]

which is the probabilities of being at the vertices \( s_1 \) and \( s_2 \) given, after one step, that we started at \( s_1 \) for \( t = 0 \). Once again, since \( \vec{\mu}(1) \) is a probability distribution, the sum of the probabilities of \( \vec{\mu}(1) \) is
1—i.e. $\sum_{j=1}^{2} \mu_j^{(1)} = 0.75 + 0.25 = 1$.

Next, we show that we can use the initial distribution and the transition matrix $P$ of a Markov digraph $D$ to compute the distributions $\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(\zeta)}$, of the Markov chain, where $\mu^{(i)}; i \in \mathbb{N}$ is the distribution of the vertices of the Markov chain at time $i$.

Before we proceed to the next theorem, we define $P^0 = I$ where $I$ is the identity matrix that has the same dimensions as $P$. Furthermore, we will also use $(X_0, X_1, \ldots)$ to denote a Markov chain where $X_i$ is a vertex that we are situated at at time $t = i$.

**Theorem 2.2.4.** For a Markov chain $(X_0, X_1, \ldots)$, with vertices $\{s_1, s_2, \ldots, s_n\}$ in the Markov digraph $D$, let $\mu^{(0)}$ be the initial distribution and let $P$ be the transition matrix. For any $l$, $l \in \mathbb{Z}; l \geq 0$, $\mu^{(l)}$ at time $l$ satisfies the following

$$\mu^{(l)} = \mu^{(0)} P^l$$

**Proof.** Let $\mu_j^{(l)} = P(s_j(l))$.

**Case $l = 0$:**
Consider the following:

$$\mu^{(0)} = \mu^{(0)} I \quad \text{(where } I \text{ is a } n \times n \text{ identity matrix)}$$

$$= \mu^{(0)} P^0$$

Hence $\mu^{(0)} = \mu^{(0)} P^0$.

**Inductive hypothesis:**
Assume that for $l - 1$ and for all $1 \leq j \leq n$,

$$\mu_j^{(l-1)} = \left( \mu^{(0)} P^{l-1} \right)_j$$

Now consider $\mu_j^{(l)}$.

$$\mu_j^{(l)} = P(s_j(l)) \quad \text{(2.2.2)}$$

$$= \sum_{i=1}^{n} P(s_j(l) \mid s_i(l-1)) P(s_i(l-1)) \quad \text{(2.2.3)}$$

from the law of total probability.

From the inductive hypothesis, we have

$$\mu_j^{(l-1)} = P(s_i(l-1)) = \left( \mu^{(0)} P^{l-1} \right)_j$$

(2.2.4)
Therefore, \( \mu_j^{(l)} \) becomes
\[
\mu_j^{(l)} = \sum_{i=1}^{n} P(s_j(l) \mid s_i(l-1)) \mu_j^{(l-1)} \\
= \sum_{i=1}^{n} P(s_j(1) \mid s_i(0)) \mu_j^{(l-1)} \quad \text{(from the Markov property)} \\
= \sum_{i=1}^{n} \mu_j^{(l-1)} p_{ij} \quad \text{(since multiplication is commutative and)} \\
\quad p_{ij} = P(s_j(1) \mid s_i(0)) \\
= \left[ \mu_1^{(l-1)} \ldots \mu_n^{(l-1)} \right] \left[ \begin{array}{c} p_{1j} \\ \vdots \\ p_{nj} \end{array} \right] \\
= \left( \mu^{(l-1)} P \right)^{j}
\]

Since this is true for each \( 1 \leq j \leq n, j \in \mathbb{N} \), it follows that
\[
\mu^{(l)} = \mu^{(l-1)} P
\]

From the inductive hypothesis, we have \( \mu^{(l-1)} = \mu^{(0)} P^{l-1} \).

Hence,
\[
\mu^{(l)} = \mu^{(0)} P^l = \mu^{(0)} P^{l-1} P
\]

and the proof is complete. \( \square \)

### 2.3 Inhomogeneous Markov Chains

The above result has been proven for a homogeneous Markov chain. We want to know what happens when the process is inhomogeneous. That is, we want to know if the probability distribution can be determined at a time \( k \). An example of such a process is the probability of rainfall in each season.

Let \( P^{(1)}, P^{(2)}, \ldots \) be a set of \( n \times n \) transition matrices where \( p_{ij}^{(w)} \geq 0 \) for all \( 1 \leq i, j \leq n; i, j \in \mathbb{N} \) and
\[
\sum_{j=1}^{n} p_{ij}^{(w)} = 1 \text{ for all } w \geq 1; w \in \mathbb{N} \text{ and } 1 \leq i \leq n; i \in \mathbb{N}.
\]

We define a random process \( (X_0, X_1, \ldots) \) with the following set of vertices \( \{s_1, s_2, \ldots, s_n\} \) as an inhomogeneous Markov chain with transition matrices \( P^{(1)}, P^{(2)}, \ldots \) if for all \( k, i, j \in \{1, \ldots, n\} \) and all \( i_0, \ldots, i_{k-1} \in \{1, \ldots, n\} \)
\[
P(s_j(k+1) \mid s_{i_0}(0), s_{i_1}(1), \ldots, s_{i_{k-1}}(k-1), s_i(k)) \\
= P(s_j(k+1) \mid s_i(k)) \\
= p_{ij}^{(k+1)}
\]

32
Theorem 2.3.1. Let \((X_0, X_1, \ldots)\) be an inhomogeneous Markov process with the Markov digraph \(D\) with the vertex set \(\{s_1, \ldots, s_n\}\). Furthermore, let \(\mu^{(0)}\) denote the initial distribution and let \(P^{(1)}, P^{(2)}, \ldots\) be the transition matrices of \(D\). For any \(k \in \mathbb{N}; k \geq 1\)

\[
\mu^{(k)} = \mu^{(0)} P^{(1)} P^{(2)} \ldots P^{(k)}
\]

Proof. Since the above Markov chain is inhomogeneous, the transition matrices are dependent on time.

i) For \(k = 1\) [We follow a similar procedure as the proof in Theorem 2.2.4]

For each \(j = 1, 2, \ldots, n\)

\[
\mu_j^{(1)} = P(s_j(1)) = \sum_{i=1}^{n} P(s_j(1) | s_i(0)) P(s_i(0)) \quad \text{[The law of total probability]}
\]

\[
= \sum_{i=1}^{n} p_{ij}^{(1)} \mu_i^{(0)}
\]

where \(p_{ij}^{(1)}\) refers to the \((i, j)\)th entry in \(P^{(1)}\).

Since multiplication is commutative,

\[
\mu_j^{(1)} = \sum_{i=1}^{n} \mu_i^{(0)} p_{ij}^{(1)} = \left( \mu^{(0)} P^{(1)} \right)_j
\]

(This indicates the \(j\)th entry in this vector.)

The above is true for all the components in \(\mu^{(1)}\) and therefore,

\[
\mu^{(1)} = \left[ \mu^{(0)} P^{(1)} \right]_1 \left( \mu^{(0)} P^{(1)} \right)_2 \ldots \left( \mu^{(0)} P^{(1)} \right)_n
\]

ii) Assume that

\[
\mu^{(k-1)} = \mu^{(0)} P^{(1)} P^{(2)} \ldots P^{(k-1)}
\]

iii) Consider \(\mu_j^{(k)}\).

\[
\mu_j^{(k)} = P(s_j(k)) = \sum_{i=1}^{n} P(s_j(k) | s_i(k-1)) P(s_i(k-1))
\]

33
\[\sum_{i=1}^{n} p_{ij}^{(k)} \mu_i^{(k-1)} = \sum_{i=1}^{n} \mu_i^{(k-1)} p_{ij}^{(k)} = \left[ \mu_1^{(k-1)} \ldots \mu_n^{(k-1)} \right] \left[ p_{1j}^{(k)} \ldots p_{nj}^{(k)} \right] = \left( \overrightarrow{\mu}^{(k-1)} P^{(k)} \right)_j\]

As before, it follows
\[\overrightarrow{\mu}^{(k)} = \overrightarrow{\mu}^{(k-1)} P^{(k)}\]

From the inductive hypothesis
\[\overrightarrow{\mu}^{(k)} = \overrightarrow{\mu}^{(0)} P^{(1)} P^{(2)} \ldots P^{(k)}\]

Therefore, from the principle of Mathematical Induction, the result is proven. \(\square\)

Some other important results follow in the next subsection.

### 2.4 Other Generalizations for Markov Chains

**Theorem 2.4.1. Chapman-Kolmogorov Equation**

Let \((X_0, X_1, \ldots)\) be a finite homogeneous Markov process with vertex set \(V(D) = \{s_1, s_2, \ldots, s_n\}\) and transition matrix \(P\). For any integer \(l_1\) such that \(0 \leq l_1 \leq l\), where \(l \in \mathbb{Z}\) and \(l \geq 0\), we have
\[p_{ij}(l) = \sum_{x=1}^{n} p_{ix}(l) p_{xj}^{(l_1)} \quad 1 \leq i, j \leq n\]

where \(p_{ij}(l) = P(s_j(l) \mid s_i(0))\).

**Proof.** [We present two ways of proving the above.]

**Proof (1):** Considering \(l\) as in the above, we have
\[p_{ij}(l) = P(s_j(l) \mid s_i(0)) \quad (2.4.1)\]

Using the law of total probability, we get
\[p_{ij}(l) = P(s_j(l) \mid s_i(0)) = \sum_{x=1}^{n} P(s_j(l) \mid s_x(0), s_x(l-l_1)) P(s_x(l-l_1) \mid s_i(0)) \quad (2.4.2)\]
where \( P(s_j(l) \mid s_i(0), s_x(l-l_1)) = P(s_j(l) \mid s_i(0) \cap s_x(l-l_1)) \).

From the property of a Markov process; since \( 0 \leq l-l_1 \leq l \), we have
\[
P(s_j(l) \mid s_i(0), s_x(l-l_1)) = P(s_j(l) \mid s_i(0) \cap s_x(l-l_1)) \quad \text{(since } l-l_1 \geq 0\text{)}
\]
where \( l - (l-l_1) \) is the length of the walk from \( s_x \) to \( s_j \)
\[
= p_{xj}(l-l_1)
\]
and
\[
P(s_x(l-l_1) \mid s_i(0)) = p_{ix}(l-l_1) \quad \text{(2.4.4)}
\]
Substituting the above results into (2.4.3), we get
\[
p_{ij}(l) = \sum_{x=1}^{n} p_{xj}(l_1) p_{ix}(l-l_1)
\]
Since multiplication is commutative, we have
\[
p_{ij}(l) = \sum_{x=1}^{n} p_{ix}(l-l_1) p_{xj}(l_1)
\]
and so we have our result.
Alternatively, we can do the following.

Proof (2): From Theorem 2.2.3, \( p_{ij}(l) \) is the entry in the \( i^{th} \) row and \( j^{th} \) column of \( P^l \).

We have \( P^l = P^{l-l_1} P^{l_1} \).

Now consider row \( i \) of \( P^{l-l_1}, [p_{i1}(l-l_1) \ldots p_{in}(l-l_1)] \), and column \( j \) of \( P^{l_1}, \begin{bmatrix} p_{1j}(l_1) \\ \vdots \\ p_{nj}(l_1) \end{bmatrix} \) of the above.

Hence,
\[
p_{ij}(l) = \sum_{x=1}^{n} p_{ix}(l-l_1) p_{xj}(l_1)
\]

\[\square\]

2.5 Some other interesting results

Consider the Markov process \((X_0, X_1, \ldots)\) with the transition matrix \( P \). If we consider every second entry in the above Markov chain, then the resulting chain is still a Markov chain. However, it will have
the transition matrix $P^2$.

To see this, let $Y_k = X_{2k}$, i.e. $Y_0 = X_0$, $Y_1 = X_2$, ..., $Y_k = X_{2k}$. Now consider the following.

\[
E_1 = P(Y_k = s_j \mid Y_{k-1} = s_i) = P(X_{2k} = s_j \mid X_{2k-2} = s_i)
\]

Since $(X_0, X_1, \ldots)$ is a Markov chain, it satisfies the Markov condition and hence, the most recent state in time will determine where we will go to next in our process. Therefore, $X_{2k-2} = s_i$ is our most recent state in time and so,

\[
E_1 = P(X_{2k} = s_j \mid X_{2k-2} = s_i) = P(Y_k = s_j \mid Y_{k-1} = s_i)
\]

Therefore, $(Y_0, Y_1, Y_2, \ldots)$ is a Markov chain.

Now for each $1 \leq i, j \leq n$,

\[
P(Y_k = s_j \mid Y_{k-1} = s_i) = P(X_{2k} = s_j \mid X_{2k-2} = s_i) = P(s_j(2k) \mid s_i(2k-2))
\]

(From Theorem 2.2.3, this is the probability to transition from $i$ to $j$ in a walk of length 2.)

These are the entries in the transition matrix $P^2$.

Alternatively,

\[
P(Y_k = s_j \mid Y_{k-1} = s_i) = P(X_{2k} = s_j \mid X_{2k-2} = s_i)
\]

\[
= \sum_{x=1}^{n} P(X_{2k} = s_j \mid X_{2k-1} = s_x, X_{2k-2} = s_i) P(X_{2k-1} = s_x \mid X_{2k-2} = s_i)
\]

\[
= \sum_{x=1}^{n} P(s_j(2k) \mid s_x(2k-2)) P(s_x(2k-1) \mid s_i(2k-2))
\]

(Using the Markov property)

\[
= \sum_{x=1}^{n} p_{xj} p_{ix}
\]

(Multiplication is commutative)

Therefore,

\[
P(Y_k = s_j \mid Y_{k-1} = s_i) = \left[ \begin{array}{c} p_{i1} \\ \vdots \\ p_{in} \end{array} \right] \left[ \begin{array}{c} p_{1j} \\ \vdots \\ p_{nj} \end{array} \right] = (P^2)_{ij}
\]
Since the above is true for all \( i \) and \( j \); where \( 1 \leq i, j \leq n; i, j \in \mathbb{Z} \), the transition matrix of \((Y_0, Y_1, \ldots)\) is \( P^2 \).

We can generalize the above for every \( l^{th} \) entry instead of every second entry.

**Proof.** We define \((Y_0, Y_1, \ldots)\) by setting \( Y_k = X_{lk} \) for each \( k = 1, 2, \ldots \). Now consider the following.

\[
E_2 = P (Y_k = s_j \mid Y_0 = s_{a_0}, Y_1 = s_{a_1}, \ldots, Y_{k-2} = s_{a_{k-2}}, Y_{k-1} = s_a)
= P (X_{lk} = s_j \mid X_0 = s_{a_0}, X_1 = s_{a_1}, \ldots, X_{lk-2l} = s_{a_{k-2}}, X_{lk-l} = s_a)
\]

Since \((X_0, X_1, \ldots)\) is a Markov chain, it satisfies the Markov condition and so

\[
E_2 = P (X_{lk} = s_j \mid X_{lk-l} = s_a)
= P (Y_k = s_j \mid Y_{k-1} = s_a)
\]

Therefore, \((Y_0, Y_1, Y_2, \ldots)\) is a Markov chain.

To determine the transition matrix for the above, we do the following.

We consider

\[
E_3 = P (Y_k = s_j \mid Y_{k-1} = s_i)
= P (X_{lk} = s_j \mid X_{lk-l} = s_i)
\]

From time homogeneity, it follows that

\[
E_3 = P (X_l = s_j \mid X_0 = s_i)
= p_{ij}(l)
\]

The above is the probability to transition from \( i \) to \( j \) in a walk of length \( l \). Therefore, the transition matrix for the Markov chain \((Y_0, Y_1, Y_2, \ldots)\) is \( P^l \).

We will now show that not all functions of a Markov chain are Markov chains themselves.

For our example, we need the following results from probability theory.

Let \( A \) and \( B \) be two events. Then

\[
P (A \cup B) = P (A) + P (B) - P (A \cap B)
\]

(2.5.1)

Furthermore, if \( A \) and \( B \) are disjoint events, then \( P (A \cap B) = 0 \). Hence, (2.5.1) becomes

\[
P (A \cup B) = P (A) + P (B)
\]

(2.5.2)

The other result that we need is the “multiplication law”.

37
Let $A$ and $B$ be two events with $P(B) \neq 0$. Then

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \quad (2.5.3)$$

Now, we consider the following example.

**Example 2.5.1.** It is given that $(X_0, X_1, \ldots)$ is a Markov chain with the vertex set/state space \{s_1, s_2, s_3\}. It is further stated that its transition matrix and initial distribution are as follows:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mu^{(0)} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

For $k \geq 0$, define

$$Y_k = \begin{cases} 0 & \text{if } X_k = s_1 \\ 1 & \text{otherwise} \end{cases}$$

Using $P$, we get the following Markov digraph for $(X_0, X_1, \ldots)$.

![Markov Digraph](image)

**Figure 2.5:** The Markov digraph $D$ as a result of $P$.

In order to prove that $(Y_0, Y_1, \ldots)$ is not a Markov chain, we must show that the Markov condition/property is no longer satisfied—i.e.

$$P(Y_k = s_j \mid Y_{k-1} = s_a, Y_{k-2} = s_b, \ldots) \neq P(Y_k = s_j \mid Y_{k-1} = s_a)$$

The proof is by contradiction, so assume that $(Y_0, Y_1, \ldots)$ is a Markov chain.

Firstly, we need to find the transition probabilities for the process $(Y_0, Y_1, \ldots)$.

We consider $P(Y_1 = 1 \mid Y_0 = 0)$. Since $Y_0 = 0$ when $X_0 = s_1$, we have

$$P(Y_1 = 1 \mid Y_0 = 0) = P(Y_1 = 1 \mid X_0 = s_1)$$
We also take note that \( Y_1 = 1 \) if \((X_1 = s_2) \cup (X_1 = s_3)\). Therefore,

\[
P(Y_1 = 1 \mid Y_0 = 0) = P((X_1 = s_2) \cup (X_1 = s_3) \mid X_0 = s_1) = P(X_1 = s_2 \mid X_0 = s_1) + P(X_1 = s_3 \mid X_0 = s_1) - P((X_1 = s_2) \cap (X_1 = s_3) \mid X_0 = s_1) \quad \text{(from (2.5.1))}
\]

Since the process cannot be in both \( s_2 \) and \( s_3 \) at time 1,

\[
((X_1 = s_2) \cap (X_1 = s_3)) = \emptyset
\]

Consequently,

\[
P(Y_1 = 1 \mid Y_0 = 0) = P(X_1 = s_2 \mid X_0 = s_1) + P(X_1 = s_3 \mid X_0 = s_1)
\]

From our Markov digraph \( D \), it follows that

\[
P(X_1 = s_2 \mid X_0 = s_1) = 1 \quad P(X_1 = s_3 \mid X_0 = s_1) = 0
\]

Hence, we have \( P(Y_1 = 1 \mid Y_0 = 0) = 1 \).

Now, let us consider

\[
P(Y_1 = 0 \mid Y_0 = 1) = P(X_1 = s_1 \mid (X_0 = s_2) \cup (X_0 = s_3)) = \frac{P((X_1 = s_1) \cap (X_0 = s_2) \cup (X_0 = s_3)))}{P((X_0 = s_2) \cup (X_0 = s_3))} \quad \text{(using the multiplication law)} \]

\[
= \frac{P((X_1 = s_1) \cap (X_0 = s_2)) \cup ((X_1 = s_1) \cap (X_0 = s_3)))}{P((X_0 = s_2) \cup (X_0 = s_3))} \quad \text{(since } A \cap (B \cup C) = (A \cap B) \cup (A \cap C)) \]

\[
= \frac{P(((X_1 = s_1) \cap (X_0 = s_2)) \cup ((X_1 = s_1) \cap (X_0 = s_3)))}{P((X_0 = s_2) \cup (X_0 = s_3))}
\]

(R1)

Note that the events \((X_1 = s_1) \cap (X_0 = s_2)\) and \((X_1 = s_1) \cap (X_0 = s_3)\) are disjoint events. Similarly, \( X_0 = s_2 \) and \( X_0 = s_3 \) are also disjoint and so

\[
P(((X_1 = s_1) \cap (X_0 = s_2)) \cup ((X_1 = s_1) \cap (X_0 = s_3))) = 0 \quad \text{and} \quad P((X_0 = s_2) \cap (X_0 = s_3)) = 0
\]

Furthermore, since \( P(A \cup B) = P(A) + P(B) \) when \( A \) and \( B \) are disjoint events, we have

\[
P(((X_1 = s_1) \cap (X_0 = s_2)) \cup ((X_1 = s_1) \cap (X_0 = s_3))) = P((X_1 = s_1) \cap (X_0 = s_2)) + P((X_1 = s_1) \cap (X_0 = s_3)) \quad \text{(from the above remark)}
\]

\[
= P(X_1 = s_1 \mid X_0 = s_2) P(X_0 = s_2) + P(X_1 = s_1 \mid X_0 = s_3) P(X_0 = s_3) \quad \text{(from the multiplication law)} \quad \text{(R2)}
\]

39
Similarly,

\[ P ((X_0 = s_2) \cup (X_0 = s_3)) = P (X_0 = s_2) + P (X_0 = s_3) \]

From \( \bar{\mu}^{(0)} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix} \), we have \( P (X_0 = s_2) = \frac{1}{3} \) and \( P (X_0 = s_3) = \frac{1}{3} \). Therefore,

\[ P ((X_0 = s_2) \cup (X_0 = s_3)) = \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \tag{R3} \]

Furthermore, from the transition matrix \( P \)

\[ P (X_1 = s_1 \mid X_0 = s_2) = 0 \]
\[ P (X_1 = s_1 \mid X_0 = s_3) = 1 \]

Using the above, (R2) becomes

\[ P (X_1 = s_1 \mid X_0 = s_2) P (X_0 = s_2) + P (X_1 = s_1 \mid X_0 = s_3) P (X_0 = s_3) = 0 \left( \frac{1}{3} \right) + 1 \left( \frac{1}{3} \right) = \frac{1}{3} \tag{R4} \]

Using (R3) and (R4) in (R1), we have

\[ P (Y_1 = 0 \mid Y_0 = 1) = \frac{1}{2} \]

We only have one transition probability left to consider, \( P (Y_1 = 1 \mid Y_0 = 1) \). Using the complementary rule, we find

\[ P (Y_1 = 1 \mid Y_0 = 1) = 1 - P (Y_1 = 0 \mid Y_0 = 1) = 1 - \frac{1}{2} = \frac{1}{2} \]

From all the above, our “transition matrix” is

\[ [Q] = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \]

The Markov digraph of the above is presented below.
Figure 2.6: The Markov digraph $D_1$ as a result of $[Q]$.

Now consider

$$P (Y_4 = 0 \mid Y_1 = 0, Y_2 = 1, Y_3 = 1)$$

$$= \begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$= \frac{1}{4}$$

However when using the Markov condition,

$$P (Y_4 = 0 \mid Y_1 = 0, Y_2 = 1, Y_3 = 1)$$

$$= P (Y_4 = 0 \mid Y_3 = 1)$$

$$= \frac{1}{2} \neq \frac{1}{4}$$

So we have a contradiction and therefore, $(Y_0, Y_1, \ldots)$ is not a Markov chain.

In the next chapter, a classification of the vertices using the reachability matrix of a Markov digraph will be investigated.
Chapter 3

Classification of Vertices and Types of Markov Chains

In this chapter, we concern ourselves with the classification of vertices, sets of vertices (induced directed subgraphs of Markov digraphs) and Markov chains according to some property/properties. The following references [4] and [35] can be consulted in addition to the previous chapter’s references.

3.1 Vertex/State Classification

A vertex \( s_i \) is called recurrent if for every vertex \( s_j \), the existence of a positive integer \( r_j \), such that the probability to transition from \( s_i \) to \( s_j \) is strictly greater than 0 (i.e. \( p_{ij}(r_j) > 0 \)), implies the existence of an integer \( r_i \) such that \( p_{ji}(r_i) > 0 \).

Since the reachability matrix indicates which vertices are reachable from each other in a digraph, we can use it to define a recurrent state.

Let \( R \) be the reachability matrix of a Markov digraph \( D \) for the Markov process \( (X_0, X_1, \ldots) \). Furthermore, let \( V(D) = \{s_1, s_2, \ldots\} \). A vertex \( s_i \in V(D) \) is called recurrent if for all \( j \) when \( r_{ij} = 1 \), then \( r_{ji} = 1 \)—where \( r_{ij} \) and \( r_{ji} \) are the entries in the reachability matrix \( R \).

Simply stated: “All vertices that are reachable from a vertex \( s_i \) can also reach \( s_i \)”. Therefore, we can always return to \( s_i \).

Suppose that we have at least one vertex \( s_j \) such that it is reachable from \( s_i \), but \( s_i \) is not reachable from \( s_j \), then \( s_i \) is a transient state. It can also be referred to as transient. Next we redefine the above definition in terms of the reachability matrix.

Let \( R \) be the reachability matrix of a Markov digraph \( D \) and let \( V(D) = \{s_1, s_2, \ldots, s_n\} \). A vertex \( s_i \in V(D) \) is said to be transient if there exists at least one integer \( j_1 \); \( 1 \leq j_1 \leq n \) such that \( r_{ij_1} = 1 \), but \( r_{j_1i} = 0 \).

The following two definitions are concerned with some terminology and notations that occur often and, consequently, they need to be highlighted.

Let \( D \) be a Markov digraph with \( V(D) = \{s_1, s_2, \ldots, s_n\} \). For integers \( i \) and \( j \), if a vertex \( s_j \) is reachable from a vertex \( s_i \) in \( D \), then \( s_i \) is said to communicate with \( s_j \) and is denoted by \( s_i \rightarrow s_j \).
The above is, in fact, a directed walk—hence the notation that is used. In fact, we can say: “If there exists a directed walk from $s_i$ to $s_j$, then $s_i$ communicates with $s_j$.”

Now we consider the other definition. Let $D$ be a Markov digraph with $V(D) = \{s_1, s_2, \ldots, s_n\}$. For integers $i$ and $j$, if a vertex $s_j$ is reachable from $s_i$ in $D$ and $s_i$ is reachable from $s_j$, then $s_i$ and $s_j$ *intercommunicate* and this is denoted by $s_i \leftrightarrow s_j$.

The above can be simply stated as: “If there exists at least one directed walk from $s_i$ to $s_j$ and at least one directed walk from $s_j$ to $s_i$, then $s_i \leftrightarrow s_j$.”

We can also redefine the above definitions in terms of the entries of the reachability matrix.

Let $R$ be the reachability matrix of a Markov digraph $D$ and $r_{ij}$ be the entry in the $i^{th}$ row and $j^{th}$ column of $R$.

(a) **For communication:** If $r_{ij} = 1$ in the reachability matrix $R$, then $s_i \rightarrow s_j$.

(b) **For intercommunication:** If $r_{ij} = r_{ji} = 1$ in the reachability matrix $R$, then $s_i \leftrightarrow s_j$.

In turn, the above terms can also be used to define transient and recurrent states.

We now present our own proof of a theorem that is a characteristic of finite Markov chains.

**Theorem 3.1.1.** In a finite Markov chain, not all vertices can be transient.

*Proof.* [The proof is by contradiction.] Let $(X_0, X_1, \ldots)$ be a Markov chain with the Markov digraph $D$, which has the vertex set $\{s_1, s_2, \ldots, s_n\}$. Assume that all the vertices are transient—i.e. $s_1, s_2, \ldots, s_n$ are such that for each $s_i \in V(D)$, there exists at least one $t \in V(D)$ such that $s_i \rightarrow t$ and $t \rightarrow s_i$.

We will now consider a walk in $D$ which will contain some or all of the vertices in $V(D)$. Suppose that we start at some vertex $s_i \in V(D)$. Call it $s_1^*$. Since $s_1^*$ is transient, there exists an $s_j \in V(D)$ such that $s_1^* \rightarrow s_j$ and $s_j \rightarrow s_1^*$. Let $s_2^* = s_j$.

Hence, we have some walk $W_1$ from $s_1^*$ to $s_2^*$. 

$$W_1 : s_1^* \rightarrow s_2^*$$

Similarly for $s_2^*$, there exists an $s_3^*$ such that $s_2^* \rightarrow s_3^*$ and $s_3^* \rightarrow s_2^*$.

Assume $s_3^* \rightarrow s_1^*$. However, $s_1^* \rightarrow s_2^*$ and so we have the contradiction, $s_3^* \rightarrow s_2^*$. Therefore, $s_3^* \nrightarrow s_1^*$.

Hence,

$$W_2 : s_1^* \rightarrow s_2^* \rightarrow s_3^*$$

Assume that we can continue the above to some $s_{(k-1)^*}$.

For $s_{(k-1)^*}$, there exists an $s_k^*$ such that $s_{(k-1)^*} \rightarrow s_k^*$ and $s_k^* \rightarrow s_{(k-1)^*}$. Consider two cases for $s_k^*$:

- It is the last vertex in $V(D)$.
- It is the last vertex in a weakly-connected or disconnected component of $D$.

Let $s_k^*$ be the last vertex in $V(D)$—i.e. $s_k^* = s_n^*$.

We have $s_1^* \rightarrow s_2^* \rightarrow s_3^* \rightarrow \ldots \rightarrow s_{(n-1)^*} \rightarrow s_n^*$.

Since $s_n^*$ is transient, this implies that there is at least one vertex $t \in V(D)$ such that $s_n^* \rightarrow t$ and...
However, if $s_{n^*}$ communicates with any of the vertices in $V(D)$, then we will contradict some or all of the transient properties of the vertices in $V(D)$. Hence, we have a contradiction and so not all of the vertices in $V(D)$ can be transient—so at least one vertex is recurrent.

Similarly, we consider the case when $s_{k^*}$ is the last vertex in a subdigraph of $D$ that is weakly-connected or disconnected.

We have $s_1 \to s_2 \to s_3 \to \cdots \to s_{(k-1)^*} \to s_{k^*}$ where $1 \leq k^* < n; k^* \in \mathbb{N}$.
Let $s_{k^*}$ be a vertex in the subdigraph induced by $U = \{s_1, s_2, s_3, \ldots, s_{(k-1)^*}, s_{k^*}\}$.
Since $s_{k^*}$ does not communicate with any of the vertices in $V(D-U)$, it communicates with itself or the other vertices in $U$—which contradicts the transient property of some or all of the vertices in this set. Hence, at least one vertex in the set $U$ must be recurrent and therefore, there is at least one recurrent vertex in $D$.

Therefore, in a finite Markov chain, not all vertices can be transient.

However, all vertices in a Markov digraph $D$ can be recurrent. We consider a few examples with regard to the above.

**Example 3.1.2.**

\[
D:
\begin{array}{cccc}
1 & 2 & \cdots & n \\
\end{array}
\]

Figure 3.1: The Markov digraph $D$ with $V(D) = \{1, 2, \ldots, n\}$ and recurrence.

A set of vertices that are disjoint with loops that each have a transitional probability of 1.

**Example 3.1.3.**

\[
C_n:
\begin{array}{cccc}
1 & 2 & 3 & \\
1 & 1 & 1 & \\
\end{array}
\]

Figure 3.2: The Markov digraph $C_n$ which is a directed $n$-cycle.

A directed $n$-cycle of vertices, where $V(C_n) = \{1, 2, \ldots, n\}$, is also an example in which all vertices are recurrent.
Obviously, for a vertex to be recurrent it must be contained in some directed cycle that is a subdigraph of its Markov digraph.

### 3.2 Reducibility and Irreducibility of Markov Chains

A Markov chain \((X_0, X_1, \ldots)\), whose Markov digraph \(D\) has a vertex set \(\{s_1, s_2, \ldots, s_n\}\) and a transition matrix \(P\), is *irreducible* if, for all \(s_i\) and \(s_j\), we have \(s_i \leftrightarrow s_j\). Otherwise, the chain is *reducible*.

From the definition of intercommunication, we have the following.

**Theorem 3.2.1.** Characterization

A Markov chain is irreducible if and only if its Markov digraph is strong.

**Proof.**

i) To prove the forward implication, assume that a Markov chain is irreducible. Let \(D\) be the Markov digraph with the vertex set \(V(D) = \{s_1, s_2, \ldots, s_n\}\). For all pairs of vertices \(s_i, s_j\) with \(1 \leq i, j \leq n\), we have \(s_i \leftrightarrow s_j\). More specifically, this means that there exists at least one directed walk from \(s_i\) to \(s_j\) and at least one directed walk from \(s_j\) to \(s_i\). Since this is true for all vertices in \(V(D)\), this implies that \(D\) is a strongly-connected digraph.

ii) To prove the reverse implication, assume that the Markov digraph \(D\) of a Markov chain \((X_0, X_1, \ldots)\) is strong. Let \(V(D) = \{s_1, \ldots, s_n\}\). Since \(D\) is strong; by definition, for every two vertices \(s_i\) and \(s_j\) with \(1 \leq i, j \leq n\), we have at least one directed \(s_i \rightarrow s_j\) walk and at least one directed \(s_j \rightarrow s_i\) walk. This is the same as saying that \(s_j\) is reachable from \(s_i\) and \(s_i\) is reachable from \(s_j\). Therefore, we have \(p_{ij}(l_1) > 0\) and \(p_{ji}(l_2) > 0\), where \(l_1\) and \(l_2\) are some positive integers. This is the definition of intercommunication and so \(s_i \leftrightarrow s_j\) for all \(1 \leq i, j \leq n\). Consequently, the Markov chain is irreducible.

The following results also indicate when a Markov chain is irreducible.

**Theorem 3.2.2. Irreducibility in terms of the Reachability Matrix**

A Markov chain \((X_0, X_1, \ldots)\) with \(V(D) = \{s_1, \ldots, s_n\}\) is irreducible if and only if its reachability matrix is an \(n \times n\) matrix of ones.

**Proof.**

i) To prove the forward implication, let the Markov process \((X_0, X_1, \ldots)\) be an irreducible Markov chain. Let \(D\) be the Markov digraph with the vertex set \(V(D) = \{s_1, \ldots, s_n\}\). Since the Markov chain is irreducible, it follows by definition that \(s_i \leftrightarrow s_j\) for all \(1 \leq i, j \leq n\). Consequently, we define the reachability matrix \(R\) of the Markov chain. Since \(s_i \leftrightarrow s_j\) for all \(i\) and \(j\), this implies \(r_{ij} = r_{ji} = 1\) for \(1 \leq i, j \leq n\) and therefore, \(R\) is an \(n \times n\) matrix of ones.

ii) To prove the reverse implication, let \(R\) be the reachability matrix of a Markov chain \((X_0, X_1, \ldots)\). Furthermore, assume that \(R\) is an \(n \times n\) matrix that has 1’s for all its entries. Thus, \(r_{ij} = 1\) for \(1 \leq i, j \leq n\); \(i, j \in \mathbb{Z}\). Define \(V(D) = \{s_1, \ldots, s_n\}\). Since \(r_{ij} = r_{ji} = 1\) for all \(i\) and \(j\), this implies that \(s_i \leftrightarrow s_j\) for all \(i\) and \(j\) and so this Markov chain is irreducible.

The above is an important tool in proving the reducibility of a Markov chain. Consider the forward implication of the above theorem: “If a Markov chain is irreducible, then its reachability matrix is an \(n \times n\) matrix of ones.”

Now consider the contraposition of the above: “If the reachability matrix of a Markov chain is an \(n \times n\) matrix of ones, then the Markov chain is irreducible.”
matrix whose entries are not all ones, then the Markov chain is reducible."

Consider the following example that is concerned with the above statements.

**Example 3.2.3.** Consider the Markov chain \((X_0, X_1, \ldots)\) with the Markov digraph \(D\) and the vertex set \(V(D) = \{1, 2, 3, 4\}\).

\[
D:
\]

![Markov digraph D](image)

Figure 3.3: The Markov digraph \(D\) and reducibility.

We can see that the above Markov chain is the union of two disjoint Markov digraphs. The Markov chain is not irreducible as \(\{1, 2\}\) can only reach each other. Similarly, only 3 and 4 can reach each other. Therefore, the Markov chain is reducible.

The relevant matrices that are involved are as follows.

\[
A(D) = \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
\end{bmatrix} = R(D)
\]

Since the above reachability matrix is not a matrix of only ones, the Markov chain is reducible.

The transition matrix is

\[
P = \begin{bmatrix}
0.3 & 0.7 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 \\
0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0.1 & 0.9 \\
\end{bmatrix}
\]

Let \(D_1\) be the directed subgraph of \(D\) that is induced by \(\{1, 2\}\). Therefore, \(D_1\) is depicted as follows.

\[
D_1:
\]

![Directed subgraph D1](image)

Figure 3.4: The directed subgraph of \(D, D_1\).
Furthermore, the transition matrix of $D_1$ is $[P(D_1)] = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{bmatrix}$. Clearly, the adjacency and reachability matrices of $D_1$ are both equal to $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Notice that $D_1$ is also a Markov chain. In fact, by Theorem 3.2.2, it is an irreducible Markov chain.

Similarly, if we consider $D_2 = \langle \{3, 4\} \rangle$, we will find that $D_2$ is also an irreducible Markov chain.

Therefore, for long term behaviour (which we will explore later) of the initial Markov chain $D$, we can just consider the long term behaviours of $D_1$ and $D_2$ as they are disjoint.

Keeping the above example in mind, we now consider a corollary in which the Markov digraph is the union of disjoint digraphs.

**Corollary 3.2.4.** Let $(X_0, X_1, \ldots)$ be a Markov chain with the Markov digraph $D$ and vertex set $\{s_1, \ldots, s_n\}$. If $D$ is the union of disjoint digraphs, then its Markov chain is reducible.

**Proof.** Assume that $(X_0, X_1, \ldots)$ is a Markov chain with the Markov digraph $D = \bigcup_{i=1}^{k} D_i$ and the vertex set $\{s_1, \ldots, s_n\}$.

Consider the vertices $s_{i^*} \in V(D_i)$ and $s_{j^*} \in V(D_j)$, where $i^* \neq j^*$ and $D_i$ and $D_j$ are disjoint directed subgraphs of $D$. Since $D_i$ and $D_j$ are disjoint, $s_{i^*}$ and $s_{j^*}$ cannot communicate with each other. Therefore, if we now consider the reachability matrix of $D$, we will have $r_{i^*, j^*} = r_{j^*, i^*} = 0$ and by Theorem 3.2.2, the Markov chain is not irreducible and is, therefore, reducible.

The converse of the above is not true and will be illustrated by the following example.

**Example 3.2.5.** Let $(X_0, X_1, \ldots)$ be a Markov process with the following Markov digraph $D$ and the vertex set $\{1, 2, 3, 4\}$. 

![Figure 3.5: The Markov digraph D.](image)

The adjacency matrix $A(D)$ of the above is 

$$
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

Similarly, the reachability matrix $R(D) = A(D)$ and is clearly not a matrix of only ones. Therefore, by Theorem 3.2.2, the above Markov chain is not irreducible. In the long-term behaviour, (no matter in which state we started) we will eventually be in 4 and remain in 4. The vertex 4 is known as an **absorbing state**.

From this example, we can see that if a Markov chain is reducible, then it is not always true that its Markov digraph is the union of disjoint digraphs.

Let $(X_0, X_1, \ldots)$ be a Markov chain with the Markov digraph $D$ and vertex set $V(D) = \{s_1, \ldots, s_n\}$. 

47
A *single chain* $W$ is a set of recurrent vertices that intercommunicate with each other. Vertices in a single chain $W$ may be reached from vertices in $D - W$.

Consider the following example from Drake [13].

**Example 3.2.6.** Consider the following Markov digraph $D$.

![Figure 3.6: The Markov digraph D.](image)

The subset of recurrent vertices of $V(D)$ is the set $\{s_1, s_3, s_4, s_6\}$. $s_2$ and $s_5$ are transient vertices. To verify these statements, consider the reachability matrix of $D$, $R(D)$.

$$R(D) = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 
\end{bmatrix}$$

By Theorem 3.2.2, the reachability matrix contains 0’s and so the above Markov chain is reducible.

To verify that $s_2$ and $s_5$ are transient, we observe that all entries $r_{i2} = 0$ for $i \neq 2$ or 5 and $r_{55} = 0$; $i \neq 2$ or 5. Therefore, $s_2$ and $s_5$ are not reachable from $\{s_1, s_3, s_4, s_6\}$ and consequently, they are transient.

The vertices $s_1$ and $s_4$ intercommunicate with each other as $r_{14} = r_{41} = 1$. Furthermore, there does not exist a walk from $s_1$ or $s_4$ to any of the remaining vertices in $D$ such that we are not able to return to $s_1$ or $s_4$. Therefore, $s_1$ and $s_4$ are recurrent.

Similarly, $s_3$ and $s_6$ are also recurrent and intercommunicate with each other.

Using the definition of a single chain, we have two chains:

- $W_1 = \{s_1, s_4\}$ and
- $W_2 = \{s_3, s_6\}$

as $s_1$ and $s_4$ are recurrent states that intercommunicate with each other and $s_3$ and $s_6$ are recurrent states that also intercommunicate with each other.

It is not always true to say: “If a Markov chain has one chain, then it is irreducible.” We now consider an example which appeared earlier in this chapter.

48
Example 3.2.7.

\[ D: \]

\[ \begin{array}{cccc}
1 & 1 & 2 & 3 \\
2 & 3 & 4 & 1 \\
3 & 4 & 1 & 2 \\
4 & 1 & 2 & 3 \\
\end{array} \]

Figure 3.7: The Markov digraph \( D \) with its reachability matrix below.

\[ R(D) = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \]

From the definition of a single chain, 1, 2, and 3 are transient and therefore cannot be in a chain. Since 4 is absorbing, it is recurrent. Furthermore, it is also a single chain. As a consequence of Theorem 3.2.2, \( R(D) \) contains 0's and therefore, the Markov chain is reducible. Clearly, we have one chain, but the Markov chain is reducible. However, it would be true to state: “If a Markov chain has a single chain \( W \) which contains all the vertices in the Markov digraph \( D \), then it is irreducible.”

Proof. Assume that we have a Markov chain with a single chain \( W = V(D) \). From the definition of a single chain, all vertices are recurrent and communicate with each other. Therefore, by Theorem 3.2.1, the Markov digraph \( D \) is strong and so the Markov chain is irreducible.

3.3 Periodicity and Aperiodicity of States and Markov Chains

Firstly, we concern ourselves with the periodicity of a vertex \( s_i \) in a Markov digraph \( D \) of a Markov chain.

We remind the reader that for a finite or an infinite set of numbers, \( \{a_1, a_2, \ldots\} \), \( gcd \{a_1, a_2, \ldots\} \) is the greatest common divisor of \( a_1, a_2, \ldots \).

Let \((X_0, X_1, \ldots)\) be a Markov chain with the Markov digraph \( D \), vertex set \( V(D) = \{s_1, \ldots, s_n\} \) and transition matrix \( P \). Define the period \( d(s_i) \) of a recurrent vertex \( s_i \) to be \( gcd \{k \geq 1 \mid p_{ii}(k) > 0\} > 1 \), where \( p_{ii}(k) \) is the entry in the \( i^{th} \) row and \( i^{th} \) column of \( P^k \). Since \( s_i \) is recurrent, the set \( gcd \{k \geq 1 \mid p_{ii}(k) > 0\} \) will always be non-empty. Alternatively, we can define a vertex \( s_i \) to be periodic if it is recurrent and there only exists \( s_i - s_i \) walks of lengths \( k = d, 2d, \ldots \) where \( d > 1 \).

Say \( d = gcd \{k \geq 1 \mid p_{ii}(k) > 0\} \). \( s_i \) is recurrent as we can eventually return to \( s_i \) in walks of length \( qd \), where \( q \geq 1; q \in \mathbb{N} \). For all \( l \in \mathbb{N} - \{qd \mid q \geq 1; q \in \mathbb{N}\} \), \( p_{ii} = 0 \) if \( d > 1 \). If \( d = 1 \), then we have no periodicity and consequently, vertex \( s_i \) is defined to be aperiodic.

As a result of the above, we have the following definitions.
A Markov chain is aperiodic if all the vertices in the Markov digraph are aperiodic. If one or more vertices are periodic, then the Markov chain is defined to be periodic.
Before we proceed, we need the following Lemma from number theory.

**Lemma 3.3.1.** from [19]

Let \( A = \{a_1, a_2, \ldots \} \) be a set of positive integers that has the following properties:

a) \( \gcd \{a_1, a_2, \ldots \} = 1 \) [the \( a_i \)'s have no common factors].

b) It is closed under addition (i.e. if \( a \in A \) and \( a' \in A \), then \( a + a' \in A \)).

Then there exists an integer \( N \) such that \( k \in A \) for all \( k \geq N \).

We now consider the theorem in which the above is used.

**Theorem 3.3.2.** Let \((X_0, X_1, \ldots)\) be an aperiodic Markov chain which has the Markov digraph \( D \) with \( V(D) = \{s_1, \ldots, s_n\} \). For \( s_i \in V(D) \) when \( \{i = 1, \ldots, n\} \), there exists an integer \( N \) such that there is always at least one walk of length \( k \) for every integer \( k \geq N \) that starts and ends at \( s_i \) in the Markov digraph \( D \).

**Proof.** For the proof, we will denote the length of a walk \( W \) by \( l(W) \).

For each \( s_i \in V(D) \), let \( A_i = \{k \geq 1 \mid \exists(s_i \rightarrow s_i); l(s_i \rightarrow s_i) = k\} \). Therefore, \( A_i \) is the set of all lengths of all the walks from vertex \( s_i \) back to \( s_i \).

i) We want to prove that \( \gcd \{k_1^{(i)}, k_2^{(i)}, \ldots\} = 1 \), where \( k_j^{(i)} \in A_i \) for all \( j \).

We assume the contrary that \( \gcd \{k_1^{(i)}, k_2^{(i)}, \ldots\} = d; d > 1 \). Now consider a vertex \( s_i \) which is periodic if it is recurrent and there only exists \( s_i - s_i \) walks of lengths \( k = d, 2d, \ldots \) where \( d > 1 \).

Since we have assumed that all the lengths have a \( \gcd \) of \( d \), \( s_i \) is periodic. However, if the Markov chain is aperiodic, then all vertices \( s_i; i \in \{1, \ldots, n\} \) are aperiodic. Therefore, we have a contradiction and thus \( \gcd \{k_1^{(i)}, k_2^{(i)}, \ldots\} = 1 \).

ii) We now have to prove that \( A_i \) is closed under addition. Let \( a, a' \in A_i \). Therefore, we have \( s_i \rightarrow s_i \) walks of lengths \( a \) and \( a' \). Let \( W_1 \) be the walk of length \( a \) and \( W_2 \) be the walk of length \( a' \).

Let

\[
W_1 : s_i, \nu_1, \nu_2, \ldots, \nu_{a-1}, s_i \quad l(W_1) = a
\]

\[
W_2 : s_i, \mu_1, \mu_2, \ldots, \mu_{a'-1}, s_i \quad l(W_2) = a'
\]

Now define \( W' \) as the walk along \( W_1 \) and then along \( W_2 \); that is,

\[
W' : s_i, \nu_1, \nu_2, \ldots, \nu_{a-1}, s_i, \mu_1, \mu_2, \ldots, \mu_{a'-1}, s_i
\]

Clearly, \( l(W') = l(W_1) + l(W_2) = a + a' \). Furthermore, \( W' \) is an \( s_i \rightarrow s_i \) walk and therefore, \( a + a' \in A_i \).

Since \( A_i \) satisfies the conditions of Lemma 3.3.1, there exists an integer \( N_i \) such that there is always at least one walk of length \( k \geq N_i \) that starts and ends at \( s_i \).

For each \( i \in \{1, \ldots, n\} \), we have an \( N_i \). Therefore, we let \( N = \max \{N_1, \ldots, N_n\} \) and the proof is complete. 

\[\square\]
Now let us consider two corollaries of the above theorem.

**Corollary 3.3.3.** Let \((X_0, X_1, \ldots)\) be an irreducible aperiodic Markov chain with the Markov digraph \(D\) which has the vertex set \(V(D) = \{s_1, \ldots, s_n\}\). Then there exists an \(M\) such that for all integers \(k \geq M\), there is always at least one walk from \(s_i \in V(D)\) to \(s_j \in V(D)\); \(i, j \in \{1, 2, \ldots, n\}\).

**Proof.** Since the Markov chain is aperiodic; by Theorem 3.3.2, there exists an integer \(N\) such that for all \(k \geq N\), there is at least one walk of length \(k\) from \(s_i\) to \(s_i\) for all \(i \in \{1, 2, \ldots, n\}\).

Consider two vertices \(s_i, s_j \in V(D)\) and keep them fixed. From the hypothesis, the Markov chain is irreducible; therefore, by Theorem 3.2.1, \(s_j\) is reachable from \(s_i\) and as a result, we can find some positive integer \(n_{i,j}\) such that it is the length of the walk from \(s_i\) to \(s_j\).

Define \(M_{i,j} = N + n_{i,j}\). We want to prove that for all \(m \geq M_{i,j}\), there is always at least one walk from \(s_j\) to \(s_j\) such that the length of the walk is \(m\).

Consider the (directed) walk from \(s_i\) to \(s_i\) of length \(l = m - n_{i,j} \geq N\) and then the walk from \(s_i\) to \(s_j\) which has the length \(n_{i,j}\). Clearly, this is also a walk from \(s_i\) to \(s_j\), the length of which is \(l + n_{i,j} \geq N + n_{i,j} = M_{i,j}\).

Since this is true for all \(i, j \in \{1, 2, \ldots, n\}\), we let \(M = \max \{M_{1,1}, M_{1,2}, \ldots, M_{n,n}\}\) and the proof is complete. □

**Corollary 3.3.4.** If a Markov chain is irreducible and its Markov digraph \(D\) with \(V(D) = \{s_1, \ldots, s_n\}\) has a vertex \(s_i\) with a loop, then the Markov chain is aperiodic.

**Proof.** Since the Markov chain is irreducible, by Theorem 3.2.2, the reachability matrix \(R(D)\) consists of only ones and hence all vertices are reachable from each other.

We consider a proof by contradiction. Suppose that the Markov chain is periodic. Therefore, at least one vertex \(s_j\) is periodic. Say \(d\) is the period of \(s_j\). Since all vertices intercommunicate, \(s_j\) can reach \(s_i\) and \(s_i\) can reach \(s_j\). Therefore, we have a walk from \(s_i\) to \(s_j\) of some length that is a multiple of \(d\) by assumption, say the length of this walk is \(qd\). Furthermore, define the walk to be \(W_1: s_j, \nu_1, \nu_2, \ldots, \nu_{t_1}, s_i, \mu_1, \ldots, \mu_{t_2}, s_j\) where \(s_i\) occurs only once. Now let us consider the walk \(W_2\), which is nearly the same as \(W_1\), except that we stay at the vertex \(s_i\) for one of the steps—i.e.

\[
W_2: s_j, \nu_1, \nu_2, \ldots, \nu_{t_1}, s_i, s_i, \mu_1, \ldots, \mu_{t_2}, s_j
\]

This is also a \(s_j \rightarrow s_j\) walk. However, it has a length of \(qd + 1\) which contradicts the periodicity of \(s_j\) and so the assumption is wrong. Therefore, the Markov chain is aperiodic. □

Let us consider an exercise from [19] on page 27.

**Example 3.3.5.** Orientated Random Walk on a Torus

Let \(a\) and \(b\) be positive integers. Consider the Markov chain with Markov digraph \(D\) where

\[
V(D) = \{(x, y) \mid x \in \{0, 1, \ldots, a - 1\}, y \in \{0, 1, \ldots, b - 1\}\}
\]

and the following transition mechanism: If the chain is at vertex \((x, y)\) at time \(k\), then at time \(k + 1\), it moves either to \(((x + 1) \mod a, y)\) or to \((x, (y + 1) \mod b)\), each with probability \(\frac{1}{2}\).

Firstly, we prove that this Markov chain is irreducible.
i) The Markov digraph $D$ is depicted below.

![Diagram of the Markov digraph](image)

Figure 3.8: The Markov digraph of the defined transition mechanism.

We also take note of the fact that the above Markov digraph has $|V(D)| = ab$ vertices as this will be used in the R-code.

Initially, we give the R-code for Warshall’s Algorithm and then use it in the R-code on some directed toruses which have been constructed in R. Please note that all the R-code that is presented throughout the discussion on Markov chains is the work of the author.

The code for Warshall’s Algorithm as well as an example.

```r
library(igraph)

# Adjacency matrix A
A <- matrix(c(1,0,0,1,0,0,1,1,1,0,1,1,
              rep(0,5),1,1,rep(0,6),1,
              0,1,0,1,0,0,1,rep(0,3)), nrow=6, ncol=6, byrow=T)

G1 <- graph.adjacency(A)

booladdRow <- function(RVec, Vec)
{
  temp <- Vec + RVec
  for (i in 1:length(Vec))
    return(temp)
}
```

52
```r
if(temp[i]>=2)
{
    temp[i]<-1
}

return(temp)

R<-function(G)
{
    Q<-get.adjacency(G)
    N<-length(Q[,1])
    for(i in 1:N)
    {
        for(j in 1:N)
        {
            if(Q[j,i]==1)
            {
                Q[j,]<-booladdRow(Q[i,],Q[j,])
            }
        }
    }
    return(Q)
}

Output:
> R(G1)
[1,] 1 0 0 1 0 0
[2,] 1 1 1 1 1 1
[3,] 0 0 1 0 0 1
[4,] 1 0 0 1 0 0
[5,] 1 1 1 1 1 1
[6,] 0 0 1 0 0 1
```

The R–code for a torus, which includes a couple of examples.

R–code for examples 1 and 2:

```r
library(igraph)

Torus<-function(a,b)
{
    Vec_from<-rep("",2*a*b) #There are ab vertices with outdegrees of 2
    Vec_to<-rep("",2*a*b) # There are ab vertices with indegrees of 2
    k<-1 # a counter
```
for(i in 0:(a-1))
{
  for(j in 0:(b-1))
  {
    vert1<-paste("",i%%a,"",j%%b,"") # mod(a) is % a
    vert2<-paste("",(i+1)%%a,"",j%%b,"")
    vert3<-paste("",i%%a,"",(j+1)%%b,"")
    Vec_from[k:(k+1)]<-rep(vert1,2)
    Vec_to[k]<-vert2
    Vec_to[k+1]<-vert3
    ##update k
    k<-k+2
  }
}
VerLabel<-unique(Vec_from)
d<-data.frame(from=Vec_from,to=Vec_to,Labels=Vec_from)
G<-graph.data.frame(d,directed=TRUE)
G <- set.vertex.attribute(G, "label", value=VerLabel)
G <- set.edge.attribute(G, "weight", value=rep(1/2,ecount(G)))
tkplot(G,edge.label=get.edge.attribute(G,"weight"))

return(G)

Output:

#Example 1
#-----------------

> G1<-Torus(2,3)

In the above digraph, the labels on the vertices represent positional x– and y–coordinates; the num-
bers on the arcs are transitional probabilities.

```r
> which(R(G1)==0)  # checks for zeros
> # Output: integer(0) indicates none are zero. Therefore, all values are 1's.
> # Markov Chain is irreducible

#Example 2
#-------------------------------------------
> G2<-Torus(4,8)
```

Using Warshall’s algorithm, we find that the reachability matrix \( R(D) \) is always a matrix of only ones. This can be demonstrated by observing that for each \( i; i \in \{0, \ldots, a-1\} \), the elements of \{\((i,0), (i,1), \ldots, (i,b-1)\)\} all communicate with each other. Similarly, for \( j; j \in \{0, \ldots, b-1\} \), the elements of \{\((0,j), \ldots, (a-1,j)\)\} also communicate with each other. Consequently, all elements communicate and \( R(D) \) is a matrix of ones. Therefore, by Theorem 3.2.2, the Markov chain is irreducible.

ii) Now we prove that the given Markov chain of an \((a \times b)\)-directed torus is aperiodic if and only if \( \gcd(a,b) = 1 \).

a) Proving: If a Markov chain of an \((a \times b)\)-directed torus is aperiodic, then \( \gcd(a,b) = 1 \). To start, we will prove the following lemma.

**Lemma 3.3.6.** Let \( a \) and \( b \) be positive integers. Let \( D \) be a digraph and define \( V(D) = \{(x,y) \mid x \in \{0, 1, \ldots, a-1\}, y \in \{0, 1, \ldots, b-1\}\} \) with outward arcs to \([(x+1) \mod a, y]\) and \([(x, (y+1) \mod b)]\). Then the length of any walk from \((i,j)\) to \((i,j)\) for \( i \in \{0, \ldots, a-1\} \) and \( j \in \{0, \ldots, b-1\} \) is a linear combination of \( a \) and \( b \). That is, if \( l \) is the length of the walk, then \( l = k_1a + k_2b; k_1, k_2 \in \mathbb{N}_0 \).

**Proof.** From the hypothesis, each step is either vertical (the second coordinate changes) or horizontal (the first coordinate changes). Suppose that we start at some vertex \((i,j)\).
If the total number of vertical steps is not a multiple of $b$, then we will not end with a second coordinate of $j$. That is, if we have $l = k_2b + c$ vertical steps where $k_2 \in \mathbb{N}_0$ and $0 < c < b$; $c \in \mathbb{N}$, then we end with a second coordinate of $(j + l) \mod b = (j + c) \mod b \neq j$.

Similarly, if the total number of horizontal steps is not a multiple of $a$, then we will not end with a first coordinate of $i$. That is, if we have $l = k_1a + c$ horizontal steps where $k_1 \in \mathbb{N}_0$ and $0 < c < a$; $c \in \mathbb{N}$, then we end with a first coordinate coordinate of $(i + l) \mod a = (i + c) \mod a \neq i$.

From the above, we find that all self-walks must be a linear combination of $a$ and $b$.

Going back to a). Assume that the Markov chain is aperiodic. Furthermore, assume the contrary that $\gcd(a, b) = c \neq 1$. Hence $a = q_1c$ and $b = q_2c$, where $q_1, q_2, c \in \mathbb{N}$. Consider any vertex $(i, j) \in V(D)$.

From Lemma 3.3.6, we note that the length of any walk from $(i, j)$ back to $(i, j)$ is a linear combination of $a$ and $b$—i.e. length($(i, j) \rightarrow (i, j)) = k_1a + k_2b$. Using our assumption $a = q_1c$ and $b = q_2c$, we get length($(i, j) \rightarrow (i, j)) = (k_1q_1 + k_2q_2)c$. Consequently, all walks from $(i, j)$ back to $(i, j)$ have lengths that are multiples of $c$. This contradicts the fact that the given Markov chain is aperiodic. Hence, our assumption that $\gcd(a, b) = c$ is wrong and so $\gcd(a, b) = 1$.

b) Proving: If $\gcd(a, b) = 1$, then the above Markov chain is aperiodic.

We consider a proof by contraposition. That is, if the above Markov chain is periodic, then $\gcd(a, b) \neq 1$.

Therefore, assume that the Markov chain is periodic. From the definition of a periodic Markov chain, at least one vertex $(i, j)$ is periodic. Let the period of it be $d$, where $d > 1$. $a$ and $b$ are possible lengths of self-walks for $(i, j)$. So $a = q_1d$ and $b = q_2d$, where $q_1, q_2 \in \mathbb{N}$. Hence, $\gcd(a, b) \geq d \neq 1$. Therefore, $\gcd(a, b) \neq 1$.

From the above, we have our result.

From a) and b), we have that the given Markov chain of an $(a \times b)$-directed torus is aperiodic if and only if $\gcd(a, b) = 1$.

We still need to consider a discussion on reversible Markov chains, but before that is considered we need to understand some things about stationary distributions, which will be presented in the next chapter.
Chapter 4

Stationary Distributions and Reversible Markov Chains

For this chapter, we will present the necessary theory for stationary distributions and reversible Markov chains (see [1], [13], [19], [28] and [29]). Thereafter, we will see how these apply to undirected graphs—refer to [1], [4], [21] and [35].

4.1 Stationary Distributions

Let \((X_0, X_1, \ldots)\) be a Markov chain with the Markov digraph \(D\) which has the transition matrix \(P\). We have seen and demonstrated in chapter 2 that \(\pi^{(k)} = \pi^{(0)}P^k\). Sometimes as \(k \to \infty\), we get a stationary vector say \(\pi\) where \(\pi P = \pi\).

We now define a stationary distribution for a Markov chain. Let \((X_0, X_1, \ldots)\) be a Markov chain with the Markov digraph \(D\), where \(V(D) = \{s_1, \ldots, s_n\}\) and \(P\) is the transition matrix. A row vector \(\pi = [\pi_1, \pi_2, \ldots, \pi_n]\) is said to be a stationary distribution for the Markov chain if \(\pi P = \pi\) and the sum of its components is 1—i.e. \(\sum_{i=1}^{n} \pi_i = 1\).

Now consider taking the transpose of the last equation. We get \(P^t \pi^t = \pi^t\).

For convenience, we recall the following definitions about eigenvectors and eigenvalues in Linear Algebra (see [26]).

Let \(A\) be an \(n \times n\) matrix. A scalar \(\lambda\) of \(A\) is called an eigenvalue of \(A\) if there exists a nonzero vector \(\vec{v}\) (a column vector) such that \(A\vec{v} = \lambda\vec{v}\). The vector \(\vec{v}\) is called the eigenvector corresponding to the eigenvalue \(\lambda\).

Looking at \(P^t \pi^t = \pi^t\), notice that \(\lambda = 1\) is the eigenvalue and that \(\vec{v} = \pi^t\) is the eigenvector that corresponds to this eigenvalue. Therefore, to calculate the stationary distribution, we must find the eigenvectors of \(P^t\) that are associated with the eigenvalue of \(\lambda = 1\) such that the sum of all the components of an eigenvector is equal to 1. The transpose of this eigenvector is a stationary distribution.

Next we define the concepts of hitting times, mean hitting times and mean return times.
Let \((X_0, X_1, \ldots)\) be a Markov chain with the vertex set \(V(D) = \{s_1, \ldots, s_n\}\). We define the hitting time as
\[ T_{i,j} = \min \{ k \geq 1 \mid X_k = s_j \} \text{ given that } X_0 = s_i \]
If two vertices in a digraph never reach each other, then the distance/length between them is defined to be \(\infty\). Therefore, it is convention to write \(T_{i,j} = \infty\) when \(s_j\) is not reachable from \(s_i\).

The mean hitting time is defined to be the expected value of the hitting time. In other words,
\[
\tau_{i,j} = \mathbb{E}[T_{i,j}] = \sum_{k=1}^{\infty} kP(T_{i,j} = k) = \sum_{k=1}^{\infty} P(T_{i,j} \geq k)
\]
where \(\mathbb{E}\) denotes the expectation function.

If \(j = i\) in the definition of the mean hitting time, then \(\tau_{i,i}\) is known as the mean return time.

To clarify the alternative form of the expectation in the above definition, we state Tonelli’s Theorem for series and present a lemma which will use the latter theorem to prove it.

**Theorem 4.1.1. Tonelli’s Theorem for series**
For \(n, m \in \mathbb{N}\), let \((x_{n,m})\) be a doubly infinite sequence of extended non-negative real numbers; i.e.
\[
\sum_{(n,m) \in \mathbb{N}^2} x_{n,m} \in [0, \infty]
\]

Then
\[
\sum_{(n,m) \in \mathbb{N}^2} x_{n,m} = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} x_{n,m} = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} x_{n,m}
\]

We will now give our own complete proof of the alternative form of the expectation.

**Lemma 4.1.2.**
\[
\mathbb{E}(X) = \sum_{k=1}^{\infty} kP(X = k) = \sum_{n=1}^{\infty} P(X \geq n)
\]

**Proof.**
Firstly, we define the following sequence of non-negative real numbers \((x_{n,k})\) as follows:
\[
x_{n,k} = \begin{cases} 
P(X = k) & \text{ for } 1 \leq n \leq k; \ n, k \in \mathbb{N} \\ 0 & \text{ for } n \geq k + 1; \ n, k \in \mathbb{N} \end{cases} \tag{4.1.1}
\]
Now consider the following

\[
\sum_{k=1}^{\infty} kP(X = k) = \sum_{k=1}^{\infty} \left( \sum_{n=1}^{k} P(X = k) + \sum_{n=k+1}^{\infty} 0 \right)
\]

The part contained in the brackets is the sum of our sequence defined in (4.1.1). Therefore,

\[
\sum_{k=1}^{\infty} kP(X = k) = \sum_{k=1}^{\infty} \sum_{n=1}^{\infty} x_{n,k}
\]

Since the summands are non-negative and we have a double infinite summation, we can apply Tonelli’s Theorem for series to the above—i.e. reversing the order of summation. Hence, we have

\[
\sum_{k=1}^{\infty} kP(X = k) = \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} x_{n,k} \quad (4.1.2)
\]

Using the bounds in (4.1.1), we can rewrite (4.1.2) as follows:

\[
\sum_{k=1}^{\infty} kP(X = k) = \sum_{n=1}^{\infty} \left( \sum_{k=1}^{n-1} x_{n,k} + \sum_{k=n}^{\infty} x_{n,k} \right) \quad (4.1.3)
\]

From (4.1.1), we have \(x_{n,k} = 0\) when \(n \geq k + 1; n, k \in \mathbb{N}\), which is the same as saying that

\(x_{n,k} = 0\) when \(1 \leq k \leq n - 1; n, k \in \mathbb{N}\) \quad (4.1.4)

Similarly, \(x_{n,k} = P(X = k)\) when \(1 \leq n \leq k; n, k \in \mathbb{N}\). This is the same as saying that

\(x_{n,k} = P(X = k)\) when \(k \geq n; n, k \in \mathbb{N}\) \quad (4.1.5)
Substituting (4.1.4) and (4.1.5) into (4.1.3), we have

\[
\sum_{k=1}^{\infty} kP(X = k) = \sum_{n=1}^{\infty} \left( \sum_{k=1}^{n-1} 0 + \sum_{k=n}^{\infty} P(X = k) \right) = \sum_{n=1}^{\infty} \sum_{k=n}^{\infty} P(X = k) = \sum_{n=1}^{\infty} P(X \geq n)
\]

Soon we shall see how useful the above can be.

We now consider a lemma involving the mean hitting time of an aperiodic irreducible Markov chain.

**Lemma 4.1.3.** Let \((X_0, X_1, \ldots)\) be an aperiodic irreducible Markov chain with the Markov digraph \(D\) where \(V(D) = \{s_1, \ldots, s_n\}\) and \(P\) is its transition matrix. Then for any two vertices \(s_i, s_j \in V(D)\) where \(s_i\) is the initial vertex of the walk, the mean hitting time is finite.

**Proof.** Since the Markov chain is irreducible and aperiodic, from Corollary 3.3.3, there exists an integer \(M\) such that for \(l \geq M; l \in \mathbb{Z}^+\), there is at least one directed walk of length \(l\) between any two vertices \(s_i, s_j \in V(D)\).

From definition, the mean hitting time from \(s_i\) to \(s_j\) is given by

\[
\tau_{i,j} = \mathbb{E}[T_{i,j}] = \sum_{k=1}^{\infty} kP(\min\{k \geq 1 | X_k = s_j\} \text{ given that } X_0 = s_i)
\]

Since \(\min\{k \geq 1 | X_k = s_j\} \text{ given that } X_0 = s_i\) is unique, let it be integer \(l_1\). Furthermore, we know \(M \geq l_1\) and \(P(\min\{k \geq 1 | X_k = s_j\} \text{ given that } X_0 = s_i) \leq 1\). All other probabilities are zero since a minimum value is unique. Therefore,

\[
\mathbb{E}[T_{i,j}] \leq (l_1)(1) \leq M
\]

Since \(M\) is finite and is an upper bound of \(\mathbb{E}[T_{i,j}]\), \(\tau_{i,j} = \mathbb{E}[T_{i,j}]\) is finite. \(\square\)

For an alternative proof of the above, we can consult [19] on pages 30 and 31.

**Theorem 4.1.4.** For any irreducible aperiodic Markov chain, there exists at least one stationary distribution.

**Proof.** Let \((X_0, X_1, \ldots)\) be a Markov chain with the Markov digraph \(D\). Let \(V(D) = \{s_1, \ldots, s_n\}\), \(P\) be the transition matrix and \(A(D)\) be the adjacency matrix of the Markov digraph \(D\). Suppose that we start at the initial vertex \(s_i\). Define

\[
\rho_i = \sum_{k=0}^{\infty} P(s_i(k) \cap (T_{l,1} > k))
\]
Using the other convention for writing intersections,

\[ \rho_i = \sum_{k=0}^{\infty} P(s_i(k), T_{l,l} > k) \]

Therefore, \( \rho_i \) is the expected number of visits to vertex \( s_i \) up to time \( T_{l,l} - 1 \). Since the Markov chain is irreducible and aperiodic from the hypothesis, by Lemma 4.1.3, we have \( \tau_{l,l} = \mathbb{E}[T_{l,l}] \) is finite. Furthermore,

\[
\begin{align*}
\rho_i &= \sum_{k=0}^{\infty} P(s_i(k), T_{l,l} > k) \\
&< \sum_{k=0}^{\infty} P(T_{l,l} > k) \\
&= \sum_{k=1}^{\infty} P(T_{l,l} \geq k) \\
&= \mathbb{E}[T_{l,l}] \\
&= \tau_{l,l}
\end{align*}
\]

Since \( \tau_{l,l} \) is finite, \( \rho_i \) is also finite from the above argument.

Consider the following distribution vector

\[ \pi = [\pi_1, \pi_2, \ldots, \pi_n] \]

where \( \pi_j = \frac{\rho_j \tau_{l,l}}{\tau_{l,l}} \). We want to prove that \( \pi \) is a stationary distribution—therefore, we need to prove that \( \pi \mathbb{P} = \pi \) and \( \sum_{j=1}^{n} \pi_j = 1 \).

i) Proving \( \pi \mathbb{P} = \pi \)

We have

\[ \pi_j = \frac{\rho_j}{\tau_{l,l}} = \frac{1}{\tau_{l,l}} \sum_{k=0}^{\infty} P(s_j(k), T_{l,l} > k) \]

Since we have \( j \neq l \), \( P(s_j(0), T_{l,l} > 0) = 0 \). This implies that

\[
\begin{align*}
\pi_j &= \frac{1}{\tau_{l,l}} \sum_{k=1}^{\infty} P(s_j(k), T_{l,l} > k) \\
&= \frac{1}{\tau_{l,l}} \sum_{k=1}^{\infty} P(s_j(k), T_{l,l} > k - 1)
\end{align*}
\]

The above is as a result of \( P(s_j(k), T_{l,l} = k) = 0 \), since \( l \neq j \); if at time \( k \) you are in state
s_j \neq s_i$, then $T_{l,l} \neq k$. Using the partition rule,

$$\pi_j = \frac{1}{\tau_{l,l}} \sum_{k=1}^{\infty} \sum_{i=1}^{n} P(s_j(k), T_{l,l} > k-1 \mid s_i(k-1)) P(s_i(k-1))$$

$$= \frac{1}{\tau_{l,l}} \sum_{k=1}^{\infty} \sum_{i=1}^{n} P(s_j(k), T_{l,l} > k-1, s_i(k-1))$$

We can think about the above as all the expected number of visits to state $s_i$ and then transitioning to $s_j$ in the next step.

Consider

$$P(s_j(k), T_{l,l} > k-1, s_i(k-1))$$

For the above, we will now use $P(A, B) = P(A \mid B) P(B)$. Therefore,

$$P(s_j(k), T_{l,l} > k-1, s_i(k-1)) = P(s_j(k) \mid T_{l,l} > k-1, s_i(k-1)) P(T_{l,l} > k-1, s_i(k-1))$$

Since $T_{l,l} > k-1$, given $s_i(k-1)$, only informs us that we have not returned to $s_i$ at time $k-1$, it does not give us any information about our position at time $k$. Furthermore, from the Markov property, the best information that we have is from the current state at time $k-1$; namely, $s_i(k-1)$. Hence,

$$P(s_j(k) \mid T_{l,l} > k-1, s_i(k-1)) = P(s_j(k) \mid s_i(k-1))$$

Therefore,

$$P(s_j(k), T_{l,l} > k-1, s_i(k-1)) = P(s_j(k) \mid s_i(k-1)) P(T_{l,l} > k-1, s_i(k-1))$$

$$= p_{ij} P(s_i(k-1), T_{l,l} > k-1)$$

As a result of the above, we have

$$\pi_j = \frac{1}{\tau_{l,l}} \sum_{k=1}^{\infty} \sum_{i=1}^{n} p_{ij} P(s_i(k-1), T_{l,l} > k-1)$$

$$= \frac{1}{\tau_{l,l}} \sum_{i=1}^{n} \sum_{k=1}^{\infty} p_{ij} P(s_i(k-1), T_{l,l} > k-1)$$

as the addition of real numbers is commutative. Since $p_{ij}$’s indices are independent of $k$,

$$\pi_j = \frac{1}{\tau_{l,l}} \sum_{i=1}^{n} p_{ij} \sum_{k=1}^{\infty} P(s_i(k-1), T_{l,l} > k-1)$$

Consider a change of index for $\sum_{k=1}^{\infty} P(s_i(k-1), T_{l,l} > k-1)$. Letting $m = k-1$, we have $m = 0$ when $k = 1$. Since the upper bound of the summation is infinite, it is unaffected.
Therefore,
\[
\sum_{k=1}^{\infty} P(s_i(k-1), T_{l,l} > k - 1) = \sum_{m=0}^{\infty} P(s_i(m), T_{l,l} > m)
\]

The above summation is the definition of \(\rho_i = \sum_{k=0}^{\infty} P(s_i(k), T_{l,l} > k)\); we simply replace the index \(k\) with the index \(m\) in its definition. Consequently, \(\pi_j\) becomes
\[
\pi_j = \frac{1}{\tau_{l,l}} \sum_{i=1}^{n} p_{ij} \sum_{m=0}^{\infty} P(s_i(m), T_{l,l} > m) = \frac{1}{\tau_{l,l}} \sum_{i=1}^{n} p_{ij} \rho_i = \sum_{i=1}^{n} p_{ij} \frac{\rho_i}{\tau_{l,l}}
\]

Substituting \(\pi_i = \frac{\rho_i}{\tau_{l,l}}\) in the above, we get
\[
\pi_j = \sum_{i=1}^{n} p_{ij} \pi_i = \sum_{i=1}^{n} \pi_i p_{ij}
\]

The above result is true for all \(j\) except \(j = l\). We now consider the case \(j = l\).

ii) Case: \(j = l\)

\[
\rho_l = \sum_{k=0}^{\infty} P(s_l(k), T_{l,l} > k)
\]

where \(\rho_l\) is the expected number of visits to state \(s_l\) up to time \(T_{l,l} - 1\). Since \(s_l\) was the initial vertex, we have visited it once at \(k = 0\) and so \(\rho_l = P(s_l(0), T_{l,l} > 0) = 1\). Furthermore, \(P(s_l(k), T_{l,l} > k) = 0\) if \(k > 0\).

We have
\[
\rho_l = 1 = P(T_{l,l} > 0) = \sum_{k=1}^{\infty} P(T_{l,l} = k)
\]

Using the law of total probability on the above, we have
\[
\rho_l = \sum_{k=1}^{\infty} \sum_{i=1}^{n} P(T_{l,l} = k \mid s_i(k-1)) P(s_i(k-1))
\]
\[
= \sum_{k=1}^{\infty} \sum_{i=1}^{n} P(s_i(k-1), T_{l,l} = k)
\]
\[
= \sum_{k=1}^{\infty} \sum_{i=1}^{n} P(s_i(k-1), T_{l,l} > k - 1, s_l(k)) \quad (R1)
\]

We will verify that
\[
P(s_i(k-1), T_{l,l} = k) = P(s_i(k-1), T_{l,l} > k - 1, s_l(k))
\]
\( P(s_i(k - 1), T_{l,l} = k) \) is the probability of being at vertex \( s_i \) in \( k - 1 \) steps and then to return to \( s_i \) at the next step. The above is the same as saying that \( T_{l,l} > k - 1 \) and the walk reaches \( s_i \) in the next step from \( s_i \) at time \( k \). Clearly, \( ((T_{l,l} > k - 1) \cap s_i(k)) \equiv (T_{l,l} = k) \).

Therefore,

\[
P(s_i(k - 1), T_{l,l} > k - 1, s_i(k)) = P(s_i(k) \mid s_i(k - 1), T_{l,l} > k - 1) P(s_i(k - 1), T_{l,l} > k - 1)
\]

By a similar argument as in the case for \( j \neq l \), we have from the Markov property

\[
P(s_l(k) \mid s_i(k - 1), T_{l,l} > k - 1) = P(s_l(k) \mid s_i(k - 1))
\]

From the above we have,

\[
P(s_l(k) \mid s_i(k - 1), T_{l,l} > k - 1) P(s_i(k - 1), T_{l,l} > k - 1) = P(s_i(k - 1) \mid T_{l,l} > k - 1) p_{il}
\]

Substituting the above in (R1), we get

\[
\rho_l = \sum_{k=1}^{\infty} \sum_{i=1}^{n} p_{il} P(s_i(k - 1), T_{l,l} > k - 1)
\]

Using the definition, \( \rho_i = \sum_{m=0}^{\infty} P(s_i(m), T_{l,l} > m) \), we get

\[
\rho_l = \sum_{i=1}^{n} p_{il} \rho_i
\]

\[
= \sum_{i=1}^{n} \rho_i p_{il}
\]

We now consider \( \pi_l = \frac{\rho_l}{\tau_{l,l}} \) and substitute \( \rho_l = \sum_{i=1}^{n} \rho_i p_{il} \) in it. Therefore,

\[
\pi_l = \sum_{i=1}^{n} \frac{\rho_i p_{il}}{\tau_{l,l}}
\]

\[
= \sum_{i=1}^{n} \pi_i p_{il} \quad \text{(as } \pi_i = \frac{\rho_i}{\tau_{l,l}} \text{)}
\]
Therefore, for \( j = l \) and \( j \neq l \), we have

\[
\pi_j = \sum_{i=1}^{n} \pi_i p_{ij} \quad j = 1, \ldots, n
\]

\[
= [\pi_1 \ldots \pi_n] \begin{bmatrix} p_{1j} \\ \vdots \\ p_{nj} \end{bmatrix}
\]

\[
= \pi \text{col}_j (P)
\]

where \( \text{col}_j (P) \) is the \( j \)th column of \( P \).

Furthermore,

\[
[\pi_1 \pi_2 \ldots \pi_n] = [\pi \text{col}_1 (P) \pi \text{col}_2 (P) \ldots \pi \text{col}_n (P)]
\]

\[
= \pi \left[ \text{col}_1 (P) \text{col}_2 (P) \ldots \text{col}_n (P) \right]
\]

Simplifying the above, we get \( \pi = \pi P \).

- Now, we need to prove \( \sum_{i=1}^{n} \pi_i = 1 \).

Using \( \pi_i = \frac{\rho_i}{\tau_{i,l}} \), we have

\[
\sum_{i=1}^{n} \pi_i = \sum_{i=1}^{n} \frac{\rho_i}{\tau_{i,l}} = \frac{1}{\tau_{i,l}} \sum_{i=1}^{n} \rho_i
\]  

(4.1.6)

Considering the definition of the mean return time, we find the following.

\[
\tau_{i,l} = \mathbb{E}[T_{i,l}]
\]

\[
= \sum_{k=1}^{\infty} P(T_{i,l} \geq k)
\]

\[
= \sum_{k=0}^{\infty} P(T_{i,l} > k)
\]

\[
= \sum_{k=0}^{\infty} \sum_{i=1}^{n} P(T_{i,l} > k | s_i(k)) P(s_i(k)) \quad \text{(from the law of total probability)}
\]

\[
= \sum_{i=1}^{n} \sum_{k=0}^{\infty} P(s_i(k), T_{i,l} > k) \quad \text{(as } P(A | B)P(B) = P(A \cap B) = P(B \cap A))
\]

\[
= \sum_{i=1}^{n} \rho_i \quad \text{(from the definition of } \rho_i)
\]

Substituting the above into (4.1.6), we get

\[
\sum_{i=1}^{n} \pi_i = \frac{\sum_{i=1}^{n} \rho_i}{\sum_{i=1}^{n} \rho_i} = 1
\]

Therefore, \( \pi \) is a stationary distribution and the theorem is proven.
In fact, there is a unique stationary distribution for an aperiodic irreducible Markov chain. We will leave this proof for now and return to it later in this chapter. We now prove an interesting result that is an exercise in [19].

**Theorem 4.1.5.** If there are two stationary distributions in a Markov chain, then there exist infinitely many.

**Proof.** 1) Let $\pi$ and $\pi'$ be two different stationary distributions in a Markov chain. Furthermore, let $P$ be the $n \times n$ transition matrix of the Markov digraph $D$ of the above Markov chain which has the vertex set $V(D) = \{s_1, \ldots, s_n\}$.

Now consider a scalar $p \in (0, 1)$. Using $p$, $\pi$ and $\pi'$, we define

$$\pi'' = p\pi + (1 - p)\pi' \quad (4.1.7)$$

We want to prove that $\pi''$ is also a stationary distribution of this Markov chain. Therefore, we are required to prove two things:

a) $\pi'' P = \pi''$

b) The sum of the components of $\pi''$ is 1. That is, $\sum_{i=1}^{n} \pi_i'' = 1$.

We proceed with a).

a) Considering $\pi'' P$ and substituting (4.1.7) into it, we get

$$(p\pi + (1 - p)\pi') P; \quad p \in (0, 1)$$

Since matrix multiplication is distributive with respect to matrix addition, the above equation can be written as

$$p\pi P + (1 - p)\pi' P \quad (4.1.8)$$

Since $\pi$ and $\pi'$ are stationary distributions in our hypothesis, we have

$$\pi P = \pi \quad \pi' P = \pi'$$

Using the above results, (4.1.8) simplifies to

$$p\pi + (1 - p)\pi'$$

which is $\pi''$ in (4.1.7).

Therefore, $\pi'' P = \pi''$ and thus a) has been proven.

We now consider b).
b) From (4.1.7), \[ \pi'' = p\pi + (1 - p)\pi' \] and since matrix addition is componentwise, we have for \( i = 1, 2, \ldots, n \)

\[ \pi''_i = p\pi_i + (1 - p)\pi'_i \quad (4.1.9) \]

So our row vector \( \pi'' \) can be written as follows

\[
\begin{bmatrix}
\pi''_1 \\
\vdots \\
\pi''_n
\end{bmatrix} =
\begin{bmatrix}
p\pi_1 + (1 - p)\pi'_1 \\
\vdots \\
p\pi_n + (1 - p)\pi'_n
\end{bmatrix}
\]

Summing over the components of \( \pi'' \), we get

\[
\sum_{i=1}^{n} \pi''_i = \sum_{i=1}^{n} (p\pi_i + (1 - p)\pi'_i) \quad \text{(from (4.1.9))}
\]

\[
= \sum_{i=1}^{n} p\pi_i + \sum_{j=1}^{n} (1 - p)\pi'_j \quad \text{(addition of real numbers is commutative and associative)}
\]

\[
= p \sum_{i=1}^{n} \pi_i + (1 - p) \sum_{j=1}^{n} \pi'_j \quad \text{(as } p \text{ and } (1 - p) \text{ are independent of the summation indices)}
\]

Since \( \pi \) and \( \pi' \) are stationary distributions, \( \sum_{i=1}^{n} \pi_i = 1 \) and \( \sum_{i=1}^{n} \pi'_i = 1 \). Therefore,

\[
\sum_{i=1}^{n} \pi''_i = p + (1 - p) = 1
\]

Since \( \pi'' \) satisfies a) and b), it is also a stationary distribution of the given Markov chain.

ii) Proof by contradiction.

Assume that we have \( k \) different stationary distributions. Consider \( \pi^{(k_1)} \) and \( \pi^{(k_2)} \) from the set of stationary distributions \( \{ \pi, \pi', \pi'', \ldots, \pi^{(k-1)} \} \) such that \( \pi^{(k_1)} \neq \pi^{(k_2)} \). Let \( p^{(k)} \in (0, 1) \) and set

\[
\pi^{(k)} = p^{(k)}\pi^{(k_1)} + (1 - p^{(k)})\pi^{(k_2)}
\]

Following the same procedure as in i), we find that \( \pi^{(k)} \) is a different stationary distribution. So our assumption is not valid and therefore, we have an infinite number of different stationary distributions. Thus the result is proven.

Before proving the Markov Chain Convergence Theorem, we need to define a suitable metric to be used in evaluating the convergence of an initial distribution to a stationary one. We now define the total variation distance.
Let $S = \{v_1, v_2, \ldots, v_n\}$ be our state space and $\vec{\nu}^{(1)} = \begin{bmatrix} \nu_1^{(1)} & \nu_2^{(1)} & \ldots & \nu_n^{(1)} \end{bmatrix}$ and $\vec{\nu}^{(2)} = \begin{bmatrix} \nu_1^{(2)} & \nu_2^{(2)} & \ldots & \nu_n^{(2)} \end{bmatrix}$ be probability distributions on $S$. Then the total variation distribution between $\vec{\nu}^{(1)}$ and $\vec{\nu}^{(2)}$ is

$$d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right) = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right|$$

**Proof.** Let us prove that $d_{TV}$ is in fact a metric.

i) Since $\vec{\nu}^{(1)}$ and $\vec{\nu}^{(2)}$ are probability distributions, we have $\sum_{i=1}^{n} \nu_i^{(1)} = 1$ and $\sum_{i=1}^{n} \nu_i^{(2)} = 1$. Furthermore, $\nu_i \geq 0$ for $i = 1, \ldots, n$ and for all probability distributions $\vec{\nu}$ on $S$. Clearly, $\left| \nu_i^{(1)} - \nu_i^{(2)} \right| \geq 0$ and is finite. This implies that $\frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right|$ is also finite, non-negative and real-valued. The above is in fact $d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right)$.

ii) a) Now consider

$$d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right) = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right| = 0$$

This implies that $\sum_{i=1}^{n} \nu_i^{(1)} = 0$. Since $\left| \nu_i^{(1)} - \nu_i^{(2)} \right| \geq 0$ for $i = 1, \ldots, n$,

$$\sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right| = 0 \text{ if and only if } \nu_i^{(1)} = \nu_i^{(2)} \text{ for } i = 1, 2, \ldots, n.$$ Therefore, $\nu_i^{(1)} - \nu_i^{(2)} = 0$ and so $\nu_i^{(1)} = \nu_i^{(2)}$ for $i = 1, \ldots, n$. We then have $\vec{\nu}^{(1)} = \vec{\nu}^{(2)}$.

b) Now consider when $\vec{\nu}^{(1)} = \vec{\nu}^{(2)}$. Therefore,

$$d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right) = d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(1)} \right) = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(1)} \right| = \frac{1}{2} \left( 0 \right) = 0$$

From a) and b), $d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right) = 0$ if and only if $\vec{\nu}^{(1)} = \vec{\nu}^{(2)}$.

iii) We also have the following

$$d_{TV} \left( \vec{\nu}^{(1)}, \vec{\nu}^{(2)} \right) = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right|$$

$$= \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(2)} - \nu_i^{(1)} \right| \quad \text{(as } |a - b| = |b - a|)$$

$$= d_{TV} \left( \vec{\nu}^{(2)}, \vec{\nu}^{(1)} \right)$$

iv) For $d_{TV}$ to be a metric, it still needs to satisfy the triangle inequality. Based on this, let $\vec{\nu}^{(3)}$ be another probability distribution on $S$. 68
\[ d_{\text{TV}}\left(\overrightarrow{\nu}(1), \overrightarrow{\nu}(2)\right) \]
\[ = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(2)} \right| \]
\[ = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(3)} + \nu_i^{(3)} - \nu_i^{(2)} \right| \]
\[ \leq \frac{1}{2} \sum_{i=1}^{n} \left( \left| \nu_i^{(1)} - \nu_i^{(3)} \right| + \left| \nu_i^{(3)} - \nu_i^{(2)} \right| \right) \quad \text{(from the triangle inequality of real numbers)} \]
\[ = \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(1)} - \nu_i^{(3)} \right| + \frac{1}{2} \sum_{i=1}^{n} \left| \nu_i^{(3)} - \nu_i^{(2)} \right| \quad \text{(since multiplication of real numbers is distributive with respect to addition)} \]
\[ = d_{\text{TV}}\left(\overrightarrow{\nu}(1), \overrightarrow{\nu}(3)\right) + d_{\text{TV}}\left(\overrightarrow{\nu}(3), \overrightarrow{\nu}(2)\right) \]

From i), ii), iii) and iv), \( d_{\text{TV}} \) satisfies the conditions to be a metric on \( S \).

Now consider a sequence of probability distributions \( \overrightarrow{\nu}(1), \overrightarrow{\nu}(2), \ldots \), and the probability distribution \( \overrightarrow{\nu} \). If \( d_{\text{TV}}\left(\overrightarrow{\nu}(k), \overrightarrow{\nu}\right) \to 0 \) as \( k \to \infty \), then we say \( \overrightarrow{\nu}(k) \) converges to \( \overrightarrow{\nu} \) in total variation. We will denote this by \( \overrightarrow{\nu}(k) \xrightarrow{\text{TV}} \overrightarrow{\nu} \).

We now draw our attention to a very important theorem which assures us of the convergence of probability distributions for irreducible aperiodic Markov chains.

**Theorem 4.1.6. The Markov Chain Convergence Theorem**

Let \( (X_0, X_1, \ldots) \) be an irreducible aperiodic Markov chain with the Markov digraph \( D \), vertex set \( V(D) = \{s_1, \ldots, s_n\} \), transition matrix \( P \) and an arbitrary initial distribution \( \overrightarrow{\mu}(0) \). Then the sequence of probability distributions \( \{\overrightarrow{\mu}(k)\} \) will converge in total variation to a stationary distribution (eigenvector) \( \overrightarrow{\pi} \) for the transition matrix \( P \); i.e.

\[ \overrightarrow{\mu}(k) \xrightarrow{\text{TV}} \overrightarrow{\pi} \]

**Proof.** Let \( (X_0, X_1, \ldots) \) be an irreducible aperiodic Markov chain with the Markov digraph \( D \) and the initial distribution \( \overrightarrow{\mu}(0) \). Similarly, let \( (X'_0, X'_1, \ldots) \) be another Markov chain with the same Markov digraph \( D \) and the stationary distribution as its initial distribution.

We now want to prove that these two chains will meet. Therefore, we define

\[ T = \min \{ k \in \mathbb{N} \mid X_k = X'_k \} \]

From our hypothesis, the Markov chain \( (X_0, X_1, \ldots) \) is irreducible and aperiodic. Therefore, from Corollary 3.3.3, there exits a positive integer \( M \) such that there is always a walk of length \( M \) between any two vertices \( s_i \) and \( s_j \in V(D) \). Therefore in \( P^M \), \( p_{ij}(M) > 0 \) for \( i, j \in \{1, \ldots, n\} \).

Let \( \alpha = \min \{ p_{ij}(M) \mid i \in \{1, \ldots, n\} \} \); hence, \( \alpha \) is equal to some \( p_{ij}(M) \) and so \( \alpha > 0 \). Thus the
probability of these two chains to meet in less than or equal to $M$ steps is given by

$$P(T \leq M) \geq P(X_M = X'_M) \geq P(X_M = s_l, X'_M = s_l) \quad \text{for any given } l$$

$$= P(X_M = s_l) \cdot P(X'_M = s_l)$$

The last step is as a result of the two Markov chains being independent of each other.

Using the law of total probability, we get

$$P(T \leq M) \geq \left( \sum_{i=1}^{n} P(X_M = s_l \mid X_0 = s_i) \cdot P(X_0 = s_i) \right) \cdot \left( \sum_{i=1}^{n} P(X'_M = s_l \mid X'_0 = s_i) \cdot P(X'_0 = s_i) \right)$$

$$= \left( \sum_{i=1}^{n} p_{il}(M) \cdot P(X_0 = s_i) \right) \cdot \left( \sum_{i=1}^{n} p_{il}(M) \cdot P(X'_0 = s_i) \right)$$

$$\geq \left( \sum_{i=1}^{n} \alpha P(X_0 = s_i) \right) \cdot \left( \sum_{i=1}^{n} \alpha P(X'_0 = s_i) \right)$$

$$= \alpha^2 \left( \sum_{i=1}^{n} P(X_0 = s_i) \right) \cdot \left( \sum_{i=1}^{n} P(X'_0 = s_i) \right)$$

Since $\sum_{i=1}^{n} P(X_0 = s_i) = \sum_{i=1}^{n} P(X'_0 = s_i) = 1$, the above becomes

$$P(T \leq M) \geq \alpha^2$$

Taking the complement of the above inequality, we have

$$P(T > M) \leq 1 - \alpha^2 \quad (4.1.10)$$

Let us prove the following result

$$P(X_{2M} \neq X'_{2M} \mid T > M) \leq 1 - \alpha^2$$

This is the same as considering $P(X_{2M} = X'_{2M} \mid T > M) \geq \alpha^2$.

Let us begin

$$P(X_{2M} = X'_{2M} \mid T > M) \geq P(X_{2M} = s_l, X'_{2M} = s_l \mid T > M)$$

$$= P(X_{2M} = s_l \mid T > M) \cdot P(X'_{2M} = s_l \mid T > M) \quad \text{(the Markov chains are independent)}$$
Using the law of total probability on the above, we find

\[
P(X_{2M} = X'_{2M} \mid T > M) \geq \left( \sum_{i=1}^{n} P(X_{2M} = s_i \mid T > M, X_M = s_i) \right) P(X_M = s_i \mid T > M) \]

\[
= \left( \sum_{i=1}^{n} P(X'_{2M} = s_i \mid T > M, X'_M = s_i) \right) P(X'_M = s_i \mid T > M) \]

\[
= \left( \sum_{i=1}^{n} P(X_M = s_i \mid X'_M = s_i) P(X_M = s_i \mid T > M) \right) \]

\[
= \left( \sum_{i=1}^{n} P(X'_M = s_i \mid X'_0 = s_i) P(X'_M = s_i \mid T > M) \right) \]

\[
\geq \alpha^2 \sum_{i=1}^{n} P(X_M = s_i \mid T > M) \sum_{i=1}^{n} P(X'_M = s_i \mid T > M)
\]

\[
= \alpha^2 (1)(1)
\]

\[
= \alpha^2
\]

The above is as a result of \( \sum_{i=1}^{n} P(X_M = s_i \mid T > M) = 1 \). To see this, consider the following

\[
\sum_{i=1}^{n} P(X_M = s_i \mid T > M)
\]

\[
= \sum_{i=1}^{n} \frac{P(X_M = s_i, T > M)}{P(T > M)}
\]

\[
= \frac{1}{P(T > M)} \sum_{i=1}^{n} (P(T > M \mid X_M = s_i) P(X_M = s_i))
\]

\[
= \frac{1}{P(T > M)} P(T > M) \quad \text{(from the law of total probability)}
\]

\[
= 1
\]

Thus, we have

\[
P(X_{2M} = X'_{2M} \mid T > M) \geq \alpha^2
\]

and so

\[
P(X_{2M} \neq X'_{2M} \mid T > M) \leq 1 - \alpha^2
\]
We then use the above result as follows.

\[
P(T > 2M) = P(T > 2M | T > M) P(T > M) \\
\leq P(X_{2M} \neq X'_M | T > M) (1 - \alpha^2) \\
\leq (1 - \alpha^2) (1 - \alpha^2) \\
= (1 - \alpha^2)^2
\]

Assume that the following is true for the integer \( z - 1 \),

\[
P(T > (z - 1)M) \leq (1 - \alpha^2)^{z-1}
\]

Next we prove that \( P(T > zM) \leq (1 - \alpha^2)^z \). We have

\[
P(T > zM) = P(T > zM | T > (z - 1)M) P(T > (z - 1)M) \\
\leq P(T > zM | T > (z - 1)M) (1 - \alpha^2)^{z-1}
\]

The above proof follows exactly the same way as for \( P(T > 2M) \), we simply substitute \( z \) in place of \( 2 \) and \( z - 1 \) in place of \( 1 \). This then gives us the following results.

\[
P(T > zM | T > (z - 1)M) \\
= 1 - P(T \leq zM | T > (z - 1)M) \\
\leq 1 - \alpha^2
\]

From this argument,

\[
P(T > zM) \leq (1 - \alpha^2)^z \leq (1 - \alpha^2)^{z-1} = (1 - \alpha^2)^z
\]

Therefore, by the principle of Mathematical Induction, the above is true for all \( z \in \mathbb{N} \).

Since \( \alpha \) is a probability after the \( M^{th} \) time step, \( \alpha \leq 1 \). Furthermore, from the definition of \( \alpha \), \( \alpha > 0 \).

Now consider the following limit

\[
\lim_{k \to \infty} P(T > k) = \lim_{z \to \infty} P(T > zM) \\
\leq \lim_{z \to \infty} (1 - \alpha^2)^z \\
= 0 \quad (\text{as } 0 \leq (1 - \alpha^2) < 1)
\]

Thus the probability that the chains never meet is 0. Therefore, it is certain that these chains will meet at some finite step \( T \).

We now construct a third Markov chain \((X''_0, X''_1, \ldots)\) by setting \( X''_0 = X_0 \) and letting

\[
X''_{k+1} = \begin{cases} 
X'_{k+1} & \text{if } X''_k \neq X'_k \\
X_k & \text{if } X''_k = X'_k
\end{cases}
\]  

(4.1.11)
for each $k$.

Since we are still working with the same Markov digraph, we are just realizing another outcome, which is still a Markov chain. $X_k'' = X_0$ implies that we start with the initial distribution $\mu^{(0)}$.

Before we proceed, we need to prove another auxiliary result:

$$P(X_k'' = s_i) - P(X_k' = s_i) \leq P(X_k'' = s_i, X_k' \neq s_i)$$

Consider the following.

$$\mathcal{P} = P((X_k'' = s_i) \cup (X_k' \neq s_i)) = P(X_k'' = s_i) + P(X_k' = s_i) - P(X_k'' = s_i, X_k' = s_i)$$

The above implies that

$$\mathcal{P} + P(X_k'' = s_i, X_k' \neq s_i) - 1 = P(X_k'' = s_i) - P(X_k' = s_i)$$

Furthermore, since $\mathcal{P}$ is a probability, $\mathcal{P} \leq 1$. Therefore, we have

$$1 + P(X_k'' = s_i, X_k' \neq s_i) - 1 \geq \mathcal{P} + P(X_k'' = s_i, X_k' \neq s_i) - 1$$

and so

$$P(X_k'' = s_i, X_k' \neq s_i) \geq \mathcal{P} + P(X_k'' = s_i, X_k' \neq s_i) - 1 = P(X_k'' = s_i) - P(X_k' = s_i)$$

Continuing with the proof, for $i = \{1, \ldots, n\}$

$$\mu^{(k)}_i - \pi_i = P(X_k'' = s_i) - P(X_k' = s_i) \leq P(X_k'' = s_i, X_k' \neq s_i) \leq P(X_k' \neq X_k'') \leq P(T > k) \to 0 \quad \text{as } k \to \infty$$

(R1)

Similarly,

$$\pi_i - \mu^{(k)}_i = P(X_k' = s_i) - P(X_k'' = s_i) \leq P(X_k' = s_i, X_k'' \neq s_i) \leq P(X_k' \neq X_k'') \leq P(T > k) \to 0 \quad \text{as } k \to \infty$$

(R2)

(R1) and (R2), then gives us

$$\lim_{k \to \infty} \left| \mu^{(k)}_i - \pi_i \right| = 0$$
From the above, we get
\[
\lim_{k \to \infty} d_{TV}((\vec{\mu}^{(k)}, \pi)) = \lim_{k \to \infty} \left( \frac{1}{2} \sum_{i=1}^{n} |\mu_i^{(k)} - \pi_i| \right)
= \frac{1}{2} \sum_{i=1}^{n} \lim_{k \to \infty} |\mu_i^{(k)} - \pi_i| = \lim_{x \to \infty} ax = a \lim_{x \to \infty} x
\]
and the limit of the sum is the sum of the limits)
\[
= \frac{1}{2} \sum_{i=1}^{n} 0
= \frac{1}{2}(0)
= 0
\]
Therefore, any initial distribution of an irreducible aperiodic Markov chain converges to a stationary distribution.

In fact, there is only one stationary distribution for an irreducible aperiodic Markov chain.

**Theorem 4.1.7. Uniqueness of a Stationary Distribution**

Any irreducible aperiodic Markov chain has a unique stationary distribution.

**Proof.** From Theorem 4.1.6, there exists at least one stationary distribution for an irreducible aperiodic Markov chain. Suppose that we have two different stationary distributions, say \(\vec{\pi}'\) and \(\vec{\pi}''\). Next we set \(\vec{\mu}^{(0)} = \vec{\pi}'\) and according to the last theorem
\[
\lim_{k \to \infty} d_{TV}(\vec{\mu}^{(k)}, \vec{\pi}) = 0.
\]
As a result of \(\vec{\pi}'\) being a stationary distribution, \(\vec{\mu}^{(k)} = \vec{\pi}'\) and so
\[
\lim_{k \to \infty} d_{TV}(\vec{\pi}'', \vec{\pi}) = 0
\]
Since \(\vec{\pi}'\) and \(\vec{\pi}''\) do not depend on \(k\), we deduce
\[
\vec{\pi}' = \vec{\pi}''
\]
Therefore, we only have one stationary distribution and our result is proven.

In fact, we can use the stationary distribution of a Markov chain to find its mean return times.

**Theorem 4.1.8.** Let \((X_0, X_1, \ldots)\) be an irreducible aperiodic Markov chain with the Markov digraph \(D\), which has the vertex set \(\{s_1, s_2, \ldots, s_n\}\). Let \(\vec{\pi} = [\pi_1 \quad \pi_2 \quad \ldots \quad \pi_n]\) be the stationary distribution of the above Markov chain. Then
\[
\tau_{i,i} \pi_i = 1
\]
where \(\tau_{i,i}\) is the mean return time to vertex \(s_i \in V(D)\).
Proof. Let $T_i$ be the time to hit vertex $s_i$ for the first time. Therefore, $T_{i,i} = (T_i \mid X_0 = s_i)$.

From the above, we have

$$
\tau_{i,i} \pi_i = \mathbb{E}[T_{i,i}] P(X_0 = s_i)
= \mathbb{E}[T_i \mid X_0 = s_i] P(X_0 = s_i)
= \sum_{k=1}^{\infty} P(T_i \geq k \mid X_0 = s_i) P(X_0 = s_i)
= \sum_{k=1}^{\infty} P((T_i \geq k), (X_0 = s_i))
$$

For $k = 1$, $P((T_i \geq 1), (X_0 = s_i)) = P(X_0 = s_i)$.

For $k \geq 2$, the event $T_i \geq k$ is the same as saying that the process does not visit $s_i$ for time steps $t = 1, \ldots, (k - 1)$. Hence,

$$
P((T_i \geq k), (X_0 = s_i)) = P((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i), (X_0 = s_i))
$$

Now consider the following.

$$
P(A) = P(A \cap B) + P(A \cap B^c)
$$

Hence, $P(A \cap B) = P(A) - P(A \cap B^c)$. Let $A = ((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i))$ and $B = (X_0 = s_i)$. Therefore,

$$
P(((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i)) = P((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_2 \neq s_i), (X_1 \neq s_i)) -
P((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i), (X_0 \neq s_i))
= P((X_{k-2} \neq s_i), (X_{k-3} \neq s_i), \ldots, (X_1 \neq s_i), (X_0 \neq s_i)) -
P((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i), (X_0 \neq s_i))
$$

The last step is as a result of time homogeneity.

Define $a_k = P((X_k \neq s_i), (X_{k-1} \neq s_i), \ldots, (X_0 \neq s_i))$. Hence,

$$
P((X_{k-1} \neq s_i), (X_{k-2} \neq s_i), \ldots, (X_1 \neq s_i), (X_0 = s_i)) = a_{k-2} - a_{k-1}
$$

Therefore, from $k = 1$ and $k \geq 2$, we have

$$
\tau_{i,i} \pi_i = P(X_0 = s_i) + \sum_{k=2}^{\infty} (a_{k-2} - a_{k-1})
$$

We can identify that the series for the summation is the telescoping series. Therefore, the $m^{th}$ partial
sum is $a_0 - a_m$. Using this fact, we have

$$
\tau_{i,i} \pi_i = P (X_0 = s_i) + \lim_{m \to \infty} (a_0 - a_m)
$$

$$
= P (X_0 = s_i) + a_0
$$

$$
= P (X_0 = s_i) + P (X_0 \neq s_i) \quad \text{(from the way in which } a_0 \text{ is defined)}
$$

$$
= 1
$$

In the above, $a_m = P ((X_m \neq s_i), (X_{m-1} \neq s_i), \ldots, (X_0 \neq s_i)) \to 0$ as $m \to \infty$. This is as a result of all our vertices being recurrent—we return to each vertex with certainty. Hence, the probability of not ever returning to a vertex tends towards zero.

From Theorem 4.1.8, we have

$$
\tau_{i,i} = \frac{1}{\pi_i}
$$

The above will be used in two later examples.

We now consider a special type of Markov chain; namely, the reversible Markov chain. This will be applied to random walks on (undirected) graphs.

## 4.2 Reversible Markov Chains

*Reversible Markov chains* are Markov chains that behave the same in both times (one that moves forward and the other that moves backward). We define the above formally.

Let $(X_0, X_1, \ldots)$ be a Markov chain with the Markov digraph $D$, the vertex set $V(D)$ and the transition matrix $P$. A probability distribution $\pi$ on $V(D)$ is said to be *reversible* for the transition matrix $P$ and therefore, for the Markov chain, if for all $i, j \in \{1, \ldots, n\}$, we have

$$
\pi_i p_{ij} = \pi_j p_{ji}
$$

where $p_{ij}$ and $p_{ji}$ are the defined entries in $P$ (4.2.1)

If there exists a reversible distribution for a Markov chain, then it is reversible.

Equation (4.2.1) is more commonly known as the *balance equation* or *strong equilibrium*. Since we see the term equilibrium, we think about stationary distributions in a Markov chain. The following theorem deals with this.

**Theorem 4.2.1.** Let $(X_0, X_1, \ldots)$ be a Markov chain with the Markov digraph $D$, the vertex set $V(D) = \{s_1, \ldots, s_n\}$ and the transition matrix $P$. If $\overline{\pi}$ is a reversible distribution for the Markov chain, then it is also a stationary distribution of the mentioned Markov chain.

**Proof.** We let $\overline{\pi} = [\pi_1 \quad \pi_2 \quad \ldots \quad \pi_n]$. There are two things that we need to prove:

i) $\sum_{i=1}^{n} \pi_i = 1$ and

ii) $\overline{\pi} P = \overline{\pi}$

i) follows from the hypothesis that $\overline{\pi}$ is a probability distribution.

We now prove ii).

For all $j \in \{1, \ldots, n\}$, we must prove that $\pi_j = \sum_{i=1}^{n} \pi_i p_{ij}$. Since the Markov chain is a stochastic
process, the sum of all the components in any row of $P$ must be 1. In particular, consider row $j$ of $P$. Therefore,

$$\sum_{i=1}^{n} p_{ji} = 1$$  \hspace{1cm} (4.2.2)

Now we use the fact that $\pi_j = \pi_j(1)$ and substitute (4.2.2) into it, giving us

$$\pi_j = \pi_j \sum_{i=1}^{n} p_{ji}$$

Since $\pi_j$ is independent of the index in the summation, the above can be written as

$$\pi_j = \sum_{i=1}^{n} \pi_j p_{ji}$$  \hspace{1cm} (4.2.3)

Since $\pi$ is reversible, we have

$$\pi_j p_{ji} = \pi_i p_{ij}$$

By substituting the above into (4.2.3), we get

$$\pi_j = \sum_{i=1}^{n} \pi_i p_{ij}$$

From the last equation, we have

$$\begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_n \end{bmatrix} = \begin{bmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{nj} \end{bmatrix}$$

$$= \begin{bmatrix} \pi c_1(P) \\ \pi c_2(P) \\ \vdots \\ \pi c_n(P) \end{bmatrix}$$

Therefore, $\overline{\pi} = \overline{\pi} P$.

So i) and ii) are satisfied and therefore, $\overline{\pi}$ is a stationary distribution by definition.

Keeping the above in mind, we revisit the Markov chain in Example 2.1.2 to calculate its mean return times.

**Example 4.2.2.** From Example 2.1.2, define $D$ to be the Markov digraph with the vertex set $V(D) = \{s_0, s_1, \ldots, s_m\}$.

Suppose that we want to find the mean return time for container $A$ to have $r \ (0 \leq r \leq m)$ gas molecules—i.e. we want to find the mean return time for say $s_r$. For this problem, we will consider a different approach to find the stationary distribution; we will use the balance equation. Therefore, we will be looking for a reversible distribution. (From Theorem 4.2.1, this distribution will also be a stationary one.)
Without loss of generality, assume \( j > i \). Let \( \vec{\pi} = [\pi_0 \quad \pi_1 \quad \ldots \quad \pi_m] \) be the reversible distribution.

We have the following through the repetitive use of the detailed balance equation \( \pi_i p_{i \rightarrow i} = \pi_r p_{r \rightarrow i} \).

\[
\begin{align*}
\pi_i p_{i(i+1)} p_{(i+1)(i+2)} \cdots p_{(j-1)j} &= \pi_i p_{i(i+1)} p_{(i+1)(i+2)} \cdots p_{(j-1)j} \quad \text{(using \( \pi_i p_{i(i+1)} = \pi_{i+1} p_{i+1(i+1)} \))} \\
&= P(i+1)i p_{i+1(i+2)} \cdots p_{(j-1)j} \quad \text{(using \( \pi_{i+1} p_{i+1(i+2)} = \pi_{i+2} p_{i+2(i+1)} \))} \\
&\vdots \\
&= P(i+1)i p_{i+2(i+1)} \cdots \pi_{j-1} p_{(j-1)j} \quad \text{(using \( \pi_{j-1} p_{(j-1)j} = \pi_j p_{j(j-1)} \))} \\
&= \pi_j p_{j(j-1)} p_{(j-1)(j-2)} \cdots p_{i+1}i
\end{align*}
\]

Using (2.1.1) and (2.1.2) on the above, we have

\[
\pi_i \left( \frac{m-i}{m} \right) \left( \frac{m-(i+1)}{m} \right) \cdots \left( \frac{m-(j-1)}{m} \right) = \pi_j \left( \frac{j}{m} \right) \left( \frac{j-1}{m} \right) \cdots \left( \frac{i+1}{m} \right)
\]

We can write the above as

\[
\pi_i \left( \frac{m-i}{m} \right) \left( \frac{m-(i+1)}{m} \right) \cdots \left( \frac{m-(j-1)}{m} \right) = \pi_j \left( \frac{j}{m} \right) \left( \frac{j-1}{m} \right) \cdots \left( \frac{i+1}{m} \right)
\]

Simplifying and rearranging the above, we have

\[
\pi_i \frac{(m-i)(m-(i+1))(m-(i+2)) \cdots (m-(j-1))}{(j)(j-1) \cdots (i+2)(i+1)} = \pi_j
\]

Multiplying both sides of the above equation by \( \frac{(m-j)!}{i!} \), we get

\[
\frac{\pi_i (m-i)!}{j!} = \pi_j \frac{(m-j)!}{i!}
\]

By multiplying both sides of the equation by \( \frac{j!}{m!} \), we will get

\[
\pi_i \frac{(m-i)!}{m!} = \pi_j \frac{(m-j)!}{m!}
\]

Using the above, we now consider the ratio of \( \frac{\pi_i}{\pi_j} \).

\[
\frac{\pi_i}{\pi_j} = \binom{m}{i} \binom{m}{j} \quad \text{where} \quad \binom{m}{i} = \frac{m!}{(m-i)!i!} \quad (4.2.4)
\]
Now consider the binomial distribution, where we define a success to be the selecting of container A; hence, a failure will be defined to be the selecting of container B. Since we can select either container with equal probability, we have

\[ P(\text{selecting container } A) = P(\text{selecting container } B) = \frac{1}{2} \]

The number of molecules would then be our number of trials. Hence, if we have \( i \) successes, we have

\[ P(X = i) = \binom{m}{i} \left( \frac{1}{2} \right)^i \left( \frac{1}{2} \right)^{m-i} = \binom{m}{i} \left( \frac{1}{2} \right)^m \]

Similarly, for \( j \) successes, we have

\[ P(X = j) = \binom{m}{j} \left( \frac{1}{2} \right)^m \]

Therefore, by multiplying the right-hand side of (4.2.4) by \( \left( \frac{1}{2} \right)^m \), we find

\[ \frac{\pi_i}{\pi_j} = \frac{\binom{m}{i} \left( \frac{1}{2} \right)^m}{\binom{m}{j} \left( \frac{1}{2} \right)^m} \]

Hence, \( \pi_i = \binom{m}{i} \left( \frac{1}{2} \right)^m \) for \( 0 \leq i \leq m \).

Therefore, the mean return time for \( s_r \) is

\[ \tau_{r,r} = \frac{1}{\binom{m}{r} \left( \frac{1}{2} \right)^m} \]

Now let us consider some of the above properties and see if they still apply to undirected graphs.

### 4.3 Random Walks on Undirected Graphs

We begin this subsection with a few definitions and an example. Thereafter, we consider more general results.
4.3.1 Some Definitions

Remember that an undirected graph $G$ also has a vertex set, which is also defined to be $V(G)$. However, we no longer have an arc set, we have an edge set that is denoted by $E(G)$. Furthermore, two vertices $u, v \in V(G)$ are said to be neighbours if $u$ and $v$ have an edge between them. The set of all neighbours of a vertex $u \in V(G)$ is defined to be the neighbourhood of $u$ and is denoted by $N_G(u)$.

Consider the following example.

Example 4.3.1. We are given the graph $G$ below.

$$
\begin{align*}
E(G) &= \{v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_6, v_6v_7, v_7v_1, v_1v_6, v_2v_7, v_2v_5, v_2v_6\}
\end{align*}
$$

We will need the degrees of each of the vertices in order to compute the transition matrix $P$. The degrees are as follows:

$$
\begin{align*}
\deg_G(v_1) &= 3 & \deg_G(v_2) &= 5 & \deg_G(v_3) &= 2 & \deg_G(v_4) &= 2 \\
\deg_G(v_5) &= 3 & \deg_G(v_6) &= 4 & \deg_G(v_7) &= 3
\end{align*}
$$

Say that we start at a vertex $v_l$. We can then move to any of its neighbours with equal probability. This probability can be written as:

$$
P(v_l(k+1) \mid v_l(k)) = \begin{cases} 
\frac{1}{\deg_G(v_l)} & \text{if } v_l \in N_G(v_l) \\
0 & \text{if } v_l \notin N_G(v_l)
\end{cases}
$$

where $l \in \{1, 2, \ldots, n\}$ and $N_G(v_l) = \{v_l \mid v_l v_l \in E(G)\}$.

Clearly, the above is the transition probabilities between the vertices in $G$ in our transition matrix $P$. Furthermore, these vertices in $V(G)$ are also the vertices in $V(D)$, where $D$ is the Markov digraph of $G$. In the Markov digraph of a graph $G$, we have two arcs in opposite directions in place of each edge of $G$. [We consider the undirected graphs to be simple undirected graphs—i.e. there are no multiple edges or loops.] Therefore, if $v_i v_j \in E(G)$, then $v_i v_j$ and $v_j v_i \in E(D)$.

In terms of our example, the above is depicted as:
We now consider a possible reversible distribution \( \vec{\pi} \) which is defined as follows:

We have

\[
2m = \sum_{i=1}^{n} \deg_G(v_i) = \sum_{i=1}^{7} \deg_G(v_i) = 22 \quad \text{and} \quad \pi_i = \frac{\deg_G(v_i)}{2m} \quad \text{for} \quad i = 1, 2, \ldots, n
\]

Therefore, according to our problem:

\[
\vec{\pi} = \begin{bmatrix}
3 & 2 & 2 & 1 & 1 & 3 & 2 & 3 \\
22 & 22 & 11 & 11 & 22 & 11 & 22 & 22
\end{bmatrix}
\]

Firstly, we need to show that \( \vec{\pi} \) is a stationary distribution. We are, therefore, required to prove:

i) \( \sum_{i=1}^{n} \pi_i = 1 \) and

ii) \( \vec{\pi} P = \vec{\pi} \)

We will prove the results in a more general way and then verify them with a small programme written in R.

i) is as follows:

Consider the potential reversible distribution

\[
\vec{\pi} = \begin{bmatrix}
\frac{\deg_G(v_1)}{2m} & \frac{\deg_G(v_2)}{2m} & \cdots & \frac{\deg_G(v_n)}{2m}
\end{bmatrix}
\]

Therefore,

\[
\sum_{i=1}^{n} \pi_i = \sum_{i=1}^{n} \frac{\deg_G(v_i)}{2m} = \frac{1}{2m} \sum_{i=1}^{n} \deg_G(v_i) = \frac{1}{2m} (2m) = 1
\]
So i) is proven and now we prove ii).

For $(\pi P)_i$, $i = 1, \ldots, n$

$$
\pi \col_i(P) = \left[ \frac{\deg_G(v_1)}{2m} \quad \frac{\deg_G(v_2)}{2m} \quad \cdots \quad \frac{\deg_G(v_n)}{2m} \right] \begin{bmatrix} p_{1i} \\ p_{2i} \\ \vdots \\ p_{ni} \end{bmatrix}
$$

$$
= \frac{1}{2m} \sum_{j=1}^{n} \deg_G(v_j) p_{ji}
$$

There are $\deg_G(v_i)$ vertices that are adjacent to $v_i$. The probability to transition from each of these vertices to $v_i$ is $p_{ji} = \frac{1}{\deg_G(v_j)}$. This results in the summands being equal to 1 in each of these cases.

Each non–adjacent vertex to $v_i$ transitions to $v_i$ in one step with a probability of 0. Hence, each of the summands associated with non–adjacent vertices of $v_i$ will be equal to 0. Using these in the above, we have

$$
\pi \col_i(P) = \frac{1}{2m} \deg_G(v_i) = \pi_i
$$

From the above, we get

$$
\pi P = \pi \left[ \col_1(P) \quad \cdots \quad \col_n(P) \right]
= \left[ \pi \col_1(P) \quad \cdots \quad \pi \col_n(P) \right]
= \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_n \end{bmatrix}
= \pi
$$

Therefore, ii) is proven and so $\pi$ is a stationary distribution. Next we prove that $\pi$ is reversible.

$$
\pi_i p_{ij} = \begin{cases} 
\frac{\deg_G(v_i)}{2m} \cdot \frac{1}{\deg_G(v_j)} = \frac{1}{2m} \quad \text{if } v_i v_j \in E(G) \\
0 = \pi_j p_{ji} \quad \text{if } v_i v_j \notin E(G)
\end{cases}
$$

(4.3.1)

Since the above is true for $i, j \in \{1, \ldots, n\}$, $\pi$ is reversible and therefore, $G$ is a reversible Markov chain.

We produce the transition matrix $P$ and verify that $\pi = \begin{bmatrix} 3 & 5 & 1 & 1 & 3 & 2 & 3 \\ 22 & 22 & 11 & 11 & 22 & 11 & 22 \end{bmatrix}$ is a stationary distribution of $G$ in the following code that has been written in R.

```R
## Example Chapter 4: Undirected Graphs ##
## ------------------------------------ ##
library(igraph)

# constructing the graph G from the vectors vec_from and vec_to
vec_from <- c(1,2,3,4,5,6,7,1,2,2,2)

# producing the transition matrix

# verifying that pi is a stationary distribution
```

82
vec_to <- c(2,3,4,5,6,1,6,7,5,6)
d<-data.frame(from=vec_from,to=vec_to)
G<-graph.data.frame(d,directed=FALSE)

# Relabelling the vertices agree with the graph
Ver_lab<-NULL
for(i in 1:length(V(G)))
{
    Ver_lab<-cbind(Ver_lab,c(paste("v",i,sep="")))
}
G <- set.vertex.attribute(G, "label", value=Ver_lab)

Plotting the graph G in R:
> tkplot(G)

# A function to compute the transition matrix of an undirected graph G
Transition_mat<-function(G)
{
    D_ <-as.directed(G, mode = c("mutual", "arbitrary"))
    edge_weights<-rep(0,ecount(D_))
    D_<-set.edge.attribute(D_,"weight",value=edge_weights)
    for(i in 0:(length(V(G))-1))
    {
        deg <- neighbors(G,i,mode=1)
        E(D_)[i%>%deg]$weight <- 1/length(deg)
    }
    P<-matrix(0,nrow=length(V(D_)),ncol=length(V(D_)))
    for(i in 1:(length(V(G))))
    {
        for(j in 1:(length(V(G))))
        {
            if(length(E(D_)[(i-1)%->%(j-1)])!=0)
            {
                P[i,j]<-E(D_)[(i-1)%->%(j-1)]$weight
            }
        }
    }
}
\text{Output:}
\begin{verbatim}
> P<-Transition_mat(G)
> P
[1,] 0.0000000 0.3333333 0.0 0.0000000 0.00 0.3333333 0.3333333
[2,] 0.2000000 0.0000000 0.2 0.0000000 0.20 0.2000000 0.2000000
[3,] 0.0000000 0.5000000 0.0 0.5000000 0.00 0.0000000 0.0000000
[4,] 0.0000000 0.0000000 0.5 0.0000000 0.50 0.0000000 0.0000000
[5,] 0.0000000 0.3333333 0.0 0.3333333 0.00 0.3333333 0.0000000
[6,] 0.2500000 0.2500000 0.0 0.0000000 0.25 0.0000000 0.2500000
[7,] 0.3333333 0.3333333 0.0 0.0000000 0.00 0.3333333 0.0000000
>
> v<-t(c(3/22,5/22,1/11,1/11,3/22,2/11,3/22))
> rowSums(v) # sum is 1
[1] 1
> v%*%P # we can see that vP=v. Therefore, v is a stationary distribution
[1,] 0.1363636 0.2272727 0.09090909 0.09090909 0.1363636 0.1818182 0.1363636
> v
[1,] 0.1363636 0.2272727 0.09090909 0.09090909 0.1363636 0.1818182 0.1363636
\end{verbatim}

Since our graph $G$ in the above example is connected, its Markov chain is strongly-connected—i.e. for any two vertices $u$ and $v \in V(D)$, there exists both $u\rightarrow v$ and $v\rightarrow u$ directed walks. From Theorem 3.2.2, the Markov chain is irreducible. Therefore, we can say that a connected graph $G$ is an irreducible Markov chain.

If we now consider any disconnected undirected graph $G$, we will now find that it is a reducible Markov chain. We use the same procedure as in the above example to construct the Markov digraph $D$ and its transition matrix $P$. Since the graph $G$ is disconnected, it implies that there exist two vertices $u$ and $v \in V(G)$ such that there does not exist an undirected $u\rightarrow v$ path. As there is no $u\rightarrow v$ path, there does not exist an undirected $u\rightarrow v$ walk. From the construction of the Markov digraph $D$, we have arcs only where there were edges before and so those two vertices $u$ and $v$ cannot reach each other. Therefore, the reachability matrix $R(D)$ will contain zeros and so as a result of Theorem 3.2.2, the Markov chain is reducible.

We would also like to know that when we are given some initial distribution of the vertices whether or not this distribution converges to a stable one; i.e. a stationary distribution.

\textbf{Example 4.3.2.} Let us consider an even cycle $C_{2l}$, where \(l \in \mathbb{Z} \mid l \geq 2\). $C_{2l}$ is connected because for any two vertices $u$, $v \in V(C_{2l})$, there exists a $u\rightarrow v$ path. Let $\overrightarrow{\mu}^{(0)}$ be an initial distribution that is uniformly distributed among every second vertex in $C_{2l}$. Say $\overrightarrow{\mu}^{(0)}$ has the following form.
\[
\overrightarrow{\mu}^{(0)} = \begin{bmatrix}
1 \\
\frac{1}{l} \\
0 \\
\frac{1}{l} \\
0 \\
\vdots \\
\frac{1}{l} \\
0 \\
\end{bmatrix}
\] (4.3.2)
Since we are dealing with a cycle, $\text{deg}_{C_{2l}}(v_i) = 2$ for $i \in \{1, \ldots, n\}$ and therefore, the entries in our transition matrix $P$ are the following for $1 \leq i \leq 2l$

$$p_{ij} = \begin{cases} 
\frac{1}{2} & \text{if } v_iv_j \in E(C_{2l}) \\
0 & \text{if } v_iv_j \notin E(C_{2l}) 
\end{cases}$$

In terms of the indices, the above translates into the following.

$$p_{ii} = 0$$

$$p_{ij} = \frac{1}{2} \quad \text{for } j = (i-1) \mod 2l \text{ and } j = (i+1) \mod 2l$$

Heuristically from the above, we can argue as follows: If we start at the evenly-labelled vertices, then based on the transition matrix, we will walk to the oddly-labelled vertices in the next step. Similarly, we would move to the even vertices thereafter, etc. Therefore, we have an even period as we can start and return to $v_i$ in an even number of steps—at least 2. Therefore, the period is $d(v_i) = 2$ for $1 \leq i \leq 2l$.

Therefore, our Markov chain is periodic and so Theorem 4.1.6 cannot be applied and we are not guaranteed convergence.

Consider the following for $1 \leq j \leq 2l$.

$$\mu_j^{(1)} = \mu_j^{(0)} \col_j(P) = \sum_{i=1}^{2l} \mu_j^{(0)} p_{ij}$$

Using (4.3.2) and (4.3.3) in the above argument, we should find two different entries in the distribution $\mu^{(1)}$.

Case: $j$ is even.

Let $j = 2x$.

$$\mu_{2x}^{(1)} = \sum_{i=1}^{2l} \mu_i^{(0)} p_{i(2x)}$$

$$= \left( \frac{1}{7} \right) p_{1(2x)} + (0)p_{1(2x)} + \ldots + \left( \frac{1}{7} \right) p_{(2x-1)(2x)} + (0)p_{(2x)(2x)} + \ldots + \left( \frac{1}{7} \right) p_{(2l-1)(2x)} + (0)p_{(2l)(2x)}$$

Using (4.3.3), the above becomes

$$\mu_{2x}^{(1)} = 0 + 0 + \ldots + \frac{1}{7} p_{(2x-1)(2x)} + 0 + \frac{1}{7} p_{(2x+1)(2x)} + 0 + \ldots + 0 + 0$$

$$= \frac{1}{7} \left( \frac{1}{2} \right) + \frac{1}{7} \left( \frac{1}{2} \right)$$

$$= \frac{1}{7}$$

85
Case: $j$ is odd.

Letting $j = 2x + 1$, we get the following.

$$\mu_{2x+1}^{(1)} = \sum_{i=1}^{2l} \mu_i^{(0)} p_i(2x+1)$$

$$= \left(\frac{1}{l}\right) p_1(2x+1) + (0)p_1(2x+1) + \ldots + (0)p_1(2x+1) + \left(\frac{1}{l}\right) p_{(2x-1)}(2x+1) +$$

$$(0)p_{(2x)}(2x+1) + \left(\frac{1}{l}\right) p_{(2x+1)}(2x+1) + (0)p_{(2x+2)}(2x+1) + \ldots +$$

$$\left(\frac{1}{l}\right) p_{(2l-1)}(2x+1) + (0)p_{(2l)}(2x+1)$$

$$= 0 + 0 + \ldots + 0 + 0 + 0 + 0 + \ldots + 0 + 0$$

$$= 0$$

From the above two cases, we have

$$\overline{\mu}^{(1)} = \begin{bmatrix} 0 & \frac{1}{l} & 0 & \frac{1}{l} & 0 & \ldots & 0 & \frac{1}{l} \end{bmatrix}$$

Similarly, $\overline{\mu}^{(2)} = \overline{\mu}^{(0)}$ from the above argument. Therefore, we are alternating between the two distributions:

$$\begin{bmatrix} \frac{1}{l} & 0 & 0 & \ldots & 0 & \frac{1}{l} \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & \frac{1}{l} & 0 & \ldots & 0 & \frac{1}{l} \end{bmatrix}.$$

As a result of this, the initial distribution does not converge. So we can say that not all initial distributions converge for an even cycle.

It is interesting to note that for an odd cycle $C_{2k+1}$, any initial distribution will converge. We will prove a more general result that entails the above. For now, we consider the following result that will contain our even-cycle example. The following corollary is an exercise from [4].

**Corollary 4.3.3.** Not every initial distribution of a connected bipartite graph converges to a stationary distribution.

**Proof.** Let $G$ be a connected bipartite graph. From the hypothesis, the graph is connected and consequently, its Markov digraph $D$ is irreducible (as was seen in a previous construction).

Since our graph is 2-chromatic, we can partition the vertices in $V(G)$ into two partitioned sets—one set will be all the vertices of one colour and the other of the other colour. Let the two colours be $c_0$ and $c_1$.

Furthermore, let $U_0$ be the set of vertices coloured with $c_0$ and $U_1$ be the set of vertices coloured with $c_1$. Assume $|U_0| = l$. Hence, $|U_1| = n - l$.

Next relabel the vertices such that the vertices with the indices from 1 to $l$ are the vertices that are coloured in $c_0$—i.e. $U_0 = \{v_1, \ldots, v_l\}$. Therefore, the remaining vertices with the indices from $(l + 1)$ to $n$ will be the set of vertices that are coloured with the colour $c_1$—i.e. $U_1 = \{v_{l+1}, \ldots, v_n\}$.

Thus, $V(G)$ will be the following.

$$V(G) = \{v_1, v_2, \ldots, v_n\}$$

$$= \{v_1, \ldots, v_l\} \cup \{v_{l+1}, \ldots, v_n\}$$

$$= U_0 \cup U_1$$
We now consider the initial distribution over all the vertices in \( U_0 \) which is uniformly distributed; i.e.

\[
\mu_j^{(0)} = \frac{1}{l} \quad \text{for } 1 \leq j \leq l; \ j \in \mathbb{N}
\]

\[
\mu_j^{(0)} = 0 \quad \text{for } l + 1 \leq j \leq n; \ j \in \mathbb{N}
\]

As a result of the graph being bipartite, we have no loops and so our Markov digraph \( D \) will also not have loops. In any partitioned set, all the vertices are independent of each other. So we only have edges between \( U_0 \) and \( U_1 \). This means that we will only have arcs between the two partitioned sets. We also have

\[
d_D^+(v_i) = \deg_G(v_i)
\]

Since we can only transition to adjacent vertices and each case is as likely as the other to occur, we have the following for our transition matrix \( P \) for \( 1 \leq i, j \leq n \)

\[
p_{ij} = \begin{cases} 
1 & \text{if } v_iv_j \in E(G) \\
0 & \text{if } v_iv_j \notin E(G) 
\end{cases}
\]

Given that we start at a vertex \( v_j \), where \( 1 \leq j \leq l \), with a probability of \( \frac{1}{l} \), we can transition to \( \deg_G(v_j) \) vertices each with a probability of \( \frac{1}{\deg_G(v_j)} \), i.e.

\[
P(\text{moving to a vertex with the colour } c_1 \mid \text{we start with a vertex with the colour } c_0) = \frac{\deg_G(v_j)}{\deg_G(v_j)} = 1
\]

Now let us compute \( \mu^{(1)} \).

Case: Index \( j \in \mathbb{N}; \ l + 1 \leq j \leq n \).

\[
u_j^{(1)} = \mu^{(0) \ col} (P) = \sum_{i=1}^{n} \mu_i^{(0)} p_{ij}
\]

We know that we can only transition to \( v_j \) if it is adjacent from \( v_i \) in \( V(D) \). Therefore,

\[
p_{ij} = \begin{cases} 
1 & \text{if } v_iv_j \in E(G) \\
0 & \text{if } v_iv_j \notin E(G) 
\end{cases}
\]

From our construction, all \( v_i \) such that \( v_iv_j \in E(G) \) are in \( U_0 \). Furthermore, we transition from one partite set to the other with a probability of one. If this were not the case, then it would imply
that there would be two adjacent vertices in a partite set—contradicting that our graph is bipartite. 
Based on the above, we have
\[
\mu_j^{(1)} = \frac{1}{l} \sum_{i=1}^{n} \mathbb{I}(v_i v_j) \frac{1}{\deg_G(v_i)}
\]
where \( \mathbb{I} \) is the indicator function defined as
\[
\mathbb{I}(v_i v_j) = \begin{cases} 
1 & \text{if } v_i v_j \in E(G) \\
0 & \text{if } v_i v_j \notin E(G)
\end{cases}
\]

Case: Index \( j \in \mathbb{N} \); \( 1 \leq j \leq l \).

Since we cannot transition to vertices in the same partite set and we do not start at any of the
vertices in \( U_1 \); for all the vertices in \( U_0 = \{v_1, \ldots, v_l\} \), we have
\[
\mu_j^{(1)} = \mu_j^{(0)} \text{col}_j (P)
\]
\[
= \sum_{i=1}^{n} \mu_i^{(0)} p_{ij}
\]
\[
= 0
\]

We cannot be at a vertex of \( U_0 \) at the walk of length 1.

Similarly for walks of length 2, we cannot be at a vertex in \( U_1 \). Therefore, we alternate between
the distributions in which one case all non–zero probabilities are in \( U_0 \) and the other in which the
vertices of \( U_1 \) have all the non–zero probabilities. Hence, \( \mu^{(0)} \) does not converge and the result is
proven.

We now consider proving some results for connected non–bipartite graphs. The following theorem and
corollary are a consequence of an exercise from [4].

**Theorem 4.3.4.** All connected non–bipartite graphs are aperiodic Markov chains.

**Proof.** Let \( G \) be a connected non–bipartite graph with \( V(G) = \{v_1, \ldots, v_n\} \). Furthermore, we let \( D \) be
the Markov digraph of the graph \( G \) with \( V(D) = V(G) \). To prove the theorem, we need to show that
we have two self–walks with a greatest common divisor of 1—i.e. it suffices to prove that one walk has
an odd length and the other has an even length.

Since the graph is connected, its Markov digraph is strong. Therefore, every walk that exists in \( G \) also exists in its Markov digraph \( D \).

We also know that a nontrivial graph \( G \) is bipartite if and only if it contains no odd cycles. Con-
sidering the contraposition of the reverse implication, we have: “If \( G \) is non–bipartite, then it contains
an odd cycle.”

Similarly, the Markov digraph of the above will also contain a directed odd cycle \( C \).

Starting at a vertex \( v \in V(D) \), we transition to one of its neighbours \( u \) at time \( t = 1 \). Then at
time \( t = 2 \), we transition from \( u \) back to \( v \). Clearly, \( W_1 : v, u, v \) is a self–walk of \( v \) with a length
\( l(W_1) = 2 \).
For the odd–self walk $W$ of $v \in V(D - C)$: Start at $v$, then transition to any vertex $u \in C$. Go around $C$ once, back to $u$ and then transition from $u$ back to $v$. Since the cycle $C$ has an odd length and the sum of the lengths of the walks from $v$ to $u$ and from $u$ to $v$ is even, $W$ will have an odd length. For the odd–self walk $W$ of $v \in C$: Start at $v$ and go around $C$ once, back to $v$. Since the cycle has an odd length, $W$ will also have an odd length—$W = C$.

Hence, all $v \in V(D)$ are aperiodic. Therefore, by definition the Markov chain is aperiodic and the result is proven.

We now have the following result as a consequence of the above.

**Corollary 4.3.5.** All initial distributions of a connected non–bipartite graph $G$ converge to a stationary distribution.

**Proof.** Let $G$ be a graph that is connected and non–bipartite. Since $G$ is connected, it is an irreducible Markov chain. From Theorem 4.3.4, $G$ is an aperiodic Markov chain. As a result of Theorem 4.1.6, we find that any initial distribution of $G$ will converge to a stationary distribution.

We will now consider an example concerning a connected non–bipartite graph $G$.

**Example 4.3.6.** Let $G = (V, E)$ be the following.

\[
G : \begin{array}{c}
v_1 \\
v_2 \\
v_3 \\
v_4 \\
v_1
\end{array}
\]

\[
D : \begin{array}{c}
v_1 \\
v_2 \\
v_3 \\
v_4 \\
v_1
\end{array}
\]

Figure 4.3: An undirected connected non–bipartite graph $G$.

Therefore, its Markov digraph $D$ is depicted as follows.

Figure 4.4: The Markov digraph $D$ of the connected non–bipartite graph $G$. 89
Say that we want to find the mean return time for \( v_1 \). Earlier in this section, we demonstrated that we have the following stationary distribution for an undirected graph \( G \):

\[
\pi = \left[ \frac{\text{deg} G(v_1)}{2m} \quad \frac{\text{deg} G(v_2)}{2m} \quad \ldots \quad \frac{\text{deg} G(v_n)}{2m} \right]
\]

where \( m \) is \( |E(G)| \).

For this particular problem,

\[
\pi = \left[ \frac{3}{10} \quad \frac{2}{10} \quad \frac{3}{10} \quad \frac{2}{10} \right] = \left[ \frac{3}{10} \quad \frac{1}{5} \quad \frac{3}{10} \quad \frac{1}{5} \right]
\]

Since \( G \) is a connected non–bipartite graph, it is an irreducible aperiodic Markov chain. (The aperiodicity follows from Theorem 4.3.4.) Therefore, from Theorem 4.1.8, we have that the mean return time for any \( v_i \) is \( \tau_{i,i} = \frac{1}{\pi_i} \). Hence,

\[
\tau_{1,1} = \frac{1}{\pi_1} = \frac{10}{3}
\]

is the mean return time for \( v_1 \)

In the next chapter, we will simulate some Markov chains with the aid of the computer language R.
Chapter 5

Computer Simulation of Markov Chains

One of the most interesting things about Markov chains is that they can be simulated on a computer. More precisely, the Markov Chain Monte Carlo methods will be of particular interest in our discussion—[1], [3], [19], [20] and [22]. These methods were originally developed by physicists during the 1950’s. However, it was not until the 1980’s that this area enjoyed an increase in interest, when it was used in image analysis. With further improvements of computer systems, this method became very important to the field of statistics—namely, Bayesian statistics. We will see how to use this method to simulate a random $q$-colouring of a graph (see [9] and [19]).

Furthermore, these methods and other related methods can be used to simulate the Ising model. The Ising model is a special case of the more general Potts model. One way of classifying this is that the Ising model has two spin configurations, namely $\{-1, 1\}$; whereas, Potts model has $q$ different spin configurations, where $q$ can be any natural number. A more in-depth discussion on this matter will appear in the next chapter. We will see that the Ising model is a specifically defined 2–colouring of a $k \times k$ grid—i.e. a specific undirected graph. Its implementation can be found in Appendix A. Note that a configuration (for a graph $G = (V, E)$) is one outcome of a defined colouring method.

5.1 Computer Simulation of Markov Chains

We will firstly outline the main idea and then discuss each of the functions that have been implemented in R. It is important to note that all these functions require the igraph library in R. Furthermore, all the R–code has been done by the author.

Firstly, we define two functions:

• an initialization function $\Psi$ and
• an update function $\phi$.

Let $\Psi : [0, 1] \rightarrow V(D)$ [this is the mapping of the unit interval to the vertex set associated with the Markov chain]. This function is used to generate the initial point $X_0$ in a simulation of our Markov chain. To make the above association, we divide the unit interval into $|V(D)| = n$ subintervals. The length of each of these intervals is dependent on each $s \in V(D)$—the total length of the interval on which $\Psi(x) = s$ equals $\mu^{(0)}(s)$; where $x \in [0, 1]$ and $\mu^{(0)}(s)$ is the probability of being at vertex $s$ in
From the above, we would construct $\Psi$ for our Markov chain as follows.

$$
\Psi(x) = \begin{cases}
  s_1 & \text{for } x \in [0, \mu^{(0)}(s_1)) \\
  s_2 & \text{for } x \in [\mu^{(0)}(s_1), \sum_{i=1}^2 \mu^{(0)}(s_i)) \\
  \vdots & \\
  s_j & \text{for } x \in \left[\sum_{i=1}^{j-1} \mu^{(0)}(s_i), \sum_{i=1}^j \mu^{(0)}(s_i)\right) \\
  \vdots & \\
  s_n & \text{for } x \in \left[\sum_{i=1}^{n-1} \mu^{(0)}(s_i), 1\right]
\end{cases}
$$

Clearly, $\Psi : [0,1] \rightarrow V(D)$—this can simply be observed from the above construction. To verify the lengths of the intervals for this initialization function, we can simply do the following.

Let $\mathbb{I}(\Psi(x) = s)$ be the indicator function defined as

$$
\mathbb{I}(\Psi(x) = s) = \begin{cases}
  1 & \text{if } \Psi(x) = s \\
  0 & \text{otherwise}
\end{cases}
$$

Next we integrate piecewise over the unit interval.

$$
\int_0^1 \mathbb{I}(\Psi(x) = s_j) \, dx = \int_0^{\mu^{(0)}(s_1)} \mathbb{I}(\Psi(x) = s_j) \, dx + \int_{\mu^{(0)}(s_1)}^{\sum_{i=1}^2 \mu^{(0)}(s_i)} \mathbb{I}(\Psi(x) = s_j) \, dx + \ldots + \int_{\sum_{i=1}^{j-1} \mu^{(0)}(s_i)}^{\sum_{i=1}^j \mu^{(0)}(s_i)} \mathbb{I}(\Psi(x) = s_j) \, dx + \int_{\sum_{i=1}^j \mu^{(0)}(s_i)}^1 \mathbb{I}(\Psi(x) = s_j) \, dx
$$

$$
= \sum_{i=1}^j \mu^{(0)}(s_i) - \sum_{i=1}^{j-1} \mu^{(0)}(s_i)
$$

So the $\Psi$ we constructed for our Markov chain is a valid initialization function.

From the above, we can generate our first value $X_0$ from a random uniform number $U_0 \in [0,1]$ by setting $X_0 = \Psi(U_0)$—

$$
P(s(0)) = P(X_0 = s) = P(\Psi(U_0) = s)
$$

$$
= \int_0^1 \mathbb{I}(\Psi(x) = s) \, dx
$$

$$
= \mu^{(0)}(s)
$$
Next we discuss how to generate $X_{n+1}$ from $X_n$ so that we can construct a chain $(X_0, X_1, \ldots)$. We will use a random number from a uniform distribution on the interval $[0, 1]$ to get from $X_n$ to $X_{n+1}$. We now define an update function $\phi : V(D) \times [0, 1] \to V(D)$ that needs to satisfy the following properties.

i) If the vertex $s_j$ is fixed, then $\phi(s_j, x)$ is a piecewise constant of $x$.

ii) For each vertex $s_i$ and $s_j \in V(D)$ that are fixed, the length of the interval of $\phi(s_i, x) = s_j$ is $p_{ij}$.

Keeping the above in mind, we construct such a function $\phi$ and then verify that these properties are satisfied. Consider the following construction for a vertex $s_i \in V(D)$,

$$\phi(s_i, x) = \begin{cases} 
  s_1 & \text{for } x \in [0, p_{i1}) \\
  s_2 & \text{for } x \in [p_{i1}, \sum_{j=1}^{2} p_{ij}) \\
  \vdots \\
  s_l & \text{for } x \in [\sum_{j=1}^{l-1} p_{ij}, \sum_{j=1}^{l} p_{ij}) \\
  \vdots \\
  s_n & \text{for } x \in [\sum_{j=1}^{n-1} p_{ij}, 1] 
\end{cases}$$

Clearly, if $s_i$ is fixed, then all transition probabilities in the matrix $P$ are constants and thus property i) is satisfied. In the same spirit as the initialization function, we can define an indicator function,

$$I(\phi(s_i, x) = s_l) = \begin{cases} 
  1 & \text{if } \phi(s_i, x) = s_l \\
  0 & \text{otherwise} 
\end{cases}$$

and integrate piecewise over the interval $[0, 1]$ as follows:

$$\int_0^1 I(\phi(s_i, x)) \, dx = \int_0^{p_{i1}} I(\phi(s_i, x)) \, dx + \int_{p_{i1}}^{\sum_{j=1}^{2} p_{ij}} I(\phi(s_i, x)) \, dx + \ldots + \int_{\sum_{j=1}^{l-1} p_{ij}}^{\sum_{j=1}^{l} p_{ij}} I(\phi(s_i, x)) \, dx + \ldots + \int_{\sum_{j=1}^{n-1} p_{ij}}^{1} I(\phi(s_i, x)) \, dx$$

$$= \sum_{j=1}^{l} p_{ij} = l - \sum_{j=1}^{l-1} p_{ij} = p_{il} \quad \text{for } l = 1, 2, \ldots, n$$

Thus property ii) is satisfied and we have a valid update function.
Using these two functions, we can simulate the Markov chain as follows.

\[
X_0 = \Psi(U_0) \\
X_1 = \phi(X_0, U_1) \\
X_2 = \phi(X_1, U_2) \\
\vdots
\]

where \( U_0, U_1, \ldots \in [0,1] \) are all random uniform variables.

The above functions have been slightly modified in the upcoming code.

The initialization function: \textit{initial\_psi}

- **Input:**
  - \( u_0 \) which is a vector of the initial distribution.
  - A Markov digraph \( G \).
- **Output:** A random vertex in the Markov digraph \( G \).
- **Procedure:**
  - Firstly, we generate a single random number, \( u \), from a uniform distribution that is on the interval \([0,1]\). One way of doing this in R is to use the \textit{runif} function, which can clearly be observed in the code below.
  - The following initializations must be done:
    * \( n = |V(G)| \)
    * \( \text{count} = 0 \)
    * \( \text{lbound} = 0 \), where \( \text{lbound} \) will be the lower bound in a “moving” interval.
    * Furthermore, set \( \text{upbound} \) equal to the first entry in our initial distribution \( u_0 \).
    - We now utilize the following loop.

    From the first entry to the last entry in \( u_0 \), if our random number is less than \( \text{upbound} \) and greater than or equal to \( \text{lbound} \), then we select the vertex of \( G \) that corresponds to the value of \( \text{count} \). Take note that the package \textit{igraph} numbers the vertices with integers starting from zero as it is simpler and faster to compute with. To avoid a potential problem from occurring when our random number is equal to one, we return the number of the last vertex in our set \( V(G) \).

    If none of the above are satisfied, then we increment \( \text{count} \) by one and we add \( u_0[i] \) to \( \text{lbound} \), where \( u_0[i] \) is the \( i \)th entry in our initial distribution. Similarly, we add \( u_0[i+1] \) to \( \text{upbound} \) and repeat the last bullet again.

The implementation of the above algorithm in R is given below.

```R
# Takes an initial distribution u_0 and a directed graph G
initial_psi<-function(u_0,G)
{
  u<-runif(1,0,1)
  n<-length(V(G))

  count<- 0
  lbound<- 0
  upbound<-u_0[1]
  \vdots

  while(count < n)
  {
    if(u < upbound && u >= lbound)
    {
      selected_vertex <- V(G)[count]
    }
    else
    {
      lbound <- u
      upbound <- u
    }
    count <- count + 1
  }
}
```
for(i in 1:length(u_0))
{
    if(u >=lbound & u < upbound )
    {
        return(count)
    }

    if(u==1)
    {
        return(n-1)
    }

    count<-count+1
    lbound<-lbound+u_0[i]
    upbound<-upbound+u_0[i+1]
}
}

Similarly, the update function $\phi$ is defined in the proof of the Markov Chain Convergence Theorem; however, we will need to use the transition matrix $P$. We will now discuss the function Transition_mat_dir which will give us our transition matrix $P$ for our Markov digraph—referred to as $G$ in the code and algorithm.

- **Input:** A Markov digraph $G$ with arc–weights/directed edge weights.
- **Output:** Our transition matrix $P$.
- **Procedure:**
  - We initialize an $n \times n$ matrix $P$ of zeros.
  - For every two vertices in our graph, we check if there is an arc between them. One way is to see if the length of such an arc is different from zero. If our result is zero, then it implies that the arc does not exist and we then move to the next comparison. Otherwise, for the vertices $(i-1)$ and $(j-1)$, we place the weight of the arc between these two in the $i^{th}$ row and $j^{th}$ column of $P$.

The following R–code corresponds to the above.

```r
## transition matrix for a digraph G
Transition_mat_dir<-function(G)
{
    P<-matrix(0,nrow=length(V(G)),ncol=length(V(G)))
    for(i in 1:length(V(G)))
    {
        for(j in 1:length(V(G)))
        {
            if(length(E(G)[(i-1)%>%>(j-1)])!=0)
            {
                P[i,j]<-E(G)[(i-1)%>%>(j-1)]$weight
            }
        }
    }
}
```

95
Next we define the function \textit{update\_phi}, which will be used to progress in our Markov chain. This function is very similar to \textit{initial\_psi}. The main differences are:

- Instead of using the initial distribution \(u_0\), we use the row vector of our transition matrix that is related to our current vertex in the process. Thereafter, the remaining procedure is identical.

For the sake of completeness, we now present the algorithm and the code associated with the above.

- Function name: \textit{update\_phi}
- Input:
  - A Markov digraph \(G\)
  - The current vertex in our Markov chain, which is an integer.
- Output: A random neighbour of our current state.
- Procedure: Let \(X\) be equal to the current vertex in our process/chain. Set \(P\) equal to the transition matrix of \(G\) and let the vector \(vec\) be equal to \(X + 1\)th row of \(P\). We add one to the above as matrices in R start indexing from 1, whereas \textit{igraph} indices the vertices from 0. The rest of the procedure follows identically to \textit{initial\_psi}, the only difference is that we use the vector \(vec\) instead of \(u_0\).

The associated R–code is given below.

\begin{Verbatim}
update_phi<-function(G,process)
{
  X_i<-process
  P<-Transition_mat_dir(G)
  vec<-P[X_i+1,]
  u<-runif(1,0,1)
  n<-length(V(G))

  count<- 0
  lbounbd<- 0
  upbound<-vec[1]

  for(i in 1:length(vec))
  {
    if(u >=lbounbd & u < upbound )
    {
      return(count)
    }

    if(u==1)
    {
      return(n-1)
    }
  }
}
\end{Verbatim}
Next we define the function \emph{MarkovSim} which uses the above functions.

- **Function name**: \emph{MarkovSim}
- **Input**:
  - The initial distribution vector $u_0$.
  - The Markov digraph $G$ which has its vertices labelled.
  - The number of steps in our simulated Markov chain.
- **Output**: A list which contains the simulated Markov chain and the distribution of the visitations to the vertices of our Markov digraph $G$.
- **Procedure**:
  - We set the variable $\text{temp}$ equal to the output of \emph{initial_psi}. We also set $\text{process}$ equal to \emph{NULL} so that it does not contain any elements.
  - For each step, we add a new vertex at the end of $\text{temp}$ using the \emph{update_phi} function.
  - Thereafter, we get the labels that are associated with these vertices in $\text{temp}$ and place these results in $\text{process}$.
  - We let $n$ be $|V(G)|$, $\text{total}$ be the number of elements in $\text{temp}$ and set $\text{visitations}$ equal to a vector of zeros that has $n$ elements in it.
  - We then count the number of times that each vertex is observed in $\text{temp}$ and divide each by the value $\text{total}$. These entries are placed in $\text{visitations}$ while they are being read.
  - Return the results: $\text{process}$ and $\text{visitations}$. These results are our simulation and the distribution of visitations.

The code of the above is:

```r
MarkovSim<-function(u_0,G,k=5)
{
  temp<-initial_psi(u_0,G)
  P<-Transition_mat_dir(G)
  process<-NULL

  for(i in 1:k)
  {
    temp<-cbind(temp,update_phi(G,temp[length(temp)]))
  }

  for(j in 1:length(temp))
  {
    process<-cbind(process,V(G)$label[temp[j]+1])
  }

  n<-length(V(G))
  total<-length(temp)

  # Remaining code
}
```
visitations<-rep(0,n)
for(i in 1:n)
{
  visitations[i]<-sum(temp==(i-1))/total
}
return(list(process,visitations))
}

We now consider a short function for an undirected graph which will give us its Markov digraph. This
procedure was outlined earlier in chapter 4. We now continue with this particular algorithm.

- Function name: `MarkovDigraph`
- Input: An undirected graph $G$ with no loops.
- Output: The Markov digraph of the above graph $G$.
- Procedure:
  - We construct $D$ by replacing each undirected edge in $G$ with a forward and a backward arc.
  - For all the arcs, we define the attribute “weight” and set them equal to zero temporarily.
  - For each vertex, we obtain its outdegree and assign its inverse to each of its arcs that are leaving
    it. More simply, let $v \in V(D)$. Then for each arc that is incident with the out–neighbours of $v$,
    we associate the weight $\frac{1}{\text{deg}_D(v)}$.
  - Afterwards, we return the resulting digraph $D$.

The R–code of the above is now given.

```r
#gives the Markov digraph for an undirected graph G
MarkovDigraph<-function(G)
{
  D_ <-as.directed(G, mode = c("mutual", "arbitrary"))
  edge_weights<-rep(0,ecount(D_))
  D_<-set.edge.attribute(D_,"weight",value=edge_weights)
  for(i in 0:(length(V(G))-1))
  {
    deg <- neighbors(G,i,mode=c("out"))
    E(D_)[i%->%deg]$weight <- 1/length(deg)
  }
  return(D_)
}
```

For the Markov chain simulation, we could run it for a large number of steps, and then use the propor-
tion of visitations to approximate our stationary distribution (if there is one).

Two examples are now considered. The first example is a Markov digraph which is constructed and
displayed in R. Using the initial distribution \[
\begin{bmatrix}
1 & 0 & 0 & 1 \\
2 & 0 & 0 & 0 \\
\end{bmatrix},
\]
we simulate a Markov chain and output the distribution of visitations. We have only simulated 5 steps to illustrate the principle of this method.
The code and output are displayed as follows:

```r
> Vec_from<-c(0,1,2,3)
> Vec_to<-c(1,3,1,1)
>
> VerLabel<-c("v1","v2","v3","v4")
> d1<-data.frame(from=Vec_from,to=Vec_to,Labels=Vec_from)
> G1<-graph.data.frame(d1,directed=TRUE)
> G1 <- set.vertex.attribute(G1, "label", value=VerLabel)
> G1 <- set.edge.attribute(G1, "weight", value=rep(1,4))
> tkplot(G1,edge.label=get.edge.attribute(G1,"weight"))
```

![Graph visualization](image)

```r
> set.seed(200608804)
> MarkovSim(c(1/2,0,0,1/2),MarkovDigraph(G1))
```

```r
[[1]]
[1,] "v4" "v2" "v4" "v2" "v4" "v2"

[[2]]
[1] 0.0 0.5 0.0 0.5
```

Therefore, the distribution of visitations after 5 steps is $[0 \ 0.5 \ 0 \ 0.5]$.

In example 2, we construct an undirected graph $G$ and label its vertices. Thereafter, we run our `MarkovSim` function for 50 steps with the initial distribution $\begin{bmatrix}
\frac{1}{2} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\
0 & 0.5 & 0 & 0 & 0 & 0
\end{bmatrix}$. The relevant code and output are displayed as follows:

```r
> # constructing the graph G from the vectors vec_from and vec_to
> vec_from <- c(1,2,3,4,5,6,7,1,2,2,2)
> vec_to <- c(2,3,4,5,6,7,1,6,7,5,6)
> d<-data.frame(from=vec_from,to=vec_to)
> G<-graph.data.frame(d,directed=FALSE)
>
> # Relabelling the vertices agree with the graph
```

99
Ver_lab<-NULL
for(i in 1:length(V(G)))
  { Ver_lab<-cbind(Ver_lab,c(paste("v",i,sep="")))
  }
G <- set.vertex.attribute(G, "label", value=Ver_lab)
#Plotting the graph G
tkplot(G)

set.seed(200608804)
Mar
kovSim(c(1/2,0,0,1/4,0,1/4,0),MarkovDigraph(G),50)

[[1]]
[,1] "v4" "v5" "v6" "v5" "v2" "v3" "v2" "v1" "v6" "v7" "v1" "v2" "v6" "v1"

[[2]]
[1] 0.19607843 0.23529412 0.01960784 0.03921569 0.15686275 0.23529412 0.11764706

Therefore, the distribution after 50 steps is

[0.19607843 0.23529412 0.01960784 0.03921569 0.15686275 0.23529412 0.11764706]
5.2 Markov Chain Monte Carlo (MCMC) Methods

The term “Monte Carlo” alludes to the name of a famous casino where random numbers are generated by croupiers. In light of this, Monte Carlo methods refers to a wide variety of methods that use randomly generated values from some kind of distribution to solve a particular problem numerically. Usually, there are boundary constraints that these random numbers must fall into. The main idea is to generate a large number of random numbers subject to these constraints and to approximate the solution(s) to our problem (usually, these cannot be solved analytically) between these bounds.

In other words, MCMC is the Monte Carlo method applied to Markov chains that are irreducible and aperiodic. The reason for this is that we want our algorithms to converge—for irreducible aperiodic Markov chains, we have any initial distribution converging to the unique stationary distribution. Our random numbers in this case must be the random configurations.

We will now discuss how the Monte Carlo methods can be used to solve the Ising model—a physical system in which atoms are assigned one of two types of spins (an up–spin and a down–spin). This model will be discussed in more detail in the next chapter.

The average of an observable $A(s)$ for the Ising model is given by

$$\langle A \rangle = \sum_s A(s) \pi_s = \frac{1}{Z_{G,\beta}} \sum_s A(s) \exp[-\beta U(s)]$$

where $Z_{G,\beta}$ is the partition function, $\beta = \frac{1}{kT}$ ($k$ is the Boltzmann constant and $T$ is the temperature), $s$ is a spin configuration and $\pi_s$ is the entry of vertex/configuration $s$ in the stationary distribution $\pi$.

In general, these methods are used to solve problems that cannot be solved analytically. A numerical approach using Monte Carlo methods is usually used to solve such problems. In these simulations, we generate a random sample of spin configurations say $s_1, s_2, \ldots, s_M$ from some distribution $\mu$. We then estimate $\langle A \rangle$ as follows:

$$\overline{A} = \frac{\sum_{m=1}^{M} A(s_m) \exp[-\beta U(s_m)]}{\sum_{m=1}^{M} \exp[-\beta U(s_m)] \mu_{s_m}}$$

$$= \frac{\sum_{m=1}^{M} A(s_m) W(s_m)}{\sum_{m=1}^{M} W(s_m)}$$

where $W(s) = \frac{\pi_s}{\mu_s}$ is some kind of “weight”.

For the random variable $\overline{A}$ to be a good estimate of the number $\langle A \rangle$, we must increase the total number of $M$ configurations being generated. However, $M$ is dependent on the choice of $\mu$. When $\mu$ is very different from $\pi$, it will predominantly sample configurations that are not representative of the thermal equilibrium. By making $M$ larger, the simulation becomes very time-consuming. Furthermore, there is no guarantee that a maximum $M$ will outweigh the error incurred by a poor choice of $\mu$.

To get meaningful results, there are two types of methods that can be considered.
(a) Static Monte Carlo methods.
In these methods, a sequence of statistically independent configurations is generated from the distribution \( \mu \). One shortfall for this algorithm is that the “weights” \( W(s) \) must be monitored and adjusted.

(b) Dynamic Monte Carlo methods.
These methods generate a sequence of correlated configurations through some Markov process which has a unique stationary distribution \( \pi \). Since the Markov process has no memory, a future configuration \( s' \) is generated from the current configuration \( s \).

From the above, Markov Chain Monte Carlo methods are examples of dynamic Monte Carlo methods.

Let the configuration space be discrete and the Markov process evolve in discrete time steps. The time evolution of this Markov chain may be characterized by

\[
P(s(t+1)) - P(s(t)) = \sum_{s \neq s'} [P(s(t+1) | s'(t)) P(s'(t)) - P(s'(t+1) | s(t)) P(s(t))] \tag{5.2.1}
\]

The above expresses the balance between the fluxes. The first term on the last line of (5.2.1) gives the flux of all states/configurations towards \( s \), resulting in an increase in \( P(s(t+1)) \). Similarly, the second term on the last line of (5.2.1) is the flux away from \( s \) towards all other configurations; hence, \( P(s(t)) \) decreases. By including the missing terms \( s = s' \) in (5.2.1), we can use the fact that a transition from \( s' \) to some \( s \) (including \( s' \) itself) will occur with certainty. Hence, we have the familiar result

\[
\sum_{s'} P(s(t+1) | s'(t)) = 1 \tag{5.2.2}
\]

Furthermore, we recall the following from the law of total probability

\[
P(s(t+1)) = \sum_{s'} P(s(t+1) | s'(t)) P(s'(t)) \tag{5.2.3}
\]

For the application of these results to statistical physics, we need \( P(s(t)) \) to converge to a unique stationary value \( \pi_s \) in the unique stationary distribution \( \pi \). Hence, \( P(s(t+1)) = P(s(t)) = \pi_s \). Therefore, the last line of (5.2.1) must be equal to zero when we obtain equilibrium. Thus, all the terms in the summation must be equal to zero. As a result, we have

\[
P(s(t+1) | s'(t)) \pi_{s'} = P(s'(t+1) | s(t)) \pi_s
\]

since \( P(s'(t)) = \pi_{s'} \) and \( P(s(t)) = \pi_s \) at the limit. We recall that the above is the detailed balance equation. The detailed balance equation is a stringent criterion. A less stringent criterion is that of stationarity \( \pi_s = \sum_{s'} P(s(t+1) | s'(t)) \pi_{s'} \). This is the reason why we require our Markov chain to be aperiodic and irreducible.

We will now verify that sequential updating is a valid updating scheme if the individual steps satisfy the detailed balance equation or the property of stationarity for each unit interval of time.

Consider

\[
P(s(N+1) | s'(0)) = \sum_{x_N} \ldots \sum_{x_2} \sum_{x_1} P(s(N+1) | x_N(N)) \ldots P(x_2(2) | x_1(1)) P(x_1(1) | s'(0))
\]
The above is the probability to transition from \( s' \) at time \( t = 0 \) to \( s \) at time \( N + 1 \). Multiplying the above equation by \( \pi_{s'} \) and then, summing over all \( s' \), we get

\[
\sum_{s'} P(s(N + 1) \mid s'(0)) \pi_{s'} = \sum_{x_N} \ldots \sum_{x_2} \sum_{x_1} P(s(N + 1) \mid x_N(N)) \ldots P(x_2(2) \mid x_1(1)) \sum_{s'} P(x_1(1) \mid s'(0)) \pi_{s'}
\]

From stationarity, \( \pi_{x_1} = \sum_{s'} P(x_1(1) \mid s'(0)) \pi_{s'} \). Therefore,

\[
\sum_{s'} P(s(N + 1) \mid s'(0)) \pi_{s'} = \sum_{x_N} \ldots \sum_{x_2} \sum_{x_1} P(s(N + 1) \mid x_N(N)) \ldots P(x_2(2) \mid x_1(1)) \pi_{x_1}
\]

By repetitively applying the stationarity property, we have

\[
\sum_{s'} P(s(N + 1) \mid s'(0)) \pi_{s'} = \pi_s
\]

Therefore, sequential updating preserves the stationarity of \( \pi \) and so the above is a correct simulation procedure.

### 5.2.1 Gibbs Sampler

This section is concerned with the implementation of the Gibbs sampler algorithm (a Markov Chain Monte Carlo method) for simulating Markov chains given in [19]. These methods are concerned with a system of states—like a graph—in which each state is assigned a value of \( \lambda_i \) from a set of values \( \Lambda \) according to a set of rules. We should keep in mind that algorithms are guidelines and so we need to do a little more work to program them.

The Gibbs sampler is an algorithm that simulates samples of a target Gibbs distribution by sampling its conditional marginal distribution. The conditional marginal distribution is as follows:

\[
p(x_s \mid x_N(s)) = \frac{p(x)}{\sum_{\lambda_i \in \Lambda} p(x) \mid x_s = \lambda_i}
\]

The left-hand side of the above can be read as: “The probability to change the value of vertex \( s \) in the configuration \( x \) given the values of the neighbours(\( N(s) \)) of \( s \).” The above is equal to the right-hand side of the equation, which is read as: “The probability of being at configuration \( x \), divided by the sum of probabilities of all possible configurations that we can transition to by changing (or not changing) the value of \( s \) in \( x \).”

Therefore, while sampling, the Gibbs sampler changes a site value (vertex) at each discrete time interval in a configuration. Letting \( x^{(t)} \) be the configuration at time \( t \) and \( s \) be the vertex that we are visiting (selected randomly), the new configuration \( x^{(t+1)} \) will be

\[
x_r^{(t+1)} = \begin{cases} 
\text{a sample from } p(x^{(t+1)}_s \mid x^{(t)}_N(s)) & r = s \\
\text{from } x_r^{(t)} & r \neq s
\end{cases}
\]

103
Firstly, we recall a familiar function: the norm of a vector. This function will be used to check a condition in two of the upcoming functions.

One particular norm of a vector $\overrightarrow{v} \in \mathbb{R}^n$, the Euclidean norm is as follows:

If $\overrightarrow{v} = [v_1 \ v_2 \ \ldots \ v_n]$, then the norm of $\overrightarrow{v} \in \mathbb{R}^n$, denoted by $\|\overrightarrow{v}\|$ is

$$\|\overrightarrow{v}\| = \left(\sum_{i=1}^{n} v_i^2\right)^{\frac{1}{2}}$$

To program this in R, we use “vectorization”. This means that R can work directly with vectors (it is more efficient than using loops).

- Function name: norm
- Input: Any numerical vector $v$
- Output: The Euclidean norm of $v$
- Procedure: Simply return the square-root of the sum of the elements of $v$ to the power of 2.

The R–code follows below:

```r
# Calculates the norm of a vector v
norm<-function(v)
{
    return(sqrt(sum(v^2)))
}
```

We now concern ourselves with a colouring rule for an undirected graph $G = (V, E)$ which will be used to imitate the behaviour of an ideal gas—here, the value 1 represents a particle and 0 indicates an empty space. Therefore, for each vertex $v \in V(G)$, we assign the value 0 or 1, where no two adjacent vertices may have the value of 1. We will call a particular assignment of 0’s and 1’s to the vertices a configuration. A configuration in which no two adjacent vertices have the value 1, is said to be feasible. Otherwise, it is non–feasible.

To pick a feasible configuration randomly means that we select one of these configurations with equal probability—i.e. we have a probability of $\frac{1}{Z_G}$ of selecting a particular feasible configuration $\xi$, where $Z_G$ is the total number of feasible configurations of $G$. Most of the time, some graphs can get large and complicated and so it would be extremely time–consuming to find the number of feasible configurations by hand. The solution is to construct an irreducible aperiodic Markov chain which has as its vertex set $V(D)$ the set of all feasible configurations of $G$. Therefore, let

$$V(D) = \left\{ \xi \in \{0, 1\}^{\left| V \right|} \mid \xi \text{ is feasible}\right\}$$

where $\{0, 1\}^{\left| V \right|}$ indicates the assignment of 0’s and 1’s to the vertices.

Before we continue, we define $X_{k+1}(v)$ to be the colour of the vertex $v$ at time $k + 1$.

The main idea of the algorithm is as follows for each step $k + 1$:
1) Pick a vertex \( v \in V(G) \) randomly.
2) Toss a fair coin (or generate a random uniform number \( u \) between 0 and 1, which can also be denoted by \( u_{[0,1]} \)).
3) If we obtain a heads (or if \( u \leq 0.5 \)) and all the neighbours of \( v \) have the value 0 at \( k \), then we let \( X_{k+1}(v) = 1 \). Otherwise, \( X_{k+1}(v) = 0 \).
4) We leave all the vertices other than \( v \) unchanged.

This algorithm is an example of a Gibbs sampler, which is a Markov Chain Monte Carlo method. Preferably, we continue the above for a large number of steps so that we will go through all the different configurations. We will also take note of the number of times that we visit each configuration. The idea is as follows.

- For each \( \xi \in \{0, 1\}^{|V|} \) \( \mu_G(\xi) = \frac{\text{number of times } \xi \text{ is visited}}{\text{total number of visitations}} \)
- As we increase the number of steps, the Markov Chain Convergence Theorem says that we will tend towards our unique stationary distribution. Therefore, we will be able to approximate this with the above argument.
- For the Markov Chain Convergence Theorem, we will need to prove that this Markov chain is irreducible and aperiodic.

Informally, we verify that our Markov chain is irreducible.

**Proof.** Say that we start with the configuration in which all vertices are assigned 0. Following the above algorithm for \( k \) steps, we end up by a configuration \( \xi \). Say we have \( k_1 \) non–zero assignments. Clearly, we have a probability of \( \frac{k_1}{|V(G)|} \) of selecting one of these. Furthermore, we have a probability of 0.5 to get “tails”. Iteratively, we have the following probability to get back to the zero configuration.

\[
P(\text{transition from } \xi \text{ to the zero configuration in } k_1 \text{ steps}) = \left( \frac{0.5}{|V(G)|} \right)^{k_1} \prod_{i=1}^{k_1} \left( 0.5 - \frac{(k_1 - i)}{|V(G)|} \right) = \left( \frac{0.5}{|V(G)|} \right)^{k_1} k_1! > 0 \quad : k_1 > 0
\]

This means that there exists a walk from \( \xi \) to the zero configuration. By the same argument, we can walk to another configuration \( \xi' \) and then go back again. We note that \( \xi \) and \( \xi' \) are arbitrary. Furthermore, \( \xi \) and \( \xi' \) can reach the zero configuration and the zero configuration can in turn reach them. Therefore, our Markov chain is irreducible.

Let us prove that the above is aperiodic.

**Proof.** To do this, we can show that one of these configuration “vertices” has a self–walk of length 1 (a loop). For time \( k \), let \( v \) be our random vertex in \( V(G) \) and let our coin toss yield “heads”. Furthermore, let all the neighbours of \( v \) have the value 0 assigned to them. Therefore, \( X_k(v) = 1 \). Call this configuration \( \xi' \). Now say that we have the following for time \( k + 1 \): We pick \( v \) again and our coin
lands on heads again. The neighbours of \( v \) are still zero; therefore, \( X_{k+1}(v) = 1 \) again. We are still at \( \xi' \). Therefore, our Markov digraph for this Markov chain would have a loop at the vertex that corresponds to \( \xi' \). So our Markov chain is aperiodic.

We can also verify the above by inspecting the output from our R–code. Furthermore, since the Markov chain is irreducible and aperiodic; we may use the Markov Chain Convergence Theorem.

To prove that \( \vec{\mu} \) is the stationary distribution, we will use Theorem 4.2.1—i.e. we need to prove that \( \vec{\mu} \) is reversible. In terms of the notation from earlier, we need to prove that

\[
\mu_G(\xi)p_{\xi\xi'} = \mu_G(\xi')p_{\xi'\xi}
\]

(5.2.4)

where \( \xi \) and \( \xi' \) are two feasible configurations and \( p_{\xi\xi'} \) and \( p_{\xi'\xi} \) are the entries in our transition matrix which has all the feasible configurations as its state space.

**Proof.** Three cases that depend on the distance \( d \) between \( \xi' \) and \( \xi \) need to be considered.

**Case 1:** \( d = 0 \), which implies that \( \xi = \xi' \). Using the above together with the fact that \( \mu_G(\xi)p_{\xi\xi} = \mu_G(\xi)p_{\xi\xi} \), we get \( \mu_G(\xi)p_{\xi\xi'} = \mu_G(\xi')p_{\xi'\xi} \).

**Case 2:** \( d \geq 2 \). Since we only change one vertex at a time in our algorithm—the distance between \( \xi \) and \( \xi' \) can only be 1. Therefore,

\[
p_{\xi\xi'} = p_{\xi'\xi} = 0
\]

and so (5.2.4) is satisfied.

**Case 3:** \( d = 1 \) implies that \( \xi \) and \( \xi' \) differ in value at one vertex, say \( v \). For \( v \) to be 0 in one of these two configurations and to be assigned the value 1 in the other implies that the neighbours of \( v \) must have the value 0 assigned to each one of them. Consequently, its colour is determined by “heads” or “tails”. Therefore

\[
p_{\xi\xi'} = \frac{1}{2}\left(\frac{1}{n}\right) = p_{\xi'\xi}
\]

where \( \frac{1}{2} \) is the probability to change to the other colour and \( \frac{1}{n} \) is the probability of selecting \( v \).

Since all vertices are selected randomly (uniformly), every configuration is as likely to be visited. Therefore, every configuration is visited an equal number of times. Hence, \( \mu_G(\xi) = \mu_G(\xi') \) and so (5.2.4) is satisfied again.

From the above cases, \( \vec{\mu} \) is reversible and so it is stationary.

Next we will give the algorithms of the functions that are needed and then show their implementations in R.

The function \texttt{equal_vecs} is used to check if a vector \( \vec{v} \) is a row vector in a matrix \( M \). If there exists such a row, then we will return the position of the first row that is equal to \( \vec{v} \) in the matrix \( M \). Otherwise, we return \texttt{false}. The code follows below:

```r
equal_vecs<-function(v1,M)
{
  for(i in 1:nrow(M))
  {
    if(sum(v1==M[i,])==length(v1))
    {
```
Next we discuss the function \textit{rand\_feasible} which is used to get a random feasible configuration.

- Function name: \textit{rand\_feasible}
- Input:
  - A digraph $D$
  - The number of steps, $k$ to be used (this is optional as a default value of 1000 will be set)
- Output: The same Markov digraph $D$ with a random feasible configuration.
- Procedure: Initially, we assign the colour “0” to all the vertices. We then use a previous function called \textit{initial\_psi} to find an initial random vertex. For each of the $k$ steps, we randomly pick an “h” or a “t” and a vertex $v$. If our outcome is “h” and all the neighbours of our vertex $v$ have a colour 0, then we colour $v$ with a value of 1. Otherwise, colour it with 0. We repeat the above procedure moving randomly from neighbour to neighbour using the \textit{update\_phi} function. Thereafter, we return our random configuration.

The R–code of the above algorithm is presented below.

```r
rand_feasible<-function(D_,k=1000)
{
  n<-length(V(D_))
  D_<-set.vertex.attribute(D_,"colour",value=rep(0,n))
  ## set up a uniform initial distribution
  vec<-rep(1/n,n)
  ## use initial\_psi to get a starting vertex
  X_0 <- initial\_psi(vec,D_)
  temp<-X_0
  for(i in 1:k)
  {
    res<-sample(c("h","t"),1,replace=T,prob=c(0.5,0.5))
    # 0 -- heads  1 -- tails
    temp<-update\_phi(D_,temp)
    if((res="h")& (norm(V(D_)[neighbors(D_,temp)]$colour)==0))
    {
      V(D_)[temp]$colour<-1
    }
    else
    {
      V(D_)[temp]$colour<-0
    }
  }
}
```

107
Earlier, we gave the main idea of the Gibbs sampler for a hardcore model—the ideal gas model in which each space is assigned the value of 1 or 0; where 1 represents the occupation of a space by a gas molecule and 0 represents no occupation of a space by any gas molecule. Now we will consider the implementation of it.

- Function name: GIBBS
- Input:
  - A Markov digraph of our graph \( G \) with a random feasible configuration.
  - The number of steps to run the algorithm for—we refer to it as \( k \).
- Output: Three .csv files will be written:
  - “configs_gibbs” which will give all the different configurations that our algorithm has encountered.
  - “visited_gibbs” which will indicate the number of times each configuration has been encountered. It will also display the proportion of each configuration.
  - “MC_gibbs” will be used to keep track of our Markov chain.

These three files can be written to a flash drive. In my case, it appears as “J:/”. The last configuration on \( D \), all the different configurations and the distribution of the configurations are also returned for convenience.

- Procedure:
  1) We initialize a matrix called \( configs \), where we will place every new configuration that we encounter.
  2) For each step from 1 to \( k \), we pick “h” or “t” randomly.
  3) Furthermore, we use the function initial_psi to pick one of the vertices of \( D \) randomly and place it in a vector called temp.
  4) If we obtained an “h” and all the neighbours of \( v \) have the colour 0 assigned to them, then we colour \( v \) with 1.
  5) Next we use equal_vecs to check if the current configuration of colours is contained in \( configs \). There are two possible outcomes:
    - If the current configuration is contained in \( configs \), then we have to update the corresponding position in \( visited \) by adding 1 more to its previous entry. Next, we update \( MC \) which will contain the positions corresponding to the rows in \( configs \)—this is our Markov chain.
    - If it is not contained in \( configs \), then we update \( configs \) so that it will include this “new” configuration. Similarly, the vector \( visited \)’s size must be increased by 1 and the value 1 must be placed in that last position. \( MC \) is also updated with this result as in the case of the previous bullet.
  6) If we obtained any result different from that in step 3, then \( v \) must be assigned the colour 0. Thereafter, 4) is followed again.
  7) The appropriate results are then written to the files that are given in output.

The R-code for the above is presented next.

```r
GIBBS<-function(D_,k=1000)
{
  return(D_)
}
```
n<-length(V(D_))

configs<-matrix(V(D_)$colour,nrow=1,byrow=TRUE)
visited<-1
## forming a uniform initial distribution
vec<-rep(1/n,n)

temp<-NULL
MC<-NULL
for(i in 1:k)
{
  res<-sample(c("h","t"),1,replace=T,prob=c(0.5,0.5))
  # 0 -- heads  1 -- tails

  temp<-initial_psi(vec,D_) # pick a random vertex

  if((res=="h")& (norm(V(D_)[neighbors(D_,temp)]$colour)==0))
  {
    V(D_)[temp]$colour<-1
    v_new<-V(D_)$colour

    if(equal_vecs(v_new,configs)[[1]]==TRUE)
    {
      pos<-equal_vecs(v_new,configs)[[2]]
      visited[pos]<-visited[pos]+1
      MC<-rbind(MC,pos)
    }
  }
  else
  {
    configs<-rbind(configs,v_new)
    visited<-cbind(visited,1)
    MC<-rbind(MC,nrow(configs))
  }
}
else
{
  V(D_)[temp]$colour<-0
  v_new<-V(D_)$colour

  if(equal_vecs(v_new,configs)[[1]]==TRUE)
  {
    pos<-equal_vecs(v_new,configs)[[2]]
    visited[pos]<-visited[pos]+1
    MC<-rbind(MC,pos)
  }
}
else
{
    configs<-rbind(configs,v_new)
    visited<-cbind(visited,1)
    MC<-rbind(MC,nrow(configs))
}
}
}

library(MASS)
write.csv(configs,file="J://configs_gibbs.csv")
write.csv(matrix(c(visited,visited/sum(visited)),ncol=2,byrow=FALSE),
    file="J://visited_gibbs.csv")
write.csv(MC,file="J://MC_gibbs.csv")
return(list(D_,configs,visited/sum(visited)))

Two examples involving the above code are presented and their relevant outputs are outputted. The .csv files for these examples may be found in the folder entitled “GIBBS”. For the best possible results the above should be implemented for a large number of steps.

```r
> ############## some graph example
> # constructing the graph G from the vectors vec_from
> # and vec_to
> vec_from <- c(1,2,3,4,5,6,7,1,2,2,2)
> vec_to <- c(2,3,4,5,6,7,1,6,7,5,6)
> d<-data.frame(from=vec_from,to=vec_to)
> G<-graph.data.frame(d,directed=FALSE)
>
> # Relabelling the vertices agree with the graph
> Ver_lab<-NULL
> for(i in 1:length(V(G)))
+ {
+     Ver_lab<-cbind(Ver_lab,c(paste("v",i,sep="")))
+ }
>
> G <- set.vertex.attribute(G, "label", value=Ver_lab)
> D_<-MarkovDigraph(G)
> set.seed(200608804)
> D_<-rand_feasible(D_)
> G1<-GIBBS(D_,2000)
> V(G1[[1]])$colour
[1] 0 1 0 0 0 0 0
> tkplot(as.undirected(G1[[1]]),
+ vertex.label=get.vertex.attribute(G1[[1]],"colour"))
```
> G1[[2]]


text

> G1[[3]]


> set.seed(200608804)
> D2 <- MarkovDigraph(G)
> D2 <- set.vertex.attribute(D2, "colour", value = rep(0, length(V(D2))))
> G2 <- GIBBS(D2, 2000)
> V(G2[[1]])$colour
[1] 0 0 1 0 1 0 1
> tkplot(as.undirected(G2[[1]]),
  + vertex.label = get.vertex.attribute(G2[[1]], "colour"))

> G2[[2]]
0 0 0 0 0 0 0
v_new 0 0 0 0 1 0
v_new 0 0 0 1 0 1 0
v_new 0 0 0 1 0 0 0
v_new 0 0 0 0 0 0 1
v_new 0 0 1 0 0 0 1
v_new 0 0 1 0 0 0 0
v_new 1 0 1 0 0 0 0
v_new 1 0 0 0 0 0 0
v_new 0 0 0 0 1 0 1
v_new 0 0 1 0 1 0 1
v_new 0 0 1 0 1 0 0
v_new 0 0 1 0 1 0 0
v_new 0 0 0 0 1 0 0
v_new 1 0 1 0 1 0 0
v_new 1 0 0 1 0 0 0
v_new 0 0 0 1 0 0 1
v_new 1 0 0 0 1 0 0
v_new 1 0 0 0 1 0 0
v_new 0 1 0 0 0 0 0
v_new 0 1 0 1 0 0 0
5.2.2 Random $q$–colourings

We now consider a generalization of the above in which $q$ different colours are involved. Furthermore, no two adjacent vertices may have the same colour. The field of random $q$–colourings is vast and has a large number of applications. Therefore, we shall limit ourselves to the discussion of its implementation.

We might ask ourselves right now: “what is the minimum number of colours to be used so that we are guaranteed to run through all possible configurations of colours by taking a large number of steps?” In this question, the minimum number of colours is the least number of colours that are needed to have a proper colouring (no two adjacent vertices are assigned the same value/colour). Furthermore, the term “large” is used to refer to a substantial number of steps—statistically, it is usually at least 1000.

Letting our undirected graph be $G = (V,E)$, I would propose to use $\Delta(G) + 2$ colours. The reasoning is as follows. We want at least one colour that is not used by the vertex with the largest degree and its neighbours. As a result, when we run our simulation, the colouring of the graph can change and so we can get all possible configurations. In other words, it assures us that this algorithm would be an irreducible Markov chain. Say that we decided to use $\Delta(G) + 1$ colours and that $G = K_n$. Clearly, once we have coloured $G$, there would be no more colours remaining and so we would be confined to that specific colour configuration—i.e. we would be stuck in one state. This means that this Markov chain would be reducible. Anything less than $\Delta(G) + 1$ colours would result in us not being able to colour $G = K_n$. Therefore, if we were to pick $q \geq \Delta(G) + 2$, then we would be able to move between all the configurations.

Keeping this in mind, the main idea is as follows.

Let $G = (V,E)$, $q \geq \Delta(G) + 2$ and $\{1, 2, \ldots, q\}$ be our set of colours. Let $D$ be the Markov digraph of the Markov chain which has all possible colour configurations as its vertices:

$$V(D) = \{\xi \in \{1, 2, \ldots, q\}^{|V|} \mid \text{no two adjacent vertices have the same colour in } \xi\}$$

Suppose that we start with an arbitrary $q$–colouring of $G$; that is, we are at some state in the Markov digraph $D$. For each step $k + 1$, we will do the following:

1) Pick a $v \in V(G)$ randomly.
2) Next, we remove all the colours of $v$'s neighbours from our set of colours.
3) For $X_{k+1}(v)$, we pick one of the remaining colours randomly and assign it to $v$.
4) Furthermore, all the colour assignments of the remaining vertices are left unchanged.
5) Continue the above for a large number of steps.

We will now prove that the qcoloring algorithm is an irreducible Markov chain.

**Proof.** Consider the digraph $D$ of the above Markov chain.

In the above algorithm, we can select any vertex $v$ in the configuration $\xi$ with equal probability ($\frac{1}{n}$ if $|V(G)| = n$) and change its colour with probability $\frac{1}{q - c(v)}$, where $c(v)$ is the number of unique colours of the neighbours of $v$ and $q \geq \Delta(G) + 2$; $q \in \mathbb{N}$. At worst, $c(v) = \Delta(G)$ and so we change the colour of $v$ with a probability of $\frac{1}{q - \Delta(G)}$. Hence, this colour configuration $\xi$ is adjacent to and, similarly, adjacent from some other configuration $\xi'$. In turn, $\xi'$ is also adjacent to some other configuration. In fact, we can continue to change the colours of the vertices in $G$ by changing the colour configuration of one appropriate vertex at a time until we reach a desired colour configuration of the graph $G$. Since this is true for all the colour configurations (the vertices in $V(D)$), we have that all vertices in $V(D)$ intercommunicate (i.e. $D$ is strongly-connected) and so the reachability matrix of this Markov digraph consists of only ones. Therefore, the Markov chain is irreducible.

To prove that the above Markov chain is aperiodic, we can simply show that one of the vertices in $V(D)$ has a loop. Let $\xi_1$ be such a colour configuration or rather a vertex in $V(D)$ and let $v \in V(G)$. Say $q \geq \Delta(G) + 2$. We have at most $\deg_G(v)$ colours for the neighbours of $v$. The remaining number of colours with which to colour $v$ is, therefore, at least $q - \deg_G(v)$. If $\deg_G(v) = \Delta(G)$, then $q - \Delta(G) \geq 2$. Therefore, we will have one other colour to choose from in case $G = K_n$. Let $r_1$ be the number of remaining colours which includes the previous colour of $v$. Clearly, we can select the same colour for $v$ again with the probability of $\frac{1}{r_1}$. Consequently, we are at the same configuration $\xi_1$ and so our Markov digraph would have a loop at $\xi_1$. Therefore, our Markov chain is aperiodic.

Now consider the following initial distribution:

$$\overline{\rho}_{G,q} = \left[ \begin{array}{cccc} \frac{1}{Z_{G,q}} & \frac{1}{Z_{G,q}} & \cdots & \frac{1}{Z_{G,q}} \end{array} \right]$$

where $Z_{G,q}$ is the total number of configurations of $q$-colourings for our graph $G$. We will prove that $\overline{\rho}_{G,q}$ is a stationary distribution by using Theorem 4.2.1.

Let $d$ be the number of vertices by which two adjacent colour configurations $\xi_1$ and $\xi_2$ differ. Similarly to the argument of the previous section, we consider three cases of $d$: i) $d = 0$ ii) $d \geq 2$ iii) $d = 1$

i) $d = 0$: This means that $\xi_1 = \xi_2$ and so

$$\rho_{G,q}(\xi_1)p_{\xi_1,\xi_2} = \rho_{G,q}(\xi_2)p_{\xi_2,\xi_1}$$

ii) $d \geq 2$: According to the main idea of this algorithm, no two adjacent vertices in $V(D)$ may differ by more than one colour at one vertex $v$.

iii) $d = 1$: Suppose that we select a vertex $v$ in $\xi_1$ randomly and then we select one of the remaining colours that are different from our colour at $v$ in $\xi_1$ to get $\xi_2$. According to our algorithm none
of the colours of the neighbours of \( v \) has changed. Therefore, we can move from \( \xi_2 \) to \( \xi_1 \) with the same probability as we have the same number of remaining colours to choose from randomly. Therefore, by letting \( r_1 \) be the number of remaining colours, we have

\[
p_{\xi_1, \xi_2} = \frac{1}{|V(G)| r_1} = p_{\xi_2, \xi_1}
\]

Furthermore, \( \rho_{G,q}(\xi_1) = \rho_{G,q}(\xi_2) = \frac{1}{Z_{G,q}} \) from the selection of \( \bar{\rho}_{G,q} \). Therefore,

\[
\rho_{G,q}(\xi_1) p_{\xi_1, \xi_2} = \rho_{G,q}(\xi_2) p_{\xi_2, \xi_1} = \frac{1}{Z_{G,q}} \frac{1}{|V(G)| r_1}
\]

and so \( \bar{\rho}_{G,q} \) is reversible and therefore, stationary by Theorem 4.2.1.

Since our Markov chain is aperiodic and irreducible, we have a unique stationary distribution \( \bar{\rho}_{G,q} \) by the Markov Chain Convergence Theorem.

We will now implement the above in R. Firstly, we will define two functions; namely, \( \text{get}_\text{col} \) and \( \text{rand}_\text{col} \). We will also use the previous functions: \( \text{equal}_\text{vecs} \), \( \text{initial}_\psi \) and \( \text{update}_\phi \).

\( \text{get}_\text{col} \) will be used to select a random colour from a set of remaining colours which was defined earlier.

- Function name: \( \text{get}_\text{col} \)
- Input:
  * A vertex \( v \).
  * A Markov digraph of our graph \( G \), which has the attribute “colour” defined for each vertex.
  * A set of colours called \( \text{set}_\text{col} \).
- Output: A random colour
- Procedure:
  i) Identify the unique colours of the neighbours of \( v \).
  ii) Remove the above set of colours from \( \text{set}_\text{col} \).
  iii) If there is more than one colour, randomly select one and return it. Otherwise, we return that one colour that is remaining.

The R-code is implemented as follows.

```r
get_col<-function(v,G,set_col)
{
  vec<-unique(V(G)[neighbors(G,v)]$colour)
  temp<-set_col
  for(i in 1:length(vec))
  {
    if(length(which(temp==vec[i]))!=0)
    {
      temp<-temp[-which(temp==vec[i])]
    }
  }
  if(length(temp)>1)
  {
```

115
Secondly, we define the function `randcol`.

- **Function name:** `randcol`
- **Input:**
  * The Markov digraph $D$
  * The number $q$—this is optional as a default value has been set.
- **Output:** The Markov digraph $D$ with a random $q$-colouring.
- **Procedure:**
  i) Assign all the vertices with the value 0.
  ii) Using a for–loop, run through each of the vertices in $V(D)$ once; assigning to each one a colour by using the function `getcol`.
  iii) Return $D$ with its random colouring.

The implementation of the above is as follows.

```r
# a function that gives a random q colouring to start with
randcol<-function(D_,q_col=max(degree(D_, v=V(D_),
               mode = c("out"), loops=TRUE))+2)
{
  n<-length(V(D_))
  D_<-set.vertex.attribute(D_,"colour",value=rep(0,n))
  set_col<-seq(1,q_col,by=1)
  for(i in 1:n)
  {
    colour<-get_col(V(D_)[i-1],D_,set_col)
    V(D_)[i-1]$colour<-colour
  }
  return(D_)
}
```

We now draw our attention to the function `qcoloring`, which is an adaptation of the above. Provision for recording the different configurations and their number of visitations will also be considered in the following algorithm and R–code.

- **Function name:** `qcoloring`
- **Input:**
  * A Markov digraph $D$
  * The number of colours that are being considered. This input is optional as there is already a default value which has been set.
  * The number of steps ($k$) to be considered. Once again, this input is optional.
Output: Three .csv files will be outputted:
  i) “configs_qcoloring” which contains all the colour configurations that we have encountered in our simulation.
  ii) “visited_qcoloring” which contains the number of visitations and the distribution of the visitations.
  iii) “MC_qcoloring” will contain the Markov chain that has been simulated. Each number represents a particular row which is a configuration in “configs_qcoloring”.

  Furthermore, we also return $D$ with the last configuration, the configurations that we have encountered and their distribution.

Procedure: [This is a combination of the functions randcol and GIBBS.]
  i) We initialize the matrix configs with the given random $q$-colouring. Furthermore, we label it as 1 and place a 1 in its position in the vector visited.
  ii) Form the set of colours—\setcol = \{1, 2, \ldots, q\}.
  iii) For each step $1$ to $k$, we pick a random vertex $v$ in $V(D)$. We then obtain a random colour using get_col and assign it to $v$. Next we use step 4) in the procedure from the function GIBBS.
  iv) Write the relevant variables to the appropriate files in the previous section “Output”.
  v) Return $D$ with its last configuration, the matrix of configurations and their distribution.

The R-code is presented as follows:

```r
qcoloring <- function(D_, q_col = max(degree(D_, v = V(D_), mode = c("out"), loops = TRUE))+2, k = 1000)
{
  n <- length(V(D_))
  configs <- matrix(V(D_)$colour, nrow = 1, byrow = TRUE)
  visited <- 1
  # set up a uniform initial distribution
  vec <- rep(1/n, n)
  ## use initial_psi to get a starting vertex
  temp <- NULL
  set_col <- seq(1, q_col, by = 1)
  MC <- NULL
  for (i in 1:k)
  {
    temp <- initial_psi(vec, D_)
    colour <- get_col(temp, D_, set_col)
    V(D_)[temp]$colour <- colour
    v_new <- V(D_)$colour
    if (equal_vecs(v_new, configs)[[1]] == TRUE)
    {
      pos <- equal_vecs(v_new, configs)[[2]]
      visited[pos] <- visited[pos] + 1
      MC <- rbind(MC, pos)
    }
    else
    {
      
    }
  }
}
```
configs<-rbind(configs,v_new)
visited<-cbind(visited,1)
MC<-rbind(MC,nrow(configs))
}
}
library(MASS)
write.csv(configs,file="J://configs_qcoloring.csv")
write.csv(matrix(c(visited,visited/sum(visited)),ncol=2,byrow=FALSE),
   file="J://visited_qcoloring.csv")
write.csv(MC,file="J://MC_qcoloring.csv")
return(list(D_,configs,visited/sum(visited)))
}

An example that uses the above is given next. The relevant files can be found in the folder called “randomq”.
> set.seed(200608804)
> D_<-randcol(MarkovDigraph(G))
> tkplot(as.undirected(D_),vertex.label=get.vertex.attribute(D_,"colour"))

> G1<-qcoloring(D_,k=2000)
> tkplot(as.undirected(G1[[1]]),vertex.label=get.vertex.attribute(G1[[1]],
   + "colour"))

Figure 5.1: The input colour configuration for the function qcoloring.
Systematic Sweep Gibbs Sampler

We now consider an adaptation of the above where we select the vertices systematically in the following manner.

<table>
<thead>
<tr>
<th>vertex</th>
<th>time selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>$1, n+1, 2n+1, \ldots$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$2, n+2, 2n+2, \ldots$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$v_n$</td>
<td>$n, 2n, 3n, \ldots$</td>
</tr>
</tbody>
</table>

Since we move through the vertices systematically, this Gibbs sampler is known as the systematic sweep Gibbs sampler. Apart from the above procedure of selecting vertices, the algorithm is identical to that of the function qcoloring.

Once again the set of vertices of our Markov chain is all the different colour configurations where no two adjacent vertices may have the same colour. The proof that this Markov chain is aperiodic and irreducible is the same as that of the previous section which dealt with random $q$-colourings.

- Function name: Sweep
- Input:
  * The Markov digraph $D$ with an initial random colouring.
  * The number of colours that are being considered. This input is optional as there is already a default value that has been set.
  * The number of steps to be considered. This input is also optional.
- Output: The following three .csv files will be written:
  i) “configs_Sweep” which will contain all the encountered configurations.
  ii) “visited_Sweep” which will contain a record of the visitations and their distribution.
  iii) “MC_Sweep” records the Markov chain of the configurations.
- Procedure: This procedure is similar to that of the procedure of qcoloring. The only main difference is that vertices are selected cyclically as opposed to being selected randomly.

The R-code of the above can be found in the Code section in Appendix A under the subsection “Implementation of the Systematic Sweep Gibbs Sampler”.

An example that uses this R-code is presented below. Results for this example may be found in the folder called “Sweep”.

Figure 5.2: A possible $q$-colouring of graph $G$ after 2000 steps.
5.3 The Propp–Wilson and the Sandwiching Algorithms

5.3.1 The Propp–Wilson Algorithm

The Propp–Wilson Algorithm was introduced in 1996 by Jim Propp and David Wilson. This algorithm involves the running of Markov chain simulations starting at each vertex in our Markov digraph at some point in the past. The exiting condition is coalescence of all these Markov chains to a stationary configuration—i.e. they will merge into one Markov chain.

The advantage of this algorithm is that it can automatically terminate, so we do not need to pick the number of steps needed. Furthermore, this algorithm produces a sample from the stationary distribution.

One major drawback is that this algorithm is extremely slow when our graph contains a large number of vertices as we would have to run a Markov chain simulation for each vertex and wait for coalescence. Another disadvantage occurs when we do not get coalescence at time $t = 0$; we have to go further back in time and restart our simulations at each vertex. Clearly, this algorithm is computationally intensive.

For a computer with 1 gigabytes RAM, and a dual–processor of 1.6 Ghz each, it takes more than 2 hours to terminate for a $3 \times 3$ grid. I terminated the running of this program for the above graph due to it being impractical. Nine vertices is a small number, so there has to be another reason. Recall from earlier that we are not guaranteed convergence for a connected bipartite graph. Well, a $3 \times 3$ grid is a connected bipartite graph. With further investigation it was found that the Markov chains were alternating periodically.

It is also important to take note that depending on the random numbers that are being generated (from a uniform distribution), we could still obtain convergence. Other reasons for non–termination will be explored after the details of this algorithm has been given.
The algorithm will depend on the following functions from earlier:
- `MarkovDigraph`
- `Transition_mat_dir`

Furthermore, we will implement the following new functions:
- `PWphi` which is similar to `update_phi` from earlier with the exception that it works with a vector.
- `check_equal`

We proceed with the algorithm for `PWphi`.
- Function name: `PWphi`
- Input:
  * A random number `U` from a uniform distribution on the interval `[0, 1]`.
  * The Markov digraph `G`
  * The current set of vertices that the Markov chains are at—referred to as `process` in the code.
- Output: A set of random vertices that is referred to as `result` in the code.
- Procedure:
  i) For each vertex in `process`, obtain the row vector in the transition matrix that corresponds to the vertex that we are currently considering; i.e. if we set `X_i` equal to our current vertex, then row `X_i+1` in `P` would correspond to it.
  ii) Divide the unit interval `[0, 1]` according to the row vector `X_i + 1` in `P`.
  iii) Check which interval `U` falls into and save the vertex that is associated with that particular interval in the vector `result`.
  iv) Continue the above procedure for each of the remaining elements in `process` and return the vector `result`.

One way of implementing the above algorithm in R is presented below.
```r
PWphi<-function(U,G,process)
{
  u<-U
  result<-NULL
  for(i in 1:length(process))
  {
    X_i<-process[i]
    P<-Transition_mat_dir(G)
    vec<-P[X_i+1,]
    n<-length(V(G))
    count<- 0
    lbound<- 0
    upbound<-vec[1]
    for(j in 1:length(vec))
    {
      if(u >=lbound & u < upbound )
      {
        result<-cbind(result,count)
      }
    }
  }
}
```
if(u==1)
{
    result<-cbind(result,n-1)
}

count<-count+1
lbound<-lbound+vec[j]
upbound<-upbound+vec[j+1]
}
}
return(result)

The function check_equal is used to check if a vector’s components are all the same. The algorithm that is presented is one possible way.

- Function name: check_equal
- Input: A numerical vector v
- Output: Returns true or false
- Procedure: If all the remaining components are equal to the first component of the vector v, then return true. Otherwise, return false.

In the R-code, we compare the length of the vector of components that are equal to the first component to the length of v minus 1. If this criterion is met, return true. Otherwise, return false. The R-code is given next:

check_equal<-function(v)
{
    if(length(which(v[-1]==v[1]))==length(v)-1)
    {
        return(TRUE)
    }
    return(FALSE)
}

Consider the following algorithm for Propp-Wilson.

- Function name: ProppWilson
- Input:
  
  * The Markov digraph of a graph; referred to as G in the code.
  * The maximum number of steps that we do not want to surpass. We can actually remove this input as the code will terminate automatically without it. Alternatively, if we pick a really large number, then it would not come into effect.

- Output:
  * A matrix of all the simulated Markov chains.
- Procedure:
  i) Set m = 0, MC = NULL and Times = 1.
  ii) Generate the first random number from a uniform distribution on the interval [0,1] to be used at time = −1 and call it U.
iii) From 1 to the value of $Times$, simulate a Markov chain for each vertex in $V(G)$ by using $PWphi$. Thereafter, place these in $MC$.

iv) Increase $m$ by 1 and set $Times = 2m$.

v) Generate $(2m - \text{number of elements in } U)$ random numbers from a uniform distribution between 0 and 1 and add these to $U$.

vi) Reverse the ordering of $U$ and use these in iii).

vii) Set $MC$ equal to $\text{NULL}$ again.

viii) Steps iii) - vii) are repeated until all the Markov chains are at the same state at time $t = 0$—the function $check\_equal$ is used for this purpose.

The same random numbers are used by each of the $n$ chains. Furthermore, if $U_0, U_{-1}, \ldots, U_{\text{Times}+1}$ is our sequence of random numbers from a uniform distribution, then we reuse these random numbers at the times they were used at previously.

The R-code for the above as well as a few examples are given below.

```R
ProppWilson<-function(G,N=1000)
{
    vec_initial<-V(G)
    n<-length(V(G))
    vec_start<-vec_initial
    m<-0
    MC<-NULL
    U<-runif(1,0,1)
    revU<-U
    Times<-1
    count<-0
    while(check_equal(vec_start)==FALSE & count<=N)
    {
        # positive instead of a negative as it is easier to work with
        MC<-matrix(vec_initial,nrow=n,ncol=1)
        temp<-vec_initial
        for(i in 1:Times)
        {
            MC<-cbind(MC,t(PWphi(revU[i],G,MC[,i])))
        }
        m<-m+1
        vec_start<-MC[,Times+1]
        Times<-2*m
        tempU<-runif(Times-length(U),0,1)
        U<-c(U,tempU)
        revU<-rev(U)
        count<-count+1
    }
    return(list(MC))
}
```

The construction of a digraph $G_1$ in R together with some output:
> Vec_from<-c(1,1,2,2)
> Vec_to<-c(1,2,1,2)
> d1<-data.frame(from=Vec_from,to=Vec_to)
> G1<-graph.data.frame(d1,directed=TRUE)
> G1 <- set.edge.attribute(G1, "weight", value=rep(0.5,4))
> tkplot(G1, edge.label=get.edge.attribute(G1,"label"))

![Diagram of a Markov digraph G1.](image)

**Figure 5.3:** The Markov digraph G1.

> set.seed(200608804)
> ProppWilson(G1)

```
[,1] [,2]
count 0 1
count 1 1
```

> # Using Grid-forhardcore: we are not guaranteed
> # convergence as any grid is bipartite
> # The code for drawing grids, its algorithm and some examples involving
> # grids can be found in Appendix A under the Code section.
> G<-'graph_grid(3)
Another reason for non–termination can be the choice of the update function. Consider the following Markov digraph $D$.

$$D :$$

![Figure 5.5: The Markov digraph $D$.](image)
The transition matrix of the above is
\[
P = \begin{bmatrix}
0.5 & 0.5 \\
0.5 & 0.5
\end{bmatrix}
\]

Now define the following update function \( \phi \) as follows:

\[
\phi(v_1, x) = \begin{cases} 
v_1 & \text{if } x \in [0, 0.5) \\
v_2 & \text{otherwise}
\end{cases}
\]

\[
\phi(v_2, x) = \begin{cases} 
v_2 & \text{if } x \in [0, 0.5) \\
v_1 & \text{otherwise}
\end{cases}
\]

where \( x \) is a random number from a uniform distribution between 0 and 1.

Since \( D \) has loops, it is aperiodic. Furthermore, there exists a \( v_1 \rightarrow v_2 \) directed path and a \( v_2 \rightarrow v_1 \) directed path; therefore, \( D \) is strongly-connected and so the Markov chain is irreducible. According to the Markov Chain Convergence Theorem, all distributions will tend towards a stationary distribution. Therefore, at some point in time these two chains should meet. Let this time be \( M \).

According to the Propp–Wilson algorithm, if we are at time \( N - M \), then at time \( t = 0 \) these chains are supposed to be at the same vertex.

Suppose that these two chains are at \( N - M \). Therefore, at time \( N - M + 1 \), we have two possible outcomes:

If \( x \in [0, 0.5) \), then

\[
\phi(v_1, x) = v_1 \\
\phi(v_2, x) = v_2
\]

So \( \phi(v_1, x) \neq \phi(v_2, x) \).

Similarly, if \( x \in [0.5, 1] \), then

\[
\phi(v_1, x) = v_2 \neq v_1 = \phi(v_2, x)
\]

Suppose that these chains meet at time \( t = 0 \) for the first time, and that \( \phi(u, x) = \phi(u', x) = v_1 \) when \( x \in [0, 0.5) \). Then \( u = u' = v_1 \), which contradicts our assumption that they met at time \( t = 0 \) since they have met at time \( t = -1 \). Continuing backwards in time, we will find that \( v_1 = v_2 \). Clearly this is false and so these two chains never meet.

5.3.2 Sandwiching

One possible way of increasing the speed of Propp–Wilson is to consider a technique that is known as sandwiching. Sandwiching only applies to Markov chains in which the vertices have some ordering or monotonic properties. As a result of the last statement, sandwiching does not work for all Markov chains.

In this case, we consider an update function that steps up or down at intermediate vertices. Furthermore, one vertex would have the highest value and another the lowest one—the ordering which we were referring to earlier. If we are at the vertex with the highest value, then we remain in that state instead of stepping up. Similarly, if we were to be in the lowest–valued vertex, then we would remain in it instead of stepping down.
From the above, we would define the ladder update function as follows. Suppose that our vertices are ordered in ascending order as follows:

\[ v_1, v_2, \ldots, v_n \]

Let \( x \) be a random number from a uniform distribution on \([0, 1]\). We then have

\[
\phi(v_1, x) = \begin{cases} 
  v_1 & \text{if } x \in \left[0, \frac{1}{2}\right) \\
  v_2 & \text{otherwise}
\end{cases}
\]

\[
\phi(v_n, x) = \begin{cases} 
  v_{n-1} & \text{if } x \in \left[0, \frac{1}{2}\right) \\
  v_n & \text{otherwise}
\end{cases}
\]

and for \( i = 2, \ldots, n - 1 \)

\[
\phi(v_i, x) = \begin{cases} 
  v_{i-1} & \text{if } x \in \left[0, \frac{1}{2}\right) \\
  v_{i+1} & \text{otherwise}
\end{cases}
\]

For this algorithm, we follow the procedure for the Propp–Wilson algorithm and use the ladder update function instead of the conventional update function defined earlier.

The term sandwiching refers to the fact that all intermediate Markov chains are “squeezed” between the upper and the lower Markov chains. Consequently, when the upper and the lower Markov chains combine, then all intermediate chains combine into one. It is important to note that when two Markov chains meet at some vertex \( v_i \), we will move to the same vertices from that point in time as we are using the same random numbers in all the simulated Markov chains.

The above algorithm is some kind of birth–death process, where \( v_1 \) is the lower population bound and \( v_n \) is the upper bound. The birth rate \( b_i = 0.5 \) for \( i = 1, \ldots, n - 1 \) and the death rate \( d_j = 0.5 \) for \( j = 2, \ldots, n \). Furthermore, we have two stationary rates \( s_1^* = 0.5 \) and \( s_n^* = 0.5 \). We can depict the above as follows.

\[
D:
\]

Figure 5.6: The Markov chain of the defined birth–death process.

\( D \) is a strongly–connected digraph and so our Markov chain is irreducible. \( v_1 \) and \( v_n \) have loops and so there is no periodicity in \( D \) and therefore, our Markov chain is aperiodic. So we will be able
to use the Propp–Wilson algorithm.

The algorithm for Sandwiching is the same as that of Propp–Wilson except that we will use the function `ladderPhi` in place of `PWphi`. The basic idea was mentioned recently and so we now present the relevant code in R together with a small example.

```r
### Sandwiching
library(igraph)
# An update function for the ladder walk MC --> birth and death processes
# This function can either go up or down at intermediary nodes
ladderPhi <- function(U, G, process)
{
  u <- U
  P <- Transition_mat_dir(G)
  result <- NULL
  n <- length(V(G))
  lower <- min(process)
  upper <- max(process)
  for (i in 1:length(process))
  {
    X_i <- process[i]
    vec <- P[X_i+1,]
    if (X_i == 0)
      {
        if (u < 0.5)
          {
            result <- cbind(result, 0)
          }
        else
          {
            result <- cbind(result, 1)
          }
      }
    else if (X_i == n-1)
      {
        if (u < 0.5)
          {
            result <- cbind(result, n-2)
          }
        else
          {
            result <- cbind(result, n-1)
          }
      }
    else
      {
      }
  }
}
if(u<0.5)
{
    result<-cbind(result,X_i-1)
}

else
{
    result<-cbind(result,X_i+1)
}

}
return(result)

check_equal<-function(v)
{
    if(length(which(v[-1]==v[1]))==length(v)-1)
    {
        return(TRUE)
    }
    return(FALSE)
}

Sandwiching<-function(G,N=1000)
{
    vec_initial<-V(G)
    n<-length(V(G))
    vec_start<-vec_initial
    m<-0
    MC<-NULL
    U<-runif(1,0,1)
    revU<-U
    Times<-1
    count<-0
    while(check_equal(vec_start)==FALSE & count<=N)
    {
        # positive instead of a negative as it is easier to work with
        MC<-matrix(vec_initial,nrow=n,ncol=1)
        temp<-vec_initial
        for(i in 1:Times)
        {
            MC<-cbind(MC,t(ladderPhi(revU[i],G,MC[,i])))
        }
        m<-m+1
        vec_start<-MC[,Times+1]
        Times<-2*m
        tempU<-runif(Times-length(U),0,1)
    }
}
U<-c(U,tempU)
revU<-rev(U)
count<-count+1

return(list(MC))

An outputted example.
> ## Queue birth-death process example
> Vec_from<-c(1,1,2,2,3,3,4,4,5,5,6,6)
> Vec_to<-c(1,2,1,3,2,4,3,5,4,6,5,6)
>
> d1<-data.frame(from=Vec_from,to=Vec_to)
> G1<-graph.data.frame(d1,directed=TRUE)
> G1 <- set.edge.attribute(G1, "weight", value=rep(0.5,12))
> tkplot(G1,edge.label=get.edge.attribute(G1,"label"))

Figure 5.7: The Markov digraph G1.

> set.seed(200608804)
> Sandwiching(G1)

[1,] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
[2,] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
[3,] 2 3 4 5 6 7 8 9 10 11 12 13 14 15
[4,] 3 4 5 6 7 8 9 10 11 12 13 14 15 16
[5,] 4 5 6 7 8 9 10 11 12 13 14 15 16 17
[6,] 5 6 7 8 9 10 11 12 13 14 15 16 17 18

After the page, the text continues with more code and example output.
<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6

The Ising Model and the Tutte Polynomial

6.1 A few definitions

We now define the more general Potts model and then consider its specialized case: the Ising model.

Let \( G = (V, E) \) be an undirected graph. The Potts model can be viewed as a \( q \)-colouring method where we assign the colours \( \{1, \ldots, q\} \) randomly to the vertices in \( V(G) \). From a physical point of view, each vertex represents an atom and each of these atoms can have one out of \( q \) spin configurations.

More formally, let \( \xi(x) \in \{1, \ldots, q\} \) be the spin of a vertex \( x \). A spin configuration of a graph \( G \) is a mapping \( \xi : V(G) \to \{1, \ldots, q\} \). In other words, it is all the possible colourings (both proper and non-proper) of the vertices in a graph \( G \).

Furthermore, the edges of our graph represent the bonds between the atoms in our model. Therefore, for each edge \( e \in E(G) \), we let \( J_e \) be the interaction strength. Then the energy \( \mathcal{H}(\xi) \) of a spin configuration \( \xi \in \{1, \ldots, q\}^{|V|} \) is defined as

\[
\mathcal{H}(\xi) = \sum_{e \in E(G)} -J_e \delta(\xi(v_x), \xi(v_y))
\]

where \( v_x \) and \( v_y \) are the vertices of an edge \( e \in E(G) \) and

\[
\delta(i, j) = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{otherwise}
\end{cases}
\]

From physics, the Boltzmann weight of a spin configuration \( \xi \) is given by \( \exp(-\beta \mathcal{H}(\xi)) \), where \( \beta \geq 0 \) is the inverse of the absolute temperature times the Boltzmann constant \( k_B = 1.38 \times 10^{-23} \) joules/Kelvin (see [4]).

Now let \( V(D) \) be the set of all configurations in our Markov digraph. We then define the partition function \( Z_{G, \beta}(q) \) as

\[
Z_{G, \beta}(q) = \sum_{\xi \in V(D)} \exp(-\beta \mathcal{H}(\xi))
\]  

(6.1.1)
Another way of expressing (6.1.1) is now shown:

We have

\[ Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D)} e^{\beta \mathcal{H}(\xi)} \]

and

\[ \mathcal{H}(\xi) = \sum_{e \in \mathcal{E}(G)} (-J_e) \delta(\xi(v_x), \xi(v_y)) \]

Therefore, \( Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D)} e^{\beta \sum_{e \in \mathcal{E}(G)} (-J_e) \delta(\xi(v_x), \xi(v_y))} \).

This can be written as

\[ Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D)} \prod_{e \in \mathcal{E}(G)} e^{\beta J_e \delta(\xi(v_x), \xi(v_y))} \] (6.1.2)

Furthermore, we recall that \( e^{a+b} = e^a e^b \) and simplify (6.1.2) to

\[ Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D)} \prod_{e \in \mathcal{E}(G)} e^{\beta J_e \delta(\xi(v_x), \xi(v_y))} \] (6.1.3)

We now consider an alternative way of writing \( e^{\gamma \delta(i, j)} \) by using \( \delta(i, j) \), which was defined earlier.

Case 1: \( i = j \)

We then have \( \delta(i, j) = 1 \) and so

\[
e^{\gamma \delta(i, j)} = e^{\gamma (1)} = e^{\gamma} = 1 + (e^{\gamma} - 1)
\]

For Case 2, when \( i \neq j \), we have \( \delta(i, j) = 0 \) and therefore,

\[ e^{\gamma \delta(i, j)} = e^{\gamma (0)} = e^{0} = 1 = 1 + (e^{\gamma} - 1) (0) = 1 + (e^{\gamma} - 1) \delta(i, j) \]

From the above two cases,

\[ e^{\gamma \delta(i, j)} = 1 + (e^{\gamma} - 1) \delta(i, j) \] (6.1.4)

Applying (6.1.4) to (6.1.3), we get the following

\[
Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D)} \prod_{e \in \mathcal{E}(G)} [1 + (e^{\beta J_e} - 1) \delta(\xi(v_x), \xi(v_y))] \] (6.1.5)

We will use (6.1.5) later. For now we will consider a special case of the Potts model in which we have two spin configurations; i.e. we have an upward and a downward orientation. We will denote these spins by \( \{-1, 1\} \) and refer to this model as the Ising model.
6.2 The Ising Model and its Simulations

For the sake of completeness, we now give some more background about the Ising model. (For more information, the reader may consult [32].)

6.2.1 The Ising Model

We will give a description of the Ising model and discuss two physical systems to which it can be applied: magnetism and liquid–gas transition.

The Ising model has a grid (lattice) of $N$ vertices which are coloured with the values $\{−1, +1\}$ (we will use $\{−1, 1\}$ for convenience). We will mostly be concerned with square grids. The Hamiltonian (energy) for the Ising model is

$$\mathcal{H} = - \sum_{(i,j) \in E(G)} J_{i,j} \xi(i) \xi(j) - H \sum_{i=1}^{N} \xi(i)$$

where $G$ is our square grid (an undirected graph) of $N$ vertices, $H$ is the magnetic field and $J_{i,j}$ is the coupling between the neighbours $i$ and $j$.

Magnetism

Originally, the Ising model was used to describe magnets. This is why the colour on each vertex is normally referred to as a spin. $H$ is called the external magnetic field and the sum $M = \sum_{i=1}^{N} \xi(i)$ is called the magnetization. Hence, the magnetization per spin is given by

$$m(T) = \frac{M(T)}{N}$$

where $T$ is the absolute temperature of the system.

The energy of two neighbouring spins is defined to be $-J_{i,j} \xi(i) \xi(j)$. There are two cases that can occur.

- If the spins of the vertices $i$ and $j$ are parallel ($i$ and $j$ have the same colour), then the energy is $-J_{i,j}$.
- If the spins $i$ and $j$ are antiparallel ($i$ and $j$ have different colours; one vertex is assigned the value 1 and the other $−1$), then the energy between the neighbouring spins is given by $+J_{i,j}$.

Now consider the cases when $J_{i,j}$ is either positive or negative.

- If $J_{i,j} > 0$ (which is the usual case), then the model favours parallel spins. Such an interaction is said to be ferromagnetic.
- If $J_{i,j} < 0$, then the spins tend to align like a chessboard at low temperatures—this is an antiferromagnetic phase.

At high temperatures, the sign of the energy does not matter as we expect entropy (lack of order or predictability) to occur. Thus, the spins will fluctuate considerably in a paramagnetic phase resulting in no net magnetization per spin.

From the above, there is a critical temperature $T_c$, where we have continuous phase transitions—we
move between up–spins and down–spins or zero–magnetization states. Hence, for the critical temperature, we have the magnetized phase for temperatures \( T < T_c \) and a zero–magnetization phase for \( T > T_c \).

**Liquid–Gas Transition**

The Ising model can also be used to model the liquid–gas transition. In this model, the up–spins \((\xi(i) = 1)\) represent atoms and the down–spins \((\xi(i) = -1)\) represent spaces without atoms. Therefore, the liquid phase occurs when we mostly have up–spins with a few down–spins and the gas phase occurs when we mostly have down–spins with a few up–spins (just like atoms in vapour).

In this case, the Ising model is useful in understanding the properties near the critical point. The critical point is where we move continuously between the liquid and the gas phases; i.e. there is a continuous phase transition.

**6.2.2 Mean Field Theory**

For this section, [20] and [24] were consulted. Before proceeding, we will note that an applied magnetic field \( \mathbf{H} \) will tend to align the spins, but thermal fluctuations will tend to create disorder.

**Non-interacting spins**

The potential energy of the system in the applied magnetic field \( \mathbf{H} \) is given by

\[
\mathcal{H}(s_1, s_2, \ldots, s_N) = \sum_{i=1}^{N} -s_i \mu H
\]

where \( \mu \) is the magnetic moment per unit cell; furthermore, \( \mu \) is a constant in our model. We will also be using \( s_i \) for the value of the spin configuration for vertex \( i \).

Recall the conservation of energy: "The total energy \( (E) \) is the sum of the potential energy \( (E_{pot}) \) and the kinetic energy\( (E_{kin}) \)."

In our model, we have

\[
E = E_{pot} + E_{kin}
\]

\[
= E_{pot} \quad \text{since} \ E_{kin} = 0
\]

\[
= \langle \mathcal{H} \rangle
\]

In the above, \( \mathcal{H} \) is the Hamiltonian in \( \mathcal{H}\psi = E\psi \), where \( \psi \) is the eigenvector (a function that describes the state of our system).

Now we are going to calculate the magnetization of our system in the applied magnetic field \( \mathbf{H} \). The magnetization is given by \( M = N \mu \langle s \rangle \), where \( \langle s \rangle \) denotes the average spin per unit cell in our lattice.

Firstly, we can write the partition function as the product of single particle functions. Letting \( Z = Z(H, T, N) \) be our partition function and \( Z_1 \) be the single partition function, we have

\[
Z = Z_1^N
\]
Furthermore, we have

\[ Z_1 = \sum_{s \in \pm 1} \exp \left( -\frac{\mu H s}{kT} \right) \]

\[ = \exp \left( \frac{\mu H(-1)}{kT} \right) + \exp \left( -\frac{\mu H(1)}{kT} \right) \]

\[ = \exp \left( \frac{\mu H}{kT} \right) + \exp \left( -\frac{\mu H}{kT} \right) \]

\[ = 2 \cosh \left( \frac{\mu H}{kT} \right) \]

where \( k \) is the Boltzmann constant and \( T \) is the absolute temperature.

Therefore, \( Z = \left[ 2 \cosh \left( \frac{\mu H}{kT} \right) \right]^N \).

Before proceeding, we note that \textit{ergodicity} and irreducibility are the same thing in a finite Markov chain. We will now state the Ergodicity Hypothesis and then use it to find the average energy.

**Ergodicity Hypothesis**

The time average and the ensemble average are the same when we have ergodicity.

We now consider our canonical ensemble—the system that we are working with.

Now consider the \textit{Helmholtz free energy}, \( F = E - TS \), where \( S \) is the entropy (measure of disorder) and the average energy \( \bar{E} = \sum_j P_j E_j \). From these, we have \( F = \sum_j P_j E_j - TS \).

The probabilities in the above \( \{P_j\} \) must minimize \( F \). Therefore,

\[ P_j = \frac{\exp \left( -\frac{E_j}{kT} \right)}{\sum_j \exp \left( -\frac{E_j}{kT} \right)} \]

This canonical distribution function gives the probability for the \( j^{th} \) distinguishable state in the ensemble (lattice). Furthermore, it minimizes \( F \) (the Helmholtz free energy which can also be written as \( F = -kT \ln Z \)); it is an equilibrium distribution.

Initially, we have

\[ \bar{E} = \sum_j P_j E_j = f(Z) \]

and by letting \( \beta = \frac{1}{kT} \), we have

\[ Z = Z(N, V, T) = \sum_j \exp \left( -\frac{E_j}{kT} \right) = \sum_j \exp \left( -\beta E_j \right) \]

Now consider

\[ \frac{\partial Z}{\partial \beta} = -\sum_j E_j \exp (-\beta E_j) \quad (6.2.1) \]
Substituting $Z$ in $P_j$, we have
\[ P_j = \frac{\exp\left(-\frac{E_j}{kT}\right)}{Z} \]

From the above, we have
\[ \exp\left(-\frac{E_j}{kT}\right) = ZP_j \]

Substituting the above into (6.2.1), we have
\[ \frac{\partial Z}{\partial \beta} = -\sum_j P_j E_j Z = -Z \sum_j P_j E_j = -ZE \]

Rearranging the above, we have
\[ \bar{E} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \left(\frac{1}{kT}\right)} \]
\[ = kT^2 \frac{\partial \ln Z}{\partial T} \]

Therefore,
\[ \bar{E} = kT^2 \frac{\partial \ln Z}{\partial T} \]

Recalling our calculation of $Z$, we can now say the following from the above.

\[ E \approx \bar{E} = kT^2 \frac{\partial \ln Z}{\partial T} \]
\[ = NkT^2 \left( \frac{2 \cosh \left( \frac{\mu H}{kT} \right)}{\partial T} \right) \]
\[ = NkT^2 \frac{1}{2 \cosh \left( \frac{\mu H}{kT} \right)} \left( \frac{2 \cosh \left( \frac{\mu H}{kT} \right)}{\partial T} \right) \]

137
\[ = N k T^2 \tanh \left( \frac{\mu H}{kT} \right) \frac{\partial}{\partial T} \left( \frac{\mu H}{kT} \right) \]
\[ = -N \mu H \tanh \left( \frac{\mu H}{kT} \right) \]

Therefore,
\[ E = -N \mu H \tanh \left( \frac{\mu H}{kT} \right) \]

Since \( E = -HM = -HN \langle s \rangle \), we have for the magnetization and the average spin value in our applied field \( \vec{H} \):
\[ M = N \mu \tanh \left( \frac{\mu H}{kT} \right) \]

and
\[ \langle s \rangle = \tanh \left( \frac{\mu H}{kT} \right) \]

For the graph of \( \langle s \rangle \) versus \( \frac{\mu H}{kT} \), we have:
- At large \( H = \| \vec{H} \| \) or low temperatures \( T \), spins are more aligned.
- For a zero magnetic field or for temperatures higher than the critical temperature \( T_c \), there is no net spin alignment.

**Interacting Spins**

We consider a simple model for interacting spins where only the nearest neighbours interact.

\[ \mathcal{H} = -\mu H \sum_{i=1}^{N} s_i - \frac{1}{2} \sum_{i,j=1}^{N} J_{i,j} s_i s_j \]

with
\[ J_{i,j} = \begin{cases} J & \text{nearest neighbours} \\ 0 & \text{otherwise} \end{cases} \]

The magnitude of the force exerted on the \( i^{th} \) spin due to the external magnetic field \( \vec{H} \) and the neighbouring spins is
\[ -\frac{\partial \mathcal{H}}{\partial s_i} = \mu H + \sum_{j} J_{i,j} s_j \]
Therefore, the net field acting on the \( i^{th} \) spin due to the external magnetic field \( \mathbf{H} \) and the \( z \) nearest neighbours is

\[
H_i = H + \sum_j J_{i,j} s_j / \mu \\
\approx H + \frac{J z \langle s_j \rangle}{\mu} \\
= H + \frac{J z \langle s_i \rangle}{\mu}
\]

(We have used the mean field approximation and the fact that \( \langle s_j \rangle = \langle s_i \rangle \).)

Using our earlier result (from non-interacting spins) and substituting \( H_i \) in place of \( H \), we have

\[
\langle s \rangle = \tanh \left( \frac{\mu H_i}{kT} \right) \\
= \tanh \left( \frac{\mu H + J z \langle s \rangle}{kT} \right) \quad \text{(since} \langle s_i \rangle = \langle s \rangle) \]

Thus, for the zero external field, we have

\[
\langle s \rangle = \tanh \left( \frac{J z \langle s \rangle}{kT} \right)
\]  

(6.2.2)

(This transcendental equation can be solved numerically for \( \langle s \rangle \).)

Therefore, we have obtained the average spin of a unit cell (mean field result for the net spin alignment) and the spontaneous (zero-field) magnetization \( M = N \mu \langle s \rangle \).

### 6.2.3 Criticality

Near the critical point (inflection point), \( \langle s \rangle \) is small. Therefore, consider the following Taylor series expansion of \( \langle s \rangle \) that is centered at 0—i.e. the Maclaurin series expansion.

\[
\langle s \rangle = \tanh \left( \frac{J z \langle s \rangle}{kT} \right) \\
= \frac{J z \langle s \rangle}{kT} - \frac{1}{3} \left( \frac{J z \langle s \rangle}{kT} \right)^3 + O \left( \langle s \rangle^5 \right)
\]

where \( O \left( \langle s \rangle^5 \right) \) indicates that this approximation has a \( 5^{th} \) order error.

From the above approximation, we get

\[
\langle s \rangle \left[ \frac{1}{3} \left( \frac{J z}{kT} \right)^3 \langle s \rangle^2 - \left( \frac{J z - kT}{kT} \right) \right] = 0
\]

We have the following solutions:

\[
\langle s \rangle = 0
\]
and

$$\langle s \rangle^2 = \frac{J_z - kT}{kT} \left[ 3 \left( \frac{kT}{J_z} \right)^3 \right]$$

Therefore,

$$\langle s \rangle = \pm \left( \frac{J_z - kT}{kT} \right)^{\frac{1}{2}} \left[ 3 \left( \frac{kT}{J_z} \right)^3 \right]^{\frac{1}{2}}$$

Real solutions for the above occur when $kT \leq J_z$.

We associate $J_z$ with the critical temperature $kT_c$. Moreover, the expansion around $\langle s \rangle = 0$ is only valid when $T \approx T_c$. Therefore, we can write the solutions as follows.

$$\langle s \rangle = \begin{cases} 
0 & T \geq T_c \\
\pm \sqrt{3} \left( \frac{T}{T_c} \right)^{\frac{1}{2}} \left( \frac{T_c - T}{T_c} \right)^{\frac{1}{2}} & T < T_c
\end{cases}$$

using $T \approx T_c$.

Therefore, $\langle s \rangle$ and $M$ show a power–low temperature dependence.

We will now consider a Gibbs sampler of the above where we will explore such results.

### 6.2.4 Modified Gibbs Sampler

We will be using *Matlab* [25] for this particular implementation as it contains the necessary graphical facilities. Furthermore, we will take advantage of “vectorized” operations for increased computational efficiency. We take note that the following algorithm is for square grids/lattices with four neighbours.

- **Function name:** Ising2
- **Input:**
  * $N$ is the number of vertices/sites for a side in our lattice.
  * The absolute temperature $T$.
  * $iter$ is the number of iterations that we want to consider.
  * $Tvec$ is a vector of temperature values that we want to display at.
  * $ivec$ is a vector of iteration values at which we want to display our $N \times N$ grid.
- **Output:**
  * The magnetizations per site (vertex)
  * The energy ($E$) per site
  * The number of clusters of up–spin states, $num$
- **Procedure:** The procedure can easily be seen in the following code.
%% Ising2.m modified from [37]
function [M, num, E] = Ising2(N,T,iter,Tvec,ivec)
J = 1; % Strength of interaction (Joules)
k = 1; % Joules per kelvin

%% Generate a random initial configuration
grid = (rand(N) > 0.5)*2 - 1;
%% Evolve the system for a fixed number of steps
for i=1:iter,
    % Calculate the number of neighbors of each cell
    neighbors = circshift(grid, [0 1]) + ...
               circshift(grid, [0 -1]) + ...
               circshift(grid, [1 0]) + ...
               circshift(grid, [-1 0]);
    % Calculate the change in energy of flipping a spin
    DeltaE = 2 * J * (grid .* neighbors);
    % Calculate the transition probabilities
    p_trans = exp(-DeltaE/(k * T));
    % Decide which transitions will occur
    transitions = (rand(N) < p_trans ).*(rand(N) < 0.1) * -2 + 1;
    % Perform the transitions
    grid = grid .* transitions;
    % Sum up our variables of interest
    M = sum(sum(grid));
    E = -sum(sum(DeltaE))/2;
    if ismember(i,ivec) && ismember(T,Tvec)
        % Display the current state of the system meeting the above criterion
        image((grid+1)*128);
        xlabel(sprintf('T = %0.4f, M = %0.4f, E = %0.4f ,
                        Iterations = %4d', T, M/N^2, E/N^2,i));
        set(gca,'YTickLabel',[],'XTickLabel',[]);
        axis square; colormap bone; drawnow;
        name = strcat('Temp = ', num2str(T,4), ' iteration = '
                       int2str(i),'.eps');
        %print -deps name.eps
        print(gcf, '-depsc2', name);
        pause;
    end
end

% Count the number of clusters of spin up states
[L, num] = bwlabel(grid == 1, 4);

In the above implementation, we place $N$ particles in any configuration in a square lattice. Afterwards, we move the particles in succession. We then calculate the change in the energy.
caused by this move; call it $\Delta E$. If $\Delta E$ is negative, then we have moved the system to a lower energy state (therefore more stable). Therefore, we will allow the move and so, we place the particle in its new position. Otherwise, $\Delta E$ is positive; we will then allow the move to occur with a probability of $\exp\left(-\frac{\Delta E}{kT}\right)$. For the purpose of calculating averages, we consider ourselves to be at a new configuration regardless of whether we have moved or not.

The following lines of code will be used to run the above code.

```matlab
%% ModifiedMonteLoop.m
%% Initial Configuration
n_grid = 50; % Size of grid
Ms = [];
Ts = [];
Ns = [];
Es = [];

T1 = [1, 0.95*2.5, 0.98*2.5, 2.499, 2.5, 2.01, 2.7, 4];
Tvec = [1, 0.95*2.5, 0.98*2.5, 2.499, 2.5, 2.01, 2.7, 4];
ivec = [1, 10, 100, 1000];
iter = 1000;
type Ising2.m

%% Monte Carlo Loop
for j=1:2000
    % Choose a temperature
    T = rand()*5+1e-10;
    % Perform a simulation
    [M, N, E] = Ising2(n_grid, T, iter, Tvec, ivec);
    % Record the results
    Ms = [Ms M/(n_grid^2)];
    Es = [Es E/(n_grid^2)];
    Ns = [Ns N];
    Ts = [Ts T];
end

for k=1:length(T1)
    % Choose a temperature
    T = T1(k);
    % Perform a simulation
    [M, N, E] = Ising2(n_grid, T, iter, Tvec, ivec);
    % Record the results
    Ms = [Ms M/(n_grid^2)];
    Es = [Es E/(n_grid^2)];
    Ns = [Ns N];
    Ts = [Ts T];
end
```

142
We now consider the output from running the above code.

In the above graph, we can see that there is a change in the magnetization per site (vertex) for some \( T \) between 2.2 and 2.5. This means that our critical temperature lies somewhere on this interval. Past the critical temperature, we can observe from the graph that there is no magnetization. For \( T < T_c \), we have magnetizations between \(-1\) and \(1\); this is due to the competition between thermal fluctuations and the alignment of spins with the magnetic field caused by the nearest neighbours. In general, at lower temperatures, the system favours magnetic alignment and then coalesces at \( T_c \).

We now look at the output of our model at different temperatures and iterations.
For $T = 1$ in the above outputs, we can clearly see that the clustering of the spin configurations is getting smoother with increasing iterations—it is becoming more aligned with the local magnetic field.
- For $T = 4$, the spin configuration stays random, which is what we expected as it lies in the zero–magnetization phase.
- For $T = 2.5$ and $T = 2.499$, both grids look like they have clustering; however, both grids are not very smooth.

We will now prove some results based on a given configuration in the Ising model.

### 6.2.5 Some Interesting Results

We will use the following to prove our results:

i) Our set of colours is $\{-1, 1\}$ and

ii) $H(\xi) = -\sum_{e \in E(G)} \xi(v_x)\xi(v_y)$, where $v_x$ and $v_y$ are the vertices of an edge $e$ in $E(G)$ and $\xi(v_x)$ and $\xi(v_y)$ are the colours at the respective vertices.

In other words, each edge adds 1 to the above if its vertices are assigned different colours. Furthermore, it subtracts 1 if their colours are the same.

For the Ising model, we pick a random spin configuration from $V(D)$ according to the following probability measure.

$$
\pi_{G, \beta}(\xi) = \frac{1}{Z_{G, \beta}} \exp (-\beta H(\xi))
$$

where $v_x$ and $v_y$ are the two vertices incident with $e$.

Using the above, we will now prove three useful results (which are exercises in [19]).

Firstly, let $G = (V, E)$ and $\beta$ be the value that was defined earlier. Let $v \in V(G)$ and $\xi \in \{-1, 1\}^{V \setminus \{v\}}$ be an arbitrary configuration of $-1$’s and $1$’s to $G \setminus v$. Let $V(D)$ be all the spin configurations in our Markov digraph $D$—i.e. $V(D) = \{-1, 1\}^{V \setminus \{v\}}$.

Define $\xi^+ \in V(D)$ to be the spin configuration of $G$ such that $v$ is assigned $1$ and all other vertices agree with $\xi$. Similarly, $\xi^- \in V(D)$ is defined to be the spin configuration of $G$ in which $v$ is assigned $-1$ and all other vertices of $G$ are assigned their respective spins/colours in $\xi$. Furthermore, define $k_+(v, \xi)$ to be the number of neighbours of $v$ that are assigned $1$ and $k_-(v, \xi)$ to be the number of neighbours of $v$ that are assigned $-1$.

**Lemma 6.2.1.** Let $G = (V, E)$ and $\beta$ be the inverse of the temperature times the Boltzmann constant. If $\xi^-$ and $\xi^+$ are two spin configurations that differ at a vertex $v$, then

$$
\frac{\pi_{G, \beta}(\xi^+)}{\pi_{G, \beta}(\xi^-)} = \exp (2\beta (k_+(v, \xi) - k_-(v, \xi)))
$$

**Proof.**

From $\pi_{G, \beta}(\xi) = \frac{1}{Z_{G, \beta}} \exp (\beta \sum_{e \in E} \xi(v_x)\xi(v_y))$, we have $\pi_{G, \beta}(\xi^+) = \frac{1}{Z_{G, \beta}} \exp (\beta \sum_{e \in E} \xi(v_x)\xi(v_y))$. 

145
Since addition is commutative and associative, the above can be written as

$$\pi_{G, \beta}(\xi^+) = \frac{1}{Z_{G, \beta}} \exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) + \sum_{\text{all edges incident to } v} \xi(v)\xi(v_y) \right) \right)$$

where \( v \) is fixed.

$$\pi_{G, \beta}(\xi^+) = \frac{1}{Z_{G, \beta}} \exp \left( -\beta \left( -\sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - \sum_{\text{all edges incident to } v} \xi(v)\xi(v_y) \right) \right)$$

Recall that

$$\mathcal{H}(\xi) = -\sum_{e \in E(G)} \xi(v_x)\xi(v_y)$$

and that each edge added 1 to the above if its vertices were assigned opposite spins and subtracted 1 if they were the same.

We consider

$$\mathcal{H}(\xi^+) = -\sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - \sum_{\text{all edges incident to } v} \xi(v)\xi(v_y)$$

For \( v, k_+(v, \xi) \) is the number of vertices adjacent to \( v \) that take the value of 1. Therefore, they have the same assignment as \( v \) and each of these subtract 1 from \( \mathcal{H}(\xi^+) \).

Furthermore, \( k_-(v, \xi) \) gives the number of neighbours of \( v \) that are assigned \(-1\) and so each of these adds 1 to \( \mathcal{H}(\xi^+) \).

As a result of the above, we have

$$\mathcal{H}(\xi^+) = -\sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - [k_+(v, \xi) - k_-(v, \xi)] \quad (6.2.3)$$

Placing (6.2.3) in \( \pi_{G, \beta}(\xi^+) \), we get

$$\pi_{G, \beta}(\xi^+) = \frac{1}{Z_{G, \beta}} \exp \left( -\beta \left( -\sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - [k_+(v, \xi) - k_-(v, \xi)] \right) \right)$$

$$= \frac{1}{Z_{G, \beta}} \exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) + k_+(v, \xi) - k_-(v, \xi) \right) \right) \quad \text{(6.2.4)}$$

For \( \pi_{G, \beta}(\xi^-) \), we follow the same procedure as the above except that \( v \) is assigned \(-1\) now and so the roles of \( k_+(v, \xi) \) and \( k_-(v, \xi) \) in \( \mathcal{H}(\xi^-) \) would be the opposite of \( k_+(v, \xi) \) and \( k_-(v, \xi) \) in \( \pi_{G, \beta}(\xi^+) \). Therefore,

$$\pi_{G, \beta}(\xi^-) = \frac{1}{Z_{G, \beta}} \exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - k_+(v, \xi) + k_-(v, \xi) \right) \right) \quad \text{(6.2.5)}$$
From (6.2.4) and (6.2.5), we have
\[
\frac{\pi_{G,\beta}(\xi^+)}{\pi_{G,\beta}(\xi^-)} = \frac{1}{Z_{G,\beta}} \exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) + k_+(v, \xi) - k_-(v, \xi) \right) \right)
\frac{1}{Z_{G,\beta}} \exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) - k_+(v, \xi) + k_-(v, \xi) \right) \right)
= \frac{\exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) \right) \right)}{\exp \left( \beta \left( \sum_{e \in E(G \setminus v)} \xi(v_x)\xi(v_y) \right) \right)} \exp \left( \beta \left( k_+(v, \xi) - k_-(v, \xi) \right) \right)
= \exp \left( 2\beta \left( k_+(v, \xi) - k_-(v, \xi) \right) \right)
\]

Therefore, we have our result. □

The following lemma shows us how the spin configuration \( \xi \) of \( V(G) \setminus \{v\} \) affects the assignment of \( \{-1, 1\} \) to the vertex \( v \).

**Lemma 6.2.2.** Let \( v \in V(G) \) be arbitrary and \( \xi \) be any \( \{-1, 1\} \) spin configuration of \( V(G) \setminus \{v\} \). Then
\[
\pi_{G,\beta}(X(v) = 1 \mid X(V \setminus \{v\}) = \xi) = \frac{\exp \left( 2\beta \left( k_+(x, \xi) - k_-(x, \xi) \right) \right)}{\exp \left( 2\beta \left( k_+(x, \xi) - k_-(x, \xi) \right) \right) + 1}
\]
where \( X(\cdot) \) is the spin configuration.

**Proof.** Consider the following
\[
\pi_{G,\beta}(X(v) = 1 \mid X(V \setminus \{v\}) = \xi) = \frac{\pi_{G,\beta}((X(v) = 1) \cap (X(V \setminus \{v\}) = \xi))}{\pi_{G,\beta}(X(V \setminus \{v\}) = \xi)} \tag{6.2.6}
\]
since \( P(A \mid B) = \frac{P(A \cap B)}{P(B)} \).

\( \xi^+ \) is the spin configuration in which \( X(v) = 1 \) and \( X(V \setminus \{v\}) = \xi \). Clearly, the event
\( \xi^+ = ((X(v) = 1) \cap (X(V \setminus \{v\}) = \xi)) \) and so
\[
\pi_{G,\beta}((X(v) = 1) \cap (X(V \setminus \{v\}) = \xi)) = \pi_{G,\beta}(\xi^+) \tag{6.2.7}
\]

According to the law of total probability, we may write
\[
\pi_{G,\beta}(X(V \setminus \{v\}) = \xi)
= \pi_{G,\beta}(X(V \setminus \{v\}) = \xi \mid X(x) = 1) P(X(x) = 1) + \pi_{G,\beta}(X(V \setminus \{v\}) = \xi \mid X(x) = -1) P(X(x) = -1)
= \pi_{G,\beta}((X(V \setminus \{v\}) = \xi) \cap (X(x) = 1)) + \pi_{G,\beta}((X(V \setminus \{v\}) = \xi) \cap (X(x) = -1))
\]

147
Substituting \( \pi_{G, \beta} (\xi^-) = \pi_{G, \beta} ((X (V \setminus \{v\}) = \xi) \cap (X(x) = -1)) \) and (6.2.7) into the above, we get

\[
\pi_{G, \beta} (X (V \setminus \{v\}) = \xi) = \pi_{G, \beta} (\xi^+) + \pi_{G, \beta} (\xi^-) \tag{6.2.8}
\]

From Lemma 6.2.1, we have

\[
\frac{\pi_{G, \beta} (\xi^+)}{\pi_{G, \beta} (\xi^-)} = \exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right)
\]

From the above, we have

\[
\pi_{G, \beta} (\xi^-) = \frac{\pi_{G, \beta} (\xi^+)}{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right)} \tag{6.2.9}
\]

Using (6.2.9), we rewrite (6.2.8)

\[
\pi_{G, \beta} (X (V \setminus \{v\}) = \xi) = \frac{\pi_{G, \beta} (\xi^+)}{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right)} \left[1 + \frac{1}{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right)} \right]
\]

Substituting the above and (6.2.7) into (6.2.6), we have

\[
\pi_{G, \beta} (X(v) = 1 \mid X (V \setminus \{v\}) = \xi) = \frac{\pi_{G, \beta} (\xi^+)}{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right) + 1} \tag{6.2.10}
\]

With some elimination and simplification, (6.2.10) yields the result

\[
\pi_{G, \beta} (X(v) = 1 \mid X (V \setminus \{v\}) = \xi) = \frac{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right)}{\exp \left(2\beta (k_+(v, \xi) - k_-(v, \xi)) \right) + 1} \tag{6.2.11}
\]

The next lemma deals with an ordering on the spin configurations.

**Lemma 6.2.3.** For any two configurations \( \xi, \eta \in \{-1, 1\}^{V \setminus \{v\}} \), let us define the following ordering relation. Let \( \xi \preceq \eta \) if \( \xi(u) \leq \eta(u) \) for all \( u \in V \setminus \{v\} \). If \( \xi \preceq \eta \), then

\[
\pi_{G, \beta} (X(v) = 1 \mid X (V \setminus \{v\}) = \xi) \preceq \pi_{G, \beta} (X(v) = 1 \mid X (V \setminus \{v\}) = \eta)
\]

**Proof.** By assumption, \( \xi \preceq \eta \) implies that \( \xi(u) \leq \eta(u) \) for all \( u \in V \setminus \{v\} \). Therefore, \( \eta \) has at least \( k_+(v, \xi) \) vertices with the value 1. In other words,

\[
k_+(v, \xi) \leq k_+(v, \eta) \tag{6.2.12}
\]
Furthermore, \( k_-(v, \xi) \geq k_-(v, \eta) \) and so

\[
-k_-(v, \xi) \leq -k_-(v, \eta)
\]  

(6.2.13)

Using (6.2.12) and (6.2.13), we have

\[
k_+(v, \xi) - k_-(v, \xi) \leq k_+(v, \eta) - k_-(v, \xi) \\
\leq k_+(v, \eta) - k_-(v, \eta)
\]  

(6.2.14)

For \( \beta > 0 \), we have

\[
2\beta \left[ k_+(v, \xi) - k_-(v, \xi) \right] \leq 2\beta \left[ k_+(v, \eta) - k_-(v, \eta) \right]
\]

and therefore,

\[
\exp(2\beta \left[ k_+(v, \xi) - k_-(v, \xi) \right]) \leq \exp(2\beta \left[ k_+(v, \eta) - k_-(v, \eta) \right])
\]  

(6.2.15)

Before we continue further, we need to prove the following auxiliary result.

Let \( a, b \in \mathbb{R}^+ \). If \( a \leq b \), then

\[
\frac{a}{1+a} \leq \frac{b}{1+b}
\]

Proof. Let \( a \leq b \), then \( 1 + a \leq 1 + b \). Taking the inverse, we get

\[
\frac{1}{1+a} \geq \frac{1}{1+b}
\]

Now consider

\[
\frac{a}{1+a} = \frac{a+1-1}{1+a} = 1 - \frac{1}{1+a}
\]

Since \( \frac{1}{1+a} \geq \frac{1}{1+b} \), we have

\[
\frac{1}{1+a} \leq -\frac{1}{1+b}
\]

Adding 1 to both sides, we get

\[
\frac{a}{1+a} = 1 - \frac{1}{1+a} \leq 1 - \frac{1}{1+b} = \frac{b}{1+b}
\]

Therefore, we have our auxiliary result.

Letting \( a = \exp(2\beta \left[ k_+(v, \xi) - k_-(v, \xi) \right]) \) and \( b = \exp(2\beta \left[ k_+(v, \eta) - k_-(v, \eta) \right]) \), we get

\[
\frac{\exp(2\beta \left[ k_+(v, \xi) - k_-(v, \xi) \right])}{1 + \exp(2\beta \left[ k_+(v, \xi) - k_-(v, \xi) \right])} \leq \frac{\exp(2\beta \left[ k_+(v, \eta) - k_-(v, \eta) \right])}{1 + \exp(2\beta \left[ k_+(v, \eta) - k_-(v, \eta) \right])}
\]

Identifying the above by using (6.2.11), we get

\[
\pi_{G,\beta} (X(v) = 1 \mid X(V \setminus \{v\}) = \xi) \\
\leq \pi_{G,\beta} (X(v) = 1 \mid X(V \setminus \{v\}) = \eta)
\]

In the next section, we draw our attention to some relations between the partition function of the Potts model and a special case of Tutte's polynomial.
6.3 The partition function of the Potts model

Recall (6.1.5) from earlier,

\[ Z_{G, \beta}(q) = \sum_{\xi \in \mathcal{V}(D) : e \in E(G)} \prod_{e \in E(G)} \left[ 1 + (\exp(\beta J_e) - 1) \delta(\xi(v_x), \xi(v_y)) \right] \]

was one of the ways of writing the partition function for a \( q \)-state Potts model. We now consider the following:

Define \( w_e = \exp(\beta J_e) - 1 \). We will see what happens when all the \( w_e \)'s are \(-1\); in other words, \( \exp(\beta J_e) = 0 \). Clearly, this only occurs when \( \beta J_e \to -\infty \). Therefore, the temperature \( T \to 0 \).

Before we progress with the above, we need to define a few concepts.

The Tutte polynomial of a multigraph \( G = (V, E) \) is a polynomial of two variables that is recursively defined (see [38]) as follows:

\[
T(G; x, y) = \begin{cases} 
1 & \text{if } E(G) = \emptyset \\
xT(G/e; x, y) & \text{if } e \in E \text{ and } e \text{ is a bridge} \\
yT(G - e; x, y) & \text{if } e \in E \text{ and } e \text{ is a loop} \\
T(G - e; x, y) + T(G/e; x, y) & \text{otherwise}
\end{cases}
\]

We can find the Tutte polynomial of any graph \( G \) by considering the cases of edge deletion and edge contraction of \( G \) (see chapter 1). Furthermore, \( K_2 \) is defined to be \( x \) and a single loop is defined to be \( y \). We now present an example in which the Tutte polynomial of a graph \( G \) is calculated.

**Example 6.3.1.** Suppose that \( G = K_4 \):

\[ G = K_4: \]

\[ \text{deletion} \quad \text{+} \quad \text{contraction} \]

\[ x(\square) + y(\bigcirc) \]

**Figure 6.1:** A graphical scheme of working out the Tutte polynomial of \( G \).

To save some time and space, we identify three graphs and work with these separately.
$G_1$:

$$
\begin{array}{c}
\begin{array}{c}
\text{x} (\quad ) \\
\quad +
\end{array}
\end{array}
\end{array}
$$

$$x^2 + (x + y)$$

Figure 6.2: Computing the Tutte polynomial of $G_1$.

$G_2$:

$$
\begin{array}{c}
\begin{array}{c}
\text{y} (\quad )
\end{array}
\end{array}
\end{array}
$$

$$y (x + y)$$

Figure 6.3: Computing the Tutte polynomial of $G_2$. 

151
Using our three polynomials $T(G_1; x, y)$, $T(G_2; x, y)$ and $T(G_3; x, y)$, we calculate the Tutte polynomial of $G$.

$$T(G; x, y) = [xT(G_1; x, y) + (T(G_1; x, y) + T(G_2; x, y))] +$$
$$[(T(G_1; x, y) + T(G_2; x, y)) + yT(G_3; x, y)]
$$
$$= xT(G_1; x, y) + 2(T(G_1; x, y) + T(G_2; x, y)) + yT(G_3; x, y)
$$
$$= x[x^2 + (x + y)] + 2[x^2 + (x + y) + xy + y^2] +
$$
$$y[x + y + y^2]
$$
$$= x^3 + x^2 + xy + 2x^2 + 2x + 2y + 2xy + 2y^2 + xy + y^2 + y^3
$$
$$= x^3 + 3x^2 + 4xy + 2x + 2y + 3y^2 + y^3$$

Another way of defining the Tutte polynomial is as follows. Let $G = (V, E)$ be an undirected graph, $A$ be a subset of $E$, and $k(A)$ be defined to be the number of connected components in the graph $(V, A)$. Define the Tutte polynomial to be

$$T(G; x, y) = \sum_{A \subseteq E} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}$$

We will now prove that this definition is equivalent to the recursive definition of the Tutte polynomial (a part of the proof can be found in [7]).

Theorem 6.3.2. Let $G = (V, E)$. Then

$$T(G; x, y) = \sum_{A \subseteq E} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}$$

152
satisfies the following:

\[
T(G; x, y) = \begin{cases} 
1 & \text{if } E(G) = \emptyset \\
xT(G/e; x, y) & \text{if } e \in E \text{ and } e \text{ is a bridge} \\
yT(G - e; x, y) & \text{if } e \in E \text{ and } e \text{ is a loop} \\
T(G - e; x, y) + T(G/e; x, y) & \text{otherwise}
\end{cases}
\]

Proof.

Proof i): Let \( G = (V, E) \), where \( E = E(G) = \emptyset \). We want to prove that
\[
T(G; x, y) = 1
\]

Since \( E = \emptyset \), there is only one \( A \subseteq E \) to consider; namely, \( A = E \). Furthermore, the number of connected components \( k(E) = |V| = n \). Hence,

\[
T(G; x, y) = \sum_{A \subseteq E} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|}
\]

\[
= (x - 1)^{k(E) - k(E)}(y - 1)^{k(E) + |E| - n}
\]

\[
= (x - 1)^0(y - 1)^n + 0^n - n
\]

\[
= (1)(y - 1)^0
\]

\[
= 1
\]

Proof ii): Let \( G = (V, E) \). Furthermore, let \( e \) be a bridge in \( E = E(G) \). We want to prove that
\[
T(G; x, y) = xT(G/e; x, y)
\]

Consider

\[
xT(G/e; x, y) = x \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) - k(E(G/e))}(y - 1)^{k(A_1) + |A_1| - |V(G/e)|}
\]

By contracting the bridge \( e \in E(G) \), we do not change the number of connected components in \( G \). Hence, \( k(E(G/e)) = k(E(G)) = k(E) \). By contracting an edge in a graph, we reduce the number of vertices by 1. Hence, \( |V(G/e)| = |V(G)| - 1 = |V| - 1 \). Therefore,

\[
xT(G/e; x, y) = x \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) - k(E)}(y - 1)^{k(A_1) + |A_1| - |V| + 1}
\]

We now use the fact that \( x = ((x - 1) + 1) \) in the above to obtain

\[
xT(G/e; x, y) = ((x - 1) + 1) \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) - k(E)}(y - 1)^{k(A_1) + |A_1| - |V| + 1}
\]
Using the right-distributivity of the multiplication of real numbers with respect to addition, we have

\[
x T(G/e; x, y) = (x - 1) \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1)} - k(E)(y - 1)^{k(A_1) + |A_1| - |V| + 1} + \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) - k(E)}(y - 1)^{k(A_1) + |A_1| - |V| + 1} = \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) + 1 - k(E)}(y - 1)^{k(A_1) + 1 + |A_1| - |V| + 1} + \sum_{A_1 \subseteq E(G/e)} (x - 1)^{k(A_1) - k(E)}(y - 1)^{k(A_1) + (|A_1| + 1) - |V|}
\]

We note that the number of subsets such that \( A_1 \subseteq E(G/e) \) is equal to the number of subsets such that \( A \subseteq E(G) \) and \( e \in A \). Since \( e \in A \), we have \( |A| = |A_1| + 1 \). Furthermore, by including \( e \) again, we have not altered the number of connected components; so, \( k(A) = k(A_1) \). Hence, we now have

\[
x T(G/e; x, y) = \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A) + 1 - k(E)}(y - 1)^{k(A) + 1 + |A| - 1 - |V|} + \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|} = \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|} + \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A - e) - k(E)}(y - 1)^{k(A - e) + |A - e| - 1 - |V|} + \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|}
\]

By letting \( A' = A - \{e\} \), the above becomes

\[
x T(G/e; x, y) = \sum_{A' \subseteq E(G), e \notin A'} (x - 1)^{k(A') - k(E)}(y - 1)^{k(A') + |A'| - |V|} + \sum_{A \subseteq E(G), e \in A} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|} = \sum_{A \subseteq E(G)} (x - 1)^{k(A) - k(E)}(y - 1)^{k(A) + |A| - |V|} = T(G; x, y)
\]
Proof iii): Let $G = (V, E)$. Furthermore, let $e$ be a loop in $E = E(G)$. We want to prove that

$$T(G; x, y) = yT(G - e; x, y)$$

Consider

$$yT(G - e; x, y) = y \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E(G - e))} (y - 1)^{k(E) + |A_1| - |V(G - e)|}$$

Since $e$ is a loop, the removal of $e$ from $G$ will not affect the number of connected components and so $k(E(G - e)) = k(E)$. Furthermore, no vertices have been removed and so, $V(G - e) = V(G) = V$. Hence, the above can be written as

$$yT(G - e; x, y) = y \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V|}$$

We now use the fact that $y = ((y - 1) + 1)$ in the above to obtain

$$yT(G - e; x, y) = ((y - 1) + 1) \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V|}$$

Using the right-distributivity of the multiplication of real numbers with respect to addition, we have

$$yT(G - e; x, y) = (y - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V|}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{k(A_1) - k(E)} (y - 1)^{k(A_1) + |A_1| - |V| + 1}$$
We note that \( A_1 \subseteq E(G - e) \) is equivalent to \( A \subseteq E \) and \( e \notin A \). Therefore,

\[
yT(G - e; x, y) = \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|} +
\]

\[
\sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|} - \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}
\]

\[
= \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|} - \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}
\]

Since \( e \) is a loop, we have \( k(A) = k(A \cup \{e\}) \). Therefore,

\[
yT(G - e; x, y) = \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A \cup \{e\})}(y - 1)^{k(A \cup \{e\}) + |A \cup \{e\}| - |V|} +
\]

\[
\sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}
\]

By letting \( A' = A \cup \{e\} \), we can change the index of the first summation from \( A \subseteq E, e \notin A \) to \( A' \subseteq E, e \in A' \). Hence,

\[
yT(G - e; x, y) = \sum_{A' \subseteq E, e \in A'} (x - 1)^{k(A')} (y - 1)^{k(A') + |A'| - |V|} +
\]

\[
\sum_{A' \subseteq E, e \in A'} (x - 1)^{k(A')} (y - 1)^{k(A') + |A'| - |V|} - \sum_{A' \subseteq E} (x - 1)^{k(A')} (y - 1)^{k(A') + |A'| - |V|}
\]

\[
= \sum_{A' \subseteq E} (x - 1)^{k(A')} (y - 1)^{k(A') + |A'| - |V|}
\]

**Proof iv):** Let \( G = (V, E) \) and \( e \in E \) such that \( e \) is neither a bridge nor a loop. We want to prove that

\[
T(G; x, y) = T(G - e; x, y) + T(G/e; x, y)
\]

Let \( A \subseteq E \). Then we can write the following from definition.

\[
T(G; x, y) = \sum_{A \subseteq E} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}
\]

Consider \( e \in E \) such that \( e \) is not a bridge, then we can rewrite the above as

\[
T(G; x, y) = \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|} +
\]

\[
\sum_{A \subseteq E, e \notin A} (x - 1)^{k(A)}(y - 1)^{k(A) + |A| - |V|}
\]

\[ (6.3.1) \]

156
Consider $G - e$. We have $|V(G - e)| = |V|$. Then we have

$$T(G - e; x, y) = \sum_{A_1 \subseteq E(G - e)} (x - 1)^{|A_1| - k(E(G - e))} (y - 1)^{k(A_1) + |A_1| - |V(G - e)|}$$

$$= \sum_{A_1 \subseteq E(G - e)} (x - 1)^{|A_1| - k(E)} (y - 1)^{k(A_1) + |A_1| - |V|} \quad (6.3.2)$$

since $k(E) = k(E(G - e))$ as the number of connected components have not changed by removing a non-bridge $e$. Furthermore, $A_1 \subseteq E(G - e)$ is equivalent to the event where $A \subseteq E$ and $e \notin A$. Using this in (6.3.2), we get

$$T(G - e; x, y) = \sum_{A \subseteq E, e \notin A} (x - 1)^{|A| - k(E)} (y - 1)^{k(A) + |A| - |V|} \quad (6.3.3)$$

Now consider $G/e$. Clearly, through edge contraction, we lose a vertex in $V$ and so $|V(G/e)| = |V| - 1$. Let $A \subseteq E$ such that $e \in A$ and let $A_2 = A - e$. We have that $|A_2| = |A - e| = |A| - 1$. Since $e$ is not a bridge, $k(A_2) = k(A)$. Therefore,

$$T(G/e; x, y) = \sum_{A_2 \subseteq E(G/e)} (x - 1)^{|A_2| - k(E(G/e))} (y - 1)^{k(A_2) + |A_2| - |V(G/e)|}$$

$$= \sum_{A_2 \subseteq E(G/e)} (x - 1)^{|A_2| - k(E)} (y - 1)^{k(A_2) + |A_2| - |V|} + 1 \quad (6.3.4)$$

$$= \sum_{A \subseteq E, e \notin A} (x - 1)^{k(A) - k(E)} (y - 1)^{k(A) + |A| - |V|}$$

Next, we use (6.3.4) and (6.3.3) in (6.3.1) to get our result

$$T(G; x, y) = T(G - e; x, y) + T(G/e; x, y)$$

From the above four proofs, we have our result.

\[\square\]

We now draw our attention to a special case of the Tutte polynomial; namely, the chromatic polynomial. We will prove that this polynomial is the Tutte polynomial evaluated at $y = 0$.

The chromatic polynomial of a graph $G$ counts the number of vertex colourings in the defined graph. To agree with the notation of the Tutte polynomial, we will use $P(G; \lambda)$ to denote the chromatic polynomial for $\lambda$ colourings. More formally, the chromatic polynomial of a graph $G$, $P(G; \lambda)$, is the number of proper $\lambda$-colourings of $V(G)$.

As in the case of the Tutte polynomial, the chromatic polynomial can be computed recursively by using edge contraction. In fact, the chromatic polynomial satisfies the following recurrence relation. Let $G = (V, E)$, $uv \in E$ and $\lambda$ be the number of colours. Then

$$P(G; \lambda) = P(G - uv; \lambda) - P(G/uv; \lambda)$$

Alternatively, if $uv \notin E$, then we can define it the opposite way around.

$$P(G; \lambda) = P(G + uv; \lambda) + P(G/uv; \lambda)$$

157
Two lemmas relating to the \( P(G; \lambda) \) are given next—see [7].

**Lemma 6.3.3.** Let \( G = (V, E) \) where \( n = |V| \) and \( \lambda^{(i)} \) denote \( P^\lambda_i = \frac{\lambda!}{(\lambda - i)!} \). If \( b_i \) is the number of ways of partitioning \( V \) into \( i \) independent sets, then \( P(G; \lambda) = \sum_{i=1}^{n} b_i \lambda^{(i)} \).

**Proof.** Suppose that we have an \( i \)-colouring of the graph \( G \). We can partition the vertex set \( V(G) \) into \( i \) independent sets according to the colours that were assigned to them. A partition of \( V \) from the above can be coloured in \( P^\lambda_i \) ways. Since we can have at most \( n \) colours, we must sum over all the cases. Therefore, we get

\[
P(G; \lambda) = \sum_{i=1}^{n} b_i \lambda^{(i)}
\]

where \( b_i \) is the number of ways of partitioning \( V(G) \) into \( i \) independent sets. \( \square \)

Before we proceed with the next lemma, we remind the reader of the Inclusion–Exclusion principle (see [36]). Firstly, we will give the simplest form and then the more general one with the aid of induction.

For any finite sets \( A \) and \( B \)

\[
|A \cup B| = |A| + |B| - |A \cap B|
\]

Considering three finite sets, we have the following:

\[
|A \cup B \cup C| = |(A \cup B) \cup C| = |A \cup B| + |C| - (A \cup B) \cap C
\]
\[
= |A| + |B| - |A \cap B| + |C| - (A \cap C) \cup (B \cap C)
\]
\[
= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| - |A \cap C \cap (B \cap C)|
\]
\[
= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|
\]

Assuming that the above is true for \( n - 1 \) finite sets, we have

\[
\left| \bigcup_{i=1}^{n-1} A_i \right| = \sum_{i=1}^{n-1} |A_i| - \sum_{j<k \leq n-1} |A_j \cap A_k| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right|
\]

Now we consider

\[
\left| \bigcup_{i=1}^{n} A_i \right| = \left| A_n \cup \left( \bigcup_{i=1}^{n-1} A_i \right) \right|
\]
\[
= |A_n| + \left| \bigcup_{i=1}^{n-1} A_i \right| - |A_n \cap \left( \bigcup_{i=1}^{n-1} A_i \right) |
\]

158
Lemma 6.3.4. We consider a more refined proof of the following lemma (which is only sketched in [7]).

For a more detailed expansion of the above, refer to “Proof of the Inclusion–Exclusion Principle" in Appendix A.) By the principle of Mathematical Induction, we have our result.

We consider a more refined proof of the following lemma (which is only sketched in [7]).

**Lemma 6.3.4.** Let $G = (V, E)$ with $|V| = n$ and $|E| = m$. Then $P(G; \lambda) = \sum_{i=1}^{m} a_i \lambda^i$ where $a_i = \sum_{r=0}^{i} (-1)^r N(i, r)$ and $N(i, r)$ is the number of spanning subgraphs of $G$ with $i$ components and $r$ edges.
Proof. Define $f : V(G) \to \{1, 2, \ldots, \lambda\}$ to be a $\lambda$-colouring of $G$. Furthermore, let $G_\lambda$ be the family of $\lambda$-colourings for $G$. Let $E(G) = \{e_1, \ldots, e_m\}$ and $q_i$ be the property of an edge $e_i = uv$ such that $f(u) = f(v)$. Since the chromatic polynomial is for a colouring in which no two adjacent vertices may have the same colour, $P(G; \lambda)$ does not satisfy any of the $q_i$'s. Let $N(q_{i_1}, q_{i_2}, \ldots, q_{i_r})$ be the number of $\lambda$-colourings that satisfy these properties—i.e. non-proper $\lambda$-colourings.

Let $H$ be any subgraph of $G$ induced by the edges that satisfy the properties of the $q_i$’s—i.e. $E(H) = \{e_{i_1}, e_{i_2}, \ldots, e_{i_r}\}$—with $p$ components. Hence, all vertices in a particular component will be coloured with the same colour. Since we have $\lambda$ colours for each component of $H$, there are $\lambda^p$ ways so that each of the vertices for an edge have the same colour. Summing over all the number of components and, letting $N(p, r)$ be the number of spanning subgraphs of $G$ with $p$ components and $r$ edges, we get

$$
\sum_{1 \leq i_1 \leq i_2 \leq \ldots \leq i_r \leq m} N(q_{i_1}, q_{i_2}, \ldots, q_{i_r}) = \sum_{p=1}^{n} N(p, r)\lambda^p
$$

We consider our $A_i$’s in the Inclusion–Exclusion Principle to be the events that the edges satisfy the properties of $q_{i_1}, \ldots, q_{i_r}$—i.e. the $A_i$’s are all the non-proper $\lambda$-colourings of $G$.

To get the chromatic polynomial, we must subtract the cardinality of the union of the above (over all possible values for $r$: $1 \leq r \leq m$) from the total number of colourings, $\lambda^n$. That is, we consider the complement of the Inclusion–Exclusion Principle. Therefore,

$$
P(G; \lambda) = \lambda^n - \left[ \sum_{p=1}^{n} (N(p, 1)\lambda^p) - \sum_{p=1}^{n} (N(p, 2)\lambda^p) + \ldots + (-1)^{m-1}\sum_{p=1}^{n} (N(p, m)\lambda^p) \right]
$$

(6.3.7)

Now consider $N(p, r)$ when $r = 0$. Clearly, there is only one empty graph (all the vertices have no edges) and so we have $n$ components. Hence, $\sum_{p=1}^{n} N(p, 0) = N(n, 0) = 1$.

Using the above in (6.3.7), we have

$$
P(G; \lambda) = (-1)^0\sum_{p=1}^{n} (N(p, 0)\lambda^p) + (-1)^1\sum_{p=1}^{n} (N(p, 1)\lambda^p) + (-1)^2\sum_{p=1}^{n} (N(p, 2)\lambda^p) + \ldots + (-1)^m\sum_{p=1}^{n} (N(p, m)\lambda^p)
$$

$$
= \sum_{r=0}^{m} (-1)^r \left( \sum_{p=1}^{n} N(p, r)\lambda^p \right)
$$

$$
= \sum_{p=1}^{n} \left( \sum_{r=0}^{m} (-1)^r N(p, r)\lambda^p \right)
$$

We now prove that the chromatic polynomial is a special case of the Tutte polynomial—the steps that were omitted in [7] are in the proof of the lemma that follows below.
Lemma 6.3.5. Let $G = (V, E)$. Then for $y = 0$,

$$P(G; \lambda) = (-1)^{|V| - k(E)} \chi^k(E) T(G; 1 - \lambda, 0)$$

Proof. Let us start with the following expression.

$$(-1)^{|V| - k(E)} \chi^k(E) T(G; 1 - \lambda, 0) \quad (6.3.8)$$

Using the following definition for the Tutte polynomial

$$T(G; x, y) = \sum_{A \subseteq E} (x - 1)^{k(A)} (y - 1)^{|A| - |V|}$$

we get

$$(-1)^{|V| - k(E)} \chi^k(E) \sum_{A \subseteq E} (1 - \lambda - 1)^{k(A) - k(E)} (0 - 1)^{k(A) + |A| - |V|}$$

$$= (-1)^{|V| - k(E)} \chi^k(E) \sum_{A \subseteq E} (-\lambda)^{k(A) - k(E)} (-1)^{k(A) + |A| - |V|}$$

$$= \sum_{A \subseteq E} \chi^k(A) (1 - 1)^{2(k(A) - k(E)) + |A|}$$

$$= \sum_{A \subseteq E} \chi^k(A) ((-1)^2)^{(k(A) - k(E))} (-1)^{|A|}$$

$$= \sum_{A \subseteq E} \chi^k(A) (-1)^{|A|}$$

Letting $N(i, r)$ be the number of spanning subgraphs of $G$ with $i$ components and $r$ edges, we sum over all possible values of $i$ and $r$ and so the above equation becomes

$$\sum_{i=1}^{|V|} \left( \sum_{r=1}^{|E|} (-1)^r N(i, r) \right) \lambda^i \quad (6.3.9)$$

This is because $1 \leq |A| \leq |E|$ and $1 \leq k(A) \leq |V|$ (we can have at least one connected component and at most $n$ connected components).

Recall from the previous lemma, that (6.3.10) is the chromatic polynomial. Therefore,

$$P(G; \lambda) = \sum_{i=1}^{|V|} \left( \sum_{r=1}^{|E|} (-1)^r N(i, r) \right) \lambda^i$$

$$= (-1)^{|V| - k(E)} \chi^k(E) T(G; 1 - \lambda, 0) \quad \square$$

We will now continue with our discussion regarding the partition function of the $q$–state Potts model.

Let

$$Z_{G, \beta} (q, \{w_e\}) = \sum_{\xi \in V(D)} \prod_{e \in E(G)} [1 + w_e \delta(\xi(v_x), \xi(v_y))] \quad (6.3.11)$$
where \( w_\varepsilon = \exp(\beta J_\varepsilon) - 1 \).

If we now consider \( T \to 0 \), we get \( w_\varepsilon \to -1 \). Substituting \( w_\varepsilon = -1 \) into (6.3.11), we have

\[
Z_{G, \beta}(q, \{-1\}) = \sum_{\xi \in V(D)} \prod_{e \in E(G)} [1 - \delta(\xi(v_x), \xi(v_y))]
\]

(6.3.12)

We can now make the following observations from (6.3.12).

i) If there are at least two adjacent vertices, say \( v_x \) and \( v_y \), in a configuration \( \xi \in V(D) \) such that \( \xi(v_x) = \xi(v_y) \), then the product in the summation will be equal to 0 for the configuration \( \xi \).

In fact, a non–proper colouring is a colour configuration in which at least two adjacent vertices have the same colour.

ii) Furthermore, if \( \xi(v_x) \neq \xi(v_y) \) for all adjacent vertices \( v_x \) and \( v_y \) in \( G \), then we will add 1 to the above summation. This kind of colour configuration in which no two adjacent vertices have the same colour is actually a proper \( q \)-colouring of the graph \( G \).

From i) and ii), we can observe that (6.3.12) is counting the number of proper \( q \)-colourings of the graph \( G \). Hence,

\[
Z_{G, \beta}(q, \{-1\}) = P(G; q)
\]

Therefore, from Lemma 6.3.5, we have

\[
Z_{G, \beta}(q, \{-1\}) = (-1)^{|V| - |E|} q^{|E|} T(G; 1 - q, 0)
\]
Concluding Remarks

One of the main objectives for this dissertation was to define the concepts and theory of Markov chains in a more Graph Theoretical manner and to compare some of these concepts to the more traditional definitions. Notably, we used the concept of reachability to define terms such as transient state, recurrent state, reducible and irreducible Markov chains, communication and periodicity/aperiodicty in a Markov chain. For some of the other concepts and theorems, we reminded or pointed out to the reader some important definitions from Graph Theory in the preliminary chapter (chapter 1). We used these together with the Markov digraphs as our main source of classifying the Markov chains.

We also used these redefined concepts to investigate stationarity, reversibility, mean return times, etc. Furthermore, we also completely demonstrated the Markov chain convergence Theorem. We used all the redefined theory and concepts to prove that undirected graphs are also Markov chains. In particular, we showed that all the previously defined concepts extended to undirected graphs.

Using all the above theory and the computer language R, we studied and implemented several Markov Chain Monte Carlo (MCMC) methods. We showed that the algorithms themselves were Markov digraphs/chains. From this we proceeded to study the relation between the partition function of the Potts model and a special case of the Tutte polynomial. We introduced the necessary definitions and results from Statistical Mechanics that were important to our investigation of the Ising model, which can also be seen as a special graph colouring.

In fact, by using block variables, we could have attempted to derive the partition function from the Tutte polynomial—however, this was not necessary for our discussion. It is also important to note that there are many other types of Markov Chain Monte Carlo methods: some use the concept of acceptance and others are designed to be more computationally efficient.
Appendix A

Additional Information

A.1 Proofs in Chapter 1

**Theorem 1.1.2. The First Theorem of Digraph Theory**

If $D$ is a digraph of order $n$ and size $m$ with the vertex set $V(D) = \{v_1, v_2, v_3, \ldots, v_n\}$, then

$$\sum_{i=1}^{n} d^+ v_i = \sum_{i=1}^{n} d^- v_i = m \quad (A.1.1)$$

**Proof.** Consider the first Theorem of Graph Theory, also known as the Handshake Lemma for undirected graphs. This theorem is stated as follows:

If $G$ is a graph of order $n$ and size $m$, where $V(D) = \{v_1, v_2, v_3, \ldots, v_n\}$, then

$$\sum_{i=1}^{n} \deg_G v_i = 2m$$

If we consider the digraph $D$ and remove all the arrowheads, we then have an undirected graph $G$, where the above theorem applies. Since there were $m$ arcs in $D$, we have $m$ edges in $G$.

Consider the following analogy:

If we have $m$ people giving handshakes, then there must be $m$ people to receive them. Letting the people be the vertices, the outdegree of $v$ be a handshake being given and the indegree of $v$ be a handshake being received, we have

$$\sum_{i=1}^{n} d^+ v_i = \sum_{i=1}^{n} d^- v_i = b \text{ where } b \in \mathbb{N}_0$$

From the Handshake Lemma,

$$\sum_{i=1}^{n} d^+ v_i + \sum_{i=1}^{n} d^- v_i = \sum_{i=1}^{n} \deg_G v_i = 2m$$

$$\Rightarrow b + b = 2m$$

$$\Rightarrow b = m$$

$$\Rightarrow \sum_{i=1}^{n} d^+ v_i = \sum_{i=1}^{n} d^- v_i = m$$
A.1.1 Other Theorems concerning Digraphs

Theorem 1.2.7.
Let $D$ be a digraph and $u$ and $v$ be two distinct vertices such that $u, v \in V(D)$. If $D$ has a closed (directed) $u$-$u$ walk, then $W$ contains a cycle $C$ through $u$ such that $E(C) \subseteq E(W)$.

Proof.
Case 1: Consider a closed (directed) $u$-$u$ walk $W$. If no other vertex other than $u$ is repeated and $u$ occurs twice in $W$, then $W$ is a cycle $C$ with $E(C) = E(W)$ and the result follows.

Case 2: Consider a closed (directed) $u$-$u$ walk $W$, in which vertices in $W$ may be repeated multiple times—i.e. they may occur twice or more times.
Let $i$ and $j$ be distinct positive integers where $i < j$ and $u_i = u_j$.
If the terms $u_i, u_{i+1}, \ldots, u_{j-1}$ are deleted from $W$, a (directed) $u$-$u$ walk $W_1$ with fewer terms is obtained. If no other vertices other than $u$ are repeated and $u$ is repeated twice, then the statement is proven. Otherwise, continue the above procedure on $W_1$ until a directed cycle $C$ through $u$ is obtained.

Theorem 1.2.8.
Let $D$ be a digraph and $u$ and $v$ be two distinct vertices such that $u, v \in V(D)$. If $D$ has a (directed) $u$-$v$ walk, then $D$ contains a (directed) $u$-$v$ path.

Proof. The proof of this theorem follows identically to the case of undirected graphs on page 17 in [6].

The first part is proven now.
Consider the case when $W$ is closed; i.e. $u = v$. From Theorem 1.2.7, we find a cycle through $u$.
Simply by removing the necessary vertices that includes $u$, we would be left with a (directed) $u$-$v$ path.
The case for an open (directed) $u$-$v$ walk in $D$ is given in [6].

A.2 Chapter 2

The Law of Total Probability

The following is an important part of probability theory; namely, “The Law of Total Probability”. For the proof of this law, refer to [28] pg 18.

Let $\Omega$ be the entire state space. Let $B_1, B_2, \ldots, B_k$ be such that $\bigcup_{i=1}^{k} B_i = \Omega$ and $B_i \cap B_j = \emptyset$ for $i \neq j$ with $P(B_i) > 0$ for all $i$.
[Note: We consider a state space and partition it into mutually disjoint sets.]
Then for any event $A$, we define

$$P(A) = \sum_{i=1}^{k} P(A \mid B_i) P(B_i)$$
Graphically, the definition can be displayed as follows.

Let the rectangle be our state space, Ω.

Figure A.1: State space, Ω.

Now we partition Ω into mutually disjoint sets $B_1, B_2, \ldots, B_k$.

Figure A.2: A partitioning of the state space, Ω.

Consider an event $A$, which is a set in our space Ω.

Figure A.3: A graphical representation of event $A$ in the state space, Ω.

To get $P(A)$, we can simply consider the intersection of each of the sets $B_1, B_2, \ldots, B_k$ with $A$ since $B_1, B_2, \ldots, B_k$ are mutually disjoint.
Using the result, \( P(A \cap B) = P(A \mid B) P(B) \), we have for each \( i = 1, \ldots, k \):
\[
P(A \cap B_i) = P(A \mid B_i) P(B_i)
\]

From the above result, we have
\[
P(A) = \sum_{i=1}^{k} P(A \cap B_i)
= \sum_{i=1}^{k} P(A \mid B_i) P(B_i)
\]

[Note: This is not the complete proof of the law of total probability.]

A.3 Chapter 6

Proof of the Inclusion–Exclusion Principle

For any finite sets \( A \) and \( B \)
\[
|A \cup B| = |A| + |B| - |A \cap B|
\]

Considering three finite sets, we have the following:
\[
|A \cup B \cup C| = |(A \cup B) \cup C| = |A \cup B| + |C| - |(A \cup B) \cap C|
= |A| + |B| - |A \cap B| + |C| - |(A \cap C) \cup (B \cap C)|
= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| - |(A \cap C) \cap (B \cap C)|
= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|
\]

Assuming that the above is true for \( n - 1 \) finite sets, we have
\[
\left| \bigcup_{i=1}^{n-1} A_i \right| = \sum_{i=1}^{n-1} |A_i| - \sum_{j<i<n} |A_i \cap A_j| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right|
\]

Now we consider
\[
\left| \bigcup_{i=1}^{n} A_i \right| = \left| A_n \cup \left( \bigcup_{i=1}^{n-1} A_i \right) \right|
= |A_n| + \left| \bigcup_{i=1}^{n-1} A_i \right| - \left| A_n \cap \left( \bigcup_{i=1}^{n-1} A_i \right) \right|
\]

(A.3.1)

\[
\left| \bigcup_{i=1}^{n} A_i \right| = |A_n| + \sum_{i=1}^{n-1} |A_i| - \sum_{j<i<n} |A_i \cap A_j| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right| - \left| A_n \cap \left( \bigcup_{i=1}^{n-1} A_i \right) \right|
\]
We have
\[
\left| A_n \cap \left( \bigcup_{i=1}^{n-1} A_i \right) \right| \\
= \left| \bigcup_{i=1}^{n-1} (A_n \cap A_i) \right| \\
= \sum_{i=1}^{n-1} |A_n \cap A_i| - \sum_{j<i<n} |A_n \cap A_i \cap A_j| + \ldots + (-1)^n |A_n \cap A_1 \cap A_2 \cap \ldots \cap A_{n-1}| \\
= \sum_{i=1}^{n-1} |A_n \cap A_i| - \sum_{j<i<n} |A_n \cap A_i \cap A_j| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n} A_i \right|
\]
(A.3.2)

Substituting the above in (A.3.1), we get
\[
\left| \bigcup_{i=1}^{n} A_i \right| \\
= |A_n| + \sum_{i=1}^{n-1} |A_i| - \sum_{j<i<n} |A_i \cap A_j| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right| \\
- \left[ \sum_{i=1}^{n-1} |A_n \cap A_i| - \sum_{j<i<n} |A_n \cap A_i \cap A_j| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right| \right] \\
= |A_n| + \sum_{i=1}^{n-1} |A_i| - \sum_{j<i<n} |A_i \cap A_j| + \ldots + (-1)^{2k+1} \sum_{i_1 < i_2 < \ldots < i_{2k+1} < n} |A_i \cap A_{i_1} \cap \ldots \cap A_{i_{2k+1}}| + \\
(-1)^{2k+2} \sum_{i_1 < i_2 < \ldots < i_{2k+1} < n} |A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_{2k+1}}| + \ldots + (-1)^n \left| \bigcap_{i=1}^{n-1} A_i \right| \\
+ \sum_{i=1}^{n-1} |A_n \cap A_i| + \sum_{j<i<n} |A_n \cap A_i \cap A_j| + \ldots + \\
(-1)^{2k+1} \sum_{i_1 < i_2 < \ldots < i_{2k} < n} |A_i \cap A_{i_1} \cap \ldots \cap A_{i_{2k-1}}| + \\
(-1)^{2k+2} \sum_{i_1 < i_2 < \ldots < i_{2k} < n} |A_n \cap A_{i_1} \cap \ldots \cap A_{i_{2k}}| + \ldots + (-1)^{n+1} \left| \bigcap_{i=1}^{n} A_i \right|
\]
\[ = \left( |A_n| + \sum_{i=1}^{n-1} |A_i| \right) - \left( \sum_{j<i \leq n} |A_i \cap A_j| + \sum_{i=1}^{n-1} |A_n \cap A_i| \right) + \ldots + \]

\[ (-1)^{2k+1} \left( \sum_{i_1<i_2<\ldots<i_{2k}<n} |A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_{2k}}| + \sum_{i_1<i_2<\ldots<i_{2k-1}<n} |A_n \cap A_{i_1} \cap \ldots \cap A_{i_{2k-1}}| \right) + \]

\[ (-1)^{2k+2} \left( \sum_{i_1<i_2<\ldots<i_{2k+1}<n} |A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_{2k+1}}| + \sum_{i_1<i_2<\ldots<i_{2k}<n} |A_n \cap A_{i_1} \cap \ldots \cap A_{i_{2k}}| \right) + \]

\[ \ldots + (-1)^{n+1} \left| \bigcap_{i=1}^{n} A_i \right| \]

By the principle of Mathematical Induction, we have our result.

A.4 Code

A.4.1 \( a \times a \) Grids

Some examples for \( a \times a \) grids will now be explored. The first example will be running the \texttt{GIBBS} function from earlier and the second one will involve the use of \texttt{qcoloring}. Since grids are not in the \texttt{igraph} package, we will have to construct such graphs ourselves. A discussion on this matter will now follow.

- Function name: \texttt{graph.grid}
- Input: An integer \( a \geq 1 \)
- Output: An \( a \times a \) grid with a “matrix–like” labelling of the vertices.
- Procedure:
  i) Form two vectors with \( 2a^2 - 2a \) elements and initialize the vector \texttt{VerLabel} as \texttt{NULL}—no elements.
  ii) For each \( i \) and \( j \) from 0 to \( a - 1 \) do the following:
      1) Form the labels \((i,j), ((i+1)mod a, j)\) and \((i, (j+1)mod a)\).
      2) We need to consider four cases:
          a) The current vertex we are dealing with is not located along the sides of the grid where \( i = a - 1 \) and \( j = a - 1 \).
          b) The current vertex is located in the top right–corner where it would be labelled as \((a - 1, a - 1)\).
          c) We are at the position where \( j = a - 1 \) and \( i < a - 1 \).
          d) We are at the position where \( i = a - 1 \) and \( j < a - 1 \).
Graphically, a)–d) may be depicted as follows.

Figure A.4: A graphical representation of an $a \times a$ grid

For a) we have some $i < a - 1$ and $j < a - 1$.

Figure A.5: A graphical depiction of case a)

Since we have started from vertex $(0, 0)$, we would already have the labels $(i - 1, j)$ and $(i, j - 1)$ for $v_1$ and $v_2$. Therefore, we only have to label the remaining three vertices $v_3$, $v_4$ and $v_5$ as $(i, j)$, $(i, j + 1)$ and $(i + 1, j)$ respectively.

We then add $(i, j)$ twice to $\text{Vert}_\text{from}$. Furthermore, $(i + 1) \mod a, j)$ and $(i, (j + 1) \mod a)$ will be added to $\text{Vert}_\text{to}$.

For c), we add $(i, j)$ to $\text{Vert}_\text{from}$ and $((i + 1) \mod a, j)$ to $\text{Vert}_\text{to}$.

For d), we add $(i, j)$ to $\text{Vert}_\text{from}$ and $(i, (j + 1) \mod a)$ to $\text{Vert}_\text{to}$.

170
iii) We then use `graph.data.frame` and construct the undirected graph by using the vectors `Vert.from` and `Vert.to`.

The R-code of the above together with two examples are presented. The relevant files can be found in the folder called “GRID”.

```r
library(igraph)
# works with MCMC directed
graph_grid<-function(a=2)
{
    Vec_from<-rep("",a*(2*a-2))
    Vec_to<-rep("",a*(2*a-2))
    k<-1
    VerLabel<-NULL
    for(i in 0:(a-1))
    {
        for(j in 0:(a-1))
        {
            vert1<-paste("\((",i",",j,"\)\)","
            vert2<-paste("\((",(i+1)%%a",",j,"\)\)","
            vert3<-paste("\((",i",",(j+1)%%a,"\)\)","
            VerLabel<-cbind(VerLabel,vert1)
            if(i!=(a-1) & j!=(a-1))
            {
                Vec_from[k:(k+1)]<-rep(vert1,2)
                Vec_to[k]<-vert2
                Vec_to[k+1]<-vert3
                k<-k+2
            }
            else if(i!=(a-1) & j==(a-1))
            {
                Vec_from[k]<-vert1
                Vec_to[k]<-vert2
                k<-k+1
            }
            else if(i==(a-1) & j!=(a-1))
            {
                Vec_from[k]<-vert1
                Vec_to[k]<-vert3
                k<-k+1
            }
            else
            {
                k<-k+1
            }
        }
    }
}
```

171
d<-data.frame(from=Vec_from,to=Vec_to,Labels=Vec_from)
G<-graph.data.frame(d,directed=FALSE)
G <- set.vertex.attribute(G, "label", value=VerLabel)
tkplot(G)
return(G)

Plotting a 3 × 3 grid in R:
> G<-graph_grid(3)

> D1<-MarkovDigraph(G)
> set.seed(200608804)
> D1<-rand_feasible(D1)
> G1<-GIBBS(D1,10000)
> V(G1[[1]])$colour
[1] 0 0 1 0 1 0 1 0 0
> tkplot(as.undirected(G1[[1]]), vertex.label=get.vertex.attribute(G1[[1]],"colour"))

Figure A.6: A 3 × 3 grid.
Figure A.7: A random feasible “2-colouring” of the previous graph.

> set.seed(200608804)
> D_ <- randcol(D1)
> tkplot(as.undirected(D_),
       vertex.label=get.vertex.attribute(D_,"colour"))

Figure A.8: A random initial $q$-colouring of the $3 \times 3$ grid.

> G2 <- qcoloring(D_,k=10000)
> tkplot(as.undirected(G2[[1]]),
       vertex.label=get.vertex.attribute(G2[[1]],"colour"))
Figure A.9: The last colour configuration on the graph after 10000 steps of \texttt{qcoloring} have been applied.

A.4.2 Implementation of the Systematic Sweep Gibbs Sampler

\begin{verbatim}
Sweep<-function(D_, q_col=max(degree(D_, v=V(D_), mode = c("out"), loops=TRUE))+2,k=3*length(V(D_)))
{
  n<-length(V(D_))
  set_col<-seq(1,q_col,by=1)
  configs<-matrix(V(D_)$colour,nrow=1,byrow=TRUE)
  visited<-1
  MC<-NULL
  for(i in 1:k)
  {
    colour<-get_col(i%%n,D_,set_col)
    V(D_)[i%%n]$colour<-colour
    v_new<-V(D_)$colour
    if(equal_vecs(v_new,configs)[[1]]==TRUE)
    {
      pos<-equal_vecs(v_new,configs)[[2]]
      visited[pos]<-visited[pos]+1
      MC<-rbind(MC,pos)
    }
    else
    {
      configs<-rbind(configs,v_new)
      visited<-cbind(visited,1)
      MC<-rbind(MC,nrow(configs))
    }
  }
}
\end{verbatim}
library(MASS)
write.csv(configs, file="J:/configs_Sweep.csv")
write.csv(matrix(c(visited, visited/sum(visited)), ncol=2, byrow=FALSE),
           file="J:/visited_Sweep.csv")
write.csv(MC, file="J:/MC_Sweep.csv")
return(list(D_, configs, visited/sum(visited)))
Bibliography


