

# The Methodology of Flowgraph Models



Yu Ren

Department of Statistics

London School of Economics and Political Science

A thesis submitted for the degree of

*Doctor of Philosophy*

8 December 2011



## Acknowledgements

First and foremost I would like to express my deepest gratitude to my supervisor, Professor Henry Wynn, for his warm encouragement and inspiring guidance throughout my Ph.D study. His expertise in statistics and enthusiasm for research have motivated me to continue my research, even during tough times in the Ph.D pursuit. It has been an honor to be his Ph.D student, and this thesis would not have been possible without his help and support.

My sincere appreciation is extended to my second supervisor Dr. Wicher Bergsma for his detailed and constructive comments on my thesis. I am also grateful to the Department of Statistics on providing me the research scholarship that made my Ph.D study possible at LSE. Amongst my fellow students, I would like to thank Noha Youssef, Hongbiao Zhao, Sujin Park, and Roy Rosemarin for the helpful discussions.

Lastly, and most importantly, I wish to thank my beloved parents, Gezheng Ren and Ruoxing Gao. It is to them that I dedicate this work.

## Abstract

Flowgraph models are directed graph models for describing the dynamic changes in a stochastic process. They are one class of multi-state models that are applied to analyse time-to-event data. The main motivation of the flowgraph models is to determine the distribution of the total waiting times until an event of interest occurs in a stochastic process that progresses through various states. This thesis applies the methodology of flowgraph models to the study of Markov and Semi-Markov processes.

The underlying approach of the thesis is that the access to the moment generating function (MGF) and cumulant generating function (CGF), provided by Mason's rule enables us to use the Method of Moments (MM) which depends on moments and cumulant. We give a new derivation of the Mason's rule to compute the total waiting MGF based on the internode transition matrix of a flowgraph. Next, we demonstrate methods to determine and approximate the distribution of total waiting time based on the inversion of the MGF, including an alternative approach using the Padé approximation of the MGF, which always yields a closed form density.

For parameter estimation, we extend the Expectation-Maximization (EM) algorithm to estimate parameters in the mixture of negative weight exponential density. Our second contribution is to develop a bias correction method in the Method of Moments (BCMM). By investigating methods for tail area approximation, we propose a new way to estimate the total waiting time density function and survival

function by showing how computation can be simplified when the traditional saddlepoint approximation is constructed based on the Padé approximation of the MGF. A bias correction method for this Padé-type saddlepoint approximation is also presented. For application, we consolidate the Method of Moments and develop our own MATLAB package called MMF to provide an interactive tool to find the total waiting time MGF, simulate flowgraph data, and incorporate the MM into large flowgraph models.

# Glossary of acronyms

BCMM	Bias corrected method of moments
CDF	Cumulative distribution function
CGF	Cumulant generating function
EM	Expectation and Maximisation
MGF	Moment generating function
MM	Method of moments
MFMM	Methodology of Flowgraph Models
MLE	Maximum likelihood estimator
MMF	Method of moments in Flowgraph
MSE	Mean square error
ME	Maximum entropy
PDF	Probability density function
SP	Saddlepoint approximation

## Notation

$p_{ij}$	The probability of transition from node $i$ to node $j$ .
$m_{ij}(s)$	The MGF of waiting time in node $i$ before arriving at adjacent node $j$ .
$q_{ij}(s)$	The transmittance for the branch connects node $i$ to node $j$ : $p_{ij}m_{ij}(s)$ .
$Q$	The branch transmittance matrix of non-closed flowgraph in Chapter 2.
$\tilde{Q}$	The branch transmittance matrix of closed flowgraph in Chapter 2.
$A$	The coefficient matrix of non-closed flowgraph in Chapter 2: $A = I - Q^T$ .
$\tilde{A}$	The coefficient matrix of closed flowgraph in Chapter 2: $\tilde{A} = I - \tilde{Q}^T$ .
$M_{1n}(s)$	The MGF of total waiting time between input node 1 and output node $n$ .
$S(t)$	The survival function $S(t) = P(T > t)$ .

# Contents

<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Mason's rule</b>	<b>7</b>
2.1 Review of the Flowgraph models . . . . .	7
2.1.1 Introduction . . . . .	7
2.1.2 A probabilistic approach to compute the MGF . . . . .	8
2.1.2.1 MGF for the series flowgraph . . . . .	8
2.1.2.2 MGF for the parallel flowgraph . . . . .	9
2.1.2.3 MGF for the flowgraph with feedback loops . . . . .	12
2.2 A flowgraph approach to compute the MGF . . . . .	14
2.2.1 Example 1 . . . . .	16
2.2.2 Example 2 . . . . .	18
2.3 A new derivation of the Mason's rule . . . . .	20
2.3.1 Example 1.1 . . . . .	35
2.3.2 Example 2.1 . . . . .	36
<b>3 Distribution theory</b>	<b>39</b>
3.1 Exact inversion of the MGF . . . . .	40
3.1.1 Example 1.3: Exponential waiting time . . . . .	41
3.2 Numerical approximation method . . . . .	44
3.2.1 The Maximum Entropy method . . . . .	44

3.2.1.1	Newton-Raphson method . . . . .	47
3.2.1.2	Implementation . . . . .	49
3.2.1.3	Example 3 . . . . .	51
3.2.1.4	Example 1.4: Non-exponential waiting time . . . . .	54
3.2.2	Padé approximation . . . . .	56
3.2.2.1	Example 4 . . . . .	58
3.2.3	The Saddlepoint approximation: a discussion . . . . .	60
3.2.3.1	Approximation error of the Saddlepoint approximation . . . . .	76
3.2.4	The Lugannani-Rice formula for survival function approximation . . . . .	77
3.3	Comparison of methods . . . . .	85
3.3.1	Example 1.5 . . . . .	85
3.3.2	Example 1.6 . . . . .	89
<b>4</b>	<b>Parameter estimation</b>	<b>93</b>
4.1	Maximum Likelihood Estimation . . . . .	94
4.2	The Expectation-Maximisation algorithm . . . . .	95
4.2.1	Case 1: Mixture with positive weights . . . . .	98
4.2.1.1	Example 5.1 . . . . .	104
4.2.1.2	Example 5.2 . . . . .	108
4.2.2	Case 2: Mixture with negative weights . . . . .	109
4.2.2.1	Example 1.3 continued . . . . .	114
4.2.2.2	Simulation . . . . .	117
4.3	Method of Moments . . . . .	120
4.3.1	Example 5.1 continued . . . . .	121
4.3.2	Example 5.2 continued . . . . .	123
4.3.3	Feedback loop flowgraph . . . . .	126
4.3.3.1	Example 6.1.1: Exponential waiting time case . . . . .	126
4.3.3.2	Example 6.1.2: Non-exponential waiting time case . . . . .	129
4.3.4	Example 1.4 continued . . . . .	133
4.3.4.1	Example 1.4.1: Exponential waiting time case . . . . .	133
4.3.4.2	Example 1.4.2: Non-exponential waiting time case . . . . .	135



4.3.5	Identifiability . . . . .	137
4.4	Bias correction in the Method of Moments . . . . .	138
4.5	Comparison of MLE and MM . . . . .	148
4.5.1	Example 7: Series network with exponential waiting time .	148
4.5.2	Computational time . . . . .	154
<b>5</b>	<b>Tail area probability approximations</b>	<b>155</b>
5.1	Tail area approximation by exponential function . . . . .	156
5.1.1	Example 1.3.1 . . . . .	158
5.1.2	Example 1.3.2 . . . . .	167
5.2	Inversion of Padé approximations . . . . .	173
5.2.1	Example 8 . . . . .	178
5.3	Saddlepoint approximation for survival function . . . . .	182
5.3.1	Example 1.3.3 . . . . .	183
5.4	Padé-type Saddlepoint approximation for survival function . . . .	186
5.5	Bias correction for the Padé-type Saddlepoint approximation . . .	190
5.5.1	Probability density function estimation . . . . .	196
5.5.2	Survival function approximation . . . . .	202
<b>6</b>	<b>Application</b>	<b>207</b>
6.1	Overview . . . . .	208
6.2	Implementation in MATLAB . . . . .	210
6.2.1	Function to compute the branch transition matrix . . . . .	210
6.2.2	Function to compute the MGF . . . . .	213
6.2.3	Sampling Methods . . . . .	214
6.2.4	Method to compute the MM estimator . . . . .	216
6.3	Example 9 . . . . .	218
	<b>Appendix</b>	<b>221</b>
	<b>Bibliography</b>	<b>241</b>

## *CONTENTS*

---

# List of Figures

1.1	Flowgraph model for modelling the total waiting time until the first occurrence of 3 consecutive heads. The $p_{ij}$ is the transition probability and $m_{ij}(s)$ is the MGF of waiting time distribution between node $i$ and node $j$ . . . . .	3
2.1	Series flowgraph model for a power generating system . . . . .	8
2.2	Parallel flowgraph model for cancer progression . . . . .	9
2.3	Single feedback loop flowgraph model . . . . .	12
2.4	An illustrated flowgraph example . . . . .	15
2.5	Flowgraph model for a reversible illness-death system . . . . .	17
2.6	A complex flowgraph . . . . .	18
2.7	Flowgraph with two input nodes . . . . .	20
2.8	A closed flowgraph with one feedback loop . . . . .	21
2.9	Solved flowgraph for the reversible illness-death system . . . . .	23
2.10	Flowgraph model for a closed network system . . . . .	24
2.11	Closed flowgraph model for a reversible illness-death system . . . . .	35
3.1	Density estimation by Maximum Entropy method. . . . .	53
3.2	Solid line: True density; Dash line: Estimated density by Maximum Entropy with 6 moment constraints. . . . .	54
3.3	Density estimation by Maximum Entropy method under 6 moment constraints in Example 1.4. . . . .	55
3.4	Rank check for matrix H . . . . .	59
3.5	Density Estimation for $T_{02}$ in Example 1.5 . . . . .	87
3.6	Survival estimation for $T_{02}$ in Example 1.5 . . . . .	87

*LIST OF FIGURES*

---

3.7	Plot $\log(f(t))$ against $t$ in Example 1.5 . . . . .	88
3.8	Plot $\log(P(T > t))$ against $t$ in Example 1.5 . . . . .	88
3.9	Estimated PDF of $T_{02}(t)$ in Example 1.6 . . . . .	91
3.10	Estimated survival function of $T_{02}$ in Example 1.6 . . . . .	91
3.11	Plot $\log(f(t))$ against $t$ in Example 1.6 . . . . .	92
3.12	Plot $\log(P(T > t))$ against $t$ in Example 1.6 . . . . .	92
4.1	Series flowgraph model for hydraulic pump system . . . . .	149
5.1	Plot of $-\log(P(T > t))$ against $t \in [0, 19.21]$ . . . . .	163
5.2	Plot of $-\log(P(T > t))$ against $t \in [0, 1.86]$ . . . . .	164
5.3	Plot of $-\log(P(T > t))$ against $t \in [19, 19.21]$ . . . . .	164
5.4	Plot of $-\log(P(T > t))$ against $t \in [29.99, 30]$ . . . . .	165
5.5	Plot of hazard function estimation against $t \in [0, 19.21]$ . . . . .	165
5.6	Plot of hazard function estimation against $t \in [0, 1.86]$ . . . . .	166
5.7	Plot of hazard function estimation against $t \in [19, 19.21]$ . . . . .	166
5.8	Plot of hazard function estimation against $t \in [29.99, 30]$ . . . . .	167
5.9	Plot of the function $G(t) = e^{at}f(t)$ against $t$ . . . . .	172
5.10	Plot of $-\log(f(t))$ and $-\log(0.1953) + 0.2t$ against $t$ . . . . .	172
5.11	Plot of $-\log(P(T > t))$ and $-\log(0.9765) + 0.2t$ against $t$ . . . . .	173
5.12	Plot of relative error in $P(T > t)$ against $t \in [0, 20]$ . . . . .	176
5.13	Plot relative error in $P(T > t)$ against $t \in [14, 20]$ . . . . .	177
5.14	Estimated error based on $PA_{[1,2]}(s)$ . . . . .	179
5.15	Estimated error based on $PA_{[1,3]}(s)$ . . . . .	180
5.16	Estimated error based on $PA_{[2,3]}(s)$ . . . . .	180
5.17	Estimated error based on $PA_{[2,4]}(s)$ . . . . .	181
5.18	Estimated error based on $PA_{[3,4]}(s)$ . . . . .	181
5.19	Survival function approximation by Lugannani-Rice formula . . . . .	185
5.20	Relative error in estimating $P(T > t)$ by Lugannani-Rice formula . . . . .	185
5.21	Plot of $-\log\hat{P}((T > t))$ against $t \in [0, 20]$ . . . . .	187
5.22	Plot of $-\log\hat{P}((T > t))$ against $t \in [1.86, 1.861]$ . . . . .	187
5.23	Plot of $-\log\hat{P}((T > t))$ against $t \in [19.21, 19.22]$ . . . . .	188
5.24	Comparison of relative error in estimating $SP_{M(s)}$ . . . . .	189
5.25	Plot of convergence test . . . . .	198

5.26	A comparison of error in SP approximation 1 . . . . .	199
5.27	A comparison of error in SP approximation 2 . . . . .	199
5.28	SP approximation for probability density function 1 . . . . .	200
5.29	SP approximation for probability density function 2 . . . . .	200
5.30	Plot of convergence test . . . . .	204
5.31	A comparison of error in tail area approximation 1 . . . . .	204
5.32	A comparison of error in tail area approximation 2 . . . . .	205
5.33	SP approximation for tail area probability 1 . . . . .	205
5.34	SP approximation for tail area probability 2 . . . . .	206
6.1	Illustration of MMF package . . . . .	208
6.2	Illustration of tranm.m function . . . . .	211
6.3	Menu for the choice of internode distribution . . . . .	212
6.4	Output of tranm.m function . . . . .	212
6.5	Illustration of inverse transform sampling . . . . .	215
6.6	Implementation of MMF package in Matlab . . . . .	219

*LIST OF FIGURES*

---

# Chapter 1

## Introduction

Multistate models are used to describe time-to-event data that result from a stochastic process. They model stochastic processes that progress through various states, and they are commonly applied to describe the events (i.e. the transitions between states) for an individual, which only occupies one of a few possible states at any time. Here, the analysis focuses on modeling the total waiting time between two states of interest for a single individual. Multistate models have a wide application in demography, economics, operations research, sociology, insurance and finance. The entire area of queuing theory is based on multistate models, starting with Johanssen (1907) as cited, for example, by Kendall (1951). The work of Fix and Neyman (1951) is one of the earliest uses of a multistate stochastic model in medical statistics. Hougaard (1999) presents a comprehensive review paper of the multistate models, and Hougaard (2000) gives more detail on the application of multistate models to handle multivariate survival data. An introduction to event history analysis via multistate models is given by Andersen and Keiding (2002). For an application to the statistical modelling and analysis of network data, see Kelly, Zachary and Ziedins (1996).

In finance and insurance, the Markov multistate models first appeared in Hoem (1969), and the first applications of semi-Markov multistate models to actuarial problems can be found in Hoem (1972). Pitacco (1995) illustrates how the multistate Markov and semi-Markov models can be used for the actuarial modeling of health insurance policies. The monograph of Haberman and Pitacco (1999)

gives a detail review of the application of multistate models in disability insurance and long-term care insurance. More recently, Sen (2008) proposes a multi-state Vasicek model for credit risk analysis, where he shows that the correlation between default and recovery can be modelled efficiently by allowing multiple loss states in the Vasicek framework. Norberg (2008) discusses the methodology of modern life insurance mathematics in the framework of a multistate model for life-history analysis. Dickson, Waters and Hardy (2009) give a modern perspective on life contingencies in terms of the multistate model. Spierdijk and Koning (2011) apply a multistate mixed proportional hazards approach to estimate a sufficient loss reserve for insurance companies.

In a stochastic system context, a flowgraph consists of nodes representing the outcomes or system states, where nodes are connected by directed line segments called branches that give the direction of state transition. Each branch is assigned a transition probability of taking this branch and waiting time distribution. The flowgraph theory was originally developed in Mason (1953, 1956) to solve systems of linear equations for finding the transfer function of signal flowgraph in electrical engineering. The work of Mason is later adapted for the computation of MGF by Sittler (1956), Huggins (1957), Lorens (1964), Pritsker and Happ (1966), Whitehouse (1970), and Butler (2000). A comprehensive presentations of the flowgraph model and its application to the analysis of time-to-event data is given in the book of Huzurbazar (2005).

Stochastic flowgraph models can be considered as a tool to analyse stochastic process via a network approach. For example, in medical survival analysis, the development of patient's illness can be considered as a process that progresses through several stages, where stage 1 is labeled as the diagnosis of disease, stage 2 is the advanced stage of the disease, and stage 3 represents the event of patient's death. A parallel flowgraph consists of 3 nodes can be applied for modeling the survival time of patients, who can either die directly from other causes (i.e. move directly from stage 1 to stage 3) or die from the advanced stage of disease (i.e. move to stage 2 before reaching stage 3). In an engineering reliability problem of 2 pump systems, the stages of the system begin with the functioning state 0



when both pumps are working properly, then proceed to the partially functioning state 1 after one pump failed and eventually move to the maintenance states 2 when both pumps failed. In this case, a series flowgraph can be applied to model the distribution of total waiting time from functioning state to maintenance state.

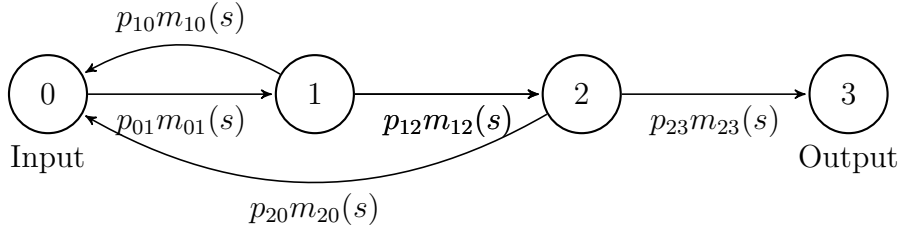


Figure 1.1: Flowgraph model for modelling the total waiting time until the first occurrence of 3 consecutive heads. The  $p_{ij}$  is the transition probability and  $m_{ij}(s)$  is the MGF of waiting time distribution between node  $i$  and node  $j$ .

The objective of flowgraph analysis is to determine the distribution of total waiting time between two nodes of interest based on the inversion of MGF. This is the starting point for computing the probability density function (PDF), cumulative distribution function (CDF), survival and hazard function of total waiting time. For illustration, let us consider a simple coin tossing experiment, where we are interested in the total waiting time to obtain three consecutive heads. Our system resulting from a sequence of independent coin tosses and the state of the system is the current number of consecutive heads. Figure 1.1 is the flowgraph for describing the outcomes of a coin tosses experiment. It starts from the input node 0 with no head and terminates in the output node 3 if we obtained 3 consecutive heads. Each branch is labeled with a quantity called the branch transmittance, which is defined as a product of the probability of taking that branch and the MGF of the waiting time. For example, the branch that connects node 1 to node 2 is assigned with branch transmittance  $q_{12}(s) = p_{12}m_{12}(s)$ , where  $p_{12}$  is the probability of transition from node 1 to node 2, and  $m_{12}(s)$  is the MGF of waiting time in node 1 before the state of system moves to node 2 (i.e. we obtain 2 consecutive heads). The structure of the flowgraph and branch transmittances summarise all information about the potential outcomes of this system.

## **Structure of the thesis**

Chapter 2 begins with a brief introduction to the flowgraph model, where three basic structures: series, parallel and feedback loop are presented. The Flowgraph model can be applied to describe any finite state stochastic network that is a Markovian system or Semi-Markov process by including non-exponential distributed waiting times between states. As the MGF plays an important role in the total waiting time density estimation, the use of Mason's rule, which allow us to compute the algebraic expression for the moment generating function of the waiting time between two nodes of interest given the internode distributions, is illustrated with examples. Our main contribution in this chapter is to propose a new derivation of Mason's rule based on matrix algebra, and the advantage of our new formula is that we can now obtain the total waiting time MGF without counting the paths and feedback loops. This has significantly improved the practicability of Mason's rule for calculating MGF in large flowgraph models. We also demonstrate the effectiveness of our formula to compute the MGF in a complicated flowgraph with combination of series, parallel and feedback loop structures.

Chapter 3 presents three different methods for inverting the MGF to obtain the PDF of total waiting time random variable. In brief, the Maximum Entropy method provides density estimation subject to the moments constraints, whereas the saddlepoint approximation is a numerical method to invert MGF by the idea of integral approximation. While the above two well-known methods work with the original given MGF, the Padé approximation approach estimates PDF based on a rational function approximation of MGF in which the direct inversion can be applied to obtain density function. Since the application of Padé approximation is much less studied in the context of flowgraph model, our main contribution in this chapter is to promote the use of Padé approximation to estimate the distribution of total waiting time between two states of interests. The implementation of each method is demonstrated in detail with examples of both exponential and non-exponential internode waiting time case, and the comparison of these methods is also given in the last section of this chapter.

---

Given a sample of total waiting times between two nodes of interest in a flowgraph, Chapter 4 compares the Maximum Likelihood method with the Method of Moment (MM) for estimating parameters in the total waiting time distribution. We start by demonstrating the application of EM algorithm for maximum likelihood estimation in the traditional mixture density (i.e. all weight are positive). As the total waiting time density is usually in the form of mixture exponential density with some negative weights, our contribution is to introduce a new systematic procedure to convert the mixture density with possible negative weights to a mixture density with positive weights, which makes the computation of maximum likelihood estimator (MLE) relatively simple using the EM algorithm.

On the other hand, we develop a easier-to-implement MGF approach for computing the bias of the MM estimator to order  $O(n^{-1})$ , where  $n$  is the size of sample data, and propose a bias correction method in MM. Although the MM is typically not as efficient as ML method in parameter estimation, it is faster and easier to implement because it does not require inverting the MGF to obtain the PDF, and hence the likelihood function. This feature is particularly useful in a big flowgraph with a complicated structure that contains large number of parameters.

Chapter 5 is devoted to the tail area approximations for the PDF and survival function of total waiting time random variable. We first review the idea of approximating tail area probabilities of PDF by an exponential function in the form of  $ce^{-at}$ , and propose a simple method to determine the asymptotic constant  $c$  and rate  $a$  based on the MGF of total waiting time. We proceed with the error analysis for the calculation of exponential function using the Padé approximation of MGF, and then derive a closed form expression for modeling the behavior of the error in estimating tail area probabilities that obtained by the Padé approximation of MGF.

Moreover, the Lugannani-Rice saddlepoint approximation for survival function is also presented. As saddlepoint approximation usually involves high computational cost if the underlying MGF is complicated, the major contribution we present here is to develop a Padé-type saddlepoint approximation, using the Padé

approximation of MGF as baseline function, to simplify the calculation in saddlepoint approximation. A bias correction method for the Padé-type saddlepoint approximation for tail probabilities is then proposed. Numerical examples for the estimation of tail area probabilities of both PDF and survival function are also illustrated to demonstrate the methods.

Chapter 6 introduces our own Matlab based computer package, the Method of Moment in Flowgraph (MMF), for computing the Method of Moments estimator for parameters in any user-defined flowgraph model. Our contribution is to develop a Matlab package that provides a convenient way to derive the MGF of total waiting time between two nodes of interest, simulate the flowgraph data, and calculate the MM estimators. This package is particularly useful to researchers and practitioners interested in applying the methodology of flowgraph to real life problems. A detailed demonstration of how to use our package is presented with an example.

# Chapter 2

## Mason's rule

### 2.1 Review of the Flowgraph models

#### 2.1.1 Introduction

Flowgraph analysis involves flowgraph algebra, which manipulates transition probabilities and the moment generating function (MGF) of internode waiting time to compute the MGF of the total waiting time and obtain the distribution of total waiting time based on the inversion of its MGF. The first step in flowgraph analysis is to identify the input node and output node from the set of nodes that represent various states of a system. The terms “input” and “output” are, here, used quite generally; the interpretation depends on the context. For example, we may consider the diagnosis of a diseases as input and death as output in medicine survival analysis. For an engineering reliability problem, the fully functioning stage of a power generating system is usually labeled as input, and the occurrence of fully failed stage is labeled as output. Secondly, we need to derive the MGF of total waiting time between the input node and output node.

This chapter will first illustrate a probabilistic approach to solve three types of flowgraph model, namely, series, parallel and single feedback loop. Next, we discuss the property of flowgraph and illustrate Mason's rule in determining the MGF between two nodes of interest. The last part of this chapter will present our new derivation of the Mason's rule based on the matrix algebra.

**Definition 1.** For a random variable  $X$  with density function  $f_X(x)$ , the moment generating function (MGF) of  $X$ ,  $M_X(s)$ , is for all  $s$ , such that

$$M_X(s) = E_X[e^{sX}] = \int_{-\infty}^{+\infty} e^{sx} f_X(x) dx$$

is convergent.

**Definition 2.** A transmittance for the branch connecting node  $i$  to node  $j$  is denoted by  $q_{ij}(s)$ , such that  $q_{ij}(s) = p_{ij}m_{ij}(s)$ , and  $i \neq j$ , where  $p_{ij}$  is the probability of transition from node  $i$  to node  $j$ , and  $m_{ij}(s)$  is the MGF of the waiting time distribution in node  $i$  before reaching node  $j$ .

## 2.1.2 A probabilistic approach to compute the MGF

### 2.1.2.1 MGF for the series flowgraph

Figure 2.1 is a series flowgraph describes, for example, the status of a power generating system with two pumps. Node 0 is the initial stage where the system is working properly with no pumps failed, node 1 indicates the event of one pump failed, node 2 represents the occurrence of two pump failed and the system breaks down. Let  $m_{01}(s)$  be the MGF of passage time  $T_{01}$  from node 0 to node 1 (i.e. the waiting time for the occurrence of one pump fails), and  $m_{12}(s)$  be the MGF of random variable  $T_{12}$ , the waiting time in node 1 before reaching node 2. The transition probability from  $0 \rightarrow 1$  and  $1 \rightarrow 2$  are all equal 1. As we are interested in the total waiting time until two pumps fail, node 0 is the input and node 2 is the output, then the total waiting time  $T_{02}$  for the power generating system to break down (i.e. node 0 to node 2) is the sum of two independent variables  $T_{01}$  and  $T_{12}$ . The MGF of  $T_{02}$  is, therefore,  $m_{01}(s)m_{12}(s)$ .

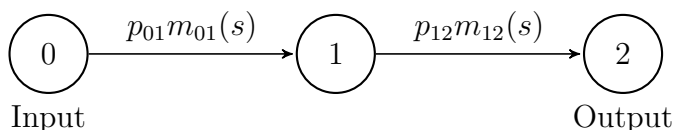


Figure 2.1: Series flowgraph model for a power generating system

Now, suppose we have a series flowgraph model with  $n$  nodes, let  $t_{i,i+1}$  be the waiting time of a particle in the node  $i$  before it gets to node  $i + 1$ ,  $i = 1 \dots n - 1$ , then the overall waiting time from the first node to the  $n$ -th node is  $T = \sum_{i=1}^{n-1} t_{i,i+1}$ . Since the MGF of the sum of independent random variables is just the product of individual MGF, the MGF of the overall waiting time distribution  $T$  is

$$M_T(s) = \prod_{i=1}^{n-1} m_{i,i+1}(s)$$

where  $m_{i,i+1}(s)$  is the MGF of  $t_{i,i+1}$ . The result shows that the MGF of total waiting time in a series flowgraph is a product of the MGFs of internode transition time.

### 2.1.2.2 MGF for the parallel flowgraph

Figure 2.2 is a parallel flowgraph model for the progression of cancer patients. Node 0 represents the initial diagnosis of cancer, node 1 is the advanced state of cancer, and node 2 is the event of death. Patients could reach node 1 with probability  $p_{01}$  or die with probability  $p_{02} = 1 - p_{01}$ . Once the patient is in node 1, the transition to node 2 is certain and  $p_{12} = 1$ . Since a path is defined as a sequence of nodes from input to output that does not pass through any intermediate nodes more than once, then there are two paths from input node 0 to output node 2 in Figure 2.2. (see Table 2.1).

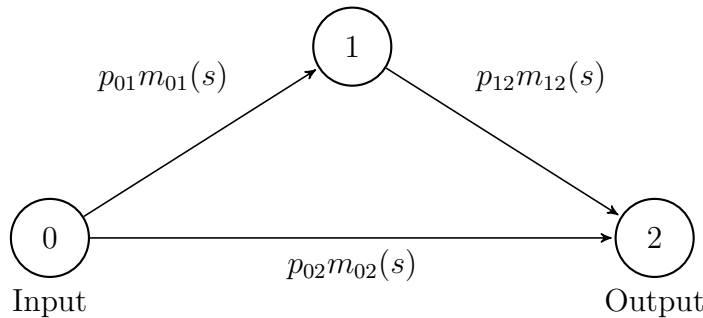


Figure 2.2: Parallel flowgraph model for cancer progression

Table 2.1: Paths for the flowgraph of cancer progression

Path $j$	Path probability $P_j$	Path MGF $M_{X_j}(s)$
1 : $0 \rightarrow 2$	$p_{02}$	$m_{02}(s)$
2 : $0 \rightarrow 1 \rightarrow 2$	$p_{01}p_{12}$	$m_{01}(s)m_{12}(s)$

Suppose we are interested in the total waiting time to the death of patient due to any cause (e.g. either dying with or without advanced state of cancer), let  $X_1, X_2$  be the waiting time to reach node 2 by taking path 1 and path 2 respectively, and let  $Y$  be the total waiting time for the occurrence of death. If we assume distribution for  $X_1$  and  $X_2$ , and the probability of taking path 1 and path 2 is  $P_1$  and  $P_2$  separately.

$$\begin{aligned} P(Y < y) &= P(X_1 \leq y|j = 1)P_1 + P(X_2 \leq y|j = 2)P_2 \\ &= P(X_1 \leq y|0 \rightarrow 2)P_1 + P(X_2 \leq y|0 \rightarrow 1 \rightarrow 2)P_2 \end{aligned}$$

Then

$$\begin{aligned} f_Y(y) &= P(X_1 = y|j = 1)P_1 + P(X_2 = y|j = 2)P_2 \\ &= f_{X_1}(y)P_1 + f_{X_2}(y)P_2 \end{aligned}$$

The MGF of the total waiting time from node 0 to node 2 is

$$\begin{aligned} M_Y(s) &= E_Y(e^{sY}) \\ &= \int_0^\infty e^{sy} (f_{X_1}(y)P_1 + f_{X_2}(y)P_2) dy \\ &= P_1 M_{X_1}(s) + P_2 M_{X_2}(s) \\ &= p_{02}m_{02}(s) + p_{01}p_{12}m_{01}(s)m_{12}(s) \end{aligned}$$

Since  $p_{01} + p_{02} = 1$  and  $p_{12} = 1$ , then  $M_Y(s)$  is a mixture of two different paths MGF: with probability  $p_{02}$  it is  $m_{02}(s)$ , and with probability  $p_{01}$  it is  $m_{01}(s)m_{12}(s)$ .



Given  $k$  possible paths between input node and output node in a parallel flowgraph, the total waiting time is the passage time from input to output, and it depends on the paths that we chose to travel to the output node. Let  $T_j$  be the random variable that represents the waiting time to reach output node by selecting path  $j$  with probability  $P_j$ , then the PDF of the total waiting time  $T$  between input and output is

$$f_T(t) = \sum_{j=1}^k P(T_j = t | j)P_j$$

where  $k$  is the total number of different paths between input node and output node. The MGF of total waiting time distribution can be expression as

$$\begin{aligned} M_T(s) &= E(e^{sT}) \\ &= \int_0^{\infty} e^{st} f_T(t) dt \\ &= \int_0^{\infty} e^{st} \sum_{j=1}^k P(T_j = t | j)P_j dt \\ &= \int_0^{\infty} e^{st} \{f_{T_1}(t)P_1 + f_{T_2}(t)P_2 + \dots + f_{T_k}(t)P_k\} dt \\ &= P_1M_{T_1}(s) + P_2M_{T_2}(s) + \dots + P_kM_{T_k}(s) \\ &= \sum_{j=1}^k P_jM_{T_j}(s) \end{aligned}$$

where  $M_{T_j}(s)$  is the MGF of total waiting time to reach output node by taking path  $j$  with probability  $P_j$ , and  $\sum_{j=1}^k P_j = 1$ . Hence the MGF of the overall waiting time in a parallel flowgraph is a finite mixture MGFs that results from taking each path between input node and output node.

### 2.1.2.3 MGF for the flowgraph with feedback loops

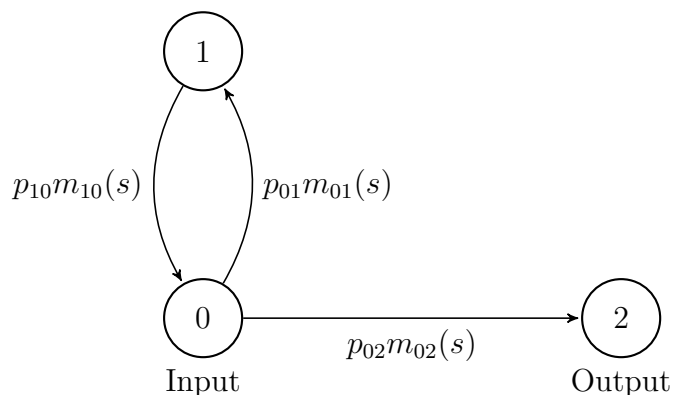


Figure 2.3: Single feedback loop flowgraph model

Figure 2.3 is a simple 3 nodes flowgraph that contains only one feedback loop. Assume a particle starts from input node 0 can either directly move to output node 2 or pass through the feedback loops  $0 \rightarrow 1 \rightarrow 0$  before transition to output node 2. let  $p_{02}$  be the probability of transition from node 0 to node 2, and  $p_{02} > 0$  (i.e the particle will eventually get to node 2), then the probability of the particle taking the feedback loop and return to node 0 (i.e. path  $0 \rightarrow 1 \rightarrow 0$ ) is  $p_{01} = 1 - p_{02}$ . Each transition is considered as a Bernoulli trials with the probability of “success”  $p_{02}$  (i.e. path  $0 \rightarrow 2$ ). Let  $N$  be the the number of times a particle takes the feedback loop before it reaches output node 2, and  $N \sim \text{Geometric}(p_{02})$ , then the probability of first success transition to node 1 after  $k$  times return to node 0 is

$$P(N = k) = (1 - p_{02})^k p_{02} \quad \text{for } k = 0, 1, 2 \dots \quad (2.1)$$

Let  $U$  be the total time of a particle spent in the path  $0 \rightarrow 1 \rightarrow 0$  and it has density function  $f_U(u)$  and MGF  $M_U(s) = m_{00}(s)$ . Since this total time is a sum of independent waiting time in  $0 \rightarrow 1$  and  $1 \rightarrow 0$ . Therefore, the distribution of  $U$  is a convolution of the distributions of the independent waiting time  $T_{01}$  and  $T_{10}$ , and we have

$$M_U(s) = m_{00}(s) = m_{01}(s)m_{10}(s)$$

Let  $V$  be the time of first transition from node 0 to node 2, it is independent of  $U$  and has probability density function  $f_V(v)$  and MGF  $M_V(s) = m_{02}(s)$ . Since  $N$  is the total number of times that a particle takes the feedback loop before it gets to output node 2, the total waiting time  $W$  from node 0 to node 2 is then the sum of time for particle to take the feedback loop,  $\sum_{i=1}^N U_i$ , and the time in the last transition from node 0 to node 2,  $V$ . Therefore, the distribution of  $W$  is the convolution of  $N$  distributions  $f_U(u)$  and a single distribution  $f_V(v)$ .

$$W = \sum_{i=1}^N U_i + V$$

The density of  $W$  is

$$f_W(w) = \sum_{k=0}^{\infty} f_W(w|N = k)p(N = k)$$

The MGF of  $W$  is

$$\begin{aligned} M_W(s) &= E(e^{sW}) \\ &= \int_0^{\infty} e^{sw} f_W(w) dw \\ &= \sum_{k=0}^{\infty} \int_0^{\infty} e^{sw} f_W(w|N = k) p(N = k) dw \\ &= \sum_{k=0}^{\infty} p(N = k) \int_0^{\infty} e^{sw} f_W(w|N = k) dw \\ &= \sum_{k=0}^{\infty} P(N = k) M_{W|N=k}(s) \end{aligned} \tag{2.2}$$

where integration and summation can be interchanged as a consequence of the monotone convergence theorem (Theorem 16.6, Billingsley (1986)). Since  $V$  is independent of  $U$  and  $U_i$  is i.i.d with MGF  $M_U(s)$ , the MGF of  $W|N = k$  is

$$M_{W|N=k}(s) = (M_U(s))^k M_V(s) \tag{2.3}$$

By substituting equation 2.3 and 2.1 into 2.2, we have

$$\begin{aligned} M_W(s) &= \sum_{k=0}^{\infty} (1 - p_{02})^k p_{02} \left( M_U(s) \right)^k M_V(s) \\ &= p_{02} M_V(s) \sum_{k=0}^{\infty} \left\{ (1 - p_{02}) M_U(s) \right\}^k \end{aligned}$$

Since  $M_U(s) \approx 1$  for  $s$  near 0 and  $1 - p_{02} < 1$ , then  $|(1 - p_{02})M_U(s)| < 1$  holds in an open neighborhood of  $s = 0$ . Therefore the MGF of  $W$  is

$$\begin{aligned} M_W(s) &= \frac{p_{02} M_V(s)}{1 - (1 - p_{02}) M_U(s)} \\ &= \frac{p_{02} m_{02}(s)}{1 - (1 - p_{02}) m_{00}(s)} \\ &= \frac{p_{02} m_{02}(s)}{1 - p_{01} m_{01}(s) m_{10}(s)} \end{aligned}$$

The above examples contain some of the main ideas. In particular we see that difference between the structure of the input-output MGF for the directed case and the case with feedback loops.

## 2.2 A flowgraph approach to compute the MGF

A flowgraph can be considered as a directed graph obtained by assigning transmittance for each edge. That is, a directed graph in which every edge is associated with a function which is a product of the transition probability and the MGF of internode waiting time distribution. In general, Mason's rule is a procedure for determining the MGF of the waiting time distribution between any two nodes of interest in flowgraph, provided that there is at least one path between those two nodes. The first step in applying Mason's rule is to identify all the distinct paths from input to output as well as the loops involved in those paths, then we need to compute the corresponding transmittance (i.e. a product of transition probability and MGF) and substitute them accordingly to the Mason's formula.

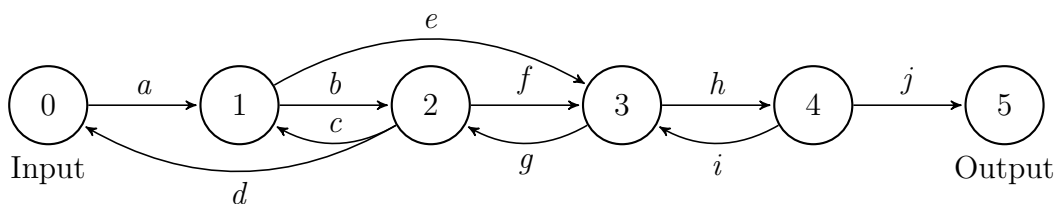


Figure 2.4: An illustrated flowgraph example

**Definition 3.** A directed graph is defined as  $G = (V, E)$  with

1. a set  $V = \{1, \dots, n\}$ , whose elements are called nodes.
2. a set  $E$  of nodes in  $V$  called directed edges.

**Definition 4.** A first-order loop is any path that returns to the starting node of the feedback loop without passing through any node more than once.

**Definition 5.** A  $j$ -th order loop consists of  $j$  non-touching first-order loops. (i.e. the loops do not share a common node). The transmittance of a  $j$ -th order loop is the product of the transmittance of  $j$  first-order loops it contains.

As the Manson's rule involves identifying paths and feedback loops based on the above definitions, in order to avoid ambiguity, we now provide a schematic representation of Definition 3, 4, and 5 in Figure 2.4. By definition 3, the flowgraph in Figure 2.4 can be defined by  $G = (V, E)$  such that

1. The set of nodes,  $V = \{0, 1, 2, 3, 4, 5\}$ .
2. The set of directed edges,  $E = \{a, b, c, d, e, f, g, h, i, j\}$ .

According to Definition 4, we have four first-order loops  $1 \rightarrow 2 \rightarrow 1$ ,  $2 \rightarrow 3 \rightarrow 2$ ,  $3 \rightarrow 4 \rightarrow 3$  and  $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$ . However, the loop  $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1$  is not a correct first-order loop because the path that returns to node 1 has passed through node 2 twice. Furthermore, by Definition 5, the feedback loops  $1 \rightarrow 2 \rightarrow 1$  and  $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$  do not form a second order loop because they share a common node at node 2. In fact, the only second order loop in Figure 2.4 is the pair of loops  $1 \rightarrow 2 \rightarrow 1$  and  $3 \rightarrow 4 \rightarrow 3$ .

The Mason's rule was originally derived by S.J. Mason for finding the transfer function in electrical engineering (see Mason (1953, 1956)). The transfer functions are commonly used within the fields of signal processing, communication theory, and control theory. It is usually a mathematical representation, in terms of MGF, of the relation between the input and output of a linear time-invariant system. Butler (1997a) gives a modernised version of proof for Mason's rule based on permutation theory and linear algebra, and introduces an cofactor formula based on the simplification of matrix systems procedure that is discussed by Pyke (1961) and Howard (1964, 1971).

**Theorem 1.** *The general form of Mason's rule gives the MGF of the total waiting time from input node to output node as*

$$M(s) = \frac{\sum_k P_k(s)[1 + \sum_j (-1)^j L_j^k(s)]}{1 + \sum_j (-1)^j L_j(s)}$$

where

1.  $P_k(s)$  is the transmittance for the  $k$ -th path from input node to output node.
2.  $L_j(s)$  the sum of the transmittances over the  $j$ -th order loops.
3.  $L_j^k(s)$  is the sum of the transmittances over  $j$ -th order loops sharing no common nodes with the  $k$ -th path (i.e. loops not touching the  $k$ -th path).

### 2.2.1 Example 1

Figure 2.5 is a flowgraph model with a combination of series, parallel, and loop structures. It describes a group of patients within a three states reversible illness-death system. Node 0 is the healthy state, where patients can transition to diseased state in node 1 with probability  $p_{01}$ , or die with probability  $p_{02} = 1 - p_{01}$ . The waiting time distribution for the transition from node 0 to node 1 has MGF,  $m_{01}(s)$ . For a patient in node 1, the next possible transition is either to node 0 with probability  $p_{10}$  or to node 2 with probability  $p_{12} = 1 - p_{10}$ , and the MGF of

each waiting times are  $m_{10}(s)$  and  $m_{12}(s)$  respectively. Our interest is to compute,  $M_{02}(s)$ , the MGF of total waiting time distribution from input node 0 to output node 2. In this flowgraph, there are two paths and one first order loop from input node 0 to output node 2.

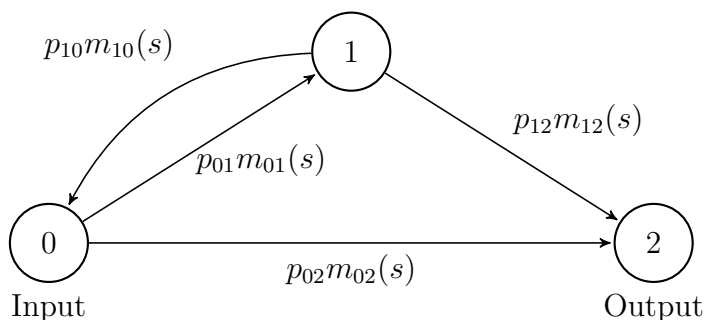


Figure 2.5: Flowgraph model for a reversible illness-death system

1. Path 1:  $0 \rightarrow 1 \rightarrow 2$ ,  $P_1(s) = q_{01}(s)q_{12}(s)$
2. Path 2 :  $0 \rightarrow 2$ ,  $P_2(s) = q_{02}(s)$
3. First order loop  $0 \rightarrow 1 \rightarrow 0$ ,  $L_1(s) = q_{01}(s)q_{10}(s)$ .

where  $q_{ij}(s) = p_{ij}m_{ij}(s)$  is the branch transmittance defined in Definition 2,  $p_{ij}$  is the transition probability from node  $i$  to node  $j$ , and  $m_{ij}(s)$  is the MGF of the waiting time between node  $i$  and node  $j$ .

By Mason's rule in Theorem 1, the MGF of the total waiting time from node 0 to node 2 is

$$\begin{aligned}
 M_{02}(s) &= \frac{q_{02}(s) + q_{01}(s)q_{12}(s)}{1 - q_{01}(s)q_{10}(s)} \\
 &= \frac{p_{02}m_{02}(s) + p_{01}p_{12}m_{01}(s)m_{12}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)}
 \end{aligned} \tag{2.4}$$

### 2.2.2 Example 2

Consider a more complicated flowgraph given in Figure 2.6.

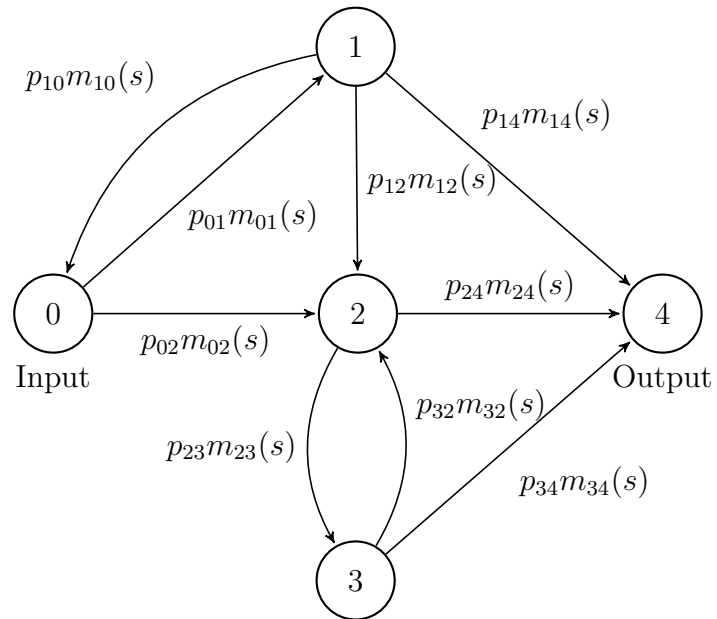


Figure 2.6: A complex flowgraph



Table 2.2 presents a list of paths, first and second order loops between input node 0 and output node 4.

Table 2.2: Summary of Paths and loops for the flowgraph

	Paths	Transmittance
1	$0 \rightarrow 2 \rightarrow 4$	$q_{02}(s)q_{24}(s)$
2	$0 \rightarrow 1 \rightarrow 4$	$q_{01}(s)q_{14}(s)$
3	$0 \rightarrow 1 \rightarrow 2 \rightarrow 4$	$q_{01}(s)q_{12}(s)q_{24}(s)$
4	$0 \rightarrow 2 \rightarrow 3 \rightarrow 4$	$q_{02}(s)q_{23}(s)q_{34}(s)$
5	$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	$q_{01}(s)q_{12}(s)q_{23}(s)q_{34}(s)$
First order loops:	$0 \rightarrow 1 \rightarrow 0$	$q_{01}(s)q_{10}(s)$
	$2 \rightarrow 3 \rightarrow 2$	$q_{23}(s)q_{32}(s)$
Second order loops:	$0 \rightarrow 1 \rightarrow 0$ and $2 \rightarrow 3 \rightarrow 2$	$q_{01}(s)q_{10}(s)q_{23}(s)q_{32}(s)$
Loop not touch path 2:	$2 \rightarrow 3 \rightarrow 2$	$q_{23}(s)q_{32}(s)$

By Mason's rule, the MGF of total waiting time between node 0 and node 4 is

$$M_{04}(s) = \frac{P_1(s) + P_2(s) [1 - L_1^2(s)] + P_3(s) + P_4(s) + P_5(s)}{1 - L_1(s) + L_2(s)}$$

where

$$P_1(s) = q_{02}(s)q_{24}(s)$$

$$P_2(s) = q_{01}(s)q_{14}(s)$$

$$P_3(s) = q_{01}(s)q_{12}(s)q_{24}(s)$$

$$P_4(s) = q_{02}(s)q_{23}(s)q_{34}(s)$$

$$P_5(s) = q_{01}(s)q_{12}(s)q_{23}(s)q_{34}(s)$$

$$L_1^2(s) = q_{23}(s)q_{32}(s)$$

$$L_1(s) = q_{01}(s)q_{10}(s) + q_{23}(s)q_{32}(s)$$

$$L_2(s) = q_{01}(s)q_{10}(s)q_{23}(s)q_{32}(s)$$

and  $q_{ij}(s) = p_{ij}m_{ij}(s)$  is the branch transmittance between node  $i$  and node  $j$

## 2.3 A new derivation of the Mason's rule

Despite the fact that Mason's rule only requires identifying all the paths and feedback loops to compute the total waiting time MGF, Phillips (1996) points out that Mason's rule must be used with extreme care, it become increasingly difficulty to identify the number of paths and loops correctly between input and output in a large complicated flowgraph, because feedback loops can easily be overlooked. In general, it can be complicated to implement the Mason's formula without making mistakes, particularly, as we can see in Figure 2.6, the existence of non-touching loops increases the complexity of the formula. To overcome this problem, we take a different approach and develop a new formula to compute the MGF based on the internode transition matrix of flowgraph.

Suppose we consider the "flow" in terms of particles, a basic property of flowgraph is the principle of mass conservation, which the outflow from a node is equal to its inflow. We show this property in the next two examples. Figure 2.7 is a flowgraph with two input nodes at node 1 and node 2. Let  $x_i$  denote the number of particles come out from node  $i$ . For convenience, we drop  $s$  from  $q_{ij}(s)$  and denote the transmittance from node  $i$  to node  $j$  by  $q_{ij}$ . Suppose each input node has 1 particle, then  $x_3$  can be expressed in terms of  $x_1$  and  $x_2$  as

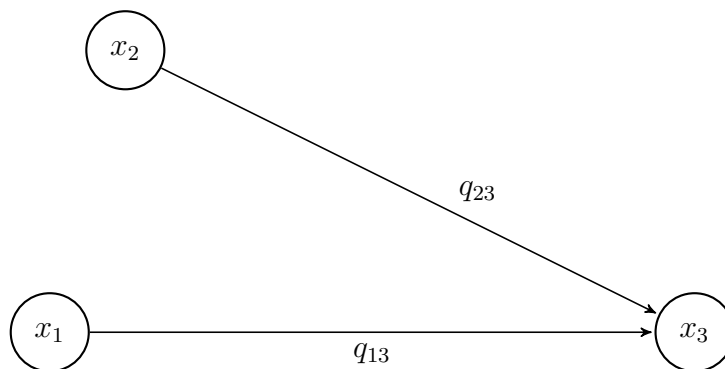


Figure 2.7: Flowgraph with two input nodes

$$\begin{aligned}x_1 &= 1 \\x_2 &= 1 \\x_3 &= q_{13}x_1 + q_{23}x_2\end{aligned}$$

In matrix form

$$AX = Y$$

where

$$A = I - Q^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -q_{13} & -q_{23} & 1 \end{pmatrix}$$

$$\text{and } Q = \begin{pmatrix} 0 & 0 & q_{13} \\ 0 & 0 & q_{23} \\ 0 & 0 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Figure 2.8 is a flowgraph with one feedback loop, where  $x_1$  is defined as the input node with 1 particle outflow. The relationship between each node can be described by linear equations shown below

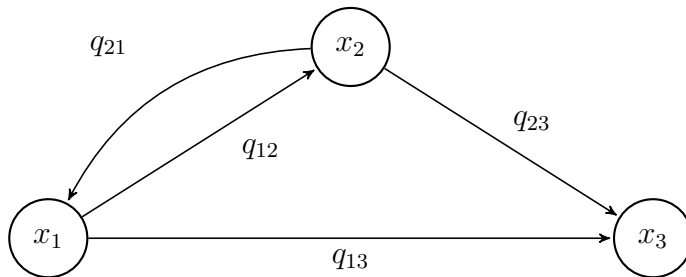


Figure 2.8: A closed flowgraph with one feedback loop

$$\begin{aligned}x_1 &= q_{21}x_2 + 1 \\x_2 &= q_{12}x_1 \\x_3 &= q_{23}x_2 + q_{13}x_1\end{aligned}$$

In matrix form

$$AX = Y$$

where

$$A = I - Q^T = \begin{pmatrix} 1 & -q_{21} & 0 \\ -q_{12} & 1 & 0 \\ -q_{13} & -q_{23} & 1 \end{pmatrix}$$

$$\text{and } Q = \begin{pmatrix} 0 & q_{12} & q_{13} \\ q_{21} & 0 & q_{23} \\ 0 & 0 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

In a general flowgraph with  $n$  nodes, we can use a set of independent simultaneous linear equations to represent the relationship between nodes.

$$\begin{aligned}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= y_1 \\a_{21}x_1 + x_2 + \cdots + a_{2n}x_n &= y_2 \\&\vdots \\a_{n1}x_1 + a_{n2}x_2 + \cdots + x_n &= y_n\end{aligned}$$

In matrix form  $AX = Y$ , where  $A$  is a  $n$ -by- $n$  coefficient matrix, such that

$$\begin{aligned}A &= I - Q^T \\&= \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ a_{21} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & 1 \end{bmatrix}\end{aligned}$$

and

$$\begin{aligned} a_{ij} &= -q_{ji} = -p_{ji}m_{ji}(s) \\ a_{ii} &= 1 \end{aligned}$$

where  $Q$  is the branch transmittance matrix,  $X$  denotes the nodes vector in flowgraph, and  $Y$  is the input node indicator vector, where  $y_i = 1$  if node  $i$  is the input node of the flowgraph (i.e. the node has only outflow), otherwise  $y_i = 0$ .

Consider modeling a  $n$  states stochastic process in terms of a flowgraph, we first need to identify two states of interest. Without any loss in generality, we set the input at node 1 and output at node  $n$  to defined the direction of transition. Secondly, we need to reduce the given flowgraph to a smaller one by excluding those nodes along with all branches connected with such nodes that are not the possible intermediate nodes during the transition from input to output. For example, if we set the input node at 3 and output node at 4 in Figure 2.6, then node 0 and node 1 are irrelevant nodes and can be removed. This process of simplifying a large flowgraph to a simple one with only two nodes and one branch is called solving a flowgraph (Figure 2.9 is the solved flowgraph for Figure 2.5 on page 19). Note that Bulter (2000) generalised this procedure to solve flowgraph in the single input and multiple outputs case (see Bulter (2000), Section 4.2).

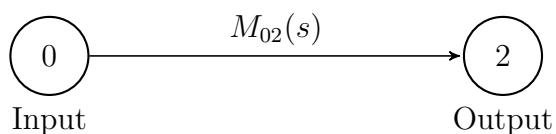


Figure 2.9: Solved flowgraph for the reversible illness-death system

After we simplified the complex flowgraph, the directed branch that connects the node 1 with node  $n$  is labeled by the equivalent transmittances  $M_{1n}$ , which represents the transmittance of the entire flowgraph from input node to output node. If the transition between these two nodes is certainly happening in a finite time, this overall transmittances is the MGF of total waiting time distribution.

To determine  $M_{1n}$ , we close the network by adding an extra branch from the output node  $n$  to the input node 1 and label the corresponding transmittance by  $w_{n1}$  (see Figure 2.10). The idea behind introducing this transmittance  $w_{n1}$  is to simplify the calculation of  $M_{1n}$  in the context of closed flowgraph. By the principle of conservation of mass, the transmittance in a closed system will remain constant over time and will not be destroyed as a result, regardless of the processes acting inside the system. In this case,  $w_{n1}$  will convert all the flow in output node  $n$  back to the input node 1, which is the reciprocal of target MGF.

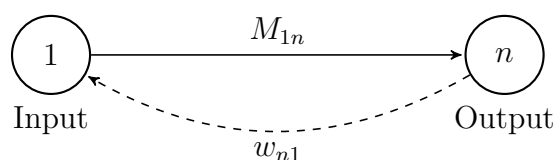


Figure 2.10: Flowgraph model for a closed network system

We are now going to show a relationship between  $M_{1n}$ , the MGF of total waiting time between input node 1 and output node  $n$ , and the extra branch transmittance  $w_{n1}$  that connects those two nodes of interest. The branch transmittance matrix  $\tilde{Q}$  for the closed flowgraph in Figure 2.10 is

$$\tilde{Q} = \begin{pmatrix} 0 & M_{1n} \\ w_{n1} & 0 \end{pmatrix}$$

As there is no input in a closed network, the input node indicator  $Y = 0$ , and the set of independent linear equations show the relationship between the nodes are a homogeneous system  $\tilde{A}X = 0$ , where the coefficient matrix  $\tilde{A}$  is

$$\begin{aligned} \tilde{A} &= I - \tilde{Q}^T \\ &= \begin{pmatrix} 1 & -w_{n1} \\ -M_{1n} & 1 \end{pmatrix} \end{aligned}$$

The determinant of coefficient matrix  $\tilde{A}$  for this closed network is

$$\det(\tilde{A}) = 1 - M_{1n}w_{n1} \quad (2.5)$$

Pritsker and Happ (1966) claims that the determinant of the coefficient matrix in a closed network is zero ( see Equation 5 in Pritsker and Happ (1966)), we will discuss this proposition and give a proof in the following section.

**Lemma 1.** *Given the branch transmittance matrix  $Q$  of a non-closed flowgraph with  $n$  nodes, we partition the transpose of matrix  $Q$  in the form of*

$$Q^T = \begin{bmatrix} B & 0 \\ v^T & 0 \end{bmatrix}$$

and where  $B$  is a  $(n-1) \times (n-1)$  matrix,  $v^T$  is  $1 \times (n-1)$  row vector. Then the MGF of waiting time between node 1 and output node  $n$ , such that  $1 \neq n$ , is

$$M_{1n}(s) = v^T (I_{n-1} - B)^{-1} Y$$

where  $Y = (y_1, \dots, y_{n-1})^T$  is a  $(n-1) \times 1$  input node indicator vector with  $y_i = 1$  for input at node  $i$ , and zeros otherwise.

*Proof.* Suppose we consider the flow as particles from input to output in a non-closed flowgraph. For ease of understanding, we set the input at node 1 and output at node  $n$ . By partitioning the transpose of branch transmittance matrix in a special way,

$$Q^T = \begin{bmatrix} B & 0 \\ v^T & 0 \end{bmatrix}$$

where  $B$  is a  $(n-1) \times (n-1)$  matrix that consists of all the  $q_{kl}$  for all  $1 \leq k \leq n-1$ ,  $1 \leq l \leq n-1$ , and  $v^T$  is  $1 \times (n-1)$  row vector such that

$$v^T = (-q_{1n}, -q_{2n}, \dots, -q_{n-1,n})$$

We are interested in the time taken for particle to reach output node  $n$ , where particles will never leave once it enters the output node. (i.e. probability  $p_{nn} = 1$ ). We assume that there is no loop on a single node to itself, and it takes at least two transitions to return a node. The key idea in our proof is to use matrix  $B$  and row vector  $v^T$  from  $Q^T$  to construct a  $n \times n$  matrix  $D(s)$  with bottom righthand corner entry equals 1 (i.e.  $d_{nn}(s) = 1$ ). To simplify the representation, we will drop  $s$  for writing convenience. Define

$$D = \begin{bmatrix} B & 0 \\ v^T & 1 \end{bmatrix}$$

Since the input is chosen at node 1, the input node indicator vector is

$$Y = (y_1, \dots, y_n)^T$$

with  $y_1 = 1$  and zeros otherwise, then  $DY$  shows the path transmittance in each node  $1, \dots, n$  after one transition,  $D^2Y$  shows the path transmittance after two consecutive transitions, then  $D^mY$  represent the path transmittance after  $m$  transitions in each node. Let  $m$  be the total number of transitions that particle has made since input node 1.

$$D^m = \begin{bmatrix} B^m & 0 \\ v^T C & 1 \end{bmatrix}$$

where

$$C = I_{n-1} + B + \dots + B^{m-1}$$

and  $I_{n-1}$  is the  $(n-1) \times (n-1)$  identity matrix.

At this point, we need to propose the following conditions

1.  $B^m \rightarrow 0$ , as  $m \rightarrow \infty$
2.  $(I_{n-1} + B + \dots + B^{m-1}) = (I_{n-1} - B)^{-1}$ , for  $m \rightarrow \infty$ , and  $|B| < 1$ .



NOTE: The first condition arises from our assumption that the particle always end up at the absorbing state (i.e. output node). The second condition is just that  $\|B\|^2 < 1$  for some suitable matrix norm  $\|\cdot\|$ .

With these two conditions,  $m \rightarrow \infty$  gives

$$D^m \rightarrow D^* = \begin{bmatrix} \mathbf{0} & \underline{\mathbf{0}} \\ v^T(I_{n-1} - B)^{-1} & 1 \end{bmatrix}$$

Since we have a single output at node  $n$ , then the MGF of total waiting time from node 1 to node  $n$  just is the  $n$ -th entry of  $D^m Y$  such that

$$\begin{aligned} D^m Y \rightarrow D^* Y &= \begin{bmatrix} \mathbf{0} & \underline{\mathbf{0}} \\ v^T(I_{n-1} - B)^{-1} & 1 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{1}} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \underline{\mathbf{0}} \\ \{v^T(I_{n-1} - B)^{-1}\}_1 \end{bmatrix} \\ &= \begin{bmatrix} \underline{\mathbf{0}} \\ d_{n1}^* \end{bmatrix} \end{aligned} \tag{2.6}$$

where  $d_{n1}^*$  is just the first component of  $1 \times (n-1)$  row vector  $v^T(I_{n-1} - B)^{-1}$ . (i.e.  $\{v^T(I_{n-1} - B)^{-1}\}_1$ ).  $\mathbf{0}$  denotes a  $(n-1) \times (n-1)$  zero matrix, and both  $\underline{\mathbf{0}}$  and  $\underline{\mathbf{1}}$  are  $(n-1) \times 1$  vector.

In general, the MGF of the total waiting time from input at node 1 to output at node  $n$  in a non-closed flowgraph is

$$M_{1n}(s) = v^T(I_{n-1} - B)^{-1} Y$$

where  $y = (y_1, y_2, \dots, y_n)^T$  with  $y_i = 1$  for input at node  $i$ , and zeros otherwise.  $\square$

Lemma 1 shows that the MGF of total waiting time between two nodes of interest can be determined based on the matrix operations. With equation 2.6, we can now prove a further result given below.

**Lemma 2.** *If a flowgraph is closed by adding a branch from output node to input node, then the determinant of coefficient matrix  $\tilde{A} = I_n - \tilde{Q}^T$  in this closed flowgraph is zero.*

*Proof.* Let  $\tilde{Q}$  be the branch transmittance matrix of closed flowgraph, we first partition the transpose of matrix  $\tilde{Q}$  as

$$\tilde{Q}^T = \begin{bmatrix} B & w \\ v^T & 0 \end{bmatrix}$$

where we take

$$w = (d_{n1}^{*-1}, 0, \dots, 0)^T \quad (2.7)$$

and  $d_{n1}^*$  is the MGF of total waiting time from the input at node 1 and output at node  $n$ . The coefficient matrix is

$$\tilde{A} = I_n - \tilde{Q}^T = \begin{bmatrix} I_{n-1} - B & -w \\ -v^T & 1 \end{bmatrix}$$

Then

$$\begin{aligned} \det(\tilde{A}) &= \det \left( \begin{bmatrix} I_{n-1} - B & -w \\ -v^T & 1 \end{bmatrix} \right) \\ &= |(I_{n-1} - B) - wv^T| \\ &= |(I_{n-1} - B)[1 - v^T(I_{n-1} - B)^{-1}w]| \end{aligned} \quad (2.8)$$

By equation 2.6 and 2.7, we have

$$\begin{aligned} 1 - v^T(I_{n-1} - B)^{-1}w &= 1 - d_{n1}^*d_{n1}^{*-1} \\ &= 1 - 1 \\ &= 0 \end{aligned} \quad (2.9)$$

Substitute result 2.9 in equation 2.8, then  $\det(\tilde{A}) = 0$ . Hence the determinant of coefficients matrix in a closed flowgraph is zero.  $\square$

By Lemma 2,  $\det(\tilde{A})$  in equation 2.5 for Figure 2.10 is zero,

$$\det(\tilde{A}) = 1 - M_{1n}w_{n1} = 0$$

Then,

$$M_{1n} = \frac{1}{w_{n1}} \quad (2.10)$$

Thus, the MGF of total waiting time from input node 1 to output node  $n$ ,  $M_{1n}$ , is the reciprocal of the branch transmittance  $w_{n1}$  that connects node  $n$  to node 1.

**Theorem 2.** *Let a finite  $n$  nodes flowgraph is closed by adding a directed edge associate with transmittance  $w_{n1}$  that connects output node  $n$  to input node 1. Let  $\tilde{A} = I_n - \tilde{Q}^T$ , where  $I_n$  is  $n \times n$  identity matrix,  $\tilde{Q}$  is the branch transmittance matrix of the closed flowgraph. Then the MGF of waiting time from node 1 to node  $n$  is*

$$M_{1n}(s) = -\frac{\frac{\partial \det(\tilde{A})}{\partial w_{n1}}}{\det(\tilde{A} |_{w_{n1}=0})}$$

*Proof.* In a closed flowgraph that consists only of loops, there is no input or output, and the input indicator vector  $Y$  becomes zero, then we have a homogeneous system to describe the closed flowgraph.

$$\tilde{A}X = 0$$

As  $w_{n1}$  is the transmittance for the directed edge that connects from node  $n$  to node 1, we consider  $\det(\tilde{A})$  as a function of  $w_{n1}$ , and  $\det(\tilde{A})$  can be separated into two parts,

$$\det(\tilde{A}) = \det(\tilde{A}) |_{w_{n1}=0} + \frac{\partial \det(\tilde{A})}{\partial w_{n1}} \cdot w_{n1}$$

By Lemma 2,  $\det(\tilde{A})$  is zero in a closed flowgraph, then

$$\det(\tilde{A}) |_{w_{n1}=0} + \frac{\partial \det(\tilde{A})}{\partial w_{n1}} \cdot w_{n1} = 0$$

$$w_{n1} = -\frac{\det(\tilde{A})|_{w_{n1}=0}}{\frac{\partial \det(\tilde{A})}{\partial w_{n1}}}$$

By equation 2.10, there is a reciprocal relationship between the MGF from node 1 to node  $n$ ,  $M_{1n}$ , and the branch transmittance  $w_{n1}$ ,

$$M_{1n} = \frac{1}{w_{n1}} = -\frac{\frac{\partial \det(\tilde{A})}{\partial w_{n1}}}{\det(\tilde{A})|_{w_{n1}=0}} \quad (2.11)$$

□

**Lemma 3.** *Let  $H$  be a  $n \times n$  square matrix, then*

$$\frac{\partial}{\partial h_{ij}} \log(\det(H)) = \frac{\frac{\partial \det(H)}{\partial h_{ij}}}{\det(H)} = \text{tr} \left[ H^{-1} \frac{\partial H}{\partial h_{ij}} \right]$$

*provided that  $\det(H) \neq 0$ , and  $\text{tr}(H) = h_{11} + h_{22} + \dots + h_{nn} = \sum_{i=1}^n h_{ii}$  is the trace of the matrix  $H$*

**Theorem 3.** *The MGF of total waiting time distribution from input node 1 to output node  $n$  in a non-closed finite  $n$  nodes flowgraph is just the  $(n,1)$ th entry of its inverse coefficient matrix  $A$ :*

$$M_{1n}(s) = [(A)^{-1}]_{n1}$$

*where  $A = I_n - Q^T$ , and  $Q$  is the branch transmittance matrix of the flowgraph.*

*Proof.* Given a  $n \times n$  coefficient matrix  $\tilde{A}$  of closed flowgraph, by equation 2.11 and Lemma 3,

$$M_{1n}(s) = -\frac{\frac{\partial \det(\tilde{A})}{\partial w_{n1}}}{\det(\tilde{A})|_{w_{n1}=0}} = -\text{tr} \left[ (\tilde{A}|_{w_{n1}=0})^{-1} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \quad (2.12)$$

Since only one entry of coefficient matrix  $\tilde{A}$  involves  $w_{n1}$  such that  $\tilde{a}_{1n} = -w_{n1}$ , then this entry becomes  $-1$  after we differentiate  $\tilde{a}_{1n}$  with respect to  $w_{n1}$ , whereas those entries without  $w_{n1}$  become zero.

Let  $A^* = \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right)$ , and  $A^*$  is a constant matrix such that

$$A^* = \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right)_{kl} = \begin{cases} -1 & \text{if } a_{kl}^* = -w_{n1} \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

Define  $\bar{A} = (\tilde{A} |_{w_{n1}=0})^{-1}$  and substitute 2.13 in equation 2.12

$$M_{1n}(s) = -\text{tr} \left[ (\tilde{A} |_{w_{n1}=0})^{-1} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] = -\text{tr}[\bar{A} \cdot A^*] \quad (2.14)$$

For  $\bar{A} \in \mathbb{R}^{n \times n}$ ,  $A^* \in \mathbb{R}^{n \times n}$ , we have

$$\begin{aligned} -\text{tr}[\bar{A} \cdot A^*] &= -\sum_{u=1}^n (\bar{A} \cdot A^*)_{uu} \\ &= -\sum_{u=1}^n \sum_{r=1}^n \bar{A}_{ur} A_{ru}^* \end{aligned}$$

Since  $w_{n1}$  is the  $(n,1)$ th entry of branch transmittance matrix  $\tilde{Q}$  and  $\tilde{A} = I_n - \tilde{Q}^T$ , then  $\tilde{A}_{1n} = -w_{n1}$ . By equation 2.13, the only non-zero entry in matrix  $A^*$  is  $A_{ru}^* = -1$  when  $r = 1$  and  $u = n$ , otherwise it is zero. Therefore

$$-\text{tr}[\bar{A} \cdot A^*] = -\bar{A}_{n1} \cdot (-1) = \bar{A}_{n1}$$

and equation 2.14 becomes

$$\begin{aligned} M_{1n}(s) &= \bar{A}_{n1} \\ &= [(\tilde{A} |_{w_{n1}=0})^{-1}]_{n1} \end{aligned}$$

Since  $\tilde{A} |_{w_{n1}=0}$  is just the coefficient matrix of a non-closed flowgraph  $A$  such that  $A = I_n - Q^T$ . Hence the MGF of waiting time from node 1 to node  $n$  in a non-closed flowgraph is the  $(n,1)$ th entry of its inverse coefficient matrix  $A$

$$M_{1n}(s) = [(A)^{-1}]_{n1}$$

□

**Corollary 1.** *Given a finite non-closed flowgraph with input at node 1 and output at node  $n$ , the  $r$ -th population moment  $\mu_r$  can be determined by*

$$\mu_r = -\text{tr} \left[ \frac{\partial \tilde{A}}{\partial w_{n1}} \cdot \frac{\partial^r \bar{A}}{\partial s^r} \right] \Big|_{s=0} \quad (2.15)$$

where  $\bar{A} = (\tilde{A} |_{w_{n1}=0})^{-1}$ , and  $\tilde{A} = I_n - \tilde{Q}^T$  is the coefficient matrix of the closed flowgraph constructed by adding a directed edge associated with transmittance  $w_{n1}$  that connects output node  $n$  to input node 1.

*Proof.* According to equation 2.14, we have

$$\begin{aligned} \frac{\partial}{\partial s} M(s) &= -\frac{\partial}{\partial s} \left\{ \text{tr} \left[ (\tilde{A} |_{w_{n1}=0})^{-1} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \right\} \\ &= -\text{tr} \left\{ \frac{\partial}{\partial s} \left[ (\tilde{A} |_{w_{n1}=0})^{-1} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \right\} \end{aligned}$$

where it leads to the following equation after applying the product rule,

$$\begin{aligned} &\frac{\partial}{\partial s} \left[ (\tilde{A} |_{w_{n1}=0})^{-1} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \\ &= \frac{\partial (\tilde{A} |_{w_{n1}=0})^{-1}}{\partial s} \cdot \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) + (\tilde{A} |_{w_{n1}=0})^{-1} \cdot \frac{\partial}{\partial s} \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \end{aligned} \quad (2.16)$$

By equation 2.13,  $\frac{\partial \tilde{A}}{\partial w_{n1}}$  is a constant matrix, which does not involve variable  $s$ , then

$$\frac{\partial}{\partial s} \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) = 0 \quad (2.17)$$

Substitute equation 2.17 into 2.16 and let  $\bar{A} = (\tilde{A} |_{w_{n1}=0})^{-1}$  then

$$\begin{aligned} \frac{\partial}{\partial s} M(s) &= -\text{tr} \left[ \frac{\partial (\tilde{A} |_{w_{n1}=0})^{-1}}{\partial s} \cdot \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \right] \\ &= -\text{tr} \left[ \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \end{aligned}$$

The first population  $\mu_1$  is

$$\mu_1 = \left[ \frac{\partial}{\partial s} M(s) \right] \Big|_{s=0} = -\text{tr} \left[ \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right] \Big|_{s=0}$$

where  $\bar{A} = (\tilde{A} |_{w_{n1}=0})^{-1}$ .

For the second population moment,

$$\begin{aligned} \frac{\partial^2}{\partial s^2} M(s) &= \frac{\partial}{\partial s} \left[ -\text{tr} \left( \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \right] \\ &= -\text{tr} \left[ \frac{\partial}{\partial s} \left( \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \right] \end{aligned}$$

By the product rule again,

$$\frac{\partial}{\partial s} \left( \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right) = \frac{\partial^2 \bar{A}}{\partial s^2} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} + \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial}{\partial s} \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \quad (2.18)$$

By equation 2.17, the second term on the left hand side of equation 2.18 is

$$\frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial}{\partial s} \left( \frac{\partial \tilde{A}}{\partial w_{n1}} \right) = \frac{\partial \bar{A}}{\partial s} \cdot 0 = 0$$

Then, equation 2.18 becomes

$$\frac{\partial}{\partial s} \left( \frac{\partial \bar{A}}{\partial s} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right) = \frac{\partial^2 \bar{A}}{\partial s^2} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}}$$

The second population moment is therefore determined as

$$\mu_2 = \left[ \frac{\partial^2}{\partial s^2} M(s) \right] \Big|_{s=0} = \left[ -\text{tr} \left( \frac{\partial^2 \bar{A}}{\partial s^2} \cdot \frac{\partial \tilde{A}}{\partial w_{n1}} \right) \right] \Big|_{s=0}$$

By iteration, we obtain a formula for computing the  $r$ -th population moment

$$\begin{aligned}
 \mu_r &= \left[ \frac{\partial^r}{\partial s^r} M(s) \right] \Big|_{s=0} \\
 &= \left\{ -\operatorname{tr} \left[ \frac{\partial \tilde{A}}{\partial w_{n1}} \cdot \frac{\partial^r \bar{A}}{\partial s^r} \right] \right\} \Big|_{s=0} \\
 &= \left\{ -\operatorname{tr} \left[ \frac{\partial \tilde{A}}{\partial w_{n1}} \cdot \frac{\partial^r}{\partial s^r} \left( \tilde{A}|_{w_{n1}=0} \right)^{-1} \right] \right\} \Big|_{s=0}
 \end{aligned}$$

□

To summarise, the MGF of the total waiting time is not easy to determine in a complicated flowgraph by Mason's rules, particularly for a flowgraph that has a larger number of different order feedback loops, where we must be extra careful to count the paths and feedback loops. Theorem 2 and 3 provide an alternative way to compute the MGF based on the branch transmittance matrix of flowgraph, which allow us to avoid finding a list of all the paths and feedback loops between the input and output node. Furthermore, formula 2.15 given in Corollary 1 makes the calculation of population moments very simple, and we will discuss more about the application of formula 2.15 to the Method of Moments for parameter estimation problem in Chapter 4.



### 2.3.1 Example 1.1

In this section, we apply Theorem 2 to compute the MGF  $M_{02}(s)$  determined in equation 2.4 on page 17, Example 1. First the flowgraph is closed by adding an extra directed edge with transmittance  $w_{20}$  that connects output node 2 to input node 1 (see Figure 2.11). The branch transmittance matrix of the flowgraph becomes

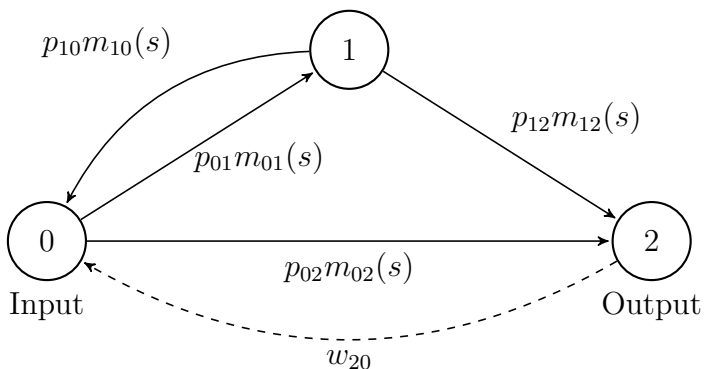


Figure 2.11: Closed flowgraph model for a reversible illness-death system

$$\tilde{Q} = \begin{pmatrix} 0 & q_{01} & q_{02} \\ q_{10} & 0 & q_{12} \\ w_{20} & 0 & 0 \end{pmatrix}$$

Define the coefficient matrix of flowgraph illustrated in Figure 2.11 as

$$\begin{aligned} \tilde{A} &= I_3 - \tilde{Q}^T \\ &= \begin{pmatrix} 1 & -q_{10} & -w_{20} \\ -q_{01} & 1 & 0 \\ -q_{02} & -q_{12} & 1 \end{pmatrix} \end{aligned}$$

we have

$$\det(\tilde{A}) = 1 - q_{01}q_{12}w_{20} - q_{01}q_{10} - q_{02}w_{20}$$

then

$$\frac{\partial \det(\tilde{A})}{\partial w_{20}} = -q_{01}q_{12} - q_{02}$$

$$\det(\tilde{A}|_{w_{20}=0}) = 1 - q_{01}q_{10}$$

By Theorem 2, the MGF of total waiting time between input node 0 and output node 2 is

$$\begin{aligned} M_{02}(s) &= -\frac{\frac{\partial \det(\tilde{A})}{\partial w_{20}}}{\det(\tilde{A}|_{w_{20}=0})} \\ &= \frac{q_{02}(s) + q_{01}(s)q_{12}(s)}{1 - q_{01}(s)q_{10}(s)} \end{aligned} \quad (2.19)$$

which gives the same expression of the MGF computed by Mason's rule in equation 2.4, Section 2.2.1.

### 2.3.2 Example 2.1

To avoid counting the paths and loops between input node 0 and output node 4 in Example 2 on page 18, we apply our formula in Theorem 3 to compute  $M_{02}(s)$  only based on the branch transmittance matrix  $Q$  such that

$$Q = \begin{pmatrix} 0 & q_{01} & q_{02} & 0 & 0 \\ q_{10} & 0 & q_{12} & 0 & q_{14} \\ 0 & 0 & 0 & q_{23} & q_{24} \\ 0 & 0 & q_{32} & 0 & q_{34} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The coefficient matrix of the flowgraph given in Figure 2.6 is

$$\begin{aligned}
 A &= I_5 - Q^T \\
 &= \begin{pmatrix} 1 & -q_{10} & 0 & 0 & 0 \\ -q_{01} & 1 & 0 & 0 & 0 \\ -q_{02} & -q_{12} & 1 & -q_{32} & 0 \\ 0 & 0 & -q_{23} & 1 & 0 \\ 0 & -q_{14} & -q_{24} & -q_{34} & 1 \end{pmatrix}
 \end{aligned}$$

By Theorem 3, the MGF of total waiting time between node 0 and node 4 is the (5,1)th entry of  $A^{-1}$

$$M_{04}(s) = \frac{q_{14}q_{01} - q_{14}q_{01}q_{23}q_{32} + q_{24}q_{01}q_{12} + q_{24}q_{02} + q_{34}q_{23}q_{01}q_{12} + q_{34}q_{23}q_{02}}{1 - q_{23}q_{32} - q_{01}q_{10} + q_{01}q_{10}q_{23}q_{32}} \quad (2.20)$$

If we want to determine the MGF of total waiting time between node 0 and node 2, we just need to look at the (3,1)th entry of  $A^{-1}$

$$(A^{-1})_{31} = \frac{q_{02} + q_{01}q_{12}}{1 - q_{23}q_{32} - q_{01}q_{10} + q_{01}q_{10}q_{23}q_{32}}$$

By our assumption, no further transition is allowed in the output node, then  $q_{23} = 0$ . Hence, the MGF of total waiting time between node 0 and node 2 is

$$M_{02}(s) = \frac{q_{02} + q_{01}q_{12}}{1 - q_{10}q_{01}}$$

which matches the result of Mason's rule in equation 2.4, Section 2.2.1.

Note that we do not need to compute the expressions 2.20 and 2.19 by hand, and in fact, all the calculation required in Theorem 2, 3 and Corollary 1 can be easily done by using symbolic algebra package MAPLE. However, we still have not obtained the total waiting time distribution. In Chapter 3 we will discuss how the distribution of total waiting time can be determined by the inversion of its MGF, and subsequently derive the corresponding CDF, PDF and survivor function.



# Chapter 3

## Distribution theory

Mason's rule allows us to compute the MGF of the total waiting time distribution between two nodes of interests in a finite flowgraph. In this chapter, the main purpose is to present methods that invert the MGF to obtain the density function and determine the distribution of the total waiting time  $T$ .

This chapter is organised as follow. The first section deals with the case where the internode waiting time follows exponential or Gamma distributions with integer-valued shape parameter (i.e. Erlang distribution). The second section considers flowgraph models with non-exponentially distributed internode waiting time, where direct inversion of the MGF is not possible. We discuss three different methods to approximate the total waiting time density function, namely, saddle-point approximation, the Maximum Entropy method and the Padé approximation method. We continue with the examples given in Chapter 2 and illustrate how to apply each method for estimating the probability density function and survivor function. The last second presents the comparison of these methods in both the exponential and non-exponential waiting time case.

### 3.1 Exact inversion of the MGF

Suppose we have a flowgraph with all the internode waiting times assumed to be exponentially distributed. The MGF of total waiting time in this case takes the form of a rational polynomial function. It can be decomposed by partial fraction method and then the inverse Laplace transform can be applied to obtain a closed form density function.

**Definition 6.** *The Laplace transform of a function  $f(t)$  for  $t > 0$  is defined by the following integral*

$$L[f](s) = \int_0^{\infty} e^{-st} f(t) dt \quad (3.1)$$

By comparing equation 3.1 to the definition of MGF on page 8 Chapter 2, we can see that  $L[f](s)$  is just the MGF with argument  $-s$  instead of  $s$ , so the inverse Laplace transform of MGF  $M_T(-s)$  is the probability density function of  $T$ . In the exponential waiting time case, the MGF is a rational function in the form of  $U(s)/P(s)$ , which can be written as the sum of rational functions by partial fractions. The probability density function can be determined by applying the inverse Laplace transform to each component of the partial fractions decomposition in  $M_T(-s)$ , and using the additive properties of the Laplace transform. This is summarised in the Heaviside method (see Dalla Valle (1931), Spiegel (1965)).

**Lemma 4.** *The Heaviside expansion formula*

*Let  $U(s)$  and  $R(s)$  be polynomials where  $U(s)$  has degree less than that of  $R(s)$ . Suppose  $R(s) = 0$  has  $n$  distinct roots  $\alpha_k$ ,  $k = 1, 2, 3, \dots, n$ , then the inverse Laplace transform of rational function  $\frac{U(s)}{R(s)}$  is*

$$L^{-1} \left[ \frac{U(s)}{R(s)} \right] = \sum_{k=1}^n \frac{U(\alpha_k)}{R'(\alpha_k)} e^{\alpha_k t} \quad (3.2)$$

where  $R'(\alpha_k) = \left. \frac{dR(s)}{ds} \right|_{s=\alpha_k}$ .

### 3.1.1 Example 1.3: Exponential waiting time

We return to the reversible illness-death system of Example 1 in Section 2.2.1. To determine the distribution of total waiting time between input at node 0 and output at node 2 in Figure 2.4, we assume all the internode waiting time are exponentially distributed (see Table 3.1). The patient in node 0 could transition to node 1 with probability  $p_{01}$  or die with probability  $p_{02} = 1 - p_{01}$ . The patient in node 1 may recover and return to node 0 with probability  $p_{10}$ , or die from the disease and transition to the output at node 2 with probability  $p_{12} = 1 - p_{10}$ .

Let  $T_{02}$  denote the total waiting time between input at node 0 and output at node 2, we show how to compute the probability density function of waiting time  $T_{02}$  by the exact inversion of its MGF,  $M_{T_{02}}(s)$ . For the purpose of illustration, we assume

1.  $\lambda_1 = 1, \lambda_2 = 1.2, \lambda_3 = 0.5, \lambda_4 = 2$ .
2. All the transition probability are equal to  $\frac{1}{2}$ , i.e.  $p_{01} = p_{02} = p_{10} = p_{12} = \frac{1}{2}$

Table 3.1: Summary of waiting time distribution

Direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$0 \rightarrow 2$	Exponential( $\lambda_3$ )	$m_{02}(s) = \frac{\lambda_3}{\lambda_3 - s}$
$1 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{12}(s) = \frac{\lambda_2}{\lambda_2 - s}$
$1 \rightarrow 0$	Exponential( $\lambda_4$ )	$m_{10}(s) = \frac{\lambda_4}{\lambda_4 - s}$

We identify two paths  $0 \rightarrow 2$ ,  $0 \rightarrow 1 \rightarrow 2$  and one feedback loop  $0 \rightarrow 1 \rightarrow 0$  between node 0 and node 2. By Mason's rule, the MGF of the total waiting time distribution from node 0 to node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\
 &= \frac{\frac{1}{4}m_{01}(s)m_{12}(s) + \frac{1}{2}m_{02}(s)}{1 - \frac{1}{4}m_{01}(s)m_{10}(s)} \\
 &= \frac{(2-s)(1.8 - 3.4s + s^2)}{(6 - 12s + 4s^2)(0.5 - s)(1.2 - s)} \tag{3.3}
 \end{aligned}$$

By Definition 6, the Laplace transform is just the MGF with argument  $-s$  instead of  $s$ , then

$$M_{T_{02}}(-s) = \frac{U(s)}{R(s)} = \frac{(2+s)(1.8 + 3.4s + s^2)}{(6 + 12s + 4s^2)(0.5 + s)(1.2 + s)}$$

solve

$$R(s) = (6 + 12s + 4s^2)(0.5 + s)(1.2 + s) = 0$$

The distinct roots are

$$\alpha_1 = -\frac{1}{2}, \alpha_2 = -\frac{6}{5}, \alpha_3 = -\frac{1}{2}(3 - \sqrt{3}), \alpha_4 = -\frac{1}{2}(3 + \sqrt{3}) \tag{3.4}$$

We apply the Heaviside expansion formula in Lemma 4 for the exact inversion of equation 3.3

$$L^{-1} [M_{T_{02}}(-s)] = L^{-1} \left[ \frac{U(s)}{R(s)} \right] = \sum_{k=1}^4 \frac{U(\alpha_k)}{R'(\alpha_k)} e^{\alpha_k t}$$

where

$$\begin{aligned}
 U(s) &= (6 + 12s + 4s^2)(0.5 + s)(1.2 + s) \\
 R'(s) &= \frac{d}{ds} R(s) = \frac{87}{5} + \frac{288}{5}s + \frac{282}{5}s^2 + 16s^3
 \end{aligned}$$



Thus, the PDF of total waiting time  $T_{02}$  from node 0 to node 2 is

$$\begin{aligned}
 f_{T_{02}}(t) &= 0.75e^{\alpha_1 t} - 0.3636e^{\alpha_2 t} - 0.1207e^{\alpha_3 t} - 0.0157e^{\alpha_4 t} \\
 &= \frac{0.75}{\alpha_1} \alpha_1 e^{\alpha_1 t} - \frac{0.3636}{\alpha_2} \alpha_2 e^{\alpha_2 t} - \frac{0.1207}{\alpha_3} \alpha_3 e^{\alpha_3 t} - \frac{0.0157}{\alpha_4} \alpha_4 e^{\alpha_4 t} \\
 &= 1.5f_1(t) - 0.303f_2(t) - 0.1904f_3(t) - 0.0066f_4(t) \\
 &= \sum_{i=1}^4 w_i f_i(t)
 \end{aligned} \tag{3.5}$$

where  $\alpha$ 's are given in equation 3.4, and  $\int_0^\infty f_{T_{02}}(t) = 1$  is verified. Since  $\sum_{i=1}^4 w_i = 1$ , the result shows that the total waiting time distribution is a mixture of exponential distribution with negative weight for some component.

The survival function is

$$\begin{aligned}
 S_{T_{02}}(t) &= P(T_{02} > t) \\
 &= 1.5e^{\alpha_1 t} - 0.3030e^{\alpha_2 t} - 0.1904e^{\alpha_3 t} - 0.0066e^{\alpha_4 t}
 \end{aligned} \tag{3.6}$$

Note that the numerical values in 3.5 and 3.6 are only decimal approximations.

The total waiting time distribution describes the time to occurrence of certain events, where the underlying process goes through a set of nodes till termination at the output node. This can be considered as the so-called phase type distribution, which is defined as the probability distribution of the time until absorption in a Markov process with a finite number of transient states and one absorbing state (see Neuts (1981)). Asmussen (1987) gives a useful introduction to the phase type distribution, and Aalen (1995) discusses the application of phase type distributions in survival analysis.

## 3.2 Numerical approximation method

The aim of this section is to introduce three different methods for approximating the waiting distribution between the input and output in a flowgraph, when the internode waiting time does not follow the exponential distribution or Gamma distributions with integer shape parameter (i.e.  $\alpha$  in  $\text{Gamma}(\alpha, \beta)$ ). In such cases, the MGF is not a rational form, and it therefore can't be inverted directly to obtain the probability density function. This motivates the use of approximation methods to estimate density functions by inverting the given MGF numerically.

### 3.2.1 The Maximum Entropy method

In a complicated flowgraph with large number of parameters, it is often difficult to determine the distribution from the given observations of total waiting time data between two nodes of interest. As we can compute the moments based on the MGF of total waiting time obtained by the Mason's rule, this forms our motivation to apply the the maximum entropy method, where we assume certain moment constraints of the total waiting time random variable and then maximise the entropy of the target density function subject to these moment constraints.

The maximum entropy method was first introduced by Jaynes (1957), where he discusses the link between statistical mechanics and information theory. Dawid and Grünwald (2004) generalise this method to apply to arbitrary decision problems and loss functions, while the applications of maximum entropy method to process information in the form of observed data and moment constraints is given in Giffin and Caticha (2007). Wagner (1995) illustrates the use of the maximum entropy method for estimating density function of random variables based on its moments. It is shown that, under appropriate moment constraints, some of the well-know distributions in statistics are maximum entropy distributions. For example, the exponential distribution can be derived from the maximum entropy distribution of nonnegative random variable  $X$  under the constraints  $E(x) = \mu_1$ , while the normal distribution is a maximum entropy distribution satisfying  $E(x) = \mu_1$  and  $E(x^2) = \mu_2$ .

In general, the maximum entropy density  $\hat{f}(t)$  can be computed by maximising the Shannon entropy

$$S = - \int_0^{\infty} \hat{f}(t) \log \hat{f}(t) dt \quad (3.7)$$

subject to a set of the moment constrains:

$$\int_0^{\infty} \hat{f}(t) t^k dt = \mu_k \quad (3.8)$$

where  $\mu_k$  is the  $k$ -th noncentral moment of the density  $f(t)$  such that

$$\mu_k = \left\{ \frac{d^k}{ds^k} M_T(s) \right\} \Big|_{s=0}$$

for  $k = 0, 1, \dots, m$ . Note that  $M_T(s)$  is the MGF of  $T$  and  $\mu_0 = 1$ .

The form of maximum entropy density  $\hat{f}(t)$  is

$$\hat{f}(t) = \exp\left(-1 - \sum_{i=0}^m c_i t^i\right) \quad (3.9)$$

where  $c_0, c_1, \dots, c_m$  are determined so that 3.9 is a proper density function and satisfies all the  $m + 1$  moment constraints. Kagan, Linnik and Rao (1973) give an simple proof of 3.9 based on convexity argument. (see Theorem 13.2.1, p.409). In order to compute the parameters in 3.9, we need to construct a set of nonlinear equations by substituting 3.9 to moment constraint in 3.8, and solve the resulting  $m + 1$  nonlinear equations 3.12 for  $m + 1$  unknown constants  $c_i$ .

Define

$$g_k(c) = \int_0^{\infty} t^k \hat{f}(t) dt \quad (3.10)$$

$$= \int_0^{\infty} t^k \exp\left(-1 - \sum_{i=0}^m c_i t^i\right) dt \quad (3.11)$$

By equation 3.8,

$$g_k(c) = \mu_k \quad \text{for } k = 0, 1, \dots, m \quad (3.12)$$

We apply the Newton-Raphson method to solve the set of nonlinear equations  $g_0, \dots, g_m$  for  $c_0, \dots, c_m$ . The first step is to compute the first order Taylor expansion of  $g_k$  for  $k = 0, 1, \dots, m$  at the initial value  $c^0 = (c_0^0, c_1^0, \dots, c_m^0)$

$$g_k(c) = g_k(c^0) + \sum_{i=0}^m (c_i - c_i^0) \frac{dg_k(c_0)}{dc_i}$$

where

$$\begin{aligned} \frac{dg_k(c)}{dc_i} &= \int_0^\infty t^k \frac{d}{dc_i} \exp(-1 - \sum_{i=0}^m c_i t^i) \\ &= - \int_0^\infty t^{k+i} \exp(-1 - \sum_{i=0}^m c_i t^i) \\ &= -g_{k+i}(c) \end{aligned}$$

Then

$$g_k(c) = g_k(c^0) - \sum_{i=0}^m (c_i - c_i^0) g_{k+i}(c^0)$$

$$g_k(c^0) - \sum_{i=0}^m (c_i - c_i^0) g_{k+i}(c^0) = \mu_k$$

$$\sum_{i=0}^m (c_i - c_i^0) g_{k+i}(c^0) = g_k(c^0) - \mu_k \quad (3.13)$$

for  $k = 0, 1, \dots, m$ .

We can rewrite the set of equations 3.13 in matrix form.

$$G\delta = V$$

$$\text{where } G = \begin{bmatrix} g_0(c_0) & g_1(c_0) & \cdots & g_m(c_0) \\ g_1(c_0) & g_2(c_0) & \cdots & g_{m+1}(c_0) \\ \vdots & \vdots & \cdots & \vdots \\ g_m(c_0) & g_{m+1}(c_0) & \cdots & g_{2m}(c_0) \end{bmatrix}, \delta = \begin{bmatrix} c_0 - c_0^0 \\ c_1 - c_1^0 \\ \vdots \\ c_m - c_m^0 \end{bmatrix}, V = \begin{bmatrix} g_0(c_0) - \mu_0 \\ g_1(c_0) - \mu_1 \\ \vdots \\ g_m(c_0) - \mu_m \end{bmatrix}$$

### 3.2.1.1 Newton-Raphson method

Newton-Raphson method is a well known iterative procedure for finding approximations to the root of a real-valued function  $f(x)$ . The basic idea of this method is to start with an initial guess value for  $x_0$ , which is reasonably close to the true root of  $f(x)$ , the function is approximated by its tangent line, then we compute the  $x$ -axis intercept of this tangent line. The  $x$ -axis intercept will usually be a better approximation to the root of function than the original guess, and the method can be iterated.

In maximum entropy method, the Newton-Raphson method can be extended to multivariate case and solved systems of  $k$  (non-linear) equations  $f_i(\mathbf{x}) = 0$ ,  $i = 1, \dots, k$ , where  $f_i(\mathbf{x})$  is a function of a  $k$ -dimensional vector  $\mathbf{x} = (x_1, \dots, x_k)$ , whose Jacobian matrix is

$$J(x) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_k} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_k} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_k(\mathbf{x})}{\partial x_1} & \frac{\partial f_k(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_k(\mathbf{x})}{\partial x_k} \end{bmatrix}$$

Given suitable initial value  $\mathbf{x}_0$  and  $(J(\mathbf{x}))^{-1}$  exists, the method is iterated for  $n = 0, 1, \dots$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (J(\mathbf{x}_n))^{-1}f(\mathbf{x}_n)$$

until the stopping criterion in equation 3.14 is satisfied.

$$|\mathbf{x}_{n+1} - \mathbf{x}_n| < \delta \tag{3.14}$$

for a user-defined tolerance  $\delta > 0$ .

**Lemma 5.** Given a  $(m + 1) \times (m + 1)$  matrix  $G$  is defined as

$$G_{ki} = -\frac{dg_k(c)}{dc_i} = g_{k+i}(c)$$

where

$$g_k(c) = \int_0^\infty t^k \hat{f}(t) dt \quad (3.15)$$

and  $\hat{f}(t) = \exp(-1 - \sum_{i=0}^m c_i t^i)$  for  $k = 0, 1, \dots, m$ ,  $i = 0, 1, \dots, m$ . Then  $G$  is a positive definite matrix and  $G^{-1}$  exists.

*Proof.* Let  $c = (c_0, c_1, \dots, c_m)^T$  be any non-zeros vector. Then

$$\begin{aligned} c^T G c &= \begin{bmatrix} c_0 & c_1 & \dots & c_m \end{bmatrix} \begin{bmatrix} g_0 & g_1 & \dots & g_m \\ g_1 & g_2 & \dots & g_{m+1} \\ \vdots & \vdots & \dots & \vdots \\ g_m & g_{m+1} & \dots & g_{2m} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix} \\ &= c_0^2 g_0 + 2c_0 c_1 g_1 + (c_1^2 + 2c_0 c_1) g_2 + \dots + c_m^2 g_{2m} \end{aligned} \quad (3.16)$$

Substitute equation 3.15 in 3.16, then

$$\begin{aligned} c^T G c &= \int_0^\infty (c_0^2 + 2c_0 c_1 t + (c_1^2 + 2c_0 c_1) t^2 + \dots + c_m^2 t^{2m}) \hat{f}(t) dt \\ &= \int_0^\infty (c_0 + c_1 t + c_2 t^2 + \dots + c_m t^m)^2 \hat{f}(t) dt \end{aligned} \quad (3.17)$$

Since the integrand in 3.17 is always positive, the  $c^T G c > 0$ , therefore the matrix  $G$  is a positive definite matrix with  $\det(G) > 0$  and  $G^{-1}$  exists. □

By Lemma 5, the matrix  $G$  satisfies the non-singular condition (i.e. the inverse of  $G$  exists) and it guarantees the convergence of the Newton-Raphson method.

$$\begin{aligned} G(c^l)(c^{l+1} - c^l) &= V(c^l) \\ c^{l+1} &= c^l + G^{-1}(c^l)C(c^l) \end{aligned}$$

Given the initial value  $c^l$  for  $c$ , we solve this system iteratively for  $c = (c_0, ..c_m)$  until  $|c^{l+1} - c^l| \leq 0.00001$ . The estimated probability density function is obtained after we substitute the resulting  $c_0, ..c_m$  in equation 3.9.

### 3.2.1.2 Implementation

The Newton-Raphson method will usually converge if the initial value of  $c$  is close enough to the true value. As the dimension of  $c$  increases with the number of moment constraints, the result of maximum entropy density estimation becomes very sensitive to the choice of initial values for  $c$ . Wu (2003) shows evidence that the sequential updating method is more robust with respect to the choice of initial values, the basic idea is to compute the constant  $c$  sequentially by adding only one higher order moment constraint at each step. Our method is a special implementation based on the sequential updating method suggested by Wu (2003) to derive an efficient algorithm for the computation of high dimensional  $c$ .

Given the first few  $c_k$  can be relatively easy computed, we choose  $c'_{k+1} = 0$  for  $c_{k+1}$  and define  $c^0 = (c_1^0, c_2^0, ...c_k^0, 0)$  as initial value in Newton method. We start with  $k = 2$  case as there is closed form solution for  $c_0, c_1$ .

$$\int_0^{\infty} \exp(-1 - c_0 - c_1 t) dt = 1 \tag{3.18}$$

$$\int_0^{\infty} t \exp(-1 - c_0 - c_1 t) dt = \mu \tag{3.19}$$

solve equation 3.18 for  $c_1$  in terms of  $c_0$  gives

$$c_1 = \exp(-1 - c_0) \tag{3.20}$$

substitute equation 3.20 into equation 3.19,

$$\begin{aligned} \int_0^{\infty} t \exp(-1 - c_0) \exp(-c_1 t) dt &= \mu \\ \int_0^{\infty} t c_1 \exp(-c_1 t) dt &= \mu \end{aligned}$$

The integral is just the definition of expected value for  $\text{Exponential}(c_1)$ , then  $c_1 = \frac{1}{\mu}$ , and  $c_0 = -1 - \log(c_1) = \log(\mu) - 1$  by equation 3.20. In general, we use the following procedure to compute  $\mathbf{c}_{(m)} = (c_0, c_1, c_2 \dots c_m)$ .

Step 1: Start with  $\mathbf{c}_{(2)}^0 = (c_0^{(0)}, c_1^{(0)}, c_2^{(0)})$ , choose the initial value  $c_0^{(0)} = \log(\mu) - 1$ ,  $c_1^{(0)} = \frac{1}{\mu}$ , and  $c_2^{(0)} = 0$ , then solve the system of equations iteratively for  $\mathbf{c}_{(2)}^1 = (c_0^{(1)}, c_1^{(1)}, c_2^{(1)})$  until  $|\mathbf{c}_{(2)}^{l+1} - \mathbf{c}_{(2)}^l| \leq 10^{-5}$ .

Step 2: Set  $\mathbf{c}_{(3)}^0 = (c_{(2)}^1, c_3^0) = (c_0^{(1)}, c_1^{(1)}, c_2^{(1)}, c_3^{(0)})$ , where  $c_{(2)}^1$  is the result from step 1 and we choose  $c_3^{(0)} = 0$ . Solve the system of equations iteratively for  $\mathbf{c}_{(3)}^2 = (c_0^{(2)}, \dots, c_3^{(2)})$  until  $|\mathbf{c}_{(3)}^{l+1} - \mathbf{c}_{(3)}^l| \leq 10^{-5}$ .

...

Step  $m - 1$ : Set  $\mathbf{c}_{(m)}^0 = (c_{(m-1)}^{m-2}, c_m^0) = (c_0^{m-2}, c_1^{m-2}, \dots, c_{m-1}^{m-2}, c_m^{(0)})$ , set  $c_m^{(0)} = 0$ , then solve the system of equations iteratively for  $\mathbf{c}_{(m)}^{m-1} = (c_0^{(m-1)}, \dots, c_m^{(m-1)})$  until  $|\mathbf{c}_{(m)}^{l+1} - \mathbf{c}_{(m)}^l| \leq 10^{-5}$ .

The sequential update procedure is particularly useful when the number of moment constraints is larger. Instead of setting  $m+1$  initial values for all  $c_0, \dots, c_m$  at the same time, this procedure reduces the multi-dimension search to one dimension by choosing initial value for  $c_k$ ,  $k = 2, \dots, m$  sequentially. The procedure has been programmed in Matlab. Our result shows that the more moment constraints we have (i.e. the larger  $m$  is), the closer is the maximum entropy density  $\hat{f}(t)$  to the true density  $f(t)$ . However, the computational cost becomes very high to solve a larger set of equations if additional moments constraints are imposed, partially because the integrals in  $g(s)$  function become increasingly difficult to evaluate. As there is no formal method to determine the number of moment constraints, further research could be conducted to estimate the number of moments constrains are needed in order to achieve a required accuracy of approximation. For the trade off between the complexity of the maximum entropy density  $\hat{f}(t)$  and the accuracy of the estimation, we only impose 6 moment constraints in the following two examples.



### 3.2.1.3 Example 3

Our first example is to apply the maximum entropy method for estimating the probability density function of waiting time in a series flowgraph. We use the hydraulic pump system example in Section 2.1.2.1, Chapter 2. The series flowgraph in Figure 2.1 describes the status of a two pump systems, the input is at node 0 which represents the system is working properly with no pumps failed, node 1 represents one pump failed, and the output is node 2 when two pump failed and the system breaks down. The waiting time distributions between each node are given in Table 3.2. We assume the transition probability from  $0 \rightarrow 1$  and  $1 \rightarrow 2$  are all equal 1 (i.e.  $p_{01} = p_{10} = 1$ ), and  $\alpha = 3$ ,  $\beta = \frac{1}{2}$ ,  $c = 1$ .

Table 3.2: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $c$ )	$m_{01}(s) = \frac{c}{c-s}$
$1 \rightarrow 2$	Gamma( $\alpha, \beta$ )	$m_{12}(s) = \left(\frac{\beta}{\beta-s}\right)^\alpha$

The MGF of total waiting time  $T_{02}$  from node 0 to node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= p_{01}m_{01}(s)p_{12}m_{12}(s) \\
 &= \left(\frac{c}{c-s}\right) \left(\frac{\beta}{\beta-s}\right)^\alpha \\
 &= \frac{1}{8(1-s)(0.5-s)^3}
 \end{aligned}$$

By estimating the probability density function of total waiting time  $T_{02}$  subject to 6 moment constraints, we use this MGF to compute the value of the moment constraints  $\mu_r$ ,  $r = 0 \dots 5$ , where  $\mu_r = \left\{ \frac{d^r}{ds^r} M_{T_{02}}(s) \right\} \Big|_{s=0}$ , gives the result  $\mu_0 = 1$ ,  $\mu_1 = 7$ ,  $\mu_2 = 62$ ,  $\mu_3 = 666$ ,  $\mu_4 = 8424$ ,  $\mu_5 = 122760$ . By substituting the general form of the maximum entropy distribution  $\hat{f}(t)$  defined in equation 3.9 and  $\mu_r$   $r = 0, \dots, 5$ , to each moment constraint in equation 3.8, we form a system of 6 nonlinear equations with 6 unknown  $c_0, c_1 \dots, c_5$ .

$$\begin{bmatrix} g_0 & g_1 & \cdots & g_5 \\ g_1 & g_2 & \cdots & g_6 \\ \vdots & \vdots & \cdots & \vdots \\ g_5 & g_6 & \cdots & g_{10} \end{bmatrix} \begin{bmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \delta_5 \end{bmatrix} = \begin{bmatrix} g_0 - \mu_0 \\ g_1 - \mu_1 \\ \vdots \\ g_5 - \mu_5 \end{bmatrix}$$

where  $\delta_i = c_i - c_i^0, i = 0 \dots 5$ ;  $g_k = g_k(c^0), k = 0, \dots, 10$ . Given the current value of value  $c^0$  at each iteration, the integrals in  $g_k(c^0)$  for  $k = 0, \dots, 10$ , defined in equation 3.11, are computed numerically by using the adaptive Simpson quadrature method, see Gander and Gautschi (2000) for “quad” function in Matlab.

The system is solved by updating the value of  $c$  iteratively until  $|\mathbf{c}^{l+1} - \mathbf{c}^l| \leq 10^{-5}$ . The estimated probability density function  $\hat{f}(t)$  is given by

$$\hat{f}_{ME}(t) = \exp(-5.8309 + 1.9342t - 0.3429t^2 + 0.0262t^3 - 0.0009t^4 + 0.000014t^5)$$

Since the waiting time distribution between node 1 and node 2 is Gamma(3, 0.5) with integer shape parameter  $\alpha = 3$ , we can still apply the Heaviside expansion formula by Lemma 4 to invert  $M_{T_{02}}(s)$  for the true probability density function  $f(t)$  such that

$$f(t) = \left(\frac{1}{8}t^2 - \frac{1}{2}t + 1\right)e^{-\frac{1}{2}t} - e^{-t}$$

The comparison of maximum entropy density under different number of moment constraints with the true density (solid line) is presented in Figure 3.1. It is clearly shown that the quality of density approximation depends critically on the number of moments constraints that we imposed, and it can be improved dramatically as the number of moment constraints increases. The plot of maximum entropy density that obtained with 6 moments constraints provides relatively the best approximation to the shape of the true density function. (see Figure 3.2).

The performance of these maximum entropy densities are assessed by computing their integrated square errors (ISE).

$$ISE = \int_0^{\infty} \{f(t) - \hat{f}(t)\}^2 dt$$

where  $f(t)$  is the true density and  $\hat{f}(t)$  is the estimated density. Table 3.3 provides numerical evidence that the accuracy of density estimation is significantly improved by adding more moment constraints.

Table 3.3: Sum of square error(ISE)

Number of constraints	3	4	5	6
ISE	0.0477	0.0212	0.0199	0.0019

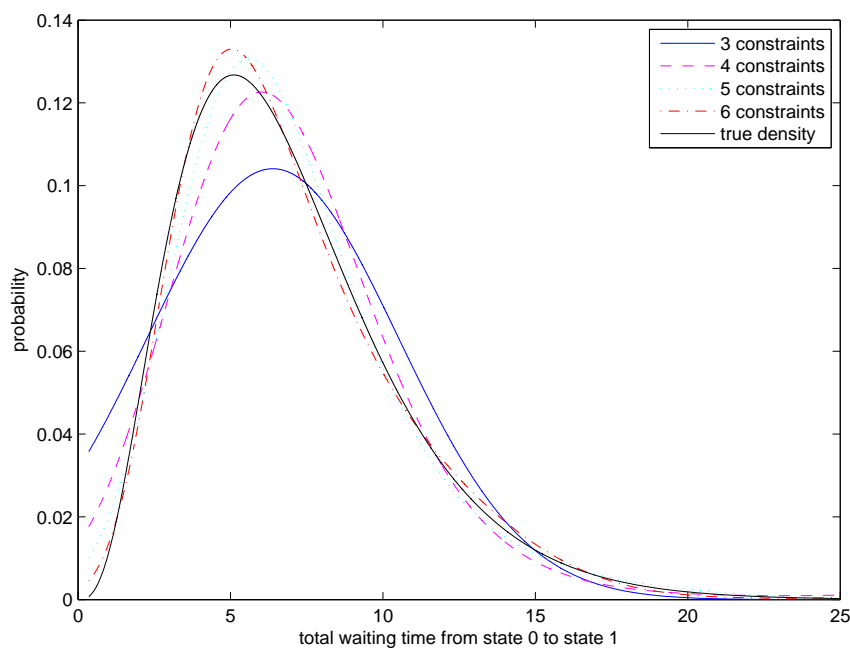


Figure 3.1: Density estimation by Maximum Entropy method.

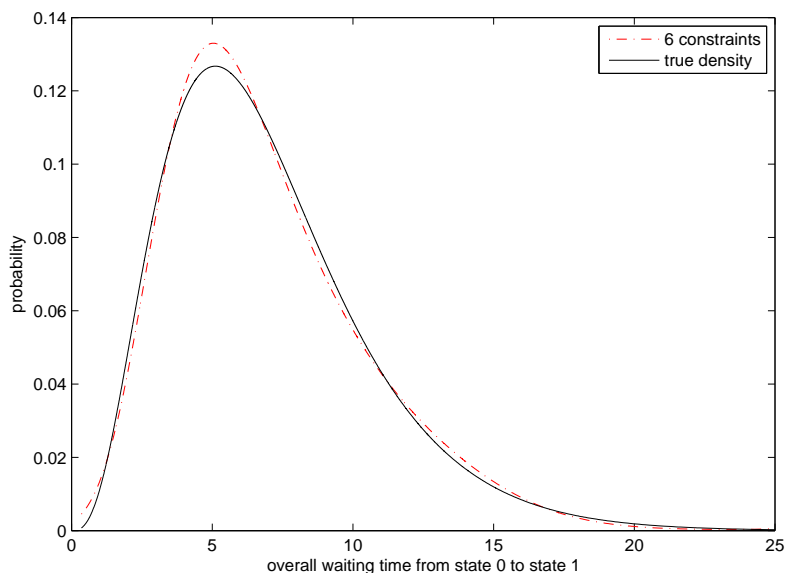


Figure 3.2: Solid line: True density; Dash line: Estimated density by Maximum Entropy with 6 moment constraints.

### 3.2.1.4 Example 1.4: Non-exponential waiting time

We consider the flowgraph model for reversible illness-death system again (see Figure 2.5 on page 17), but Gamma distributed waiting time is allowed (see Table 3.4). In the non-exponentially distributed internode waiting time case, the MGF of total waiting time can not be decomposed by partial fraction method, and we can not use the exact inversion method of MGF for a closed form true density. Maximum entropy method is then applied to estimate the density of total waiting time between node 0 and node 2 under 6 moment constraints.

Table 3.4: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	$\text{Gamma}(\alpha_1, \beta_1)$	$m_{01}(s) = \left(\frac{\beta_1}{\beta_1 - s}\right)^{\alpha_1}$
$0 \rightarrow 2$	$\text{Exponential}(\theta_1)$	$m_{02}(s) = \frac{\theta_1}{\theta_1 - s}$
$1 \rightarrow 2$	$\text{Gamma}(\alpha_2, \beta_2)$	$m_{12}(s) = \left(\frac{\beta_2}{\beta_2 - s}\right)^{\alpha_2}$
$1 \rightarrow 0$	$\text{Exponential}(\theta_2)$	$m_{10}(s) = \frac{\theta_2}{\theta_2 - s}$

For illustration we assume the following parameter values:  $p_{01} = 1 - p_{02} = 0.5$ ,  $p_{12} = p_{10} = 0.5$ ,  $\alpha_1 = 3.2$ ,  $\beta_1 = 4.8$ ,  $\alpha_2 = 2$ ,  $\beta_2 = 3$ ,  $\theta_1 = 2.2$ ,  $\theta_2 = 4$ . By Mason's rule, the MGF of overall waiting time distribution between node 0 and node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\ &= \frac{\frac{1}{4}\left(\frac{4.8}{4.8-s}\right)^{3.2}\left(\frac{3}{3-s}\right)^2 + \frac{1}{2}\left(\frac{2.2}{2.2-s}\right)}{1 - \frac{1}{4}\left(\frac{4.8}{4.8-s}\right)^{3.2}\left(\frac{4}{4-s}\right)} \end{aligned}$$

By using the sequential update procedure, we obtain  $c_0 = -0.9670$ ,  $c_1 = 1.6088$ ,  $c_2 = -1.0621$ ,  $c_3 = 0.4775$ ,  $c_4 = -0.0785$ ,  $c_5 = 0.0043$ . The resulting estimated density function  $\hat{f}(t)$  is

$$\begin{aligned} \hat{f}_{ME}(t) &= \exp(-1 - c_0 - c_1t - c_2t^2 - c_3t^3 - c_4t^4 - c_5t^5) \\ &= \exp(-0.033 - 1.6088t + 1.0621t^2 - 0.4775t^3 + 0.0785t^4 - 0.0043t^5) \end{aligned}$$

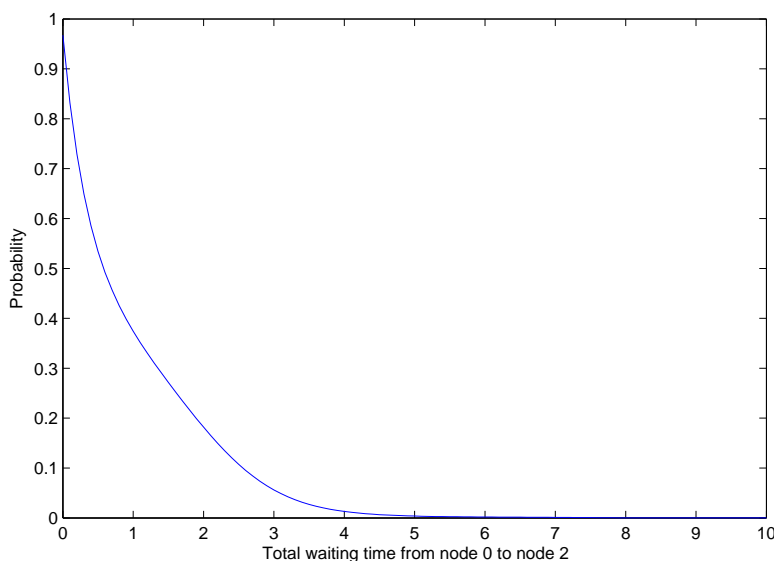


Figure 3.3: Density estimation by Maximum Entropy method under 6 moment constraints in Example 1.4.

### 3.2.2 Padé approximation

Padé approximation is the method for approximating a function by a rational function of given order, where the coefficients of its numerator and denominator match the coefficients in the Taylor series expansion of given function. The most important reason to use Padé approximation of MGF in flowgraph analysis is because it allows us to approximate the MGF as the ratio of two polynomial functions, which can be easily inverted to obtain a closed form estimation for the probability density function by Lemma 4 in Section 3.1.

Amindavar and Ritcey (1994) first introduced the Padé approximation approach to estimate probability density functions from moments. It is shown that the PDF can be easily determined as a series of exponential function, where rates are computed as the poles of Padé approximation of MGF (i.e. the singularities of  $PA_{[p,q]}(s)$ ). A comprehensive reference for the background theory and the application of Padé approximation can be found in Baker and Graves-Morris (1996), where the derivation of 3.21 is discussed.

**Definition 7.** *Given a formal Taylor series expansion of a function  $f(x)$ ,*

$$f(x) = \sum_{i=0}^{\infty} c_i x^i$$

*The Padé approximation of  $f(x)$ ,  $PA_{[p,q]}$ , is a rational function with numerator order  $p$  and denominator order  $q$  in the form of:*

$$PA_{[p,q]}(s) = \frac{\sum_{j=0}^p a_j s^j}{\sum_{k=0}^q b_k s^k} = \frac{a_0 + a_1 s + \dots + a_p s^p}{b_0 + b_1 s + \dots + b_q s^q} \quad (3.21)$$

*such that*

$$f(x) = PA_{[p,q]}(x) + O(x^{p+q+1})$$

Let  $M_X(s)$  be the MGF of a random variable  $X$  and write the Taylor expansion of  $M_X(s)$  at  $s = 0$  as

$$M_X(s) = \sum_{n=0}^{\infty} \frac{\mu_n}{n!} s^n$$

where  $\mu_n = \int_0^{\infty} x^n f(x) dx$  is the  $n$ -th noncentral moment of  $f(x)$ . By setting  $p < q$ , the Padé approximation of MGF,  $\text{PA}_{[p,q]}(s)$ , can be decomposed by partial fraction and then easily inverted to obtain a closed form density estimation by Lemma 4. The coefficients  $a_j$  and  $b_k$  are computed by equating  $\text{PA}_{[p,q]}(s)$  with the power series of MGF  $M_X(s)$ ,

$$\frac{\sum_{j=0}^p a_j s^j}{\sum_{k=0}^q b_k s^k} = \sum_{n=0}^{p+q} c_n s^n + O(s^{p+q+1}) \quad (3.22)$$

where  $c_n = \frac{\mu_n}{n!}$ . By multiplying  $\sum_{k=0}^q b_k s^k$  on both side of equation 3.22

$$\sum_{j=0}^p a_j s^j = \left( \sum_{k=0}^q b_k s^k \right) \left( \sum_{n=0}^{p+q} c_n s^n \right) + O(s^{p+q+1}) \quad (3.23)$$

We equate the coefficients of  $s^{p+1}, s^{p+2}, \dots, s^{p+q}$  on both side of 3.23 and construct a system of  $q$  equations with  $q$  unknown denominator coefficients  $b_1, b_2, \dots, b_q$ .

$$\begin{aligned} b_q c_{p-q+1} + b_{q-1} c_{p-q+2} + \dots + b_0 c_{p+1} &= 0 \\ \vdots & \\ b_q c_p + b_{q-1} c_{p+1} + \dots + b_0 c_{p+q} &= 0 \end{aligned}$$

In matrix form

$$HB = C \quad (3.24)$$

$$\text{where } H = \begin{bmatrix} c_{p-q+1} & c_{p-q+2} & c_{p-q+3} & \dots & c_p \\ c_{p-q+2} & c_{p-q+3} & c_{p-q+4} & \dots & c_{p+1} \\ c_{p-q+3} & c_{p-q+4} & c_{p-q+5} & \dots & c_{p+2} \\ \vdots & \vdots & \dots & \vdots & \\ c_p & c_{p+1} & c_{p+2} & \dots & c_{p+q-1} \end{bmatrix}, B = \begin{bmatrix} b_q \\ b_{q-1} \\ b_{q-2} \\ \vdots \\ b_1 \end{bmatrix}, C = \begin{bmatrix} -c_{p+1} \\ -c_{p+2} \\ -c_{p+3} \\ \vdots \\ -c_{p+q} \end{bmatrix}.$$

Set  $b_0 = 1$ , we solve a set of equations in 3.24 for the  $b_1, b_2, \dots, b_q$ . Given the resulting  $b_k, k = 1, 2, \dots, q$ , the numerator coefficient  $a_0, a_1, \dots, a_p$  can be computed by matching the coefficient of  $1, s, s^2, \dots, s^p$  on both side of equation 3.23:

$$\begin{aligned} a_0 &= c_0 \\ a_1 &= c_1 + b_1 c_0 \\ a_2 &= c_2 + b_1 c_1 + b_2 c_0 \\ &\vdots \\ a_p &= c_p + \sum_{n=1}^{\min(p,q)} b_n c_{p-n} \end{aligned}$$

Since we can't solve the system of equation  $HB = C$  if the determinant of  $q \times q$  square matrix H is zero (i.e.  $H^{-1}$  doesn't exist), the appropriate order of Padé approximation should guarantee that the square matrix H has full rank. We follow the method suggested by Amindavar and Ritcey (1994) to define a Padé approximation  $PA_{[q-1,q]}(s)$ , such that the order of numerator is 1 degree smaller than the order of denominator, compute the  $q \times q$  square matrix H, and plot the rank of H matrix against  $q$  ( see Figure 3.4). The optimal order of denominator in the Padé approximation is denoted by  $q^*$ , the largest integer that matrix H has full rank. The MGF is then approximated by the Padé approximation  $PA_{[q^*-1,q^*]}(s)$ .

### 3.2.2.1 Example 4

We use the Padé approach to estimate the density function of total waiting time from node 0 to node 2 in Section 3.1.1 on page 41 again. By equation 3.3, the MGF of total waiting time from node 0 to node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= \frac{(2-s)(1.8-3.4s+s^2)}{(6-12s+4s^2)(0.5-s)(1.2-s)} \\ &= \frac{1-2.39s+1.5s^2-0.28s^3}{1-4.83s+8s^2-5.22s^3+1.11s^4} \end{aligned} \tag{3.25}$$



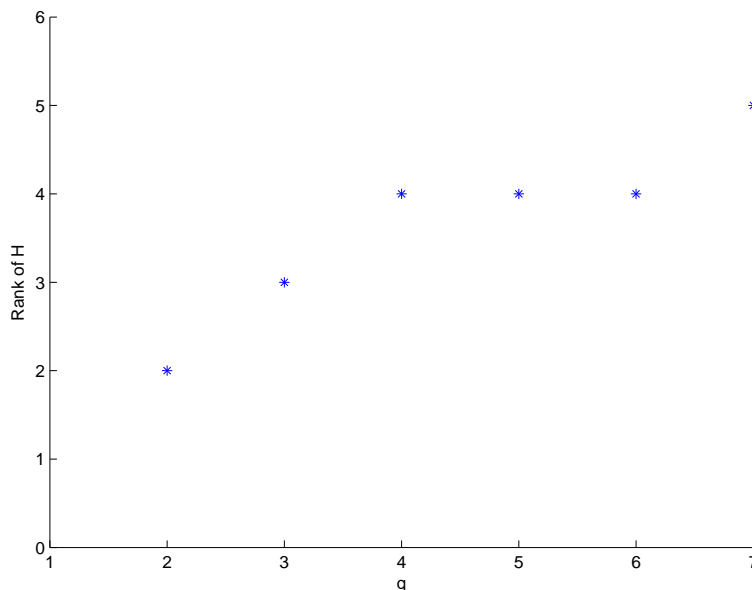


Figure 3.4: Rank check for matrix H

The optimal order of the Padé approximation is then determined by plotting the rank of square matrix H defined in equation 3.24. Figure 3.4 shows how the rank of matrix H changes as we choose different order of denominator  $q$  in  $PA_{[q-1,q]}(s)$ . As can be seen that the  $q \times q$  square matrix H becomes rank deficient if  $q \geq 5$ , and we therefore choose  $q = 4$  as the optimal order of the denominator in Padé approximation. The coefficient of  $PA_{[3,4]}(s)$  can be easily computed by using MAPLE.

$$PA_{[3,4]}(s) = \frac{1 - 2.3889s + 1.5001s^2 - 0.2778s^3}{1 - 4.8334s + 8.0003s^2 - 5.2227s^3 + 1.1113s^4} \quad (3.26)$$

By comparing the coefficient in 3.26 with 3.25,  $PA_{[3,4]}(s)$  gives a very good approximation of  $M_{T_{02}}(s)$ . We apply the Heaviside formula to invert  $PA_{[3,4]}(-s)$  and obtain the approximated probability density function of  $T_{02}$  as

$$\begin{aligned} f_{PA}(t) = & 0.75 \exp(-0.5t) - 0.3636 \exp(-1.1999t) \\ & - 0.12067 \exp(-0.6339t) - 0.01570 \exp(-2.3656t) \end{aligned} \quad (3.27)$$

Note that the form of finite mixtures of exponential distribution is much more stable than the maximum entropy density defined in equation 3.9 on page 45. The approximated survival function of  $T_{02}$  is

$$\begin{aligned} S_{PA}(t) &= P(T \geq t) \\ &= 1.5 \exp(-0.5t) - 0.3030 \exp(-1.1999t) \\ &\quad - 0.1903 \exp(-0.6339t) - 0.0066 \exp(-2.3656t) \end{aligned} \tag{3.28}$$

### 3.2.3 The Saddlepoint approximation: a discussion

Saddlepoint approximation are powerful tools for numerically inverting a MGF to estimate density and distribution function. It is originally developed by Daniels (1954) to approximate the probability density function of the mean of i.i.d random variable. For the literature review, see Daniels (1954), Jensen (1995), Barndorff-Nielsen and Cox (1997). Lugannani and Rice (1980) introduces the saddlepoint approximation for estimating the cumulative distribution function. A rather insightful discussion about the basic idea behind the derivation of saddlepoint approximation is given by Casella and Goutis (1999). The saddlepoint approximation has widespread applicability in many fields, including survival analysis (Butler and Huzurbazar (1997), Huzurbazar (1999)), system reliability (Butler (2000)), portfolio credit risk (Gordy, (2002)), insurance ruin problem (Gatto (2008)), and option pricing (Carr and Madan (2009)).

In view of the rising popularity of saddlepoint approximation in flowgraph model, this method play a key role in estimating the distribution of total waiting time random variable whose MGF is known, and it is important to thoroughly understand the proof before we propose our Padé-type saddlepoint approximation in Chapter 5. Since the original proof given by Daniels (1954) is difficult to follow and we are not aware of a complete detailed proof in the literature, this forms our motivation to give a comprehensive explanation of how the saddlepoint approximation is derived based on exponential tilting, edgeworth expansions, and Hermite polynomials that discussed in Kolassa (1994). First, we give the basic definition of saddlepoint approximation.

**Theorem 4.** *Saddlepoint approximation*

Let  $X_1, X_2, \dots, X_n$  be i.i.d random variables with density  $f(x)$ , and assume the moment generating function  $M_X(s)$ , exists in an open interval about zero. Denote the corresponding cumulant generating function as  $K_X(s) = \log(M_X(s))$ . The the saddlepoint approximation for the probability density function of  $\bar{X}$  is

$$\hat{f}_{\bar{X}}(x) = \left( \frac{n}{2\pi K_X''(\hat{s})} \right)^{1/2} \exp \left\{ n(K_X(\hat{s}) - \hat{s}\bar{x}) \right\} \quad (3.29)$$

where  $K_X''(s) = \frac{d^2 K_X(s)}{ds^2}$ , and  $K_X'(\hat{s}) = \bar{x}$

Note that the formula given in equation 3.29 is applied to determine the distribution of the sample mean  $\bar{X}$ . In the rest of this section, we present our own proof of Theorem 4 based on Edgeworth expansion, and discuss the accuracy of approximation by showing the “order of the approximation error”.

*Proof.* Let  $X_1, X_2, \dots, X_n$  be i.i.d random variable with mean  $\mu$  and finite variance  $\sigma^2$ . Define

$$Y = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

By the central limit theorem,  $Y$  is asymptotically Normal  $N(0,1)$  and its the characteristic function,  $\chi_y(t)$ , converges to the characteristic function of  $N(0,1)$

$$\chi_y(t) = E[e^{ity}] = \int_{-\infty}^{+\infty} e^{ity} f_Y(y) dy \longrightarrow e^{-t^2/2} \quad \text{as } n \rightarrow \infty$$

Since

$$\begin{aligned} \chi_y(t) &= \exp\left(-\frac{1}{2}t^2\right) \\ &= \left\{ \exp\left(-\frac{1}{2}\frac{t^2}{n}\right) \right\}^n \\ &= \left\{ \chi\left(\frac{t}{\sqrt{n}}\right) \right\}^n \end{aligned} \quad (3.30)$$

where  $\chi$  is the characteristic function of  $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$ .

It was decided to take a detour to explain the saddlepoint approximation method. The basic idea is to approximate the characteristic function  $\chi_y(t)$  by keeping the first few terms of the Taylor expansion of  $\chi_y(t)$  at 0, and invert each term of the approximation of  $\chi_y(t)$  for the distribution function of  $Y$ .

Let  $L(t) = \log(\chi(t))$ , we first expand  $L(t)$  as a Taylor series at  $t = 0$ ,

$$\begin{aligned} \log(\chi(t)) &= L(0) + L^{(1)}(0)t + \frac{L^{(2)}(0)}{2!}t^2 + \frac{L^{(3)}(0)}{3!}t^3 + \frac{L^{(4)}(0)}{4!}t^4 \\ &\quad + \frac{L^{(5)}(0)}{5!}t^5 + \dots + \frac{L^{(j)}(0)}{j!}t^j \end{aligned} \quad (3.31)$$

where

$$\frac{L^{(j)}(0)}{j!}t^j = \frac{1}{j!} \left( \frac{d^j}{dt^j} \log(\chi(t)) \right) \Big|_{t=0} t^j \quad (3.32)$$

As  $\chi(t)$  is the characteristic function of standard normal random variable  $Z$ , then we have

$$\left( \frac{d^j}{dt^j} \log(\chi(t)) \right) \Big|_{t=0} = i^j E(Z^j) \quad (3.33)$$

Let  $K(t) = \log(M_z(t))$  be the cumulant generating function of  $Z \sim N(0, 1)$ , then

$$\begin{aligned} \left( \frac{d^j}{dt^j} K(t) \right) \Big|_{t=0} &= \left( \frac{d^j}{dt^j} \log(M_z(t)) \right) \Big|_{t=0} \\ &= E(z^j) \end{aligned} \quad (3.34)$$

Since the  $j$ -th cumulant of  $Z$  is  $\kappa_j = \left( \frac{d^j}{dt^j} K(t) \right) \Big|_{t=0}$ , we have  $\kappa_j = E(z^j)$  by 3.34, and equation 3.33 becomes

$$\left( \frac{d^j}{dt^j} \log(\chi(t)) \right) \Big|_{t=0} = i^j \kappa_j \quad (3.35)$$

Substitute 3.35 in 3.32 yields

$$\frac{L^{(j)}(0)}{j!}t^j = \frac{(it)^j \kappa_j}{j!} \quad (3.36)$$

Note that  $L(0) = \log(\chi(0)) = 0$ . By substituting result 3.36 in 3.31, we have

$$\log(\chi(t)) = (it)\kappa_1 + \frac{(it)^2\kappa_2}{2!} + \frac{(it)^3\kappa_3}{3!} + \frac{(it)^4\kappa_4}{4!} + \frac{(it)^5\kappa_5}{5!} + \dots + \frac{(it)^j\kappa_j}{j!} \quad (3.37)$$

$$\chi(t) = \exp\left((it)\kappa_1 + \frac{(it)^2\kappa_2}{2!} + \frac{(it)^3\kappa_3}{3!} + \frac{(it)^4\kappa_4}{4!} + \frac{(it)^5\kappa_5}{5!} + \dots + \frac{(it)^j\kappa_j}{j!}\right) \quad (3.38)$$

To determine the  $\kappa_j$ , we need to compute the Taylor expansion of  $\chi(t)$  at  $t = 0$ . By the use of property  $\chi^{(j)}(0) = i^j E(Z^j)$ , we have

$$\begin{aligned} \chi(t) &= \chi(0) + \chi^{(1)}(0)t + \frac{\chi^{(2)}(0)t^2}{2!} + \frac{\chi^{(3)}(0)t^3}{3!} + \dots + \frac{\chi^{(j)}(0)t^j}{j!} \\ &= 1 + E(z)it + \frac{E(z^2)(it)^2}{2!} + \frac{E(z^3)(it)^3}{3!} + \dots + \frac{E(z^j)(it)^j}{j!} \end{aligned} \quad (3.39)$$

Taking logarithm on both side of equation 3.39

$$\log(\chi(t)) = \log\left(1 + E(z)it + \frac{E(z^2)(it)^2}{2!} + \frac{E(z^3)(it)^3}{3!} + \dots + \frac{E(z^j)(it)^j}{j!}\right)$$

By applying  $\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \dots$ , for  $-1 < x \leq 1$ .

$$\log(\chi(t)) = \sum_{v \geq 1} \left\{ \frac{(-1)^{v+1}}{v} \left( \sum_{j \geq 1} \frac{1}{j!} E(z^j)(it)^j \right)^v \right\} \quad (3.40)$$

Equate equation 3.37 and 3.40

$$\sum_{j \geq 1} \frac{1}{j!} (it)^j \kappa_j = \sum_{v \geq 1} \left\{ \frac{(-1)^{v+1}}{v} \left( \sum_{j \geq 1} \frac{1}{j!} E(z^j)(it)^j \right)^v \right\} \quad (3.41)$$

The cumulant  $\kappa_j$  can be computed by matching the coefficient of  $(it)^j$  in 3.41,

$$\begin{aligned} \kappa_1 &= E(z) \\ \kappa_2 &= Var(z) \\ \kappa_3 &= E\{(z - E(z))^3\} \\ \kappa_4 &= E\{(z - E(z))^4\} - 3Var(z) \end{aligned}$$

Since  $z$  is standard normal  $N(0,1)$ , substitute  $\kappa_1 = 0$  and  $\kappa_2 = 1$  into 3.38,

$$\chi(t) = \exp \left( -\frac{1}{2}t^2 + \frac{(it)^3\kappa_3}{3!} + \frac{(it)^4\kappa_4}{4!} + \frac{(it)^5\kappa_5}{5!} + O(t^6) \right) \quad (3.42)$$

Substitute 3.42 in 3.30 gives

$$\begin{aligned} \chi_y(t) &= \left\{ \chi \left( \frac{t}{\sqrt{n}} \right) \right\}^n \\ &= \left\{ \exp \left( -\frac{1}{2} \frac{t^2}{n} + \frac{1}{3!} (it)^3 \kappa_3 n^{-\frac{3}{2}} + \frac{1}{4!} (it)^4 \kappa_4 n^{-2} + \frac{1}{5!} (it)^5 \kappa_5 n^{-\frac{5}{2}} + O(n^{-3}) \right) \right\}^n \end{aligned} \quad (3.43)$$

Set  $u = it$  in 3.43, then

$$\begin{aligned} \chi_y(t) &= \exp \left( -\frac{1}{2}t^2 + \frac{1}{3!} \kappa_3 u^3 n^{-\frac{1}{2}} + \frac{1}{4!} \kappa_4 u^4 n^{-1} + \frac{1}{5!} \kappa_5 u^5 n^{-\frac{3}{2}} + O(n^{-2}) \right) \\ &= \exp \left( -\frac{1}{2}t^2 \right) \exp \left( \underbrace{\frac{1}{3!} \kappa_3 u^3 n^{-\frac{1}{2}} + \frac{1}{4!} \kappa_4 u^4 n^{-1} + \frac{1}{5!} \kappa_5 u^5 n^{-\frac{3}{2}} + O(n^{-2})}_{*} \right) \end{aligned}$$

By expanding the second term in above equation as power series of

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + O(x^4)$$

which implies

$$\begin{aligned} \exp(*) &= 1 + \frac{1}{3!} \kappa_3 u^3 n^{-\frac{1}{2}} + \frac{1}{4!} \kappa_4 u^4 n^{-1} + \frac{1}{5!} \kappa_5 u^5 n^{-\frac{3}{2}} \\ &\quad + \frac{1}{2!} \left( \frac{1}{3!} \kappa_3 u^3 n^{-\frac{1}{2}} + \frac{1}{4!} \kappa_4 u^4 n^{-1} + \frac{1}{5!} \kappa_5 u^5 n^{-\frac{3}{2}} \right)^2 \\ &\quad + \frac{1}{3!} \left( \frac{1}{3!} \kappa_3 u^3 n^{-\frac{1}{2}} + \frac{1}{4!} \kappa_4 u^4 n^{-1} + \frac{1}{5!} \kappa_5 u^5 n^{-\frac{3}{2}} \right)^3 \end{aligned}$$

Then

$$\begin{aligned}\chi_y(t) &= \exp\left(-\frac{1}{2}t^2\right) \exp(*) & (3.44) \\ &= e^{-\frac{1}{2}t^2} + r_1(u)e^{-\frac{1}{2}t^2}n^{-\frac{1}{2}} + r_2(u)e^{-\frac{1}{2}t^2}n^{-1} + r_3(u)e^{-\frac{1}{2}t^2}n^{-\frac{3}{2}} + O(n^{-2})\end{aligned}$$

where

$$\begin{aligned}r_1(u) &= \frac{1}{6}\kappa_3u^3 & (3.45) \\ r_2(u) &= \frac{1}{24}\kappa_4u^4 + \frac{1}{72}\kappa_3^2u^6 \\ r_3(u) &= \frac{1}{1296}\kappa_3^3u^9 + \frac{1}{144}\kappa_3\kappa_4u^7 + \frac{1}{120}\kappa_5u^5\end{aligned}$$

As the characteristic function of standard normal  $Y$  is

$$\chi_y(t) = \int_{-\infty}^{\infty} e^{ity}d\Phi(y) = e^{-\frac{t^2}{2}} \quad (3.46)$$

where  $\Phi(y)$  is the CDF of  $Y \sim N(0, 1)$  respectively.

We can now find the distribution function of  $Y$  by inverting each terms in equation 3.44

$$P(Y \leq y) = \Phi(y) + R_1(y)n^{-\frac{1}{2}} + R_2(y)n^{-1} + R_3(y)n^{-\frac{3}{2}} + O(n^{-2}) \quad (3.47)$$

where  $R_j(y)$  is the inverse transform of  $r_j(it)e^{-\frac{1}{2}t^2}$  in 3.44. Next, we show how to determine the expression of  $R_j(y)$ ,  $j = 1, 2, 3$ .

The integration by parts method is applied to evaluate the integration in equation 3.46, we have

$$\begin{aligned}\left[d\Phi(y)\frac{1}{it}e^{ity}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{e^{ity}}{it}d\Phi^{(1)}(y) &= e^{-\frac{t^2}{2}} \\ (-it)^{-1} \int_{-\infty}^{\infty} e^{ity}d\Phi^{(1)}(y) &= e^{-\frac{t^2}{2}}\end{aligned}$$

Repeated the integration

$$\begin{aligned}
 (-it)^{-2} \int_{-\infty}^{\infty} e^{ity} d\Phi^{(2)}(y) &= e^{-\frac{t^2}{2}} \\
 (-it)^{-3} \int_{-\infty}^{\infty} e^{ity} d\Phi^{(3)}(y) &= e^{-\frac{t^2}{2}} \\
 &\vdots \\
 (-it)^{-j} \int_{-\infty}^{\infty} e^{ity} d\Phi^{(j)}(y) &= e^{-\frac{t^2}{2}}
 \end{aligned} \tag{3.48}$$

where

$$\Phi^{(j)}(y) = \left( \frac{d}{dy} \right)^j \Phi(y)$$

$\Phi(z)$  is the CDF of standard normal random variable  $Z$ . We then multiply both side of equation 3.48 by  $(it)^j$

$$\int_{-\infty}^{\infty} e^{ity} d \left\{ \left( -\frac{d}{dy} \right)^j \Phi(y) \right\} = (it)^j e^{-\frac{t^2}{2}}$$

which implies that the inverse transform of  $(it)^j e^{-\frac{t^2}{2}}$  equals  $\left( -\frac{d}{dy} \right)^j \Phi(y)$ , such that

$$L^{-1} \left[ (it)^j e^{-\frac{t^2}{2}} \right] = \left( -\frac{d}{dy} \right)^j \Phi(y) \tag{3.49}$$

Since  $u = it$  in equation 3.45, the coefficient of  $n^{-\frac{1}{2}}$  in 3.44 is

$$r_1(it) e^{-\frac{1}{2}t^2} = \frac{1}{6} \kappa_3(it)^3 e^{-\frac{1}{2}t^2}$$

By the result of 3.49, the inverse transform of  $r_1(it) e^{-\frac{1}{2}t^2}$  can be obtained

$$R_1(y) = \frac{1}{6} \kappa_3 \left( -\frac{d}{dy} \right)^3 \Phi(y) \tag{3.50}$$

Similarly, we can derive

$$R_2(u) = \frac{1}{24} \kappa_4 \left( -\frac{d}{dy} \right)^4 \Phi(y) + \frac{1}{72} \kappa_3^2 \left( -\frac{d}{dy} \right)^6 \Phi(y) \tag{3.51}$$



$$R_3(u) = \frac{1}{1296}\kappa_3^3 \left(-\frac{d}{dy}\right)^9 \Phi(y) + \frac{1}{144}\kappa_3\kappa_4 \left(-\frac{d}{dy}\right)^7 \Phi(y) + \frac{1}{120}\kappa_5 \left(-\frac{d}{dy}\right)^5 \Phi(y) \quad (3.52)$$

Now we need to use the property of Hermite polynomials given in 3.53 for the rest of our proof

$$\left(-\frac{d}{dy}\right)^j \Phi(y) = -H_{j-1}(y)\phi(y) \quad \text{for } j \geq 1 \quad (3.53)$$

where  $\phi(y)$  is the PDF of standard normal  $N(0, 1)$ , and  $H_n(y)$  is the probabilistic Hermite polynomials. See Koornwinder (2010) for the notation of Hermite polynomials  $H_j$ . By applying formula 3.53 to equation 3.50, 3.51, and 3.52, then

$$\begin{aligned} R_1(y) &= -\frac{1}{6}\kappa_3 H_2(y)\phi(y) \\ R_2(y) &= \left(-\frac{1}{24}\kappa_4 H_3(y) - \frac{1}{72}\kappa_3^2 H_5(y)\right)\phi(y) \\ R_3(y) &= \left(-\frac{1}{1296}\kappa_3^3 H_8(y) - \frac{1}{144}\kappa_3\kappa_4 H_6(y) + \frac{1}{120}\kappa_5 H_4(y)\right)\phi(y) \end{aligned}$$

Hence we can obtain the distribution function of  $Y$  from 3.47

$$\begin{aligned} F_Y(y) &= P(Y \leq y) \\ &= \Phi(y) + R_1(y)n^{-\frac{1}{2}} + R_2(y)n^{-1} + R_3(y)n^{-\frac{3}{2}} + O(n^{-2}) \\ &= \Phi(y) + \phi(y) \left(p_1(y)n^{-\frac{1}{2}} + p_2(y)n^{-1} + p_3(y)n^{-\frac{3}{2}} + O(n^{-2})\right) \end{aligned}$$

where

$$p_1(y) = -\frac{1}{6}\kappa_3 H_2(y) \quad (3.54)$$

$$p_2(x) = -\frac{1}{24}\kappa_4 H_3(y) - \frac{1}{72}\kappa_3^2 H_5(y) \quad (3.55)$$

$$p_3(x) = -\frac{1}{1296}\kappa_3^3 H_8(y) - \frac{1}{144}\kappa_3\kappa_4 H_6(y) + \frac{1}{120}\kappa_5 H_4(y) \quad (3.56)$$

To derive the density function of  $Y$  from its CDF, we have

$$\begin{aligned}
 f_Y(y) &= \frac{d}{dy} F_Y(y) \\
 &= \phi(y) + \frac{d}{dy} \phi(y) (p_1(y)n^{-\frac{1}{2}} + p_2(y)n^{-1} + p_3(y)n^{-\frac{3}{2}} + O(n^{-2})) \\
 &= \phi(y) + \phi(y) \left\{ (p'_1(y) - yp_1(y))n^{-\frac{1}{2}} + (p'_2(y) - yp_2(y))n^{-1} + (p'_3(y) - yp_3(y))n^{-\frac{3}{2}} \right\} \\
 &= \phi(y) \left\{ 1 + (p'_1(y) - yp_1(y))n^{-\frac{1}{2}} + (p'_2(y) - yp_2(y))n^{-1} + (p'_3(y) - yp_3(y))n^{-\frac{3}{2}} \right\}
 \end{aligned}$$

Since the Hermite polynomials satisfy the following recursion relation

$$H'_n(x) = xH_n(x) - H_{n+1}(x) \quad (3.57)$$

By equation 3.54, 3.55, 3.56 and 3.57, we have

$$\begin{aligned}
 p'_1 - yp_1 &= c_1H'_2 - yc_1H_2 \\
 &= c_1yH_2 - c_1H_3 - yc_1H_2 \\
 &= \frac{\kappa_3}{6}H_3
 \end{aligned}$$

$$\begin{aligned}
 p'_2 - yp_2 &= (c_2H'_3 + c_3H'_5) - y(c_2H_3 + c_3H_5) \\
 &= c_2(yH_3 - H_4) + c_3(yH_5 - H_6) - y(c_2H_3 + c_3H_5) \\
 &= c_2yH_3 - c_2H_4 + c_3yH_5 - c_3H_6 - yc_2H_3 - yc_3H_5 \\
 &= -c_2H_4 - c_3H_6 \\
 &= \frac{\kappa_4}{24}H_4 + \frac{\kappa_3^2}{72}H_6
 \end{aligned}$$

$$\begin{aligned}
 p'_3 - yp_3 &= (c_4H'_8 + c_5H'_6 + c_6H'_4) - y(c_4H_8 + c_5H_6 + c_6H_4) \\
 &= c_4(yH_8 - H_9) + c_5(yH_6 - H_7) + c_6(yH_4 - H_5) - y(c_4H_8 + c_5H_6 + c_6H_4) \\
 &= -c_4H_9 - c_5H_7 - c_6H_5 \\
 &= \frac{\kappa_3^3}{1296}H_9 + \frac{\kappa_3\kappa_4}{144}H_7 + \frac{\kappa_5}{120}H_5
 \end{aligned}$$

Hence the probability density function of  $Y$  is

$$\begin{aligned}
 f_Y(y) = & \phi(y) \left\{ 1 + \frac{\kappa_3}{6} H_3(y) n^{-\frac{1}{2}} + \left( \frac{\kappa_4}{24} H_4(y) + \frac{\kappa_3^2}{72} H_6(y) \right) n^{-1} \right. \\
 & \left. + \left( \frac{\kappa_3^3}{1296} H_9(y) + \frac{\kappa_3 \kappa_4}{144} H_7(y) + \frac{\kappa_5}{120} H_5(y) \right) n^{-\frac{3}{2}} + O(n^{-2}) \right\} \quad (3.58)
 \end{aligned}$$

where  $\kappa_j$  are the cumulant of standard normal distribution, and  $H_j(y)$  are the probabilistic Hermit polynomials.

As  $f_Y(y)$  is the probability density function of  $Y = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$ , we need to make the transformation  $\bar{x} = \mu + Y \frac{\sigma}{\sqrt{n}}$  for the probability density function of sample mean  $\bar{X}$ . By taking only the first two terms in the edgeworth expansions in 3.58, we obtain

$$\begin{aligned}
 f_{\bar{X}}(x) &= \frac{\sqrt{n}}{\sigma} f_Y \left( \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \right) \quad (3.59) \\
 &= \frac{1}{\sigma} \phi \left( \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \right) \left[ \sqrt{n} + \frac{\kappa_3}{6} \left\{ \left( \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \right)^3 - 3 \left( \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \right) \right\} + O \left( \frac{1}{\sqrt{n}} \right) \right]
 \end{aligned}$$

To make the order of the approximation accurate to  $O(\frac{1}{\sqrt{n}})$ , we need the value of  $\bar{x}$  near  $\mu$  so that the term in the square bracket of equation 3.59 close to zero. For each  $\bar{x}$ , the method of exponential tilting allow us to create a family of densities such that the term in square bracket become zero. By introducing an extra parameter  $s$ , we can control the order of error term to give a optimal approximation.

For fixed  $s$ , the tilted density  $f_{X_s}(x)$  is defined as

$$\begin{aligned}
 f_{X_s}(x) &= \frac{e^{sx} f_X(x)}{\int_{-\infty}^{+\infty} e^{sx} f_X(x) dx} \\
 &= \frac{e^{sx} f_X(x)}{M_X(s)} \\
 &= \exp(sx - K_X(s)) f_X(x) \quad (3.60)
 \end{aligned}$$

where  $K_X(s) = \log(M_X(s))$  is the cumulant generating function (CGF) of  $X$ .

Given a random sample  $X_{s_1}, X_{s_2}, \dots, X_{s_n}$  from  $f_{X_s}(x)$ , we can derive the tilted density of its sample mean  $\bar{X}_s$ ,  $f_{\bar{X}_s}(x)$ , using its MGF. Since the MGF of the sample mean of a random sample is

$$M_{\bar{X}_s}(t) = \left\{ M_{X_s} \left( \frac{t}{n} \right) \right\}^n$$

and

$$\begin{aligned} M_{X_s}(t) &= \int_{-\infty}^{+\infty} e^{tx} f_{X_s}(x) dx \\ &= \int_{-\infty}^{+\infty} \exp(tx + sx - K_X(s)) f_X(x) dx \\ &= \exp(-K_X(s)) \int_{-\infty}^{+\infty} \exp\{(t+s)x\} f_X(x) dx \\ &= \exp(-K_X(s)) M_X(t+s) \end{aligned}$$

$$\begin{aligned} M_{\bar{X}_s}(t) &= \left\{ M_{X_s} \left( \frac{t}{n} \right) \right\}^n \\ &= \left\{ \exp(-K_X(s)) M_X \left( \frac{t}{n} + s \right) \right\}^n \\ &= \exp(-nK_X(s)) \left\{ M_X \left( \frac{t+ns}{n} \right) \right\}^n \\ &= \exp(-nK_X(s)) M_{\bar{X}}(t+ns) \\ &= \exp(-nK_X(s)) \int_{-\infty}^{+\infty} \exp\{(t+ns)\bar{x}\} f_{\bar{X}}(x) dx \\ &= \int_{-\infty}^{+\infty} \exp(t\bar{x}) \exp\{n(s\bar{x} - K_X(s))\} f_{\bar{X}}(x) dx \end{aligned} \quad (3.61)$$

Hence the PDF of  $\bar{X}_s$  is determined based on equation 3.61

$$f_{\bar{X}_s}(x) = \exp\{n(s\bar{x} - K_X(s))\} f_{\bar{X}}(x) \quad (3.62)$$

Let  $\mu_s, \sigma_s^2$  be the mean and variance of title density  $f_{X_s}$ . By the same idea in constructing equation 3.59, we can write  $f_{\bar{X}_s}(x)$  as

$$f_{\bar{X}_s}(x) = \frac{\sqrt{n}}{\sigma_s} \phi \left( \frac{\bar{x} - \mu_s}{\frac{\sigma_s}{\sqrt{n}}} \right) \left[ 1 + \frac{\kappa_3}{6\sqrt{n}} \left\{ \left( \frac{\bar{x} - \mu_s}{\frac{\sigma_s}{\sqrt{n}}} \right)^3 - 3 \left( \frac{\bar{x} - \mu_s}{\frac{\sigma_s}{\sqrt{n}}} \right) \right\} + O \left( \frac{1}{n} \right) \right] \quad (3.63)$$

Given  $\bar{x}$ , the parameter  $s$  is chosen  $s = \hat{s}$ , such that  $\hat{s}$  satisfies  $\bar{x} = \mu_{\hat{s}}$ , then

$$\bar{x} = \mu_{\hat{s}} = K'_{X_s}(0) \quad (3.64)$$

Now we need to connect the condition 3.64 to the CGF of  $X$ ,  $K_X(s)$ . By equation 3.60, the CGF of the titled density  $X_s$  is

$$\begin{aligned} K_{X_s}(t) &= \log(M_{X_s}(t)) \\ &= \log \left( \int_{-\infty}^{+\infty} e^{tx} f_{X_s}(x) dx \right) \\ &= \log \left( \int_{-\infty}^{+\infty} e^{(t+s)x - K_X(s)} f_X(x) dx \right) \\ &= \log \left( \int_{-\infty}^{+\infty} e^{(t+s)x} f_X(x) dx \right) + \log \left( e^{-K_X(s)} \right) \\ &= K_X(t+s) - K_X(s) \end{aligned}$$

As  $K_X(s)$  is a function of  $s$ , then

$$\frac{dK_{X_s}(t)}{dt} = \frac{dK_X(t+s)}{dt}$$

Since

$$\begin{aligned} \left( \frac{dK_X(t+s)}{dt} \right) \Big|_{t=0} &= \left( \frac{\frac{dM_X(t+s)}{dt}}{M_X(t+s)} \right) \Big|_{t=0} \\ &= \frac{\frac{dM_X(s)}{ds}}{M_X(s)} \\ &= \frac{dK_X(s)}{ds} \end{aligned}$$

we have

$$\left. \left( \frac{d^j K_X(t+s)}{dt^j} \right) \right|_{t=0} = \frac{d^j K_X(s)}{ds^j}$$

$$K_{X_s}^{(j)}(0) = K_X^{(j)}(s) \quad (3.65)$$

Therefore, the  $j$ -th cumulant of  $X_s$  is just the  $j$ -th derivative of CGF of  $X$  with respect to  $s$ . By condition 3.64 and equation 3.65, we have

$$\bar{x} = \underbrace{K'_{X_s}(0)}_{\mu_{\hat{s}}} = K'_X(s)$$

Hence  $\hat{s}$  is the solution of  $K'_X(s) = \bar{x}$ .

We further expand  $f_{\bar{X}_s}(x)$  given in 3.63 to the order of  $n^{-\frac{3}{2}}$

$$f_{\bar{X}_s}(x) = \frac{\sqrt{n}}{\sigma_{\hat{s}}} \phi(y) \left\{ 1 + \frac{\kappa_3}{6} H_3(y) n^{-\frac{1}{2}} + \left( \frac{\kappa_4}{24} H_4(y) + \frac{\kappa_3^2}{72} H_6(y) \right) n^{-1} \right. \\ \left. + \left( \frac{\kappa_3^3}{1296} H_9(y) + \frac{\kappa_3 \kappa_4}{144} H_7(y) + \frac{\kappa_5}{120} H_5(y) \right) n^{-\frac{3}{2}} + O(n^{-2}) \right\} \quad (3.66)$$

Since  $\mu_{\hat{s}} = \bar{x} \Rightarrow y = \frac{\bar{x} - \mu_{\hat{s}}}{\sigma_{\hat{s}}/\sqrt{n}} = 0$ . On the other hand, the first nine probabilistic Hermite polynomials are defined as  $H_0(y) = 1$ ,  $H_1(y) = y$ ,  $H_2(y) = y^2 - 1$ ,  $H_3(y) = y^3 - 3y$ ,  $H_4(y) = y^4 - 6y^2 + 3$ ,  $H_5(y) = y^5 - 10y^3 + 15y$ ,  $H_6(y) = y^6 - 15y^4 + 45y^2 - 15$ ,  $H_7(y) = y^7 - 21y^5 + 105y^3 - 105y$ ,  $H_8(y) = y^8 - 28y^6 + 210y^4 - 420y^2 + 105$ ,  $H_9(y) = y^9 - 36y^7 + 378y^5 - 1260y^3 + 945y$ .

Then we have  $\phi(0) = \frac{1}{\sqrt{2\pi}}$ ,  $H_4(0) = 3$ ,  $H_6(0) = -15$ ,  $H_3(0) = H_5(0) = H_7(0) = H_9(0) = 0$ , which implies  $\frac{\kappa_3}{6} H_3(0) = 0$ , and  $\frac{\kappa_3^3}{1296} H_9(0) + \frac{\kappa_3 \kappa_4}{144} H_7(0) + \frac{\kappa_5}{120} H_5(0) = 0$ .

Hence equation 3.66 becomes

$$\begin{aligned} f_{\bar{X}_s}(x) &= \frac{\sqrt{n}}{\sigma_{\hat{s}}} \phi(0) \left[ \left\{ 1 + \frac{1}{n} \left( \frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24} \right) \right\} + O(n^{-2}) \right] \\ &= \left( \frac{n}{2\pi K_X''(\hat{s})} \right)^{1/2} \left\{ 1 + \frac{1}{n} \left( \frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24} \right) + O(n^{-2}) \right\} \end{aligned} \quad (3.67)$$

By multiplying  $\exp \{ -n(s\bar{x} - K_X(s)) \}$  on both side of equation 3.62, and substitute 3.67

$$\begin{aligned} f_{\bar{X}}(x) &= \exp \{ -n(s\bar{x} - K_X(s)) \} f_{\bar{X}_s}(x) \quad (3.68) \\ &= \left( \frac{n}{2\pi K_X''(\hat{s})} \right)^{1/2} \exp \{ n(K_X(\hat{s}) - \hat{s}\bar{x}) \} \left\{ 1 + \frac{1}{n} \left( \frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24} \right) + O(n^{-2}) \right\} \end{aligned}$$

Since  $\kappa_j$  is only the  $j$ -th cumulant of standard normal  $N(0,1)$ , we will show how to express  $\kappa_j$  in term of the cumulant of  $X$  in the next part of our proof.

As  $Z_s = \frac{X_s - \mu_s}{\sigma_s} \sim N(0,1)$ , where  $\mu_s$  and  $\sigma_s$  are the mean and standard deviation of  $X$ ,  $\frac{dZ_s}{dX_s} = \frac{1}{\sigma_s}$ , and  $f_{Z_s}(z) = \sigma_s f_{X_s}(x)$ . Then

$$\begin{aligned} M_{Z_s}(t) &= \int_{-\infty}^{+\infty} e^{tz_s} f_{Z_s}(z) dz \\ &= \int_{-\infty}^{+\infty} \exp \left\{ t \left( \frac{x_s - \mu_s}{\sigma_s} \right) \right\} \sigma_s f_{x_s}(x) \frac{1}{\sigma_s} dx_s \\ &= \int_{-\infty}^{+\infty} \exp \left( -\frac{\mu_s t}{\sigma_s} \right) \exp \left( \frac{tx_s}{\sigma_s} \right) f_{x_s}(x) dx_s \\ &= \exp \left( -\frac{\mu_s t}{\sigma_s} \right) M_{X_s} \left( \frac{t}{\sigma_s} \right) \end{aligned}$$

which gives

$$\begin{aligned} K_{Z_s}(t) &= \log(M_{Z_s}(t)) \\ &= -\frac{\mu_s t}{\sigma_s} + K_{X_s}\left(\frac{t}{\sigma_s}\right) \end{aligned} \quad (3.69)$$

By differentiating 3.69 repeatedly with respect to  $t$ , we have

$$\begin{aligned} K'_{Z_s}(t) &= -\frac{\mu_s}{\sigma_s} + \frac{1}{\sigma_s} K'_{X_s}\left(\frac{t}{\sigma_s}\right) \\ K''_{Z_s}(t) &= \frac{1}{\sigma_s^2} K''_{X_s}\left(\frac{t}{\sigma_s}\right) \\ K'''_{Z_s}(t) &= \frac{1}{\sigma_s^3} K'''_{X_s}\left(\frac{t}{\sigma_s}\right) \\ &\vdots \\ K^{(j)}_{Z_s}(t) &= \frac{1}{(\sigma_s^2)^{\frac{j}{2}}} K^{(j)}_{X_s}\left(\frac{t}{\sigma_s}\right) \end{aligned}$$

Hence

$$K^{(j)}_{Z_s}(0) = \frac{K^{(j)}_{X_s}(0)}{(\sigma_s^2)^{\frac{j}{2}}}$$

Since  $\sigma_s^2 = K''_{X_s}(0)$ , and  $K^{(j)}_{X_s}(0) = K^{(j)}_X(s)$  by 3.65, then

$$\gamma_j = K^{(j)}_{z_s}(0) = \frac{K^{(j)}_X(s)}{(K''_X(s))^{\frac{j}{2}}}$$

Therefore the density of sample mean  $\bar{X}$  is

$$f_{\bar{x}}(x) = \left( \frac{n}{2\pi K''_X(\hat{s})} \right)^{1/2} \exp \left\{ n(K_X(\hat{s}) - \hat{s}\bar{x}) \right\} \left\{ 1 + \frac{1}{n} \left( \frac{\gamma_4}{8} - \frac{5\gamma_3^2}{24} \right) + O(n^{-2}) \right\} \quad (3.70)$$

where

$$\gamma_3 = \frac{K^{(3)}_X(s)}{(K''_X(s))^{\frac{3}{2}}}, \quad \gamma_4 = \frac{K^{(4)}_X(s)}{(K''_X(s))^2}$$



The leading term of the equation 3.70 is called the saddlepoint approximation for the density of sample mean  $\bar{X}$

$$g(x) = \left( \frac{n}{2\pi K_X''(\hat{s})} \right)^{1/2} \exp \left\{ n(K_X(\hat{s}) - \hat{s}\bar{x}) \right\} \quad (3.71)$$

where  $K_X(s)$  is the cumulant generating function of  $X$ ,

$$K_X''(s) = \frac{d^2 K_X(s)}{ds^2}$$

and  $\hat{s}$  satisfies that

$$K_X'(\hat{s}) = \bar{x}$$

□

To summarise, the key idea of this derivation is to write the characteristic function of  $\bar{X}$  in term of the characteristic function of standard normal distribution  $Y$ , expand the characteristic function of  $Y$  as Taylor series, keep the first few term and apply inversion method to recover the CDF of  $Y$ , then we compute the PDF of  $Y$  and transform back to the PDF of sample mean  $\bar{X}$ . As it is shown in equation 3.70, we can essentially consider saddlepoint approximation as a method to approximate a probability distribution in terms of its cumulant generating function (CGF). As we mentioned in Chapter 2, since the MGF can be easily obtained by the use of Mason's rule, we can then compute the corresponding CGF, this makes saddlepoint approximation for MFM (see glossary) relatively straightforward.

### 3.2.3.1 Approximation error of the Saddlepoint approximation

As the size of the approximation error is the most important concern in practical application, we give a detail discussion about the order of the saddlepoint approximation error in this section. We show that the order of error in estimating sample mean density function can be improved from  $O(n^{-\frac{1}{2}})$  to  $O(n^{-\frac{3}{2}})$  by normalising equation 3.71, where  $g(x) = O(n^{\frac{1}{2}})$ . Hence the order of the approximation of density function of sample mean  $\bar{x}$  in 3.70 is

$$\begin{aligned} f_{\bar{x}}(x) &= g(x) \left( 1 + O\left(\frac{1}{n}\right) \right) \\ &= g(x) + O(n^{-\frac{1}{2}}) \end{aligned} \quad (3.72)$$

Therefore, the error rate is  $O(n^{-\frac{1}{2}})$  if  $f_{\bar{x}}(x)$  is approximated by  $g(x)$ . However  $g(x)$  doesn't necessarily integrate to 1, we can improve this error rate by normalising function  $g(x)$ . From equation 3.72, we have

$$g(x) = \frac{f_{\bar{x}}(x)}{1 + O(\frac{1}{n})} = f_{\bar{x}}(x) \left( 1 - O\left(\frac{1}{n}\right) + \dots \right) \quad (3.73)$$

The constant term is computed based on equation 3.73

$$c = \int_{-\infty}^{+\infty} g(x) dx = 1 - O\left(\frac{1}{n}\right)$$

Then the normalized function for  $g(x)$  is

$$\begin{aligned} G(x) &= \frac{g(x)}{c} \\ &= \frac{f_{\bar{x}}(x) \left( 1 - O(\frac{1}{n}) + O(\frac{1}{n^2}) \right)}{1 - O(\frac{1}{n})} \\ &= f_{\bar{x}}(x) \left\{ 1 + O\left(\frac{1}{n^2}\right) \left( 1 + O\left(\frac{1}{n}\right) + \dots \right) \right\} \\ &= f_{\bar{x}}(x) + f_{\bar{x}}(x) O\left(\frac{1}{n^2}\right) \end{aligned} \quad (3.74)$$

Substitute equation 3.72 in 3.74 gives

$$\begin{aligned}
 G(x) &= f_{\bar{x}}(x) + \left\{ g(x) \left( 1 + O\left(\frac{1}{n}\right) \right) \right\} O\left(\frac{1}{n^2}\right) \\
 &= f_{\bar{x}}(x) + g(x) O\left(\frac{1}{n^2}\right) \\
 &= f_{\bar{x}}(x) + O(n^{-\frac{3}{2}})
 \end{aligned}$$

because of  $g(x) = O(n^{\frac{1}{2}})$  Hence the order of error in approximating  $f_{\bar{x}}(x)$  is improved to  $O(n^{-\frac{3}{2}})$  by using the normalised function  $G(x) = \frac{g(x)}{\int_{-\infty}^{+\infty} g(x) dx}$ .

### 3.2.4 The Lugannani-Rice formula for survival function approximation

As the use of saddlepoint method for tail probabilities approximation is even more important in practical application (see Daniel (1987), Reid (1988, 1991), and Terrell (2003)), our contribution here is to rewrite the proof of Lugannani-Rice formula for survival function approximation  $P(\bar{X} \geq x)$  based on the uniform asymptotic expansion (see Jensen (1995), p.67, Theorem 3.2.1).

**Theorem 5.** *The Lugannani-Rice approximation for survival function*

*Let  $X_1, X_2, \dots, X_n$  be i.i.d random variable with cumulant generating function (CGF)  $K_x(s) = \log(M_x(s))$ . Then the Lugannani and Rice approximation for the survival function of  $\bar{X}$  is given by*

$$\begin{aligned}
 P(\bar{X} > x) &\approx 1 - P(\bar{X} \leq x) \\
 &\approx 1 - \Phi(z_2) + \phi(z_2) \left( \frac{1}{z_1} - \frac{1}{z_2} + O(n^{-\frac{3}{2}}) \right) \quad (3.75)
 \end{aligned}$$

where  $\Phi$  and  $\phi$  are the CDF and PDF of standard normal  $N(0, 1)$  with  $z_1 = \hat{s} \sqrt{K_X''(\hat{s})}$ ,  $z_2 = \sqrt{2(\hat{s}x - K_X(\hat{s}))}$ , and  $\hat{s}$  is the solution to  $K_X'(s) = x$  where  $K_X'(s) = dK_X/ds$  and  $K_X''(s) = d^2K_X/ds^2$  are the first and second derivatives of the CGF of  $X$ .

*Proof.* Given equation 3.68, we have

$$\begin{aligned}
 & P(\bar{X} \geq x) \\
 &= \int_x^\infty f_{\bar{X}}(y) dy \tag{3.76} \\
 &= \int_x^\infty \left( \frac{n}{2\pi K''(\hat{s})} \right)^{1/2} \exp \left\{ n(K(\hat{s}) - \hat{s}y) \right\} \left\{ 1 + \frac{1}{n} \left( \frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24} \right) + O(n^{-2}) \right\} dy
 \end{aligned}$$

In order to apply the uniform asymptotic expansion, we need to transform equation 3.76 in the form of

$$\int_\eta^{T_\alpha} \left( \frac{\alpha}{2\pi} \right)^{1/2} \exp \left( -\frac{\alpha}{2} t^2 \right) q_\alpha(t) dt$$

Define

$$h(y) = \sqrt{2 \{ \hat{s}(y)y - K(\hat{s}(y)) \}} \tag{3.77}$$

Given data  $y$ ,  $\hat{s}(y)$  is the solution of  $K'(s) = y$  such that

$$\frac{dK(\hat{s})}{ds} = y$$

then

$$\begin{aligned}
 \frac{d}{dy} \left( \frac{dK(\hat{s})}{ds} \right) &= 1 \\
 \frac{d}{d\hat{s}} \left( \frac{dK(\hat{s})}{ds} \right) \frac{d\hat{s}}{dy} &= 1 \\
 \frac{d^2 K}{d\hat{s}^2} \frac{d\hat{s}}{dy} &= 1 \\
 \frac{d\hat{s}}{dy} &= \left( \frac{d^2 K}{d\hat{s}^2} \right)^{-1}
 \end{aligned}$$

As

$$K''(\hat{s}) = \frac{d^2 K}{d\hat{s}^2} > 0 \Rightarrow \frac{d\hat{s}}{dy} > 0$$

then  $\hat{s}(y)$  is a strictly increasing function in  $y$ .

Let

$$T = 2 \{ \hat{s}(y)y - K(\hat{s}(y)) \}$$

then

$$\frac{dT}{dy} = 2 \{ \hat{s} + \hat{s}'y - K'(\hat{s})\hat{s}' \}$$

and

$$\begin{aligned} \frac{dh}{dy} &= \frac{dh}{dT} \frac{dT}{dy} \\ &= \frac{1}{2} \{ 2(\hat{s}y - K(\hat{s})) \}^{-\frac{1}{2}} \cdot 2 \{ \hat{s} + \hat{s}'y - K'(\hat{s})\hat{s}' \} \\ &= \{ 2(\hat{s}y - K(\hat{s})) \}^{-\frac{1}{2}} \{ \hat{s} + \underbrace{\hat{s}'(y - K'(\hat{s}))}_{=0} \} \\ &= \{ 2(\hat{s}y - K(\hat{s})) \}^{-\frac{1}{2}} \hat{s} \end{aligned}$$

By equation 3.77, we have

$$h' = \frac{dh}{dy} = \frac{\hat{s}}{h} \tag{3.78}$$

As  $h(y) \geq 0$  and  $\hat{s}(y)$  is a strictly increasing function of  $y$ , then  $h$  is also a strictly increasing function of  $y$ . Therefore the range of integrand can be change from  $\int_x^\infty dy \rightarrow \int_{h(x)}^\infty dh(y)$ .

Since

$$dy = \frac{dy}{dh} \cdot dh = \frac{1}{\frac{dh}{dy}} \cdot dh = \frac{1}{h'} dh \tag{3.79}$$

By equation 3.77 and 3.79, we change the integrand variable from  $y$  to  $h(y)$  and 3.76 becomes

$$\begin{aligned} &P(\bar{X} \geq x) \tag{3.80} \\ &= \underbrace{\int_{h(x)}^\infty \left( \frac{n}{2\pi} \right)^{\frac{1}{2}} \exp\left(-\frac{n}{2}h^2\right) \frac{1}{h' \sqrt{K''(\hat{s})}}}_{(*)} \left\{ 1 + \frac{1}{n} \left( \frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24} \right) + O\left(\frac{1}{n^2}\right) \right\} dh \end{aligned}$$

Since the uniform asymptotic expansion formula is

$$\begin{aligned}
 & \int_{\eta}^{T_{\alpha}} \left(\frac{\alpha}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\alpha}{2}t^2\right) q_{\alpha}(t) dt \tag{3.81} \\
 = & \{1 - \Phi(\sqrt{\alpha}\eta)\} \left\{ q_{\alpha}(0) + \frac{1}{\alpha} \frac{q_{\alpha}^{(2)}(0)}{2} + \dots + \frac{1}{\alpha^k} \frac{q_{\alpha}^{(2k)}(0)}{2 \cdot 4 \cdot \dots \cdot (2k)} + O\left(\frac{1}{\alpha^{k+1}}\right) \right\} \\
 & + \frac{\exp(-\frac{\alpha}{2}\eta^2)}{\sqrt{2\pi\alpha}} \left\{ \frac{q_{\alpha}(\eta) - q_{\alpha}(0)}{\eta} + \frac{1}{\alpha} \frac{q_{1\alpha}(\eta) - q_{1\alpha}(0)}{\eta} + \dots + \frac{1}{\alpha^k} \frac{q_{k\alpha}(\eta) - q_{k\alpha}(0)}{\eta} \right\}
 \end{aligned}$$

By comparing the integral of (\*) term in 3.80 to 3.81, we determine  $T_{\alpha} = \infty$ ,  $\eta = h$ ,  $\alpha = n$ , and  $q_{\alpha}(h) = \frac{1}{h'\sqrt{K''(\hat{s})}}$ , and apply the uniform asymptotic expansion formula for  $k = 1$  case.

$$\begin{aligned}
 & P(\bar{X} \geq x) \\
 = & \{1 - \Phi(\sqrt{n}h)\} \left\{ q_n(0) \left(1 + \frac{1}{n} \left(\frac{\kappa_4}{8} - \frac{5\kappa_3^2}{24}\right)\right) + \frac{1}{n} \frac{q_n''(0)}{2} + O\left(\frac{1}{n^2}\right) \right\} \\
 & + \frac{1}{\sqrt{2\pi n}} \exp\left(-\frac{(\sqrt{n}h)^2}{2}\right) \left\{ \frac{q_n(h) - q_n(0)}{h} + O\left(\frac{1}{n}\right) \right\} \tag{3.82}
 \end{aligned}$$

Next, we need to find the  $q_n(h)$  function and compute  $q_n(0)$ ,  $q^{(2)}(0)$ .

By substituting result 3.78 in  $q_{\alpha}(h) = \frac{1}{h'\sqrt{K''(\hat{s})}}$ , we obtain

$$q_n(h) = \frac{1}{h'\sqrt{K''(\hat{s})}} = \frac{h}{\hat{s}\sqrt{K''(\hat{s})}} \tag{3.83}$$

Then

$$\begin{aligned}
 (q_n(h))^2 &= \left(\frac{h}{\hat{s}\sqrt{K''(\hat{s})}}\right)^2 \\
 &= \frac{2(\hat{s}y - K(\hat{s}))}{\hat{s}^2 K''(\hat{s})} \\
 &= \frac{2(\hat{s}K'(\hat{s}) - K(\hat{s}))}{\hat{s}^2 K''(\hat{s})}
 \end{aligned}$$

By the Taylor expansion of  $K(\hat{s})$  at 0, we have

$$\begin{aligned}
 K(\hat{s}) &= \underbrace{K(0)}_{=0} + K'(0)\hat{s} + \frac{K''(0)\hat{s}^2}{2} + \frac{K'''(0)\hat{s}^3}{6} + \frac{K^{iv}(0)\hat{s}^4}{24} + O(\hat{s}^5) \\
 &= \kappa_1\hat{s} + \frac{\kappa_2\hat{s}^2}{2} + \frac{\kappa_3\hat{s}^3}{6} + \frac{\kappa_4\hat{s}^4}{24} + O(\hat{s}^5)
 \end{aligned} \tag{3.84}$$

Then

$$\begin{aligned}
 K'(\hat{s}) &= \frac{dK(\hat{s})}{d\hat{s}} \\
 &= \kappa_1 + \kappa_2\hat{s} + \frac{\kappa_3\hat{s}^2}{2} + \frac{\kappa_4\hat{s}^3}{6} + O(\hat{s}^4)
 \end{aligned} \tag{3.85}$$

$$\begin{aligned}
 K''(\hat{s}) &= \frac{d^2K(\hat{s})}{d\hat{s}^2} \\
 &= \kappa_2 + \kappa_3\hat{s} + \frac{\kappa_4}{2}\hat{s}^2 + O(\hat{s}^3)
 \end{aligned} \tag{3.86}$$

As  $y = K'(\hat{s})$ , we substitute equation 3.84, 3.85, and 3.86 in equation 3.77

$$\begin{aligned}
 h^2 &= 2(\hat{s}K'(\hat{s}) - K(\hat{s})) \\
 &= 2\left\{ \left( \kappa_1\hat{s} + \kappa_2\hat{s}^2 + \frac{\kappa_3}{2}\hat{s}^3 + \frac{\kappa_4}{6}\hat{s}^4 + O(\hat{s}^5) \right) \right. \\
 &\quad \left. - \left( \kappa_1\hat{s} + \frac{\kappa_2\hat{s}^2}{2} + \frac{\kappa_3\hat{s}^3}{6} + \frac{\kappa_4\hat{s}^4}{24} + O(\hat{s}^5) \right) \right\} \\
 &= \kappa_2\hat{s}^2 + \frac{2\kappa_3\hat{s}^3}{3} + \frac{\kappa_4\hat{s}^4}{4} + O(\hat{s}^5)
 \end{aligned} \tag{3.87}$$

Further substitute equation 3.87 and 3.86 in equation 3.84 gives

$$\begin{aligned}
 q_n^2(h) &= \frac{h^2}{\hat{s}^2 K''(\hat{s})} \\
 &= \frac{\kappa_2 + \frac{2}{3}\kappa_3\hat{s} + \frac{1}{4}\kappa_4\hat{s}^2 + O(\hat{s}^3)}{\kappa_2 + \kappa_3\hat{s} + \frac{1}{2}\kappa_4\hat{s}^2 + O(\hat{s}^3)} \\
 &= \frac{1 + \frac{2}{3}\frac{\kappa_3}{\kappa_2}\hat{s} + \frac{1}{4}\frac{\kappa_4}{\kappa_2}\hat{s}^2 + O(\hat{s}^3)}{1 + \frac{\kappa_3}{\kappa_2}\hat{s} + \frac{1}{2}\frac{\kappa_4}{\kappa_2}\hat{s}^2 + O(\hat{s}^3)} \\
 &= \left(1 + \frac{2}{3}\frac{\kappa_3}{\kappa_2}\hat{s} + \frac{1}{4}\frac{\kappa_4}{\kappa_2}\hat{s}^2 + O(\hat{s}^3)\right) \left(1 - \frac{\kappa_3}{\kappa_2}\hat{s} - \frac{1}{2}\frac{\kappa_4}{\kappa_2}\hat{s}^2 + \left(\frac{\kappa_3}{\kappa_2}\hat{s} + \frac{1}{2}\frac{\kappa_4}{\kappa_2}\hat{s}^2\right)^2 + \dots\right) \\
 &= 1 - \frac{1}{3}\frac{\kappa_3}{\kappa_2}\hat{s} + \left\{\frac{1}{3}\left(\frac{\kappa_3}{\kappa_2}\right)^2 - \frac{1}{4}\frac{\kappa_4}{\kappa_2}\right\}\hat{s}^2 + O(\hat{s}^3) \\
 &= 1 - \frac{1}{3}\gamma_3\sqrt{\kappa_2}\hat{s} + \left(\frac{1}{3}\gamma_3^2 - \frac{1}{4}\gamma_4\right)\kappa_2\hat{s}^2 + O(\hat{s}^3) \tag{3.88}
 \end{aligned}$$

where  $\gamma_j = \frac{\kappa_j}{(\kappa_2)^{\frac{j}{2}}}$  for  $j = 3, 4$ . We next take logarithm on both side of 3.88, and use  $\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \dots$ , for  $-1 < x \leq 1$ ,

$$\begin{aligned}
 2\log(q_n(h)) &= \log \left[ 1 + \left\{ -\frac{1}{3}\gamma_3\sqrt{\kappa_2}\hat{s} + \left(\frac{1}{3}\gamma_3^2 - \frac{1}{4}\gamma_4\right)\kappa_2\hat{s}^2 + O(\hat{s}^3) \right\} \right] \\
 &= -\frac{1}{3}\gamma_3\sqrt{\kappa_2}\hat{s} + \left(\frac{1}{3}\gamma_3^2 - \frac{1}{4}\gamma_4\right)\kappa_2\hat{s}^2 - \frac{\left(-\frac{1}{3}\gamma_3\sqrt{\kappa_2}\hat{s}\right)^2}{2} + O(\hat{s}^3) \\
 \log(q_n(h)) &= -\frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \frac{1}{2}\left(\frac{1}{3}\gamma_3^2 - \frac{1}{4}\gamma_4\right)\kappa_2\hat{s}^2 - \frac{1}{36}\gamma_3^2\kappa_2\hat{s}^2 + O(\hat{s}^3) \\
 &= -\frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \left(\frac{5}{36}\gamma_3^2 - \frac{1}{8}\gamma_4\right)\kappa_2\hat{s}^2 + O(\hat{s}^3)
 \end{aligned}$$



Then

$$\begin{aligned}
 q_n(h) &= \exp \left\{ -\frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \left( \frac{5}{36}\gamma_3^2 - \frac{1}{8}\gamma_4 \right) \kappa_2\hat{s}^2 + O(\hat{s}^3) \right\} \\
 &= 1 - \frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \left( \frac{5}{36}\gamma_3^2 - \frac{1}{8}\gamma_4 \right) \kappa_2\hat{s}^2 + \frac{\left( -\frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \dots \right)^2}{2} + O(\hat{s}^3) \\
 &= 1 - \frac{1}{6}\gamma_3\sqrt{\kappa_2}\hat{s} + \left( \frac{11}{72}\gamma_3^2 - \frac{1}{8}\gamma_4 \right) \kappa_2\hat{s}^2 + O(\hat{s}^3)
 \end{aligned} \tag{3.89}$$

As  $q_n(h)$  is a function of  $h$ , we need to express  $\hat{s}$  in terms of  $h$ .

Define

$$\hat{s} = a_1h + a_2h^2 + O(h^3) \tag{3.90}$$

By substituting 3.90 to the first two terms of 3.87, we have

$$\begin{aligned}
 h^2 &= \kappa_2 (a_1h + a_2h^2)^2 + \frac{2}{3}\kappa_3 (a_1h + a_2h^2)^3 + O(h^4) \\
 &= \kappa_2 a_1^2 h^2 + \left( 2a_1a_2\kappa_2 + \frac{2}{3}\kappa_3 a_1^3 \right) h^3 + O(h^4)
 \end{aligned} \tag{3.91}$$

Equate the coefficient  $h^2$  and  $h^3$  on both side of 3.91 to solve  $a_1, a_2$ , then

$$a_1^2\kappa_2 = 1 \Rightarrow a_1 = \frac{1}{\sqrt{\kappa_2}} \tag{3.92}$$

$$2a_1a_2\kappa_2 + \frac{2}{3}\kappa_3 a_1^3 = 0 \Rightarrow a_2 = -\frac{1}{3} \frac{\kappa_3}{\kappa_2^2} = -\frac{1}{3} \frac{\gamma_3}{\sqrt{\kappa_2}} \tag{3.93}$$

where  $\gamma_3 = \kappa_3\kappa_2^{-\frac{3}{2}}$ , then we substitute 3.92 and 3.93 in 3.90 yields

$$\hat{s} = \frac{1}{\sqrt{\kappa_2}} \left( h - \frac{1}{3}\gamma_3 h^2 \right) + O(h^3) \tag{3.94}$$

Substitute equation 3.94 in to equation 3.89

$$\begin{aligned}
 q_n(h) &= 1 - \frac{1}{6}\gamma_3\sqrt{\kappa_2}\frac{1}{\sqrt{\kappa_2}}\left(h - \frac{1}{3}\gamma_3h^2\right) + \left(\frac{11}{72}\gamma_3^2 - \frac{1}{8}\gamma_4\right)\kappa_2\frac{1}{\kappa_2}\left(h - \frac{1}{3}\gamma_3h^2\right)^2 + O(h^3) \\
 &= 1 - \frac{1}{6}\gamma_3\left(h - \frac{1}{3}\gamma_3h^2\right) + \left(\frac{11}{72}\gamma_3^2 - \frac{1}{8}\gamma_4\right)\left(h - \frac{1}{3}\gamma_3h^2\right)^2 + O(h^3) \\
 &= 1 - \frac{1}{6}\gamma_3h + \left(\frac{1}{18}\gamma_3^2 + \frac{11}{72}\gamma_3^2 - \frac{1}{8}\gamma_4\right)h^2 + O(h^3) \\
 &= 1 - \frac{1}{6}\gamma_3h + \left(\frac{5}{24}\gamma_3^2 - \frac{1}{8}\gamma_4\right)h^2 + O(h^3)
 \end{aligned}$$

Hence

$$q_n(0) = 1, \quad q_n^{(2)}(0) = 2\left(\frac{5}{24}\gamma_3^2 - \frac{1}{8}\gamma_4\right) \quad (3.95)$$

Finally, we substitute equation 3.95 and 3.83 in equation 3.82 to obtain the Lugannani-Rice formula for survival function of  $\bar{X}$

$$\begin{aligned}
 P(\bar{X} \geq x) &= \{1 - \Phi(\sqrt{nh})\} \left\{1 + \frac{1}{n}\left(\frac{\gamma_4}{8} - \frac{5\gamma_3^2}{24}\right) + \frac{1}{n}\left(\frac{5\gamma_3^2}{24} - \frac{\gamma_4}{8}\right) + O\left(\frac{1}{n^2}\right)\right\} \\
 &\quad + \phi(\sqrt{nh})\frac{1}{\sqrt{n}}\left(\frac{q_n(h) - 1}{h} + O\left(\frac{1}{n}\right)\right) \\
 &= (1 - \Phi(\sqrt{nh}))\left(1 + O\left(\frac{1}{n^2}\right)\right) + \phi(\sqrt{nh})\left(\frac{1}{\sqrt{n}\hat{s}\sigma} - \frac{1}{\sqrt{nh}} + O\left(\frac{1}{n^{\frac{3}{2}}}\right)\right) \\
 &= 1 - \Phi(z_2) + \phi(z_2)\left(\frac{1}{z_1} - \frac{1}{z_2} + O(n^{-\frac{3}{2}})\right) \quad (3.96)
 \end{aligned}$$

Note that the saddlepoint approximation for the CDF of  $\bar{X}$  is

$$\begin{aligned}
 F_{\bar{x}}(x) &= P(\bar{x} \leq x) \\
 &= 1 - P(\bar{X} \geq x) \\
 &= \Phi(z_2) + \phi(z_2)\left(\frac{1}{z_2} - \frac{1}{z_1} + O(n^{-\frac{3}{2}})\right)
 \end{aligned}$$

where  $z_1 = \hat{s}\sqrt{nK_X''(\hat{s})}$ ,  $z_2 = \sqrt{2n[\hat{s}x - K_X(\hat{s})]}$ ;  $\Phi$  and  $\phi$  are the CDF and PDF of standard normal  $N(0, 1)$ .  $\square$

## 3.3 Comparison of methods

### 3.3.1 Example 1.5

We compare all the above methods which relies on our access to the MGF and CGF, the maximum entropy method with 5 moment constraints, saddlepoint approximation, and Padé approximation to estimate the total waiting time PDF  $f_{T_{02}}(t)$  based on  $M_{T_{02}}(s)$  in Section 3.1.1 on page 41, where all the internode waiting time follow exponential distribution (see Table 3.1 on page 41).

First, we compute the maximum entropy density based on 5 moment constraints

$$\hat{f}_{ME}(t) = \exp(-1.2160 - 0.0202t - 0.0937t^2 + 0.0073t^3 + 0.0002t^4)$$

The corresponding maximum entropy survival function,  $\hat{S}_{ME}(t) = \int_t^\infty \hat{f}_{ME}(u)du$ , can then be obtained by numerical integration.

Since we are only interested in the distribution of total waiting time  $T_{02}$  instead of the mean of total waiting time from node 0 to node 2, the saddlepoint approximation for  $f_{T_{02}}(t)$  of can be obtained by the substituting  $n = 1$  in equation 3.71.

$$\hat{f}_{sp}(t) = (2\pi K''(\hat{s}))^{-\frac{1}{2}} \exp(K(\hat{s}) - \hat{s}x)$$

where  $K(s) = \log(M_{T_{02}}(s))$ ,  $M_{T_{02}}(s)$  is the MGF of  $T_{02}$ ;  $K'(s) = \frac{d}{ds}K(s)$ , and  $\hat{s}$  satisfies  $K'(\hat{s}) = x$ .

The saddlepoint approximation for the survival function can be computed by the use of Lugannani-Rice formula in Theorem 5 with  $n = 1$ .

$$\hat{S}_{sp}(t) = 1 - \Phi(z_2) + \phi(z_2) \left( \frac{1}{z_1} - \frac{1}{z_2} \right)$$

where  $z_1 = \hat{s}\sqrt{K''(\hat{s})}$ ,  $z_2 = \sqrt{2(\hat{s}t - K(\hat{s}))}$ ;  $\Phi$  and  $\phi$  are the distribution and density function of standard normal  $N(0, 1)$ .

The estimated PDF and survival function based on the inversion of Padé approximation of  $M_{T_{02}}(s)$ ,  $PA_{[3,4]}(s)$ , are given in equation 3.27 and 3.28. Since both the true density function and the survival function are in closed form (see equation 3.6 and 3.5 on page 43), we compare these three methods by calculating the integrated square errors (ISE).

$$ISE = \int_0^{\infty} (f(t) - \hat{f}(t))^2 dt$$

where  $f(t)$  is the true density and  $\hat{f}(t)$  is the estimated density. We particularly focus on the performance of these methods when estimation is only based the simulated data on the time interval  $t = [0, 20]$ . The integrated square errors obtained for each method are given in Table 3.5. It is shown that the Padé approximation has the smallest ISE and is remarkable accurate in estimating both PDF and survival function of total waiting time between node 0 and node 2.

Table 3.5: Integrated square error(ISE)

Method	density function	survival function
Maximum Entropy	0.0422	0.0186
Saddlepoint approximation	0.0286	0.0017
Padé approximation	$7.46 \times 10^{-13}$	$3.57 \times 10^{-14}$

Figure 3.5 and 3.6 indicate that all the three methods give reasonable good approximations for both density function and survival function, particularly the density estimation based on Padé method is virtually the same as the true density. For further comparison, we take the logarithm of our density estimation and survival function estimation result, and plot each of them against  $t$  in Figure 3.7 and 3.8 respectively. Generally, we can see that both saddlepoint approximation and Padé approximation are relatively more accurate than the approximations based maximum entropy method, which has a large error in the tail area estimation of PDF and survival function. This may due to the fact the expression of maximum entropy density is an exponential of polynomial function, see equation 3.9 on page 45, and it becomes unstable when  $t$  is large. We will give more discussion about the tail area approximation in Chapter 5.

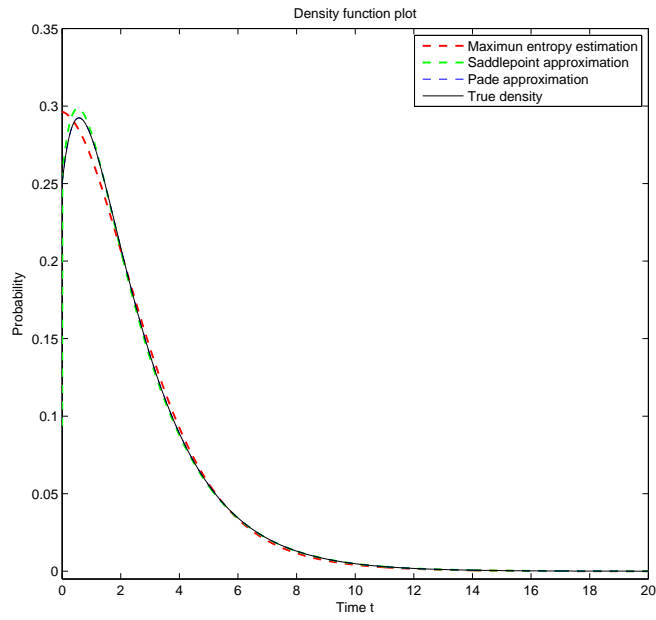


Figure 3.5: Density Estimation for  $T_{02}$  in Example 1.5

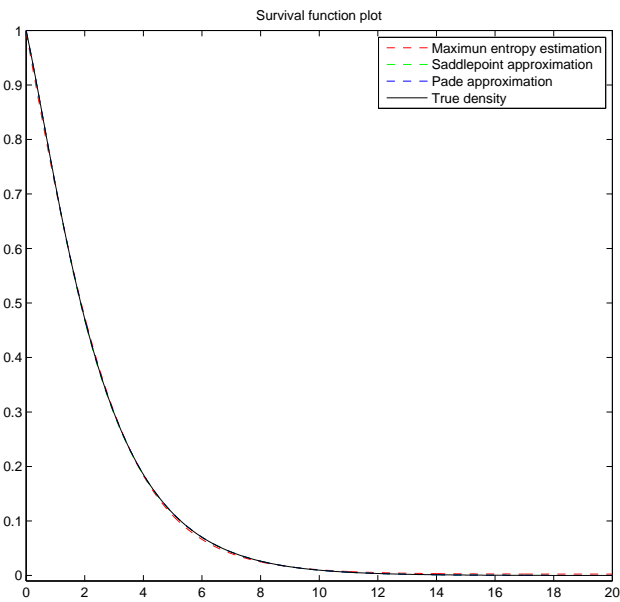


Figure 3.6: Survival estimation for  $T_{02}$  in Example 1.5

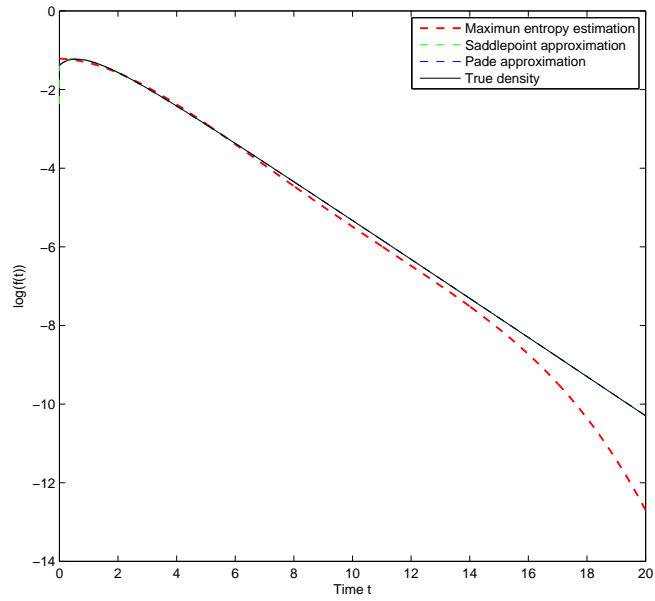


Figure 3.7: Plot  $\log(f(t))$  against  $t$  in Example 1.5

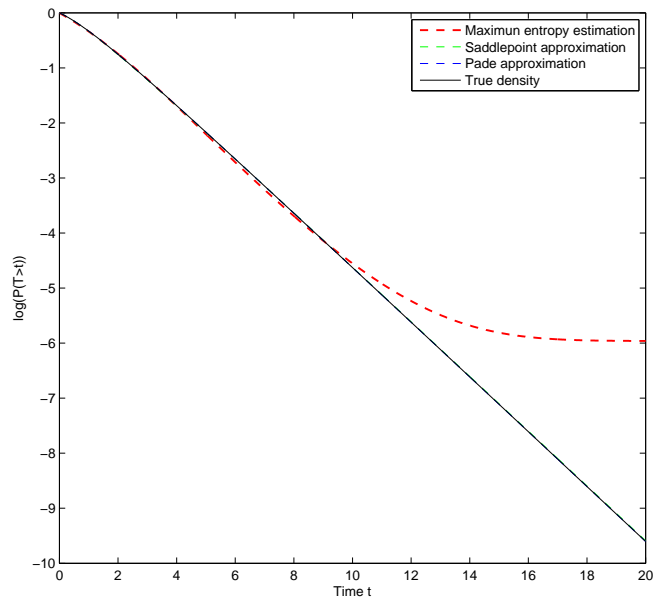


Figure 3.8: Plot  $\log(P(T > t))$  against  $t$  in Example 1.5

### 3.3.2 Example 1.6

Since a closed form expression for the true PDF in exponential case can always be obtained by directly applying the inverse Laplace transform of the MGF, we do not need to use any density approximation methods. As this is not always the case and the true PDF will not have close form if the internode waiting time is non exponentially distributed. In this section, we apply the maximum entropy method, saddlepoint approximation and Padé approximation to estimate the distribution of the total waiting time in the example of Section 3.2.1.4 on page 54, where Gamma distributed inter-node waiting time is allowed (see Table 3.4).

The MGF of the total waiting time from node 0 to node 2 is

$$M_{T_{02}}(s) = \frac{\frac{1}{4}\left(\frac{4.8}{4.8-s}\right)^{3.2}\left(\frac{3}{3-s}\right)^2 + \frac{1}{2}\left(\frac{2.2}{2.2-s}\right)}{1 - \frac{1}{4}\left(\frac{4.8}{4.8-s}\right)^{3.2}\left(\frac{4}{4-s}\right)}$$

We can see that  $M_{T_{02}}(s)$  is not a rational function and the Heaviside formula given in Lemma 4 can not be applied to compute its inverse Laplace transform for  $f_{T_{02}}(t)$ . By imposing 6 moment constraints, the maximum entropy estimated density is determined as

$$\hat{f}_{ME}(t) = \exp(-0.033 - 1.6088t + 1.0621t^2 - 0.4775t^3 + 0.0785t^4 - 0.0043t^5)$$

The estimated survival function based on the maximum entropy density is defined as  $\hat{S}_{ME}(t) = \int_t^\infty \hat{f}_{ME}(u)du$ , which requires to compute integration numerically.

Furthermore the saddlepoint approximation for the PDF and survival function of  $T_{02}$  can be obtained by formula given in 3.71 and 3.96 respectively. To apply the Padé approximation method introduced in Section 3.2.2, we first approximate  $M_{T_{02}}(s)$  as rational function  $PA_{[3,4]}(s)$

$$PA_{[3,4]}(s) = \frac{1 - 0.4508s + 0.138s^2 - 0.0117s^3}{1 - 1.5039s + 0.7323s^2 - 0.1442s^3 + 0.0109s^4}$$

Lemma 4 on page 40 is applied to invert  $PA_{[3,4]}(s)$  for the Padé estimated PDF.

$$\begin{aligned} \hat{f}_{pa}(t) = & 2e^{-4.6724t} (1.1902 \cos(2.406t) + .613995682 \sin(2.406t)) \\ & -3.876e^{-2.5763t} + 2.566e^{-1.2863t} \end{aligned}$$

The Padé estimated survival function is

$$\begin{aligned} \hat{S}_{pa}(t) = & -1.5044 \exp(-2.5764t) + 1.9948 \exp(-1.2863t) \\ & 2 \exp(-4.6725t) (0.2548 \cos(2.406t) + 0.0002 \sin(2.406t)) \end{aligned}$$

The estimation for PDF and survival function is presented in Figure 3.9 and Figure 3.10 separately. We also compare all three methods by plotting the logarithm of estimation result against  $t$ , Figure 3.11 and 3.12 reveal that the instability of maximum entropy density occurs when  $t$  becomes large.

To conclude, we find two appealing advantages of using Padé approximation for total waiting time density estimation. First, the Padé method is more informative than the other two methods in the sense that it can always provide an analytical approximation for the unknown true density, which can be easily applied to obtain closed form expression for survival function, reliability function and hazard function. Secondly, the Padé method allows us to estimate the PDF and CDF in the form of mixture of exponential densities, whose rates are determined by the poles of the rational approximation of MGF. This unique feature indicates that we can potentially analyse the total waiting time data in terms of phase type distribution (see Huzurbazar (2002)). By comparisons, the maximum entropy method can also give a closed form maximum entropy density to estimate the probability density function, however, it is not easy to perform further calculation because the high degree of polynomial in equation 3.9 leads to high computational cost in evaluating the integral. Although the saddlepoint approximation is accurate in both density and survival function estimation, the implementation could be difficult if the MGF of total waiting time is complicated, and it often requires a great deal of computational effort to compute the saddlepoint  $\hat{s}$  (see Theorem 4) and obtain estimation.



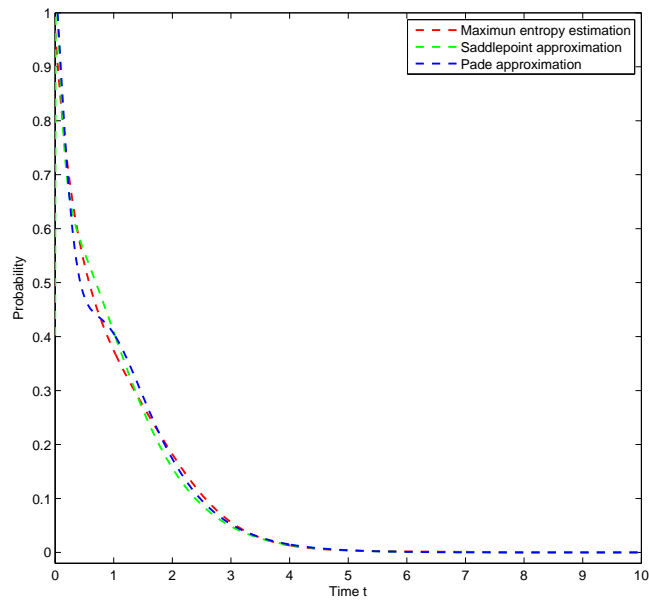


Figure 3.9: Estimated PDF of  $T_{02}(t)$  in Example 1.6

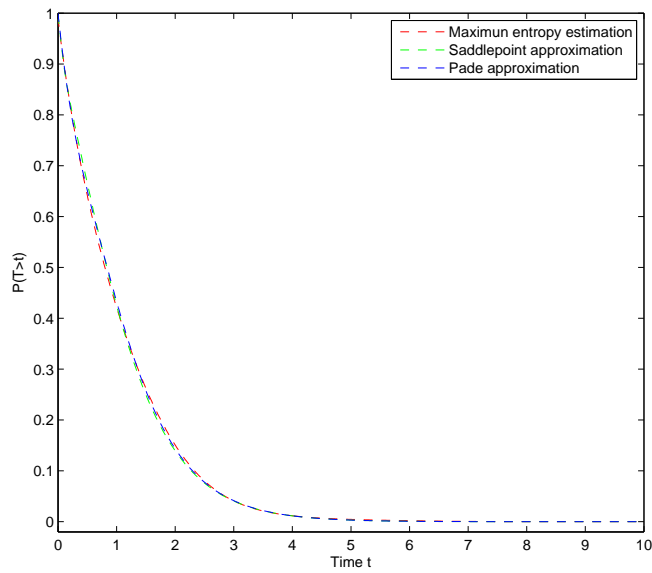


Figure 3.10: Estimated survival function of  $T_{02}$  in Example 1.6

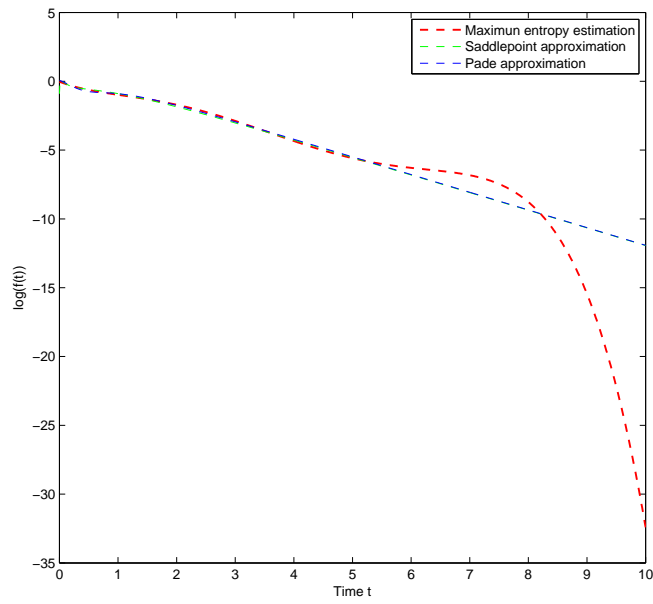


Figure 3.11: Plot  $\log(f(t))$  against  $t$  in Example 1.6

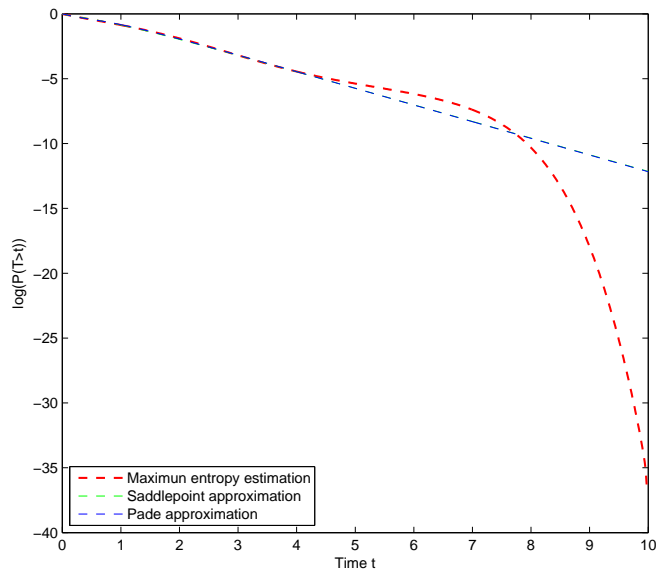


Figure 3.12: Plot  $\log(P(T > t))$  against  $t$  in Example 1.6

# Chapter 4

## Parameter estimation

In this chapter, we apply the maximum likelihood method and the method of moments (MM) to estimate parameters in the total waiting time distribution. The maximum likelihood method selects values of the parameters that maximise the likelihood function. Maximum likelihood estimators (MLE) have a number of attractive asymptotic properties, such as consistency, efficiency and asymptotic normality.

However, in a flowgraph with non-exponentially distributed internode waiting time, as shown in Section 3.3.2 Chapter 3, the probability density function of total waiting time is intractable and the analytical expression of likelihood function is not available. In this case, the maximum likelihood method is often difficult to implement and computationally costly, and the MLE has to be determined numerically by the use of optimization methods based on an approximation of the likelihood function.

The method of moments, on the other hand, does not require a closed form of the likelihood function. It constructs moment equations by equating sample moments with population moments (i.e. the derivation of MGF at  $s = 0$ ) and solving these moment equations to estimate the parameters. Moreover, with suitable parameterizations, MM is computationally efficient.

Here is how this chapter is organised. The first two sections of this chapter deal with the Maximum Likelihood Estimation. The first contribution we present is to propose a method to convert the mixture of negative weight exponential densities into a suitable form that can be adapted for applying the EM algorithm to estimate its weight and rate parameters. We then turn to the method of moments approach for parameter estimation in flowgraph, particularly the transition probability and the parameters in the internode waiting time distribution. Our second contribution is to develop a formula to estimate the bias in the MM estimator and suggest a bias correction method for the MM. The last section of this chapter gives comparison between the MLE and MM estimator in terms of their mean square error and actual computational time.

## 4.1 Maximum Likelihood Estimation

Given a sample of observations, the maximum likelihood estimation is a method to estimate the parameters of a distribution by finding the value of parameters that maximise the likelihood function.

**Definition 8.** *let  $x = \{x_1, \dots, x_n\}$  be a set of independent and identically distributed (iid) observations from a distribution with probability density function  $f(x|\theta)$ , that is indexed by the set of parameters  $\theta = (\theta_1, \dots, \theta_d)$ . Given the observations, the resulting likelihood function of the parameters  $\theta$  is:*

$$L(\theta|X) = \prod_{i=1}^n f(x_i|\theta)$$

*Then the maximum likelihood estimator (MLE) for  $\theta$  based on the observations  $x_1, \dots, x_n$  is defined as  $\hat{\theta}$  for which*

$$\hat{\theta} = \operatorname{argmax} L(\theta|x)$$

For a  $d$ -dimensional parameter  $\theta = (\theta_1, \dots, \theta_d)$ , the procedure of finding the MLE is usually about solving a system of likelihood equations, which are constructed by setting the first derivative of the logarithm of the likelihood function to zero. As it is shown in equation 3.5, Section 3.1.1, it is common to have the total waiting time density function in the form of a mixture density, and the likelihood equation of mixture density is nonlinear equation, which can not have analytical solution. Consequently, we need to apply some iterative procedures to find an approximate solution of the likelihood equations.

## 4.2 The Expectation-Maximisation algorithm

The Expectation-Maximisation (EM) algorithm is an iterative method to find the MLE when the likelihood function is impractical to differentiate or directly maximise, and the closed form solutions of likelihood function is not available. Each iteration of EM algorithm consists of two steps: the expectation step (E-step) follows by the maximisation step (M-step). This algorithm was first formally introduced by Dempster, Laird, and Rubin (1977), and it has been broadly applied to perform statistical inference when the given data set is incomplete or has missing values. For more detail background theory and extension, see McLachlan and Krishnam (1997).

In flowgraph analysis, we assume that the path of each particle chose to move from input node to output node is not observable, then the notion of “incomplete data” can be considered as the missing information about the “path” between input and output that generates each total waiting time, it is therefore useful to express the waiting time problem as an incomplete-data problem within the EM algorithm framework.

In this section, we first briefly review the basic theory of the EM algorithm, present a detail derivation of the procedure for parameter estimation in a mixture density, and propose an method to transform the mixture of negative weight density into a suitable form for implementing the EM algorithm.

Let  $X$  be a random variable with observed data  $x = (x_1, \dots, x_n)$  that are generated by some distribution having parameters  $\theta$  with parameter space  $\Theta$ . Let  $y = (y_1, \dots, y_n)$  denotes the unobservable or missing information, and then assume a complete data set  $z = (x, y)$ . The joint density function for random variable  $Z$  is

$$\begin{aligned} f(z|\theta) &= f(x, y|\theta) \\ &= f(y|x, \theta)f(x|\theta) \end{aligned}$$

We can then define the complete data likelihood function as

$$\begin{aligned} L(\theta|z) &= \prod_{i=1}^n f(z_i|\theta) \\ &= \prod_{i=1}^n f(x_i, y_i|\theta) \end{aligned}$$

Since the missing information  $y$  is unknown,  $L(\theta|z)$  can be considered as a function of  $y$  gives  $x$  and  $\theta$ . We refer to the original likelihood function  $L(\theta|x)$  as the incomplete data likelihood function. The EM algorithm consists of two steps as follow.

**The E-step (Expectation):** The first step of EM algorithm is to find the expected value of complete data log-likelihood function  $\log(L(\theta|x, y))$  with respect to the unknown data  $Y$  given the observed data  $X$  and current value of parameter  $\theta^{(m)}$ .

$$\begin{aligned} Q(\theta, \theta^{(m)}) &= E \{ \log(L(\theta|x, y)) \}_Y \\ &= \int_Y \log(L(\theta|x, y)) f(y|x, \theta^{(m)}) dy \end{aligned} \tag{4.1}$$

The second argument  $\theta^{(m)}$  in  $Q(\theta, \theta^{(m)})$  represents the current value for  $\theta$  that we use to evaluate the expectation at the  $m$ -th iteration. Note that equation 4.1 is obtained based on the following definition of conditional expectation.

$$E[h(y)|x, \theta^{(m)}] = \int_Y h(y)f(y|x, \theta^m)dy$$

The key idea in the expectation step is to understand that  $X$  and  $\theta^m$  are given constants, and  $Y$  is a random variable with marginal distribution  $f(y|x, \theta^{(m)})$ .

**The M-step (Maximisation):** The second step of EM algorithm is to maximise the expectation we obtained in equation 4.1 with respect to  $\theta$  such that

$$\theta^{(m+1)} = \operatorname{argmax} Q(\theta, \theta^{(m)})$$

We choose  $\theta^{(m+1)}$  from  $\Theta$  that

$$Q(\theta^{(m+1)}, \theta^{(m)}) \geq Q(\theta, \theta^{(m)}) \text{ for all } \theta \in \Theta$$

Given the initial value  $\theta^{(m)}$  and data  $X$ , the EM algorithm starts off with E-step and follows by M-step, the two steps are repeated until the difference in the incomplete data log-likelihood function is less than a defined small amount  $\delta$  for convergence.

$$L(\theta^{(m+1)}|x) - L(\theta^{(m)}|x) \leq \delta$$

Essentially, EM algorithm reconstructs the given incomplete-data likelihood function  $L(\theta|x)$  in terms of a complete-data likelihood function  $L(\theta|z)$ , makes use of the connection between these two function and obtains a MLE based on a potentially simpler calculation in the EM algorithm for complete-data problem. In some simple case, for example mixture of Normal distributions or mixture of Poisson distribution, the M-step of the algorithm always has analytical solution, so that the implementation of EM algorithm is easy and we have closed form MLE. In our case, the total waiting time PDF is in the form of mixture of exponential distribution, the M-step usually doesn't have analytical solution, and numerical method is required to compute the MLE.

Dempster, Laird, and Rubin (1977) shows that each iteration of EM algorithm is guaranteed to increase the incomplete data log-likelihood until it converges, the rate of convergence is linear and the rate depends on the proportion of information in the observed data. By comparing to Newton-Raphson method, the EM algorithm is numerically more stable with each iteration to increase the log-likelihood function, and it converges to a local maximiser from almost any starting point (Dennis and Schnabel (1983), page 5). A detail discussion on the convergence issue of EM algorithm is given in Wu (1983). The EM algorithm also has the advantage of lower computational cost per iteration, which it does not require to evaluate the Jacobian matrix of likelihood equations. Although the EM algorithm may converge very slowly for some initial value of parameter, this can be overcome by starting from random initial value and keep the best of those initial value of parameters that requires small number of iterations. In practise, it is common to monitor convergence by observing the increase in the plot of log-likelihood against iterations.

### 4.2.1 Case 1: Mixture with positive weights

As an important family of densities to which the EM algorithm can be applied is the mixture density, which is a weighted sum of  $k$  component densities that come from a particular distribution, such as normal distribution or exponential distribution. The goal of this section is to discuss the formulation of the EM algorithm for mixture densities.

**Definition 9.** *The finite mixture densities is defined as*

$$f(x|\Theta) = \sum_{j=1}^k w_j f_j(x|\theta_j) \quad (4.2)$$

where each  $f_j(x|\theta_j)$  is a density function,  $\sum_{j=1}^k w_j = 1$ , and each weight  $w_j \geq 0$  can be thought of as the probability of the data belongs to the  $j$ -th component density  $f_j(x|\theta_j)$  (i.e.  $P(\text{component density } j) = w_j$ ). The parameter vector is denoted by  $\Theta = (w_1, \dots, w_k; \theta_1, \dots, \theta_k)$



The rest of this section will illustrate the derivation of formulas to estimate the parameters in mixture densities (i.e. weights and parameters in component density) based on the EM algorithm. We start off by defining the incomplete data log-likelihood for  $f(x|\Theta)$  from observed data  $x = (x_1, \dots, x_n)$  as

$$\begin{aligned} \log(L(\Theta|X)) &= \log\left(\prod_{i=1}^n f(x_i|\Theta)\right) \\ &= \sum_{i=1}^n \log\left(\sum_{j=1}^k w_j f_j(x_i|\theta_j)\right) \end{aligned}$$

As  $\log(L(\Theta|X))$  is in the form of the logarithm function of the sum of mixture densities, it is difficult to be maximised by the usual maximum likelihood procedure. To apply EM algorithm, we first denote the missing information by  $y = (y_1, \dots, y_n)$ , where  $y_i \in (1, 2, \dots, k)$  for each  $i = 1, \dots, n$  such that  $y_i = j$  if the  $i$ -th data,  $x_i$  belongs to the  $j$ -th component density  $f_j(x|\theta_j)$ ,  $j = 1, 2, \dots, k$ .

The complete data log-likelihood function is

$$\begin{aligned} \log(L(\Theta|X, Y)) &= \log(f(X, Y|\Theta)) \\ &= \sum_{i=1}^n \log(f(y_i|x_i, \theta_{y_i})f(x_i|\theta_{y_i})) \end{aligned} \quad (4.3)$$

Given the initial value  $\Theta^{(0)} = (w_1^{(0)}, \dots, w_k^{(0)}; \theta_1^{(0)}, \dots, \theta_k^{(0)})$  and sample data  $X$ , we can compute the marginal density function

$$f(Y|X, \Theta^{(0)}) = \prod_{i=1}^n f(y_i|x_i, \Theta^{(0)}) \quad (4.4)$$

where

$$f(y_i|x_i, \Theta^{(0)}) = \frac{f(y_i, x_i, \Theta^{(0)})}{f(x_i, \Theta^{(0)})} = \frac{f(x_i|y_i, \Theta^{(0)})f(y_i|\Theta^{(0)})}{f(x_i|\Theta^{(0)})} \quad (4.5)$$

By equation 4.2, we have

$$f(x_i|\Theta^{(0)}) = \sum_{j=1}^k w_j f_j(x_i|\theta_j^{(0)}) \quad (4.6)$$

and

$$f(y_i|\Theta^{(0)}) = w_{y_i}^{(0)} \quad (4.7)$$

Substitute 4.6 and 4.7 in 4.5, equation 4.4 becomes

$$\begin{aligned} f(Y|X, \Theta^{(0)}) &= \prod_{i=1}^n f(y_i|x_i, \Theta^{(0)}) \\ &= \prod_{i=1}^n \frac{f(x_i|y_i, \Theta^{(0)})w_{y_i}^{(0)}}{\sum_{j=1}^k w_j f_j(x_i|\theta_i^{(0)})} \end{aligned} \quad (4.8)$$

Then the E-step gives

$$\begin{aligned} Q(\theta, \theta^{(0)}) &= \mathbb{E}_{Y|X, \Theta^{(0)}} \left\{ \log(L(\Theta|X, Y)) \right\} \\ &= \sum_{y \in Y} \log(L(\Theta|X, y)) f(y|X, \Theta^{(0)}) \end{aligned}$$

where  $\log(L(\Theta|X, y))$  and  $f(y|X, \Theta^{(0)})$  are defined in equation 4.3 and 4.8 separately. Further simplification leads to

$$\begin{aligned} Q(\theta, \theta^{(0)}) &= \sum_{y \in Y} \log(L(\Theta|X, y)) f(y|X, \Theta^{(0)}) \\ &= \sum_{y_1=1}^k \cdots \sum_{y_n=1}^k \sum_{i=1}^n \log(w_{y_i} f_{y_i}(x_i|\theta_{y_i})) f(y|X, \Theta^{(0)}) \\ &= \sum_{y_1=1}^k \cdots \sum_{y_n=1}^k \sum_{i=1}^n \sum_{j=1}^k \delta_{(j, y_i)} \log(w_j f_j(x_i|\theta_j)) f(y|X, \Theta^{(0)}) \\ &= \sum_{j=1}^k \sum_{i=1}^n \log(w_j f_j(x_i|\theta_j)) \sum_{y_1=1}^k \cdots \sum_{y_n=1}^k \delta_{(j, y_i)} f(y|X, \Theta^{(0)}) \end{aligned} \quad (4.9)$$

where

$$\delta_{(j, y_i)} = \begin{cases} 1 & \text{if } j = y_i \\ 0 & \text{otherwise} \end{cases}$$

As

$$f(y|X, \Theta^{(0)}) = \prod_{z=1}^n f(y_z|x_z, \Theta^{(0)})$$

Then we have

$$\begin{aligned} & \sum_{y_1=1}^k \cdots \sum_{y_n=1}^k \delta_{(j,y_z)} f(y|X, \Theta^{(0)}) \\ = & \sum_{y_1=1}^k \cdots \sum_{y_n=1}^k \delta_{(j,y_z)} \prod_{z=1}^n f(y_z|x_z, \Theta^{(0)}) \\ = & \left[ \sum_{y_1=1}^k \cdots \sum_{y_{i-1}=1}^k \sum_{y_{i+1}=1}^k \cdots \sum_{y_n=1}^k \prod_{z=1, z \neq i}^n f(y_z|x_z, \Theta^{(0)}) \right] f(j|x_i, \Theta^{(0)}) \\ = & \prod_{z=1, z \neq i}^n \left[ \sum_{y_z=1}^k f(y_z|x_z, \Theta^{(0)}) \right] f(j|x_i, \Theta^{(0)}) \\ = & f(j|x_i, \Theta^{(0)}) \end{aligned}$$

Because of  $\sum_{y_z=1}^k f(y_z|x_z, \Theta^{(0)}) = 1$ . Therefore, we can simplify equation 4.9 as

$$\begin{aligned} Q(\theta, \theta^{(0)}) &= \sum_{j=1}^k \sum_{i=1}^n \log(w_j f_j(x_i|\theta_j)) f(j|x_i, \Theta^{(0)}) \\ &= \sum_{j=1}^k \sum_{i=1}^n \log(w_j) f(j|x_i, \Theta^{(0)}) \\ &\quad + \sum_{j=1}^k \sum_{i=1}^n \log(f_j(x_i|\theta_j)) f(j|x_i, \Theta^{(0)}) \\ &= G(w_j) + R(\theta_j) \end{aligned}$$

As  $G(w_j)$  and  $R(\theta_j)$  are independent, we can maximise  $Q(\theta, \theta^{(0)})$  with respect to  $w_j$  and  $\theta_j$  separately in M-step. Since the maximisation of  $G(w_j)$  is subject to the constraint  $\sum_{j=1}^k w_j = 1$ , the method of Lagrange multipliers can be applied

in this case, and solve the following equation for estimating  $w_j$ ,  $j = 1, \dots, k$ . Let

$$\begin{aligned}\Lambda(w_j, \lambda) &= G(w_j) + \lambda \left( \sum_{j=1}^k w_j - 1 \right) \\ &= \sum_{j=1}^k \sum_{i=1}^n \log(w_j) f(j|x_i, \Theta^{(0)}) + \lambda \left( \sum_{j=1}^k w_j - 1 \right)\end{aligned}$$

which yields the system of equations:

$$\frac{\partial \Lambda}{\partial w_j} = \sum_{j=1}^k \sum_{i=1}^n \frac{f(j|x_i, \Theta^{(0)})}{w_j} + \sum_{j=1}^k \lambda = 0 \quad (4.10)$$

$$\frac{\partial \Lambda}{\partial \lambda} = \sum_{j=1}^k w_j - 1 = 0 \quad (4.11)$$

Note equation 4.11 is the original constraint for the weights in Definition 9.

By multiplying  $w_j$  on both side of equation 4.10, we have

$$\begin{aligned}\sum_{j=1}^k \sum_{i=1}^n f(j|x_i, \Theta^{(0)}) + \sum_{j=1}^k \lambda w_j &= 0 \\ \sum_{j=1}^k \left( \sum_{i=1}^n f(j|x_i, \Theta^{(0)}) + \lambda w_j \right) &= 0\end{aligned} \quad (4.12)$$

$$\sum_{i=1}^n \sum_{j=1}^k f(j|x_i, \Theta^{(0)}) + \lambda = 0 \quad (4.13)$$

Since  $\sum_{j=1}^k f(j|x_i, \Theta^{(0)}) = 1$ , then equation 4.13 becomes

$$\begin{aligned}\sum_{i=1}^n 1 + \lambda &= 0 \\ \lambda &= -n\end{aligned}$$

On the other hand, equation 4.12 implies that

$$\sum_{i=1}^n f(j|x_i, \Theta^{(0)}) + \lambda w_j = 0$$

$$w_j = -\frac{\sum_{i=1}^n f(j|x_i, \Theta^{(0)})}{\lambda} \quad (4.14)$$

Substituting  $\lambda = -n$  in 4.14 gives a general formula to determine the weight in mixture density

$$w_j = \frac{1}{n} \sum_{i=1}^n f(j|x_i, \Theta^{(0)}) \quad (4.15)$$

where  $f(j|x_i, \Theta^{(0)})$  is the marginal density function of  $y_i = j$ -th component density given data  $x_i$  and the current value of parameter  $\Theta^{(0)}$  such that

$$f(j|x_i, \Theta^{(0)}) = \frac{w_j^{(0)} f_j(x_i|\theta_j^{(0)})}{\sum_{l=1}^k w_l^{(0)} f_l(x_i|\theta_l^{(0)})} \quad \text{for } j = 1, \dots, k.$$

On the other hand, in order to estimate the parameters  $\theta = (\theta_1, \dots, \theta_k)$  that are specified in each component density, we need to maximise function  $R(\theta_j)$

$$R(\theta_j) = \sum_{j=1}^k \sum_{i=1}^n \log(f_j(x_i|\theta_j)) f(j|x_i, \Theta^{(0)})$$

with respect  $\theta_j$ , for  $j = 1, \dots, k$ . This involves finding the partial derivative of  $R(\theta_j)$

$$\frac{\partial R(\theta_j)}{\partial \theta_j} = \sum_{j=1}^k \sum_{i=1}^n \left( \frac{\partial \log(f_j(x_i|\theta_j))}{\partial \theta_j} \right) f(j|x_i, \Theta^{(0)}) = 0 \quad (4.16)$$

Generally, the computation of equation 4.16 depends on the expression of component densities  $f_j(x_i|\theta_j)$ , and parameters  $\theta_1, \dots, \theta_k$  need to be estimated in a case by case manner. In the best situation, each component density is from a well-known distribution and it has a simple analytical form of PDF, where we can obtain a closed form expression for  $\theta_j$  directly, however, in some case, the component density is very complicated, and we usually need to use numerical method to solve equation 4.16 accordingly.

### 4.2.1.1 Example 5.1

We consider the three nodes parallel flowgraph model for the progression of cancer patients in Section 2.1.2.2, Chapter 2 again. The distribution of internode waiting time of flowgraph in Figure 2.2, page 9, is given in Table 4.1.

Table 4.1: Summary of waiting time distribution

Flow direction	Distribution	MGF
0 → 1	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
1 → 2	Exponential( $\lambda_2$ )	$m_{12}(s) = \frac{\lambda_2}{\lambda_2 - s}$
0 → 2	Exponential( $\lambda_3$ )	$m_{02}(s) = \frac{\lambda_3}{\lambda_3 - s}$

By Mason's rule, the MGF of overall waiting time distribution between node 0 and node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s) \\ &= p_{01}\left(\frac{\lambda_1}{\lambda_1 - s}\right)\left(\frac{\lambda_2}{\lambda_2 - s}\right) + p_{02}\left(\frac{\lambda_3}{\lambda_3 - s}\right) \end{aligned}$$

By the exact inversion method discussed in Section 3.1, we obtain the probability density function of total waiting time from node 0 to node 2 in the form of mixture of two densities such that

$$\begin{aligned} f(t|\theta) &= p_{01}\frac{\lambda_1\lambda_2}{\lambda_1 - \lambda_2}(e^{-\lambda_2 t} - e^{-\lambda_1 t}) + p_{02}\lambda_3 e^{-\lambda_3 t} \\ &= (1 - p_{02})f_1(t | \theta_1) + p_{02}f_2(t | \theta_2) \\ &= \sum_{j=1}^2 w_j f_j(t | \theta_j) \end{aligned} \tag{4.17}$$

where  $f_1(t|\theta_1) = \frac{\lambda_1\lambda_2}{\lambda_1 - \lambda_2}(e^{-\lambda_2 t} - e^{-\lambda_1 t})$ ,  $f_2(t|\theta) = \lambda_3 e^{-\lambda_3 t}$ , the parameter vector is  $\theta = (w_1, w_2; \theta_1, \theta_2)$ ,  $\theta_1 = (\lambda_1, \lambda_2)$ ,  $\theta_2 = \lambda_3$ , and the mixture model weights satisfy  $\sum_{j=1}^2 w_j = w_1 + w_2 = p_{01} + p_{02} = 1$ .

Given the total waiting time data  $t_1, \dots, t_n$ , the log-likelihood function is

$$\begin{aligned} \log(L(\theta)) &= \log\left(\prod_{i=1}^n f(t_i|\theta)\right) \\ &= \sum_{i=1}^n \log\left(\sum_{j=1}^2 w_j f_j(t_i | \theta_j)\right) \end{aligned} \quad (4.18)$$

Since the log likelihood function in parallel flowgraph contains the logarithm function of the sum of mixture densities, the derivation of the sum of log function with respect to  $\theta$  leads to a system of non-linear equations, which are computationally demanding to solve for  $\theta$ .

As it is illustrated in Figure 2.2, that there are two different paths from input node 0 to output 2 in our parallel flowgraph example, path 1 is  $0 \rightarrow 1 \rightarrow 2$  and path 2 is  $0 \rightarrow 2$ , the total waiting time data  $T = (t_1, \dots, t_n)$  can be viewed as incomplete data by assuming the existence of unobserved data  $Y = (y_1, \dots, y_n)$ , whose values indicate which path “generated ” each total waiting time, the EM algorithm can be easily applied to find the MLE of mixture parameters in the above parallel flowgraph.

Since  $f_2(t|\theta_2)$  is in the form of the PDF of exponential distribution, which is much simpler than  $f_1(t|\theta_1)$ , and  $p_{01} = 1 - p_{02}$ , then it is more convenient to update  $p_{02}$  first. By the use of formula given in equation 4.15, we have

$$\begin{aligned} p_{02}^{(m+1)} &= \frac{1}{n} \sum_{i=1}^n f(2|t_i, \theta_2^{(m)}) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{w_2^{(m)} f_2(t_i|\lambda_3^{(m)})}{\sum_{l=1}^2 w_l^{(m)} f_l(t_i|\theta_l^{(m)})} \\ &= \frac{1}{n} \sum_{i=1}^n \frac{p_{02}^{(m)} f_2(t_i|\lambda_3^{(m)})}{(1 - p_{02}^{(m)}) f_1(t_i|\lambda_1^{(m)}, \lambda_2^{(m)}) + p_{02}^{(m)} f_2(t_i|\lambda_3^{(m)})} \end{aligned} \quad (4.19)$$

To find the parameter  $\theta_1 = (\lambda_1, \lambda_2)$  and  $\theta_2 = \lambda_3$  in each component density,

let

$$\begin{aligned} Q(\theta, \theta^{(m)}) &= Q(\theta, \theta_1^{(m)}) + Q(\theta, \theta_2^{(m)}) \\ &= \sum_{i=1}^n \log(f_1(t_i|\theta_1))f(1|t_i, \Theta^{(m)}) \\ &\quad + \sum_{i=1}^n \log(f_2(t_i|\theta_2))f(2|t_i, \Theta^{(m)}) \end{aligned}$$

For  $\theta_2 = \lambda_3$ , we substitute  $f_2(x|\theta_2)$  in  $Q(\theta, \theta_2^{(m)})$ ,

$$\begin{aligned} Q(\theta, \theta_2^{(m)}) &= \sum_{i=1}^n \log(\lambda_3 e^{-\lambda_3 t_i}) f(2|t_i, \Theta^{(m)}) \\ &= \sum_{i=1}^n (\log(\lambda_3) - \lambda_3 t_i) f(2|t_i, \Theta^{(m)}) \end{aligned} \quad (4.20)$$

Taking the derivative of 4.20 with respect to  $\lambda_3$  and equating to zero, we have

$$\begin{aligned} \sum_{i=1}^n \left( \frac{1}{\lambda_3} - t_i \right) f(2|t_i, \Theta^{(m)}) &= 0 \\ \frac{\sum_{i=1}^n f(2|t_i, \Theta^{(m)})}{\lambda_3} - \sum_{i=1}^n t_i f(2|t_i, \Theta^{(m)}) &= 0 \\ \lambda_3^{(m+1)} &= \frac{\sum_{i=1}^n f(2|t_i, \Theta^{(m)})}{\sum_{i=1}^n t_i f(2|t_i, \Theta^{(m)})} \end{aligned} \quad (4.21)$$

where

$$f(2|t_i, \Theta^{(m)}) = \frac{p_{02}^{(m)} f_2(t_i|\lambda_3^{(m)})}{(1 - p_{02}^{(m)}) f_1(t_i|\lambda_1^{(m)}, \lambda_2^{(m)}) + p_{02}^{(m)} f_2(t_i|\lambda_3^{(m)})}$$

Similarly, for the update of  $\theta_1 = (\lambda_1, \lambda_2)$ , we substitute  $f_1(x|\theta_1)$  in  $Q(\theta, \theta_1^{(m)})$ ,

$$\begin{aligned} Q(\theta, \theta_1^{(m)}) &= \sum_{i=1}^n \log\left(\frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}\right) \left(e^{-\lambda_2 t_i} - e^{-\lambda_1 t_i}\right) f(1|t_i, \Theta^{(0)}) \\ &= \sum_{i=1}^n \left\{ \log(\lambda_1) + \log(\lambda_2) - \log(\lambda_1 - \lambda_2) + \log(e^{-\lambda_2 t_i} - e^{-\lambda_1 t_i}) \right\} f(1|t_i, \Theta^{(0)}) \end{aligned} \quad (4.22)$$



and solve the following partial derivative equations,

$$\frac{\partial Q}{\partial \lambda_1} = \sum_{i=1}^n \left( \frac{1}{\lambda_1} - \frac{1}{\lambda_1 - \lambda_2} + \frac{t_i e^{-\lambda_1 t_i}}{e^{-\lambda_2 t_i} - e^{-\lambda_1 t_i}} \right) f(1|t_i, \Theta^{(0)}) = 0 \quad (4.23)$$

$$\frac{\partial Q}{\partial \lambda_2} = \sum_{i=1}^n \left( \frac{1}{\lambda_2} + \frac{1}{\lambda_1 - \lambda_2} - \frac{t_i e^{-\lambda_2 t_i}}{e^{-\lambda_2 t_i} - e^{-\lambda_1 t_i}} \right) f(1|t_i, \Theta^{(0)}) = 0 \quad (4.24)$$

where

$$f(1|t_i, \Theta^{(0)}) = \frac{(1 - p_{02}^{(m)}) f_1(t_i | \lambda_1^{(m)}, \lambda_2^{(m)})}{(1 - p_{02}^{(m)}) f_1(t_i | \lambda_1^{(m)}, \lambda_2^{(m)}) + p_{02}^{(m)} f_2(t_i | \lambda_3^{(m)})}$$

In this case, it will not be possible to obtain analytic expressions for  $\lambda_1$  and  $\lambda_2$  in terms of all other parameters, then numerical method such as Newton-Raphson method is required to solve equation 4.23 and 4.24 for  $\lambda_1$  and  $\lambda_2$ .

Given  $\Theta^{(m)} = (1 - p_{02}^{(m)}, p_{02}^{(m)}, \lambda_1^{(m)}, \lambda_2^{(m)}, \lambda_3^{(m)})$ , we compute  $p_{02}^{(m+1)}$  and  $\lambda_3^{(m+1)}$  by explicit formula 4.19 and 4.21 respectively, determine  $\lambda_1^{(m+1)}$  and  $\lambda_2^{(m+1)}$  by solving 4.23 and 4.24 numerically, and update  $\Theta^{(m)}$  to  $\Theta^{(m+1)}$ . The iterations will not stop until the convergence of log-likelihood function given in 4.18 occurs, where the change of  $\log(L(\theta))$  is smaller than a pre-specified amount  $\delta$

$$\log(L(\Theta^{(m+1)})) - \log(L(\Theta^{(m)})) \leq \delta \quad (4.25)$$

By assuming the true parameter values as:  $p_{02} = 0.3$ ,  $p_{01} = 1 - p_{02} = 0.7$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 2$ ,  $\lambda_3 = 3$ , and simulate a sample size of 20000 total waiting time data from node 0 to node 2 in the parallel flowgraph. Starting from suitable initial value for  $\theta$ , we set  $\delta = 10^{-6}$  in equation 4.25 as the tolerance value. The convergence of log-likelihood function occurs after 1419 iterations. The estimation result is presented in Table 4.2.

Table 4.2: Summary of MLE by EM algorithm

Sample size	$\hat{p}_{01}$	$\hat{p}_{02}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$
$n = 20000$	0.7006	0.2994	1.0254	1.9053	2.9866

### 4.2.1.2 Example 5.2

Suppose the parallel flowgraph illustrated in Figure 2.2 consists of non-exponentially distributed internode waiting time (see Table 4.3), the MGF of total waiting time between input node 0 and output node 2 can not be decomposed into sum of rational function by partial fractions, and it therefore can not be inverted directly to obtain closed form density function. In this case, the likelihood function is not in closed form and the standard procedure for computing MLE can not be applied. We need to first numerically invert the MGF based on the numerical inversion of Laplace transforms method (see De Hoog (1982)), then apply the Nelder-Mead simplex method to compute MLE numerically. For the purpose of illustration, we allow the waiting time between node 1 and node 2 to follow a Gamma distribution with non-integer shape parameter (see Table 4.3).

Table 4.3: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$1 \rightarrow 2$	Gamma( $\alpha, \beta$ )	$m_{12}(s) = \left(\frac{\beta}{\beta - s}\right)^\alpha$
$0 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{02}(s) = \frac{\lambda_2}{\lambda_2 - s}$

The MGF of total waiting time from node 0 to node 2 becomes

$$\begin{aligned}
 M_{T_{02}}(s) &= p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s) \\
 &= p_{01}m_{01}(s)m_{12}(s) + (1 - p_{01})m_{02}(s) \\
 &= p_{01} \left(\frac{\lambda_1}{\lambda_1 - s}\right) \left(\frac{\beta}{\beta - s}\right)^\alpha + (1 - p_{01}) \left(\frac{\lambda_2}{\lambda_2 - s}\right) \quad (4.26)
 \end{aligned}$$

Assuming the value of true parameters:  $p_{01} = 0.7$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\alpha = 2.5$ ,  $\beta = 3$ , we simulated a sample size of 10000 total waiting time data from node 0 to node 2 in Figure 2.2. The first step of estimation is to numerically invert the MGF given in equation 4.26 to obtain an approximated probability density function  $\hat{f}(t)$ , compute the likelihood function based on  $\hat{f}(t)$ , and then apply the Nelder-Mead simplex method to compute MLE numerically. The result is shown in Table 4.4.

Table 4.4: Summary of MLE in non-exponential case

Sample size	$\hat{p}_{01}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\alpha}$	$\hat{\beta}$
$n = 10000$	0.75	1.2	1.8	2.4	2.9

### 4.2.2 Case 2: Mixture with negative weights

As it is shown in Section 4.2.1.1, the EM algorithm significantly simplifies the calculation of Maximum likelihood estimation for finite mixture distribution. However, the PDF of total waiting time could be a mixture of exponential distribution with negative weights, for example equation 3.5 on page 43, this forms our motivation to consider parameter estimation problem in a more difficult case where the weights of component densities in a mixture density are allowed to be negative value. A reference on the maximum likelihood estimation for these more general mixed exponential densities can be found in Harris and Sykes (1987)).

**Definition 10.** *A mixture of negative weight exponential density function for a positive random variable  $X$  is defined as a weighted sum of  $k$  component densities:*

$$f(x) = \sum_{l=1}^k w_l f_l(x|\alpha_l)$$

where

$$f_l(x|\alpha_l) = \alpha_l e^{-\alpha_l x}$$

is the probability density functions of exponential distribution with  $\alpha_l > 0$  for  $l = 1, \dots, k$ , such that  $\alpha_1 < \alpha_2 < \dots < \alpha_k$ , and the weights satisfy two constraints:

1. At least one  $w_l \notin [0, 1]$  for some  $l$ .
2.  $\sum_{l=1}^k w_l = 1$

Bartholomew (1969) developed a simple condition to verify whether or not a mixture of exponentials is a proper probability density function (see Bartholomew (1969), page 2184, Theorem 1), which plays an important role in the our derivation of our EM algorithm for mixture of negative weights exponential density. We first present the original proof given in Bartholomew (1969).

**Theorem 6. Sufficient conditions (Bartholomew)**

Given a function in the form of

$$f(x) = \sum_{l=1}^k w_l \alpha_l e^{-\alpha_l x} \quad (4.27)$$

where the  $\alpha$ 's are all positive and  $\sum_{l=1}^k w_l = 1$ . The sufficient condition to ensure that  $f(x)$  in 4.27 is a proper probability density function is

$$\sum_{l=1}^r w_l \alpha_l \geq 0 \quad \text{for } r = 1, 2, \dots, k \quad (4.28)$$

*Proof.* The alternative expression of 4.27 can be derived as

$$f(x) = e^{-\alpha_k x} \sum_{l=1}^k w_l \alpha_l + \sum_{r=1}^{k-1} \left\{ (e^{-\alpha_r x} - e^{-\alpha_{r+1} x}) \sum_{l=1}^r w_l \alpha_l \right\} \quad (4.29)$$

Since  $\alpha_1 < \alpha_2 < \dots < \alpha_k$ , it follows that

$$e^{-\alpha_r x} - e^{-\alpha_{r+1} x} \geq 0 \quad \text{for } r = 1, 2, \dots, k-1.$$

Hence  $f(x)$  given in equation 4.29 is positive for all  $x$  if

$$\sum_{l=1}^r w_l \alpha_l \geq 0 \quad \text{for } r = 1, 2, \dots, k.$$

□

To apply the EM algorithm for estimating the parameters of a mixture of negative weights exponential density function, we suggest a method to convert the density defined in Definition 10 into a mixture of densities with positive weight, so that the modified mixture density will be suitable for implementing EM algorithm.

**Theorem 7.** *Given a mixture of negative weights density  $f(x)$  defined in Definition 10 with rates in ascending order  $\alpha_1 < \alpha_2 < \dots < \alpha_k$ , and  $f_1(t|\alpha_1)$  is the corresponding component density with weight  $w_1$  and the smallest rate  $\alpha_1$ . If  $f(x)$  satisfies the Bartholomew's sufficient conditions to be a PDF, then an alternative expression of  $f(x)$  can be determined as*

$$f(x) = \sum_{j=1}^k \pi_j g_j(x) \quad (4.30)$$

where  $g_1(t) = f_1(t|\alpha_1)$ ,  $\pi_1 = 1 - \sum_{j=2}^k \pi_j$ ,

$$\pi_j = w_j \left(1 - \frac{\alpha_j}{\alpha_1}\right)$$

$w_j$  is the negative weight for the  $j$ -th component densities  $f_j(x|\alpha_j)$  and

$$g_j(x) = \frac{\alpha_1 \alpha_j (e^{-\alpha_j t} - e^{-\alpha_1 t})}{\alpha_1 - \alpha_j}$$

for some  $j \in \{2, \dots, k\}$ . Then equation 4.30 satisfies the following properties

1.  $\pi_j > 0$  for all  $j = 1, \dots, k$ .

2.  $\sum_{j=1}^k \pi_j = 1$

*Proof.* By Definition 10, we have the total weight  $\sum_{l=1}^k w_l = 1$  while some  $w_l$  are negative, then there must exist at least one  $w^*$  such that  $w^* > 0$ . Furthermore, Theorem 6 states that  $f(x)$  is a valid probability density function if condition 4.28 is satisfied. As  $\alpha_1 > 0$ , in order to ensure that  $f(x)$  is a proper density function,  $w_1$  must be a nonnegative value so that we have  $w_1 \alpha_1 \geq 0$  (i.e. condition in 4.28 for  $r = 1$  case). That is, the weight for the component density with the smallest rate is always positive. Let  $v$  be the total number of component density that associates with negative weight such that  $v \leq k - 1$ , label each of them by index  $j$  for some  $j \in \{2, \dots, k\}$ .

The basic idea is to partition  $w^*$  into  $c_1, \dots, c_{v+1}$ , where  $\sum_{z=1}^{v+1} c_z = w^*$ , and redistribute them to the  $j$ -th component densities involves negative weight by constructing a nonnegative function  $c_j f_1(t) + w_j f_j(t)$  for some  $j \in \{2, \dots, k\}$ . To determine  $c_j$ , we need to have

$$\begin{aligned} c_j f_1(t) + w_j f_j(t) &\geq 0 \\ c_j &\geq \frac{-w_j f_j(t)}{f_1(t)} \quad \text{for } j \in \{2, \dots, k\} \end{aligned}$$

Now

$$-w_j \frac{f_j(t)}{f_1(t)} = -w_j \frac{\alpha_j e^{-\alpha_j t}}{\alpha_1 e^{-\alpha_1 t}} = -w_j \frac{\alpha_j}{\alpha_1} e^{(\alpha_1 - \alpha_j)t}$$

As  $0 < \alpha_1 < \alpha_j$ , then  $e^{(\alpha_1 - \alpha_j)t} \rightarrow 0$ , as  $t \rightarrow \infty$ . Since  $-w_j > 0$ , then  $-w_j \frac{f_j(t)}{f_1(t)}$  decreases as  $t$  increase, giving the least upper bound for  $-w_j \frac{f_j(t)}{f_1(t)}$  can be obtained at  $t = 0$ , therefore we choose

$$\begin{aligned} c_j &= \left( \frac{-w_j f_j(t)}{f_1(t)} \right)_{t=0} \\ &= \frac{-w_j \alpha_j e^{-\alpha_j 0}}{\alpha_1 e^{-\alpha_1 0}} \\ &= \frac{-w_j \alpha_j}{\alpha_1} \quad \text{for } j \in \{2, \dots, k\} \end{aligned} \tag{4.31}$$

Define function  $g_j(t)$  in the form of

$$g_j(t) = c_j f_1(t) + w_j f_j(t) \quad \text{for } j \in \{2, \dots, k\}$$

such that  $g_j(t) \geq 0$  for all  $t$ . Since  $f_1(t)$  and  $f_2(t)$  are the probability density function of exponential distribution, then

$$\begin{aligned} \int_0^\infty g_j(t) &= c_j \int_0^\infty f_1(t) dt + w_j \int_0^\infty f_j(t) dt \\ &= c_j + w_j \end{aligned}$$

Hence we can replace  $f_j(t)$  with the normalised function  $g_j(t)$ ,

$$g_j(t) = \frac{c_j f_1(t) + w_j f_j(t)}{c_j + w_j} \tag{4.32}$$

Define  $\pi_j = c_j + w_j$  and  $c_j$  is defined in equation 4.31,

$$\begin{aligned}
 \pi_j &= c_j + w_j \\
 &= \frac{-w_j\alpha_j}{\alpha_1} + w_j \\
 &= w_j \left( 1 - \frac{\alpha_j}{\alpha_1} \right)
 \end{aligned} \tag{4.33}$$

Since  $\alpha_1$  is the smallest rate, we have  $\alpha_1 < \alpha_j$  implies  $\left( 1 - \frac{\alpha_j}{\alpha_1} \right) < 0$ , and  $w_j < 0$ , then

$$\pi_j = w_j \left( 1 - \frac{\alpha_j}{\alpha_1} \right) > 0 \quad \text{for } j \in \{2, \dots, k\}.$$

We can also further simplify 4.32 by substituting 4.31,

$$\begin{aligned}
 g_j(t) &= \frac{\frac{-w_j\alpha_j}{\alpha_1} f_1(t) + w_j f_j(t)}{\frac{-w_j\alpha_j}{\alpha_1} + w_j} \\
 &= \frac{-w_j\alpha_j f_1(t) + w_j\alpha_1 f_j(t)}{-w_j\alpha_j + w_j\alpha_1} \\
 &= \frac{-w_j\alpha_j\alpha_1 e^{-\alpha_1 t} + w_j\alpha_1\alpha_j e^{-\alpha_j t}}{-w_j\alpha_j + w_j\alpha_1} \\
 &= \frac{\alpha_1\alpha_j(e^{-\alpha_j t} - e^{-\alpha_1 t})}{\alpha_1 - \alpha_j} \quad \text{for } j \in \{2, \dots, k\}.
 \end{aligned} \tag{4.34}$$

Hence the alternative expression of mixture negative function is

$$f(t) = \pi_1 g_1(t) + \pi_2 g_2(t) + \dots + \pi_k g_k(t)$$

where  $g_1(t) = f_1(t)$ ,  $\pi_1 = 1 - \sum_{j=2}^k \pi_j$ ,  $g_j(t)$  and  $\pi_j$  are defined by 4.34 and 4.33 respectively for  $j \in \{2, \dots, k\}$ .  $\square$

**Remark:** We can also check whether  $g_j(t)$ ,  $j = 2, \dots, k$ , is a proper probability density function by condition 4.28. Given

$$g_j(t) = -\frac{\alpha_j}{\alpha_1 - \alpha_j} \alpha_1 e^{-\alpha_1 t} + \frac{\alpha_1}{\alpha_1 - \alpha_j} \alpha_j e^{-\alpha_j t}$$

Since  $\alpha_1 < \alpha_j$ , we have

$$p_1\alpha_1 = -\frac{\alpha_j}{\alpha_1 - \alpha_j}\alpha_1 > 0 \text{ for } r = 1 \text{ case}$$

$$\sum_{i=1}^2 p_i\alpha_i = -\frac{\alpha_j}{\alpha_1 - \alpha_j}\alpha_1 + \frac{\alpha_1}{\alpha_1 - \alpha_j}\alpha_j = 0 \text{ for } r = 2 \text{ case}$$

By Theorem 6, we conclude that  $g_j(t)$  is a valid probability density function for  $j = 2, \dots, k$ .

#### 4.2.2.1 Example 1.3 continued

In this example, we apply Theorem 7 and EM algorithm to estimate parameters in PDF of total waiting time from node 0 to node 2 in the flowgraph for 3 nodes reversible illness-death system example in Section 3.1.1 Chapter 3. For the purpose of illustration, we assumed  $\lambda_1 = 1$ ,  $\lambda_2 = 1.2$ ,  $\lambda_3 = 0.5$ ,  $\lambda_4 = 2$ , and  $p_{01} = p_{02} = p_{10} = p_{12} = 0.5$  in the following flowgraph.

By equation 3.5 on page 43, the resulting probability density function  $f_{T_{02}}(t)$  is a mixture of exponential densities that contains negative weight.

$$f_{T_{02}}(t) = \sum_{j=1}^4 w_j f_j(t|\alpha_j) \quad (4.35)$$

where

$$f_j(t|\alpha_j) = \alpha_j e^{-\alpha_j t}$$

and  $\theta = (w, \alpha)$  such that  $w = (w_1, w_2, w_3, w_4) = (1.5, -0.303, -0.1915, -0.0066)$ , and  $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.5, 1.2, 0.6339, 2.366)$ .

It is important to note that we only concentrate on the estimation of parameters  $\theta = (w, \alpha)$  in the density function  $f_{T_{02}}(t)$  rather than those parameters in the flowgraph. (i.e.  $\lambda$ 's and transition probabilities in Figure 2.4). According to Theorem 7, as  $0.5 < 0.6339 < 1.2 < 2.366$ , we set  $\alpha_1 = 0.5$  and  $w^* = 1.5$ . The alternative expression can be constructed by partitioning  $1.5f_1(t)$ , and re-



distributing to other component densities that have negative weights, then

$$f_{T_{02}}(t) = \sum_{j=1}^4 \pi_j g_j(t) \quad (4.36)$$

where

$$\pi_j = w_j \left( 1 - \frac{\alpha_j}{\alpha_1} \right) \quad (4.37)$$

and  $\pi = (\pi_1, \pi_2, \pi_3, \pi_4) = (0.5, 0.4242, 0.051, 0.0248)$ . The modified component densities are defined by

$$g_j(t) = \frac{\alpha_1 \alpha_j (e^{-\alpha_j t} - e^{-\alpha_1 t})}{\alpha_1 - \alpha_j} \quad (4.38)$$

By substituting  $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.5, 0.6339, 1.2, 2.366)$  in equation 4.38 accordingly, we obtain

$$\begin{aligned} g_2(t) &= 0.8571 (e^{-0.5t} - e^{-1.2t}) \\ g_3(t) &= 2.366 (e^{-0.5t} - e^{-0.6339t}) \\ g_4(t) &= 0.6339 (e^{-0.5t} - e^{-2.366t}) \\ g_1(t) &= f_1(t) = 0.5e^{-0.5t} \end{aligned}$$

Now we can apply the EM algorithm discussed in Section 4.2 to estimate parameters  $\tilde{\theta}_j = (\pi_j, \alpha_j)$ , for  $j = 1, 2, 3, 4$ , in 4.36. The weight  $\pi_j$  can be updated iteratively by formula given in equation 4.15

$$\begin{aligned} \pi_j^{(m+1)} &= \frac{1}{n} \sum_{i=1}^n g(j|t_i, \alpha_i^{(m)}) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\pi_j^{(m)} g_j(t_i | \alpha_i^{(m)})}{\sum_{l=1}^4 \pi_l^{(m)} g_l(t_i | \alpha_i^{(m)})} \end{aligned} \quad (4.39)$$

As the first component density  $g_1(t)$  is the PDF of exponential distribution, which has relatively simple expression than the other three component densities, then

we first estimate its rate  $\alpha_1$  by the formula derived in 4.21,

$$\alpha_1^{(m+1)} = \frac{\sum_{i=1}^n g(1|t_i, \tilde{\theta}^{(m)})}{\sum_{i=1}^n t_i g(1|t_i, \tilde{\theta}^{(m)})} \quad (4.40)$$

where

$$g(1|t_i, \tilde{\theta}^{(m)}) = \frac{\pi_1^{(m)} g_1(t_i|\alpha_1^{(m)})}{\sum_{l=1}^4 \pi_l^{(m)} g_l(t_i|\alpha_l^{(m)})}$$

Since  $g_j(t)$  is a function of  $\alpha_1$  and  $\alpha_j$  (see equation 4.38), and it is in the same form of  $f_1(t | \alpha_1)$  in the mixture of density given by 4.17, we can therefore update each  $\alpha_j$  for  $j = 2, 3, 4$  by the same approach in equation 4.22. Define

$$Q(\theta, \tilde{\theta}_j^{(m)}) = \sum_{i=1}^n \log \left\{ \left( \frac{\alpha_1 \alpha_j}{\alpha_1 - \alpha_j} \right) \left( e^{-\alpha_j t_i} - e^{-\alpha_1 t_i} \right) \right\} g(j|t_i, \tilde{\theta}^{(m)}) \quad (4.41)$$

To speed up the calculation, we can replace the starting value  $\alpha_1^{(m)}$  and  $\pi_1^{(m)}$  in 4.41 by the updated value  $\pi_1^{(m+1)}$  and  $\alpha_1^{(m+1)}$  obtained from 4.39 and 4.40 respectively, and solve the following equation for  $\alpha_j$

$$\frac{\partial Q}{\partial \alpha_j} = \sum_{i=1}^n \left[ \frac{1}{\alpha_j} + \frac{1}{\alpha_1^{(m+1)} - \alpha_j} - \frac{t_i e^{-\alpha_j t_i}}{e^{-\alpha_j t_i} - e^{-\alpha_1^{(m+1)} t_i}} \right] g(j|t_i, \tilde{\theta}^{(m)}) = 0$$

where

$$g(j|t_i, \tilde{\theta}^{(m)}) = \frac{\pi_j^{(m)} g_j(t_i|\alpha_j^{(m)})}{p_1^{(m+1)} g_1(t_i|\alpha_1^{(m+1)}) + \sum_{l=2}^4 \pi_l^{(m)} g_l(t_i|\tilde{\theta}_l^{(m)})}$$

The updated value for the original weight  $w_j$  can be derived from equation 4.37.

$$w_j^{(m+1)} = \frac{\pi_j^{(m+1)}}{1 - \frac{\alpha_j^{(m+1)}}{\alpha_1^{(m+1)}}} \quad \text{for } j = 2, 3, 4. \quad (4.42)$$

As the conversion from mixture negative weight density to mixture positive weight densities is a fairly straightforward computational procedure, it can be broadly applied within the framework of the EM algorithm. Given the initial value of  $\tilde{\theta}$ ,  $\tilde{\theta}^{(m)} = (w^{(m)}, \alpha^{(m)})$ , the general procedure of parameter estimation for mixture negative weight density is as follow:

1. Convert the negative weight mixture density into positive weight mixture density by Theorem 7, define  $\theta^{(m)} \rightarrow \tilde{\theta}^{(m)} = (\pi^{(m)}, \alpha^{(m)})$
2. Update  $\tilde{\theta}^{(m)} \rightarrow \tilde{\theta}^{(m+1)}$  by EM algorithm until likelihood function converges.
3. Transform back to original parameters  $\tilde{\theta}^{(m+1)} \rightarrow \theta^{(m+1)} = (w^{(m+1)}, \alpha^{(m+1)})$ , where  $w^{(m+1)}$  can be determined by 4.42.

#### 4.2.2.2 Simulation

As the probability density function of total waiting time is obtained in the form of 4.35, in order to test our method in Theorem 7 to estimate its parameters  $w$ 's and  $\alpha$ 's, we need to simulate the total waiting time data between input at node 0 and output at node 2 of flowgraph in Figure 2.5 on page 17. To generate a sample size of  $n$  total waiting time data  $t_1, \dots, t_n$ , we need to specify the internode waiting distribution, label the flow direction between each node, as well as the corresponding transition probability (see Table 4.5). We first need to simulate the path of  $n$  particles from input node 0 to output node 2, and then compute each total waiting time  $t_i, i = 1, \dots, n$ .

Table 4.5: Summary of waiting time distribution

Label	Flow direction	Probability	Distribution
1	$0 \rightarrow 2$	$p_{02}$	Exponential( $\lambda_3$ )
2	$0 \rightarrow 1$	$p_{01}$	Exponential( $\lambda_1$ )
3	$1 \rightarrow 0$	$p_{10}$	Exponential( $\lambda_4$ )
4	$1 \rightarrow 2$	$p_{12}$	Exponential( $\lambda_2$ )

The procedure of simulation is described as follow

1. At node 0, take a size of  $n$  weighted sample with replacement from set  $\{1, 2\}$ , using a vector of probabilities  $\{p_{02}, p_{01}\}$ , where the integer 1 or 2 is selected with probability  $p_{02}$  or  $p_{01}$ . Denote the total sample size of result equal to 1 and 2 by  $s_1$  and  $s_2$  respectively, and  $s_1 + s_2 = n$ .
2. For those result equal to 2, take a size of  $s_2$  weighted sample with replacement from set  $\{3, 4\}$ , using a vector of probabilities  $\{p_{10}, p_{12}\}$ , where the

integer 3 or 4 is selected with probability  $p_{10}$  or  $p_{12}$ . Denote the total sample size of result equal to 3 and 4 by  $s_3$  and  $s_4$  respectively, and  $s_3 + s_4 = s_2$ .

3. Set  $n = s_3$ , stop simulation if  $s_3 = 0$ , otherwise go to 4.
4. Repeat Step 1, 2 and 3 until  $s_3 = s_2 = 0$ . (i.e. all the particle reach output node 2).

We construct a  $m \times n$  matrix  $D$  to record the path for each particle from input node 0 to output node 2, where the column represents index of sample data and the row represents the direction of movement that each particle made before it gets to output node 2, that is,  $d_{ij}$  denotes the flowgraph direction that the  $j$ -th particle chose in the  $i$ -th movement. Next, we generate internode waiting time for each particle in every step of movement according to the given distribution in Table 4.5, defined a  $m \times n$  matrix  $G$  such that  $g_{ij}$  represents the passage time of  $j$ -th particle in the  $i$ -th movement, then the total waiting time for the  $j$ -th particle to reach output node 2 is the column sum of matrix  $G$ .

For example, suppose we want to simulate 5 total waiting time data. First, the path of 5 particles from input node 0 to output 2 are recorded in matrix  $D$  below, the simulation of path is terminated once the particle follow either flow direction 1 (i.e.  $0 \rightarrow 2$ ) or 4 (i.e.  $1 \rightarrow 2$ ). From matrix  $D$ , we see that it took a total of 6 steps of transition for all the particles reach output node 2. Next, the passage time for each transition in matrix  $D$  are simulated and stored in matrix  $G$ . Therefore, we can then obtain 5 total waiting time  $t = (0.5803, 2.8013, 3.6601, 3.3988, 4.8463)$  after taking the column sum of matrix  $G$ .

$$D = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 1 & 2 & 2 & 2 & 2 \\ 0 & 3 & 3 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \end{matrix} \rightarrow G = \begin{pmatrix} 0.5803 & 2.5775 & 3.0027 & 0.1053 & 0.9066 \\ 0 & 0.2161 & 0.1474 & 0.1241 & 0.4028 \\ 0 & 0.0077 & 0.5100 & 2.1964 & 2.0252 \\ 0 & 0 & 0 & 0.9730 & 0.3381 \\ 0 & 0 & 0 & 0 & 0.0449 \\ 0 & 0 & 0 & 0 & 1.1288 \end{pmatrix}$$

With this procedure, we simulate a random sample of  $n = 10000$  total waiting time data  $t_1, \dots, t_n$  between node 0 and node 2 of the flowgraph in Figure 2.4 for parameter estimation. Then we choose the initial value for the EM algorithm as

$$w^{(0)} = (w_1^{(0)}, w_2^{(0)}, w_3^{(0)}, w_4^{(0)}) = (0.4971, -0.3020, -0.1933, -0.0076) \quad (4.43)$$

$$\alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}, \alpha_3^{(0)}, \alpha_4^{(0)}) = (0.5020, 1.2030, 0.6370, 2.3680) \quad (4.44)$$

The stopping criterion for EM algorithm was set based on the change in the log-likelihood function is less than a tolerance value of  $10^{-5}$ .

$$\log(L(\theta^{(m+1)}|t)) - \log(L(\theta^{(m)}|t)) \leq 10^{-5}$$

Note that the log-likelihood function  $\log(L(\theta^{(m)}|t))$  tends to have multiple local maxima in this example. Therefore we need to randomly select various initial value for parameters with constrains: 1)  $0 < \pi_j < 1$ , 2)  $\alpha_j > 0$ , and choose suitably starting point in order to avoid converging to spurious maxima. Given 4.43 and 4.44, the log-likelihood converges after 2145 iterations and the results are listed in the Tables 4.6 and 4.7. We observe that the estimation given by the EM algorithm is sufficiently close to true value of parameters, and the difference between is generally small.

Table 4.6: Summary of MLE by EM algorithm

parameter	$w_1$	$w_2$	$w_3$	$w_4$
True value	1.5	-0.3030	-0.1903	-0.0066
Estimated value	1.5029	-0.3021	-0.1932	-0.0076

Table 4.7: Summary of MLE by EM algorithm

parameter	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$
True value	0.5	1.2	0.6339	2.366
Estimated value	0.5007	1.1890	0.6385	2.6581

### 4.3 Method of Moments

The Method of Moments (MM) is based on the idea that statistical distribution can be uniquely characterised by their moments, provided the moments are finite and satisfy the Carleman's condition (see Sjödin (1987)). Although we can uniquely determine many distributions such as normal and exponential, there are some distributions, such as the Log-normal distribution, that can not be identified given their moments, see Heyde (1963), Feller(1971), Stoyanov (1997). The MM is closely related to the Stieltjes moment problem, where we find a distribution function  $F(x)$  on  $[0, \infty)$  such that

$$\mu_r = \int_0^{\infty} x^r dF(x)$$

given a sequence of finite moments  $\{\mu_r, r = 0, 1, \dots, d\}$ . The implementation of Method of Moments requires equating a set of sample moments to their population moments, and then solves the set of (generally nonlinear) equations for the parameters in the target distribution.

**Definition 11.** *let  $x_1, \dots, x_n$  be iid sample from a distribution with  $d$ -dimensional parameter  $\theta = (\theta_1, \dots, \theta_d)$ . The Method of Moment (MM) estimator  $\hat{\theta}$  is the solution to the following system of equations*

$$\mu_r(\theta) = m_r$$

where

$$\begin{aligned} \mu_r(\theta) &= \left. \frac{d^r M_x(s)}{ds^r} \right|_{s=0} \\ m_r &= \frac{1}{n} \sum_{i=1}^n x_i^r \end{aligned}$$

for  $r = 1, \dots, d$ .

Note that we use the formula 2.15 on page 32 to compute the population moment  $\mu_r$  symbolically.

In a large complicated flowgraph with non-exponentially distributed internode waiting time, it is often computationally demanding to approximate the total waiting time density function. Since the MM does not need the step to approximate the likelihood function numerically, it tends to be more easier to implement. Furthermore, the MGF of total waiting time between two nodes of interest in a complicated flowgraph can be obtained based on Theorem 3 (see page 30), and we can establish the population moment  $\mu_r$  by the use of Corollary 1 Chapter 2, which made the construction of a set of equations defined in 11 very simple.

### 4.3.1 Example 5.1 continued

We apply MM to estimate parameters in the parallel flowgraph illustrated by Figure 2.2 on page 9. The waiting time distribution is given in Table 4.1 on page 104. Since the MGF of overall waiting time distribution between node 0 and node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s) \\ &= p_{01}\left(\frac{\lambda_1}{\lambda_1 - s}\right)\left(\frac{\lambda_2}{\lambda_2 - s}\right) + p_{02}\left(\frac{\lambda_3}{\lambda_3 - s}\right) \\ &= p_{01}\left(\frac{\lambda_1}{\lambda_1 - s}\right)\left(\frac{\lambda_2}{\lambda_2 - s}\right) + (1 - p_{01})\left(\frac{\lambda_3}{\lambda_3 - s}\right) \end{aligned}$$

By the Definition 11, in order to estimate 4 parameters  $\theta = (p_{01}, \lambda_1, \lambda_2, \lambda_3)$ , we need to solve 4 equations that are constructed by equating sample moments with population moments

$$\left. \frac{d^r M_{T_{02}}(s)}{ds^r} \right|_{s=0} = \frac{1}{n} \sum_{i=1}^n t_i^r$$

for  $r = 1, \dots, 4$ , and

$$\mu_1(p_{01}, \lambda_1, \lambda_2, \lambda_3) = \frac{p_{01}}{\lambda_1} + \frac{p_{01}}{\lambda_2} + \frac{(1 - p_{01})}{\lambda_3} \quad (4.45)$$

$$\mu_2(p_{01}, \lambda_1, \lambda_2, \lambda_3) = \frac{2p_{01}}{\lambda_1^2} + \frac{2p_{01}}{\lambda_1\lambda_2} + \frac{2p_{01}}{\lambda_2^2} + \frac{2(1 - p_{01})}{\lambda_3^2} \quad (4.46)$$

$$\mu_3(p_{01}, \lambda_1, \lambda_2, \lambda_3) = \frac{6p_{01}}{\lambda_1^3} + \frac{6p_{01}}{\lambda_1^2\lambda_2} + \frac{6p_{01}}{\lambda_1\lambda_2^2} + \frac{6p_{01}}{\lambda_2^3} + \frac{6(1-p_{01})}{\lambda_3^3} \quad (4.47)$$

$$\begin{aligned} \mu_4(p_{01}, \lambda_1, \lambda_2, \lambda_3) &= \frac{24p_{01}}{\lambda_1^4} + \frac{24p_{01}}{\lambda_1^3\lambda_2} + \frac{24p_{01}}{\lambda_1^2\lambda_2^2} + \frac{24p_{01}}{\lambda_1\lambda_2^3} + \frac{24p_{01}}{\lambda_2^4} \\ &\quad + \frac{24(1-p_{01})}{\lambda_3^4} \end{aligned} \quad (4.48)$$

To make the subsequence calculation more stable, we reparametrises 4.45, 4.46, 4.47 and 4.48 by defining  $\theta_1 = p_{01}$ ,  $\theta_2 = \frac{1}{\lambda_1}$ ,  $\theta_3 = \frac{1}{\lambda_2}$ ,  $\theta_4 = \frac{1}{\lambda_3}$ , then

$$\begin{aligned} \mu_1(\theta_1, \theta_2, \theta_3, \theta_4) &= \theta_1\theta_2 + \theta_1\theta_3 + (1 - \theta_1)\theta_4 \\ \mu_2(\theta_1, \theta_2, \theta_3, \theta_4) &= 2\theta_1\theta_2^2 + 2\theta_1\theta_2\theta_3 + 2\theta_1\theta_3^2 + 2(1 - \theta_1)\theta_4^2 \\ \mu_3(\theta_1, \theta_2, \theta_3, \theta_4) &= 6\theta_1\theta_2^3 + 6\theta_1\theta_2^2\theta_3 + 6\theta_1\theta_2\theta_3^2 + 6\theta_1\theta_3^3 \\ &\quad + 6(1 - \theta_1)\theta_4^3 \\ \mu_4(\theta_1, \theta_2, \theta_3, \theta_4) &= 24\theta_1\theta_2^4 + 24\theta_1\theta_2^3\theta_3 + 24\theta_1\theta_2^2\theta_3^2 + 24\theta_1\theta_2\theta_3^3 \\ &\quad + 24\theta_1\theta_3^4 + 24(1 - \theta_1)\theta_4^4 \end{aligned}$$

To estimate the parameters in the total waiting time distribution, we use the same 10000 simulated total waiting time data in parallel flowgraph from Section 4.2.1.1, where true parameter values are assumed as:  $p_{01} = 0.7$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 2$ ,  $\lambda_3 = 3$ . The resulting sample moments based on the simulated data are computed as  $m_1 = 1.1679$ ,  $m_2 = 2.5974$ ,  $m_3 = 8.3192$ ,  $m_4 = 34.5268$ .



The method of moments estimators are determined by using the multivariate Newton-Raphson method in Section 3.2.1.1 to solve the moment equations.

$$\mu_r(\theta_1, \theta_2, \theta_3, \theta_4) = m_r$$

for  $r = 1, \dots, 4$ . The result is presented in Table 4.8.

Table 4.8: Summary of result by Method of Moments

Sample size	$\hat{p}_{01}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$
$n = 10000$	0.7060	1.0321	1.9457	3.0408

### 4.3.2 Example 5.2 continued

We continue with the parallel flowgraph example in Section 4.2.1.2, but assume the waiting time from node 1 to node 2 follows Gamma distribution. The parametric assumptions of all internode waiting time is shown in Table 4.9

Table 4.9: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$1 \rightarrow 2$	Gamma( $\alpha, \beta$ )	$m_{12}(s) = \left(\frac{\beta}{\beta - s}\right)^\alpha$
$0 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{02}(s) = \frac{\lambda_2}{\lambda_2 - s}$

The MGF of total waiting time from node 0 to node 2 is given in equation 4.26 on page 108.

$$\begin{aligned} M_{T_{02}}(s) &= p_{01} \left( \frac{\lambda_1}{\lambda_1 - s} \right) \left( \frac{\beta}{\beta - s} \right)^\alpha + (1 - p_{01}) \left( \frac{\lambda_2}{\lambda_2 - s} \right) \\ &= \frac{\beta^\alpha \lambda_1 p_{01} (\lambda_2 - s) + \lambda_2 (1 - p_{01}) (\lambda_1 - s) (\beta - s)^\alpha}{(\beta - s)^\alpha (\lambda_1 - s) (\lambda_2 - s)} \end{aligned}$$

Again, we need to construct 5 equations for estimating 5 parameters  $p_{01}$ ,  $\lambda_1$ ,  $\lambda_2$ ,  $\alpha$ ,  $\beta$  such that

$$\left. \frac{d^r M_{T_{02}}(s)}{ds^r} \right|_{s=0} = \frac{1}{n} \sum_{i=1}^n t_i^r \quad \text{for } r = 1, \dots, 5$$

where

$$\mu_1 = \frac{p_{01}}{\lambda_1} + \frac{p_{01}\alpha}{\beta} + \frac{1-p_{01}}{\lambda_2} \quad (4.49)$$

$$\mu_2 = \frac{2p_{01}}{\lambda_1^2} + \frac{2p_{01}\alpha}{\lambda_1\beta} + \frac{p_{01}\alpha^2}{\beta^2} + \frac{p_{01}\alpha}{\beta^2} + \frac{2(1-p_{01})}{\lambda_2^2} \quad (4.50)$$

$$\begin{aligned} \mu_3 = & \frac{6p_{01}}{\lambda_1^3} + \frac{6p_{01}\alpha}{\lambda_1^2\beta} + \frac{3p_{01}\alpha^2}{\lambda_1\beta^2} + \frac{3p_{01}\alpha}{\lambda_1\beta^2} + \frac{p_{01}\alpha^3}{\beta^3} + \frac{3p_{01}\alpha^2}{\beta^3} \\ & + \frac{2p_{01}\alpha}{\beta^3} + \frac{6(1-p_{01})}{\lambda_2^3} \end{aligned} \quad (4.51)$$

$$\begin{aligned} \mu_4 = & \frac{24p_{01}}{\lambda_1^4} + \frac{24p_{01}\alpha}{\lambda_1^3\beta} + \frac{12p_{01}\alpha^2}{\lambda_1^2\beta^2} + \frac{12p_{01}\alpha}{\lambda_1^2\beta^2} + \frac{6p_{01}\alpha^3}{\beta^4} + \frac{11p_{01}\alpha^2}{\beta^4} \\ & + \frac{4p_{01}\alpha^3}{\lambda_1\beta^3} + \frac{12p_{01}\alpha^2}{\lambda_1\beta^3} + \frac{8p_{01}\alpha}{\lambda_1\beta^3} + \frac{p_{01}\alpha^4}{\beta^4} + \frac{6p_{01}\alpha}{\beta^4} + \frac{24(1-p_{01})}{\lambda_2^4} \end{aligned} \quad (4.52)$$

$$\begin{aligned} \mu_5 = & \frac{120p_{01}\alpha}{\lambda_1^4\beta} + \frac{60p_{01}\alpha^2}{\lambda_1^3\beta^2} + \frac{120(1-p_{01})}{\lambda_2^5} + \frac{p_{01}\alpha^5}{\beta^5} + \frac{120p_{01}}{\lambda_1^5} \\ & + \frac{5p_{01}\alpha^4}{\lambda_1\beta^4} + \frac{20p_{01}\alpha^3}{\lambda_1^2\beta^3} + \frac{60p_{01}\alpha}{\lambda_1^3\beta^2} + \frac{60p_{01}\alpha^2}{\lambda_1^2\beta^3} + \frac{10p_{01}\alpha^4}{\beta^5} \\ & + \frac{40p_{01}\alpha}{\lambda_1^2\beta^3} + \frac{30p_{01}\alpha^3}{\lambda_1\beta^4} + \frac{55p_{01}\alpha^2}{\lambda_1\beta^4} + \frac{30p_{01}\alpha}{\lambda_1\beta^4} + \frac{35p_{01}\alpha^3}{\beta^5} \\ & + \frac{50p_{01}\alpha^2}{\beta^5} + \frac{24p_{01}\alpha}{\beta^5} \end{aligned} \quad (4.53)$$

Define  $\theta_1 = p_{01}$ ,  $\theta_2 = \frac{1}{\lambda_1}$ ,  $\theta_3 = \frac{1}{\lambda_2}$ ,  $\theta_4 = \alpha$  and  $\theta_5 = \frac{1}{\beta}$  to reparametrise equations 4.49, 4.50, 4.51, 4.52, and 4.53 as

$$\mu_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = \theta_1\theta_2 + \theta_1\theta_4\theta_5 + (1 - \theta_1)\theta_3$$

$$\begin{aligned} \mu_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= 2\theta_1\theta_2^2 + 2\theta_1\theta_2\theta_4\theta_5 + \theta_1\theta_4^2\theta_5^2 \\ &\quad + \theta_1\theta_4\theta_5^2 + 2(1 - \theta_1)\theta_3^2 \end{aligned}$$

$$\begin{aligned} \mu_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= 6\theta_1\theta_2^3 + 6\theta_1\theta_2^2\theta_4\theta_5 + 3\theta_1\theta_2\theta_4^2\theta_5^2 \\ &\quad + 3\theta_1\theta_2\theta_4\theta_5^2 + \theta_1\theta_4^3\theta_5^3 + 3\theta_1\theta_4^2\theta_5^3 \\ &\quad + 2\theta_1\theta_4\theta_5^3 + 6(1 - \theta_1)\theta_3^3 \end{aligned}$$

$$\begin{aligned} \mu_4(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= 24\theta_1\theta_2^4 + 24\theta_1\theta_2^3\theta_4\theta_5 + 12\theta_1\theta_2^2\theta_4^2\theta_5^2 \\ &\quad + 12\theta_1\theta_2^2\theta_4\theta_5^2 + 4\theta_1\theta_2\theta_4^3\theta_5^3 + 12\theta_1\theta_2\theta_4^2\theta_5^3 \\ &\quad + 8\theta_1\theta_2\theta_4\theta_5^3 + \theta_1\theta_4^4\theta_5^4 + 6\theta_1\theta_4^3\theta_5^4 \\ &\quad + 11\theta_1\theta_4^2\theta_5^4 + 6\theta_1\theta_4\theta_5^4 + 24(1 - \theta_1)\theta_3^4 \end{aligned}$$

$$\begin{aligned} \mu_5(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= 120\theta_1\theta_2^4\theta_4\theta_5 + 60\theta_1\theta_2^3\theta_4^2\theta_5^2 + 120(1 - \theta_1)\theta_3^5 \\ &\quad + \theta_1\theta_4^5\theta_5^5 + 5\theta_1\theta_2\theta_4^4\theta_5^4 + 20\theta_1\theta_2^2\theta_4^3\theta_5^3 \\ &\quad + 60\theta_1\theta_2^3\theta_4\theta_5^2 + 60\theta_1\theta_2^2\theta_4^2\theta_5^3 + 40\theta_1\theta_2^2\theta_4\theta_5^3 \\ &\quad + 30\theta_1\theta_2\theta_4^3\theta_5^4 + 55\theta_1\theta_2\theta_4^2\theta_5^4 + 30\theta_1\theta_2\theta_4\theta_5^4 \\ &\quad + 10\theta_1\theta_4^4\theta_5^5 + 35\theta_1\theta_4^3\theta_5^5 + 50\theta_1\theta_4^2\theta_5^5 \\ &\quad + 24\theta_1\theta_4\theta_5^5 + 120\theta_1\theta_2^5 \end{aligned}$$

We simulated a sample size of 10000 total waiting time from node 0 and node 2 with  $p_{02} = 0.7$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\alpha = 2.5$ ,  $\beta = 3$ . The sample moments are calculated as  $m_1 = 1.4871$ ,  $m_2 = 3.5348$ ,  $m_3 = 11.4061$ ,  $m_4 = 46.9320$ ,  $m_5 = 235.9361$ . By numerically solving the following 5 equations

$$\mu_r(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = m_r$$

for  $r = 1, \dots, 5$ , the MM estimators are presented in Table 4.10.

Table 4.10: Summary of MoM in non-exponential parallel flowgraph

Sample size	$\hat{p}_{01}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\alpha}$	$\hat{\beta}$
$n = 100000$	0.7112	1.0090	1.8765	2.6267	3.0167

### 4.3.3 Feedback loop flowgraph

#### 4.3.3.1 Example 6.1.1: Exponential waiting time case

In this section, we demonstrate the use of MM to estimate parameters in the feedback loop flowgraph of Figure 2.3 on page 12, where all the internode waiting time follow exponential distribution.

Table 4.11: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_3$ )	$m_{01}(s) = \frac{\lambda_3}{\lambda_3 - s}$
$1 \rightarrow 0$	Exponential( $\lambda_1$ )	$m_{10}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$0 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{02}(s) = \frac{\lambda_2}{\lambda_2 - s}$

By Mason's rule, the MGF of total waiting time distribution between node 0 and node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= \frac{p_{02}m_{02}(s)}{1 - p_{01}m_{01}(s)m_{10}(s)} \\
 &= \frac{p_{02} \left( \frac{\lambda_2}{\lambda_2 - s} \right)}{1 - (1 - p_{02}) \left( \frac{\lambda_3}{\lambda_3 - s} \right) \left( \frac{\lambda_1}{\lambda_1 - s} \right)} \\
 &= \frac{p_{02}\lambda_2(\lambda_3 - s)(\lambda_1 - s)}{(\lambda_2 - s)[s^2 - (\lambda_3 + \lambda_1)s + \lambda_3\lambda_1p_{02}]}
 \end{aligned}$$

To compute the MM estimator for  $p_{02}$ ,  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ , we construct 4 equations such that

$$\left. \frac{d^r M_x(s)}{ds^r} \right|_{s=0} = \frac{1}{n} \sum_{i=1}^n x_i^r$$

for  $r = 1, \dots, 4$ . where

$$\mu_1 = \frac{\lambda_3 + \lambda_1}{p_{02}\lambda_3\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_3} - \frac{1}{\lambda_1} \quad (4.54)$$

$$\begin{aligned}
 \mu_2 &= \frac{2}{\lambda_3\lambda_1} - \frac{2}{\lambda_2\lambda_3} - \frac{2}{\lambda_2\lambda_1} + \frac{2}{\lambda_2^2} - \frac{2(\lambda_3 + \lambda_1)}{p_{02}\lambda_1\lambda_3^2} - \frac{2(\lambda_3 + \lambda_1)}{p_{02}\lambda_3\lambda_1^2} \\
 &+ \frac{2(\lambda_3 + \lambda_1)}{p_{02}\lambda_2\lambda_3\lambda_1} + \frac{2(\lambda_3 + \lambda_1)^2}{p_{02}^2\lambda_3^2\lambda_1^2} - \frac{2}{p_{02}\lambda_3\lambda_1}
 \end{aligned} \quad (4.55)$$

$$\begin{aligned}
 \mu_3 &= \frac{6}{p_{02}\lambda_1\lambda_3^2} - \frac{6}{\lambda_2^2\lambda_1} - \frac{6}{\lambda_2^2\lambda_3} + \frac{6}{\lambda_2^3} - \frac{6(\lambda_3 + \lambda_1)^2}{p_{02}^2\lambda_1^2\lambda_3^2} - \frac{6(\lambda_3 + \lambda_1)^2}{p_{02}^2\lambda_3^2\lambda_1^2} \\
 &+ \frac{6(\lambda_3 + \lambda_1)}{p_{02}\lambda_2^2\lambda_3\lambda_1} - \frac{12(\lambda_3 + \lambda_1)}{p_{02}^2\lambda_3^2\lambda_1^2} + \frac{6(\lambda_3 + \lambda_1)}{p_{02}\lambda_3^2\lambda_1^2} + \frac{6}{p_{02}\lambda_3\lambda_1^2} \\
 &+ \frac{6(\lambda_3 + \lambda_1)^3}{p_{02}^3\lambda_3^3\lambda_1^3} + \frac{6(\lambda_3 + \lambda_1)^2}{p_{02}^2\lambda_2\lambda_3^2\lambda_1^2} - \frac{6(\lambda_3 + \lambda_1)}{p_{02}\lambda_2\lambda_1\lambda_3^2} - \frac{6(\lambda_3 + \lambda_1)}{p_{02}\lambda_2\lambda_3\lambda_1^2} \\
 &+ \frac{6}{p_{02}\lambda_2\lambda_3\lambda_1} + \frac{6}{\lambda_2\lambda_3\lambda_1}
 \end{aligned} \quad (4.56)$$

$$\begin{aligned}
\mu_4 = & \frac{24(\lambda_3 + \lambda_1)^2}{p_{02}^2 \lambda_3^3 \lambda_1^3} + \frac{24(\lambda_3 + \lambda_1)}{p_{02} \lambda_2 \lambda_3^2 \lambda_1^2} - \frac{24}{\lambda_2^3 \lambda_3} + \frac{24}{p_{02} \lambda_2 \lambda_1 \lambda_3^2} \\
& - \frac{24}{\lambda_2^3 \lambda_1} + \frac{24}{p_{02} \lambda_2 \lambda_3 \lambda_1^2} - \frac{24(\lambda_3 + \lambda_1)^3}{p_{02}^3 \lambda_1^3 \lambda_3^4} - \frac{24(\lambda_3 + \lambda_1)^3}{p_{02}^3 \lambda_3^3 \lambda_1^4} \\
& + \frac{24(\lambda_3 + \lambda_1)}{p_{02} \lambda_2^3 \lambda_3 \lambda_1} - \frac{48(\lambda_3 + \lambda_1)}{p_{02}^2 \lambda_2 \lambda_3^2 \lambda_1^2} + \frac{24(\lambda_3 + \lambda_1)^3}{p_{02}^3 \lambda_2 \lambda_3^3 \lambda_1^3} - \frac{72(\lambda_3 + \lambda_1)^2}{p_{02}^3 \lambda_3^3 \lambda_1^3} \\
& \frac{24(\lambda_3 + \lambda_1)^2}{p_{02}^2 \lambda_2^2 \lambda_3^2 \lambda_1^2} + \frac{24(\lambda_3 + \lambda_1)^4}{p_{02}^4 \lambda_3^4 \lambda_1^4} - \frac{24(\lambda_3 + \lambda_1)^2}{p_{02}^2 \lambda_2 \lambda_1^2 \lambda_3^3} + \frac{48(\lambda_3 + \lambda_1)}{p_{02}^2 \lambda_1^2 \lambda_3^3} \\
& - \frac{24(\lambda_3 + \lambda_1)^2}{p_{02}^2 \lambda_2 \lambda_3^2 \lambda_1^3} + \frac{48(\lambda_3 + \lambda_1)}{p_{02}^2 \lambda_3^2 \lambda_1^3} - \frac{24(\lambda_3 + \lambda_1)}{p_{02} \lambda_2^2 \lambda_1 \lambda_3^2} - \frac{24(\lambda_3 + \lambda_1)}{p_{02} \lambda_2^2 \lambda_3 \lambda_1^2} \\
& - \frac{24}{p_{02} \lambda_2^2 \lambda_3 \lambda_1} + \frac{24}{p_{02}^2 \lambda_3^2 \lambda_1^2} + \frac{24}{\lambda_2^4} - \frac{24}{p_{02} \lambda_3^2 \lambda_1^2} + \frac{24}{\lambda_2^2 \lambda_3 \lambda_1} \tag{4.57}
\end{aligned}$$

Define  $\theta_1 = \frac{1}{\lambda_1}$ ,  $\theta_2 = \frac{1}{\lambda_2}$ ,  $\theta_3 = \frac{1}{\lambda_3}$ .  $\theta_4 = \frac{1}{p_{02}}$  to reparametrise equations 4.54, 4.55, 4.56 and 4.57 as

$$\mu_1(\theta_1, \theta_2, \theta_3, \theta_4) = \theta_2 + \theta_4 \theta_3 + \theta_4 \theta_1 - \theta_3 - \theta_1$$

$$\begin{aligned}
\mu_2(\theta_1, \theta_2, \theta_3, \theta_4) = & 2\theta_3 \theta_1 - 2\theta_2 \theta_3 - 6\theta_4 \theta_3 \theta_1 - 2\theta_4 \theta_3^2 - 2\theta_2 \theta_1 - 2\theta_4 \theta_1^2 \\
& + 2\theta_4 \theta_2 \theta_1 + 2\theta_4 \theta_2 \theta_3 + 2\theta_4^2 \theta_1^2 + 4\theta_4^2 \theta_3 \theta_1 + 2\theta_4^2 \theta_3^2 + 2\theta_2^2
\end{aligned}$$

$$\begin{aligned}
\mu_3(\theta_1, \theta_2, \theta_3, \theta_4) = & 6\theta_2 \theta_3 \theta_1 - 6\theta_2^2 \theta_3 - 6\theta_2^2 \theta_1 - 18\theta_4 \theta_2 \theta_3 \theta_1 + 12\theta_4^2 \theta_2 \theta_3 \theta_1 \\
& + 12\theta_4 \theta_1 \theta_3^2 - 30\theta_4^2 \theta_3 \theta_1^2 - 30\theta_4^2 \theta_3^2 \theta_1 + 6\theta_4 \theta_2^2 \theta_1 + 6\theta_4 \theta_2^2 \theta_3 \\
& - 6\theta_4 \theta_2 \theta_1^2 + 18\theta_4^3 \theta_3 \theta_1^2 + 18\theta_4^3 \theta_3^2 \theta_1 + 6\theta_4^2 \theta_2 \theta_1^2 + 6\theta_4^2 \theta_2 \theta_3^2 \\
& - 6\theta_4^2 \theta_1^3 + 6\theta_4^3 \theta_1^3 + 6\theta_4^3 \theta_3^3 + 12\theta_4 \theta_3 \theta_1^2 - 6\theta_4 \theta_2 \theta_3^2 \\
& + 6\theta_2^3 - 6\theta_4^2 \theta_3^3
\end{aligned}$$

$$\begin{aligned}
& \mu_4(\theta_1, \theta_2, \theta_3, \theta_4) \\
= & 48\theta_4\theta_2\theta_1\theta_3^2 + 48\theta_4\theta_2\theta_3\theta_1^2 - 72\theta_4\theta_2^2\theta_3\theta_1 - 120\theta_4^2\theta_2\theta_3\theta_1^2 - 120\theta_4^2\theta_2\theta_3^2\theta_1 \\
& + 72\theta_4^3\theta_2\theta_3^2\theta_1 + 48\theta_4^2\theta_2^2\theta_3\theta_1 - 24\theta_2^3\theta_3 - 24\theta_2^3\theta_1 + 24\theta_4^4\theta_1^4 + 24\theta_4^4\theta_3^4 - 24\theta_4^3\theta_3^4 \\
& + 24\theta_2^4 - 288\theta_4^3\theta_3^2\theta_1^2 - 168\theta_4^3\theta_3^3\theta_1 + 24\theta_4\theta_2^3\theta_1 + 24\theta_4\theta_2^3\theta_3 + 24\theta_4^3\theta_2\theta_1^3 \\
& - 24\theta_4^2\theta_2\theta_3^3 - 24\theta_4\theta_3^2\theta_1^2 + 168\theta_4^2\theta_3^2\theta_1^2 + 96\theta_4^4\theta_3\theta_1^3 + 24\theta_2^2\theta_3\theta_1 + 144\theta_4^4\theta_3^2\theta_1^2 \\
& + 96\theta_4^4\theta_1\theta_3^3 + 72\theta_2^2\theta_3\theta_1^3 + 72\theta_4^2\theta_3^3\theta_1 - 168\theta_4^3\theta_3\theta_1^3 - 24\theta_4^2\theta_2\theta_1^3 - 24\theta_4\theta_2^2\theta_3^2 \\
& - 24\theta_4\theta_2^2\theta_1^2 + 24\theta_4^2\theta_2^2\theta_1^2 + 24\theta_4^2\theta_2^2\theta_3^2 + 72\theta_4^3\theta_2\theta_3\theta_1^2 - 24\theta_4^3\theta_1^4 + 24\theta_4^3\theta_2\theta_3^3
\end{aligned}$$

A sample size of 100000 total waiting time data between node 0 and node 2 are simulated by assuming the true parameters as  $p_{02} = 0.7$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 0.5$ ,  $\lambda_3 = 2$ . The sample moments are  $m_1 = 2.6203$ ,  $m_2 = 12.8252$ ,  $m_3 = 89.6013$ ,  $m_4 = 806.8391$ . Then the method of moments estimator are obtained by numerically solving the following 4 equations

$$g_r(\theta_1, \theta_2, \theta_3, \theta_4) = m_r$$

for  $r = 1, \dots, 4$ .

Table 4.12: Summary of result by Method of Moments

Sample size	$\hat{p}_{02}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$
$n = 10000$	0.6057	1.0388	0.5913	2.0076

#### 4.3.3.2 Example 6.1.2: Non-exponential waiting time case

Consider the single feedback loop flowgraph model in Figure 2.3 on page 12, where the waiting time between node 0 and node 1 is assumed to be non-exponentially distributed (see Table 4.13).

Table 4.13: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Gamma( $\alpha, \beta$ )	$m_{01}(s) = \left(\frac{\beta}{\beta-s}\right)^\alpha$
$1 \rightarrow 0$	Exponential( $\lambda_2$ )	$m_{10}(s) = \frac{\lambda_2}{\lambda_2-s}$
$0 \rightarrow 2$	Exponential( $\lambda_1$ )	$m_{02}(s) = \frac{\lambda_1}{\lambda_1-s}$

By Mason's rule, the MGF of total waiting time distribution between node 0 and node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= \frac{p_{02}m_{02}(s)}{1 - p_{01}m_{01}(s)m_{10}(s)} \\
 &= \frac{p_{02} \left(\frac{\lambda_1}{\lambda_1-s}\right)}{1 - (1 - p_{02}) \left(\frac{\beta}{\beta-s}\right) \left(\frac{\lambda_2}{\lambda_2-s}\right)}
 \end{aligned}$$

We construct a system of 5 moments equations by Definition 11, and reparametrise  $\theta_1 = \frac{1}{p_{02}}$ ,  $\theta_2 = \alpha$ ,  $\theta_3 = \frac{1}{\beta}$ ,  $\theta_4 = \frac{1}{\lambda_1}$  and  $\theta_5 = \frac{1}{\lambda_2}$  to improve the computational efficiency.

$$\mu_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = \theta_4 + \theta_1\theta_2\theta_3 + \theta_1\theta_5 - \theta_2\theta_3 - \theta_5$$

$$\begin{aligned}
 \mu_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= 2\theta_4^2 + 2\theta_4\theta_1\theta_2\theta_3 + 2\theta_4\theta_1\theta_5 - 2\theta_4\theta_2\theta_3 - 2\theta_4\theta_5 \\
 &\quad + 4\theta_1^2\theta_2\theta_3\theta_5 + 2\theta_1^2\theta_5^2 - 3\theta_1\theta_2^2\theta_3^2 - 6\theta_1\theta_2\theta_3\theta_5 - 2\theta_1\theta_5^2 \\
 &\quad + 2\theta_2\theta_3\theta_5 + \theta_1\theta_2\theta_3^2 - \theta_2\theta_3^2 + 2\theta_1^2\theta_2^2\theta_3^2 + \theta_2^2\theta_3^2
 \end{aligned}$$



$$\begin{aligned}
& \mu_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\
= & -3\theta_2^2\theta_3^2\theta_5 + 3\theta_2\theta_3^2\theta_5 + 6\theta_4^2\theta_1\theta_2\theta_3 + 6\theta_4\theta_1^2\theta_2^2\theta_3^2 - 9\theta_4\theta_1\theta_2^2\theta_3^2 \\
& + 6\theta_4\theta_2\theta_3\theta_5 + 3\theta_4\theta_1\theta_2\theta_3^2 + 18\theta_1^3\theta_2^2\theta_3^2\theta_5 + 18\theta_1^3\theta_2\theta_3\theta_5^2 - 36\theta_1^2\theta_2^2\theta_3^2\theta_5 \\
& - 30\theta_1^2\theta_2\theta_3\theta_5^2 + 21\theta_1\theta_2^2\theta_3^2\theta_5 + 12\theta_1\theta_2\theta_3\theta_5^2 + 6\theta_1^2\theta_5\theta_2\theta_3^2 - 9\theta_1\theta_5\theta_2\theta_3^2 \\
& - \theta_2^3\theta_3^3 + 3\theta_2^2\theta_3^3 - 2\theta_2\theta_3^3 + 6\theta_4^2\theta_1\theta_5 - 6\theta_4^2\theta_2\theta_3 + 6\theta_4\theta_1^2\theta_5^2 - 6\theta_4\theta_1\theta_5^2 \\
& + 3\theta_4\theta_2^2\theta_3^2 - 3\theta_4\theta_2\theta_3^2 + 6\theta_1^3\theta_2^3\theta_3^3 - 12\theta_1^2\theta_2^3\theta_3^3 + 7\theta_1\theta_2^3\theta_3^3 + 6\theta_1^2\theta_2^2\theta_3^3 \\
& - 9\theta_1\theta_2^2\theta_3^3 + 2\theta_1\theta_2\theta_3^3 + 6\theta_1^3\theta_5^3 - 6\theta_1^2\theta_5^3 - 6\theta_4^2\theta_5 + 6\theta_4^3 + 12\theta_4\theta_1^2\theta_2\theta_3\theta_5 \\
& - 18\theta_4\theta_1\theta_2\theta_3\theta_5
\end{aligned}$$

$$\begin{aligned}
& \mu_4(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\
= & 4\theta_2^3\theta_3^3\theta_5 - 12\theta_2^2\theta_3^3\theta_5 + 8\theta_2\theta_3^3\theta_5 + 24\theta_4^3\theta_1\theta_2\theta_3 + 24\theta_4^2\theta_1^2\theta_2^2\theta_3^2 - 36\theta_4^2\theta_1\theta_2^2\theta_3^2 \\
& + 24\theta_4^2\theta_2\theta_3\theta_5 + 12\theta_4^2\theta_1\theta_2\theta_3^2 + 24\theta_4\theta_1^3\theta_2^3\theta_3^3 - 48\theta_4\theta_1^2\theta_2^3\theta_3^3 + 28\theta_4\theta_1\theta_2^3\theta_3^3 \\
& - 12\theta_4\theta_2^2\theta_3^2\theta_5 + 24\theta_4\theta_1^2\theta_2^2\theta_3^3 - 36\theta_4\theta_1\theta_2^2\theta_3^3 + 12\theta_4\theta_2\theta_3^2\theta_5 + 8\theta_4\theta_1\theta_2\theta_3^3 \\
& + 48\theta_4^2\theta_1^2\theta_2\theta_3\theta_5 - 72\theta_4^2\theta_1\theta_2\theta_3\theta_5 + 72\theta_4\theta_1^3\theta_2^2\theta_3^2\theta_5 + 72\theta_4\theta_1^3\theta_2\theta_3\theta_5^2 \\
& - 144\theta_4\theta_1^2\theta_2^2\theta_3^2\theta_5 - 120\theta_4\theta_1^2\theta_2\theta_3\theta_5^2 + 84\theta_4\theta_1\theta_2^2\theta_3^2\theta_5 + 48\theta_4\theta_1\theta_2\theta_3\theta_5^2 \\
& + 24\theta_4\theta_1^2\theta_5\theta_2\theta_3^2 - 36\theta_4\theta_1\theta_5\theta_2\theta_3^2 + \theta_2^4\theta_3^4 - 6\theta_2^3\theta_3^4 + 11\theta_2^2\theta_3^4 - 6\theta_2\theta_3^4 \\
& + 24\theta_4^3\theta_1\theta_5 - 24\theta_4^3\theta_2\theta_3 + 24\theta_4^2\theta_1^2\theta_5^2 - 24\theta_4^2\theta_1\theta_5^2 + 12\theta_4^2\theta_2^2\theta_3^2 - 12\theta_4^2\theta_2\theta_3^2 \\
& + 24\theta_4\theta_1^3\theta_5^3 - 24\theta_4\theta_1^2\theta_5^3 - 4\theta_4\theta_2^3\theta_3^3 + 12\theta_4\theta_2^2\theta_3^3 - 8\theta_4\theta_2\theta_3^3 + 24\theta_4^4\theta_2^4\theta_3^4 \\
& - 60\theta_1^3\theta_2^4\theta_3^4 + 50\theta_1^2\theta_2^4\theta_3^4 - 15\theta_1\theta_2^4\theta_3^4 + 36\theta_1^3\theta_2^3\theta_3^4 - 72\theta_1^2\theta_2^3\theta_3^4 + 42\theta_1\theta_2^3\theta_3^4 \\
& + 22\theta_1^2\theta_2^2\theta_3^4 - 33\theta_1\theta_2^2\theta_3^4 + 6\theta_1\theta_2\theta_3^4 + 24\theta_4^4 - 24\theta_4^3\theta_5 + 24\theta_4^4\theta_5^4 - 24\theta_4^3\theta_5^4 \\
& + 96\theta_1^4\theta_2^3\theta_3^3\theta_5 + 144\theta_1^4\theta_2^2\theta_3^2\theta_5^2 + 96\theta_1^4\theta_2\theta_3\theta_5^3 - 240\theta_1^3\theta_2^3\theta_3^3\theta_5 \\
& - 324\theta_1^3\theta_2^2\theta_3^2\theta_5^2 - 168\theta_1^3\theta_2\theta_3\theta_5^3 + 200\theta_1^2\theta_2^3\theta_3^3\theta_5 + 228\theta_1^2\theta_2^2\theta_3^2\theta_5^2 \\
& + 72\theta_1^2\theta_2\theta_3\theta_5^3 - 60\theta_1\theta_2^3\theta_3^3\theta_5 - 48\theta_1\theta_2^2\theta_3^2\theta_5^2 + 72\theta_1^3\theta_2^2\theta_3^3\theta_5 \\
& + 36\theta_1^3\theta_5^2\theta_2\theta_3^2 - 144\theta_1^2\theta_5^2\theta_2\theta_3^3\theta_5 - 60\theta_1^2\theta_5^2\theta_2\theta_3^2 + 84\theta_1\theta_5^2\theta_2\theta_3^3\theta_5 \\
& + 24\theta_1\theta_5^2\theta_2\theta_3^2 + 16\theta_1^2\theta_5\theta_2\theta_3^3 - 24\theta_1\theta_5\theta_2\theta_3^3
\end{aligned}$$

$$\begin{aligned}
& \mu_5(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\
= & 120\theta_4^4\theta_1\theta_2\theta_3 - 180\theta_4^3\theta_1\theta_2^2\theta_3^2 + 60\theta_4^3\theta_1\theta_2\theta_3^2 + 120\theta_4^3\theta_2\theta_3\theta_5 + 140\theta_4^2\theta_1\theta_2^3\theta_3^3 - 180\theta_4^2\theta_1\theta_2^2\theta_3^3 \\
& + 40\theta_4^2\theta_1\theta_2\theta_3^3 - 60\theta_4^2\theta_2^2\theta_3^2\theta_5 + 60\theta_4^2\theta_2\theta_3^2\theta_5 - 75\theta_4\theta_1\theta_2^4\theta_3^4 + 210\theta_4\theta_1\theta_2^3\theta_3^4 - 165\theta_4\theta_1\theta_2^2\theta_3^4 \\
& + 30\theta_4\theta_1\theta_2\theta_3^4 + 20\theta_4\theta_2^3\theta_3^3\theta_5 - 60\theta_4\theta_2^2\theta_3^3\theta_5 + 40\theta_4\theta_2\theta_3^3\theta_5 + 155\theta_1\theta_2^4\theta_3^4\theta_5 - 450\theta_1\theta_2^3\theta_3^4\theta_5 \\
& + 160\theta_1\theta_2^3\theta_3^3\theta_5^2 + 385\theta_1\theta_2^2\theta_3^4\theta_5 - 240\theta_1\theta_2^2\theta_3^3\theta_5^2 - 90\theta_1\theta_2\theta_3^4\theta_5 + 80\theta_1\theta_2\theta_3^3\theta_5^2 + 120\theta_4^3\theta_1^2\theta_2^2\theta_3^2 \\
& + 120\theta_4^2\theta_1^2\theta_2^2\theta_3^3 - 240\theta_4^2\theta_1^2\theta_2^3\theta_3^3 + 120\theta_4^2\theta_1^2\theta_2^2\theta_3^3 + 120\theta_4\theta_1^4\theta_2^4\theta_3^4 - 300\theta_4\theta_1^3\theta_2^4\theta_3^4 + 250\theta_4\theta_1^2\theta_2^4\theta_3^4 \\
& - 5\theta_2^4\theta_3^4\theta_5 + 30\theta_2^3\theta_3^4\theta_5 - 55\theta_2^2\theta_3^4\theta_5 + 30\theta_2\theta_3^4\theta_5 - 360\theta_4^3\theta_1\theta_2\theta_3\theta_5 + 420\theta_4^2\theta_1\theta_2^2\theta_3^2\theta_5 \\
& - 180\theta_4^2\theta_1\theta_5\theta_2\theta_3^2 + 240\theta_4^2\theta_1\theta_2\theta_3\theta_5^2 - 300\theta_4\theta_1\theta_2^2\theta_3^3\theta_5 + 420\theta_4\theta_1\theta_2^2\theta_3^3\theta_5 - 240\theta_4\theta_1\theta_2^2\theta_3^2\theta_5^2 \\
& - 120\theta_4\theta_1\theta_5\theta_2\theta_3^3 + 120\theta_4\theta_1\theta_5^2\theta_2\theta_3^2 + 240\theta_4^3\theta_1^2\theta_2\theta_3\theta_5 + 360\theta_4^2\theta_1^2\theta_2^2\theta_3^2\theta_5 + 360\theta_4^2\theta_1^2\theta_2\theta_3\theta_5^2 \\
& - 720\theta_4^2\theta_1^2\theta_2^2\theta_3^2\theta_5 - 600\theta_4^2\theta_1^2\theta_2\theta_3\theta_5^2 + 120\theta_4^2\theta_1^2\theta_5\theta_2\theta_3^2 + 480\theta_4\theta_1^4\theta_2^3\theta_3^3\theta_5 + 720\theta_4\theta_1^4\theta_2^2\theta_3^2\theta_5^2 \\
& + 480\theta_4\theta_1^4\theta_2\theta_3\theta_5^3 - 1200\theta_4\theta_1^3\theta_2^3\theta_3^3\theta_5 - 1620\theta_4\theta_1^3\theta_2^2\theta_3^2\theta_5^2 - 840\theta_4\theta_1^3\theta_2\theta_3\theta_5^3 + 1000\theta_4\theta_1^2\theta_2^3\theta_3^3\theta_5 \\
& + 1140\theta_4\theta_1^2\theta_2^2\theta_3^2\theta_5^2 + 360\theta_4\theta_1^2\theta_2\theta_3\theta_5^3 + 360\theta_4\theta_1^2\theta_2^2\theta_3^3\theta_5 + 180\theta_4\theta_1^3\theta_5^2\theta_2\theta_3^2 - 720\theta_4\theta_1^2\theta_2^2\theta_3^3\theta_5 \\
& - 300\theta_4\theta_1^2\theta_5^2\theta_2\theta_3^2 + 80\theta_4\theta_1^2\theta_5\theta_2\theta_3^3 + 120\theta_4^5 - \theta_2^5\theta_3^5 + 10\theta_2^4\theta_3^5 - 35\theta_2^3\theta_3^5 + 50\theta_2^2\theta_3^5 - 24\theta_2\theta_3^5 \\
& + 120\theta_4^4\theta_1\theta_5 - 120\theta_4^4\theta_2\theta_3 - 120\theta_4^3\theta_1\theta_5^2 + 60\theta_4^3\theta_2^2\theta_3^2 - 60\theta_4^3\theta_2\theta_3^2 - 20\theta_4^2\theta_2^3\theta_3^3 + 60\theta_4^2\theta_2^2\theta_3^3 \\
& - 40\theta_4^2\theta_2\theta_3^3 + 5\theta_4\theta_2^4\theta_3^4 - 30\theta_4\theta_2^3\theta_3^4 + 55\theta_4\theta_2^2\theta_3^4 - 30\theta_4\theta_2\theta_3^4 + 31\theta_1\theta_2^5\theta_3^5 - 150\theta_1\theta_2^4\theta_3^5 \\
& + 245\theta_1\theta_2^3\theta_3^5 - 150\theta_1\theta_2^2\theta_3^5 + 24\theta_1\theta_2\theta_3^5 + 120\theta_4^3\theta_1^2\theta_5^2 + 120\theta_4^2\theta_1^3\theta_5^3 - 120\theta_4^2\theta_1^2\theta_5^3 \\
& - 120\theta_4\theta_1^3\theta_5^4 - 120\theta_4^4\theta_5 + 120\theta_1^5\theta_5^5 - 120\theta_1^4\theta_5^5 + 180\theta_4\theta_1^3\theta_2^3\theta_3^4 - 360\theta_4\theta_1^2\theta_2^3\theta_3^4 \\
& + 600\theta_1^5\theta_2^4\theta_3^4\theta_5 + 1200\theta_1^5\theta_2^3\theta_3^3\theta_5^2 + 1200\theta_1^5\theta_2^2\theta_3^2\theta_5^3 + 600\theta_1^5\theta_2\theta_3\theta_5^4 - 1800\theta_1^4\theta_2^4\theta_3^4\theta_5 \\
& - 3360\theta_1^4\theta_2^3\theta_3^3\theta_5^2 - 2880\theta_1^4\theta_2^2\theta_3^2\theta_5^3 - 1080\theta_1^4\theta_2\theta_3\theta_5^4 + 1950\theta_1^3\theta_2^4\theta_3^4\theta_5 + 3300\theta_1^3\theta_2^3\theta_3^3\theta_5^2 \\
& + 2220\theta_1^3\theta_2^2\theta_3^2\theta_5^3 + 480\theta_1^3\theta_2\theta_3\theta_5^4 - 900\theta_1^2\theta_2^4\theta_3^4\theta_5 - 1300\theta_1^2\theta_2^3\theta_3^3\theta_5^2 - 540\theta_1^2\theta_2^2\theta_3^2\theta_5^3 \\
& + 720\theta_1^4\theta_2^3\theta_3^4\theta_5 + 720\theta_1^4\theta_2^2\theta_3^3\theta_5^2 + 240\theta_1^4\theta_2^3\theta_3^2\theta_5 - 1800\theta_1^3\theta_2^3\theta_3^4\theta_5 - 1620\theta_1^3\theta_2^2\theta_3^3\theta_5^2 \\
& - 420\theta_1^3\theta_5^3\theta_2\theta_3^2 + 1500\theta_1^2\theta_2^3\theta_3^4\theta_5 + 1140\theta_1^2\theta_2^2\theta_3^3\theta_5^2 + 180\theta_1^2\theta_5^3\theta_2\theta_3^2 + 330\theta_1^3\theta_2^2\theta_3^4\theta_5 \\
& + 120\theta_1^3\theta_5^2\theta_2\theta_3^3 - 660\theta_1^2\theta_2^2\theta_3^4\theta_5 - 200\theta_1^2\theta_5^2\theta_2\theta_3^3 + 60\theta_1^2\theta_5\theta_2\theta_3^4 + 120\theta_1^5\theta_2^5\theta_3^5 \\
& + 390\theta_1^3\theta_2^5\theta_3^5 - 180\theta_1^2\theta_2^5\theta_3^5 + 240\theta_1^4\theta_2^4\theta_3^5 - 600\theta_1^3\theta_2^4\theta_3^5 + 500\theta_1^2\theta_2^4\theta_3^5 + 210\theta_1^3\theta_2^3\theta_3^5 \\
& - 420\theta_1^2\theta_2^3\theta_3^5 + 100\theta_1^2\theta_2^2\theta_3^5 - 360\theta_1^4\theta_2^5\theta_3^5 + 120\theta_4\theta_1^4\theta_2^4\theta_3^4 + 110\theta_4\theta_1^2\theta_2^2\theta_3^4
\end{aligned}$$

We simulate a sample size of 10000 total waiting time from node 0 and node 2 with  $p_{02} = 0.7$ ,  $\alpha = 0.5$ ,  $\beta = 1$ ,  $\lambda_1 = 1.5$ ,  $\lambda_2 = 3$ . The sample moments are  $m_1 = 1.0232$ ,  $m_2 = 2.1721$ ,  $m_3 = 7.1898$ ,  $m_4 = 32.3488$ ,  $m_5 = 186.974$ . Then the method of moments estimators are obtained by numerically solving the following 5 equations

$$\mu_r(\theta_1, \theta_2, \theta_3, \theta_4) = m_r \quad \text{for } r = 1, \dots, 5.$$

Table 4.14: Summary of result by Method of Moments

Sample size	$\hat{p}_{02}$	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}_1$	$\hat{\lambda}_2$
$n = 10000$	0.7115	0.4448	0.9548	1.4703	2.6289

### 4.3.4 Example 1.4 continued

#### 4.3.4.1 Example 1.4.1: Exponential waiting time case

We return to the three nodes reversible illness model presented in Section 4.2.2.1. Instead of maximum likelihood approach, we apply the method of moment to estimate parameters in Figure 2.5 on page 17. This is a more challenging example, where the dimension of parameter is high and the algebraic structure of moment equations are complicated.

Table 4.15: Summary of waiting time distribution

Direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$0 \rightarrow 2$	Exponential( $\lambda_3$ )	$m_{02}(s) = \frac{\lambda_3}{\lambda_3 - s}$
$1 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{12}(s) = \frac{\lambda_2}{\lambda_2 - s}$
$1 \rightarrow 0$	Exponential( $\lambda_4$ )	$m_{10}(s) = \frac{\lambda_4}{\lambda_4 - s}$

By Mason's rule, the MGF of the total waiting time distribution from node 0 to node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\ &= \frac{(\lambda_4 - s) [p_{01}p_{12}\lambda_1\lambda_2(\lambda_3 - s) + \lambda_3(1 - p_{01})(\lambda_1 - s)(\lambda_2 - s)]}{(\lambda_2 - s)(\lambda_3 - s) [(\lambda_1 - s)(\lambda_4 - s) - \lambda_1\lambda_4p_{01}(1 - p_{12})]} \end{aligned}$$

We construct a system of 6 moment equations for estimating 6 parameters  $p_{01}$ ,  $p_{12}$ ,  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ ,  $\lambda_4$  as follow

$$\left. \frac{d^r M_{T_{02}}(s)}{ds^r} \right|_{s=0} = \frac{1}{n} \sum_{i=1}^n t_i^r$$

for  $r = 1, \dots, 6$  (see Appendix). For illustration purposes, we assume

1.  $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, \lambda_4 = 4.$
2.  $p_{01} = 0.3, p_{12} = 0.8$

A sample size of  $n = 100000$  total waiting time data from node 0 to node 2 is simulated by the procedure we discussed in Section 4.2.2.2. The sample moments are computed as  $m_1 = 0.7126, m_2 = 1.3456, m_3 = 4.2553, m_4 = 18.4244, m_5 = 99.8849, m_6 = 649.9525$ . As it is not possible to solve the set of nonlinear equations given by equation 4.3.4.1 algebraically, we use the Newton-Raphson method to compute the MM estimators. To have a fast convergence of the Newton-Raphson algorithm, we set the initial values as  $p_{01}^{(0)} = 0.2, p_{12}^{(0)} = 0.72, \lambda_1^{(0)} = 0.92, \lambda_2^{(0)} = 1.9, \lambda_3^{(0)} = 3.05, \lambda_4^{(0)} = 4.0900$ . The result of the parameter estimation by MM is given in Table 4.16

Table 4.16: Summary of result by Method of Moments

parameter	$p_{01}$	$p_{12}$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$
True value	0.3	0.8	1	2	3	4
Estimated value	0.3015	0.8889	0.9658	1.9731	2.9199	4.2631

Although there is a relatively large difference from the true value of  $\lambda_4$ , the MM estimators still provide a reasonably good estimation for both the transition probabilities and the rates of exponential distribution in the flowgraph model. For high dimensional parameter estimation problems in a large flowgraph, the advantage of the method of moments is in its simplicity of working with MGFs, which can be easily computed by using any symbolic algebra package.

#### 4.3.4.2 Example 1.4.2: Non-exponential waiting time case

In this section we demonstrate the use of method of moments to compute the estimator for 8 parameters in the flowgraph illustrated in Figure 2.5 which involves non-exponentially distributed internode waiting time (see Table 4.17).

Table 4.17: Summary of waiting time distribution

Flow direction	Distribution	MGF
0 → 1	Gamma( $\alpha_1, \beta_1$ )	$m_{01}(s) = \left(\frac{\beta_1}{\beta_1 - s}\right)^{\alpha_1}$
0 → 2	Exponential( $\lambda_1$ )	$m_{02}(s) = \frac{\lambda_1}{\lambda_1 - s}$
1 → 2	Gamma( $\alpha_2, \beta_2$ )	$m_{12}(s) = \left(\frac{\beta_2}{\beta_2 - s}\right)^{\alpha_2}$
1 → 0	Exponential( $\lambda_2$ )	$m_{10}(s) = \frac{\lambda_2}{\lambda_2 - s}$

The MGF of the total waiting time distribution from node 0 to node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\
 &= \frac{(\lambda_2 - s) [p_{01}p_{12}\beta_1^{\alpha_1}\beta_2^{\alpha_2}(\lambda_1 - s) + \lambda_1(1 - p_{01})(\beta_1 - s)^{\alpha_1}(\beta_2 - s)^{\alpha_2}]}{(\lambda_1 - s)(\beta_2 - s)^{\alpha_2} [(\beta_1 - s)^{\alpha_1}(\lambda_2 - s) - p_{01}\beta_1^{\alpha_1}\lambda_2(1 - p_{12})]}
 \end{aligned}$$

In order to estimate  $\theta = (p_{01}, p_{12}, \alpha_1, \beta_1, \alpha_2, \beta_2, \lambda_1, \lambda_2)$ , a set of 8 moment equations are constructed as follow

$$\left. \frac{d^r M_{T_{02}}(s)}{ds^r} \right|_{s=0} = \frac{1}{n} \sum_{i=1}^n t_i^r \tag{4.58}$$

for  $r = 1, \dots, 8$ .

The sample moments,  $m_1 = 1.05303$ ,  $m_2 = 1.97919$ ,  $m_3 = 5.09816$ ,  $m_4 = 16.66027$ ,  $m_5 = 66.36213$ ,  $m_6 = 313.28586$ ,  $m_7 = 1714.98377$ ,  $m_8 = 10697.19494$ , are calculated based on 100000 samples of total waiting time from node 0 to node 2 (i.e. see Section 4.2.2.2 for the procedure of simulation). Again, Newton-Raphson Method is used for solving the set of moment equations derived by 4.58.

Since in multimodal situations the Newton-Raphson method may not converge from any starting point, particularly for high dimensional parameters, the convergence is sensitive to the initial value of parameters. We therefore typically use multi-start when using the algorithm. For a fast convergence of Newton-Raphson method, we start the algorithm at  $p_{01}^{(0)} = 0.56$ ,  $p_{12}^{(0)} = 0.4$ ,  $\alpha_1^{(0)} = 3.25$ ,  $\beta_1^{(0)} = 4.71$ ,  $\alpha_2^{(0)} = 1.91$ ,  $\beta_2^{(0)} = 3.1$ ,  $\lambda_1^{(0)} = 2.15$ ,  $\lambda_2^{(0)} = 3.9$ . Table 4.18 summaries the estimated value of parameters.

Table 4.18: Summary of result by Method of Moments

Parameter	$p_{01}$	$p_{12}$	$\alpha_1$	$\beta_1$	$\alpha_2$	$\beta_2$	$\lambda_1$	$\lambda_2$
True value	0.5	0.5	3.2	4.8	2	3	2.2	4
Estimated value	0.5161	0.5061	3.1610	4.8081	2.1344	3.2018	2.2258	4.0515

For high dimensional parameter estimation problems, we note that the accuracy of Method of Moment estimator depend crucially on two factors: 1) The value of sample moments, 2) The efficiency of implementing the Newton-Raphson method. Since the calculation of MM estimators is essentially a set of equation solving problem, it is important to have sample moments that are close to the theoretical moments evaluated at the true value of parameters, especially for high sample moment (i.e.  $\frac{1}{n} \sum_{i=1}^n t_i^r$  when  $r$  is large). Therefore, we usually require a large sample size data to reduce the difference between sample moments and theoretical moments. On the other hand, we can greatly improve the numerical stability of the Newton-Raphson algorithm by reparametrising the moment equations into a suitable form that does not involve rational terms. In addition, we need to fine tune to the initial value so that the Newton-Raphson algorithm can converge quickly.

### 4.3.5 Identifiability

Identifiability is a property that a model must satisfy so that we can estimate the true value of this model's underlying parameters based on the a number of observations from it. Casella and Berger (2002) explain that the identifiability is a property of the model, not of an estimator or estimation procedure. If a model is not identifiable, then there is a difficulty in performing statistical inference.

Identifiability here means the parameters of flowgaph model can be estimated by using the total waiting time data. We need to solve the set of moment equations in MM or the set of likelihood equations in ML estimation for the parameter estimator, which requires the Jacobian matrix of moment equations or the Fisher information matrix, respectively, to be invertible. Note that we could inspect the PDF for the constraints of parameter values, so that we have a valid PDF to construct likelihood function. For the MM approach, given a  $d$ -dimensional parameter  $\theta = (\theta_1, \dots, \theta_d)$ , the Jacobian matrix is a  $d \times d$  matrix  $H = \{h_{ij}\}$  such that  $h_{ij} = \frac{\partial \mu_i}{\partial \theta_j}$  for  $i, j = 1, \dots, d$ . We check this identification conditions by computing the algebraic expression of Jacobian determinant (see Appendix), and find the conditions for the value of parameters to have  $\det(H) \neq 0$ . The result is summarised in Table 4.19 and 4.20.

Table 4.19: Conditions for exponential internode waiting time case

Example	5.1 continued	6.1.1
MLE	$\lambda_1 \neq \lambda_2$	$p_{02} \neq 1$
MM	$p_{01} \neq 1$ $\lambda_1 \neq \lambda_2$	$p_{02} \neq 1$
	$\lambda_2 \neq \lambda_3$ $\lambda_1 \neq \lambda_3$	$\lambda_3 \neq \lambda_1$

Table 4.20: Conditions for Non-exponential internode waiting time case

Example	5.2 continued	6.1.2
MM	$p_{01} \neq 1$	$p_{02} \neq 1$
	$\lambda_1 \neq \beta$	$\lambda_2 \neq \beta$

## 4.4 Bias correction in the Method of Moments

This section discusses the calculation of analytical first-order bias expressions for the method of moments (MM) estimator of the parameters in the total waiting time distribution. The MM estimator may be biased when the sample size of data is small, partly because the sample moments will loss accuracy in matching the population moments for small size samples, and the difference between sample moments and population moments becomes large for higher order. The motivation is to estimate the size of bias in MM estimator given sample data, and improve the accuracy of MM estimators by removing the estimated bias.

First, we review two different approaches for bias reduction of MLE in literature. Cox and Snell (1968) develop a general expression for the bias to order  $O(n^{-1})$  of the maximum likelihood estimation for high dimensional parameter. Essentially, the Cox-Snell method is a “corrective” approach to bias adjusted MLE, where we first calculate the MLE, then correct by subtracting its estimated bias. On the other hand, Firth (1993) introduces an alternative “preventive” approach, which involves modifying the score functions before they are solved for computing the MLE. Cordeiro and McCullagh (1991) have adopted the Cox and Snell formula (see equation (20) in Cox and Snell (1968)) to derive general formula for second-order biases of MLEs of parameters in generalized linear model. For the MM estimator, further research could be done to connect Firth’s method to adjust moments equations (i.e. biased corrected the sample moments) in the method of moments.

Based on the asymptotic expansion of moment function, we propose a formula for computing the analytical bias expressions to order  $O(n^{-1})$  of the MM estimator in both univariate and multivariate case, where  $n$  is the sample size. The bias corrected MM estimator can then be determined by subtracting the bias (estimated at the MM of the parameters) from the original MM estimator. The results of a series flowgraph simulation experiment is presented, where we evaluate the performance of bias corrected MM estimators that are based on our analytical results, as well as the corresponding MM estimator and MLEs.



**Lemma 6.** Let  $G(\theta) = (G_1(\theta), \dots, G_d(\theta))^T$  be a  $d \times 1$  vector such that

$$G_r(\theta) = \mu_r(\theta) - m_r \tag{4.59}$$

where  $\mu_r(\theta)$  and  $m_r$  are defined in Definition 11. Then

$$E(G_r(\theta)) = 0 \tag{4.60}$$

for  $r = 1 \dots d$ .

*Proof.* By Definition 11, we have

$$E(m_r) = E(x^r) = \mu_r(\theta)$$

then,

$$\begin{aligned} E(G_r(\theta)) &= E(\mu_r(\theta) - m_r) \\ &= \mu_r(\theta) - E(m_r) \\ &= 0 \end{aligned}$$

for  $r = 1, \dots, d$ .

□

Let  $\hat{\theta}$  be the Method of Moment estimator. By Definition 11, we also have

$$G(\hat{\theta}) = 0 \tag{4.61}$$

**Lemma 7.** Given a vector of parameters  $\theta_1, \dots, \theta_n$  with mean  $\mu$  and covariance matrix  $\Sigma$ , let  $\bar{\theta} = \frac{1}{n} \sum_{i=1}^n \theta_i$ . The following holds

$$\sqrt{n}(\bar{\theta} - \mu) \xrightarrow{d} N(0, \Sigma)$$

We say that  $\bar{\theta}$  is asymptotically normally distributed with mean  $\mu$  and covariance matrix  $\frac{1}{n}\Sigma$ .

**Theorem 8.** Given parameter  $\theta = (\theta_1, \dots, \theta_d)$ . The closed form expression for the bias of Method of Moments estimator  $\hat{\theta}$  is

$$\begin{aligned} b(\hat{\theta}_r) &= E(\hat{\theta} - \theta) \\ &= \sum_{i=1}^d B_{ri}^{-1} A_i + O\left(\frac{1}{n^2}\right) \end{aligned}$$

for  $r = 1, \dots, d$ , where

$$B = \begin{bmatrix} E\left(\frac{\partial G_1}{\partial \theta_1}\right) & \dots & E\left(\frac{\partial G_1}{\partial \theta_d}\right) \\ E\left(\frac{\partial G_2}{\partial \theta_1}\right) & \dots & E\left(\frac{\partial G_2}{\partial \theta_d}\right) \\ \vdots & \dots & \vdots \\ E\left(\frac{\partial G_d}{\partial \theta_1}\right) & \dots & E\left(\frac{\partial G_d}{\partial \theta_d}\right) \end{bmatrix}$$

$$A_i = -\frac{1}{2} \sum_{k=1}^d \sum_{l=1}^d \text{cov}(\hat{\theta}_k, \hat{\theta}_l) E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_k \partial \theta_l}\right) = -\frac{1}{2} C V_i$$

and

$$G_r(\theta) = M_x^r(0) - m_r$$

$$V_i = [v^{(1)} | v^{(2)} | \dots | v^{(d)}]^T, \text{ where } v^{(j)} = \left[ E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_1}\right), E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_2}\right), \dots, E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_d}\right) \right].$$

$$C = [c^{(1)} | c^{(2)} | \dots | c^{(d)}], \text{ where } c^{(j)} = [\text{cov}(\hat{\theta}_j, \hat{\theta}_1), \text{cov}(\hat{\theta}_j, \hat{\theta}_2), \dots, \text{cov}(\hat{\theta}_j, \hat{\theta}_d)].$$

for  $j = 1, \dots, d$

*Proof. Univariate case*

In one dimensional parameter case (i.e.  $d = 1$  in Lemma 6, we apply second order of Taylor expansion of  $G(\hat{\theta})$  in  $\hat{\theta}$  at  $\theta$ :

$$G(\hat{\theta}) \approx G(\theta) + G'(\theta)(\hat{\theta} - \theta) + \frac{1}{2} G''(\theta)(\hat{\theta} - \theta)^2 \quad (4.62)$$

Taking expectations on both side of equation 4.62, we obtain

$$E(G(\hat{\theta})) \approx E(G(\theta)) + E\{G'(\theta)(\hat{\theta} - \theta)\} + E\left\{\frac{1}{2}G''(\theta)(\hat{\theta} - \theta)^2\right\} \quad (4.63)$$

By equation 4.61, we have  $E(G(\hat{\theta})) = 0$ , then equation 4.63 becomes

$$0 \approx E(G(\theta)) + E\{G'(\theta)(\hat{\theta} - \theta)\} + E\left\{\frac{1}{2}G''(\theta)(\hat{\theta} - \theta)^2\right\} \quad (4.64)$$

By equation 4.60, the first term on the right hand side of equation 4.64 vanishes, then

$$E\{G'(\theta)(\hat{\theta} - \theta)\} + E\left\{\frac{1}{2}G''(\theta)(\hat{\theta} - \theta)^2\right\} \approx 0 \quad (4.65)$$

where

$$\begin{aligned} G'(\theta) &= \frac{dG(\theta)}{d\theta} \\ &= \frac{dM_x^1(0)}{d\theta} - \frac{dm_1}{d\theta} \\ &= \frac{dM_x^1(0)}{d\theta} \end{aligned}$$

$$\begin{aligned} G''(\theta) &= \frac{d^2G}{d\theta^2} \\ &= \frac{d^2M_x^1(0)}{d\theta^2} - \frac{d^2m_1}{d\theta^2} \\ &= \frac{d^2M_x^1(0)}{d\theta^2} \end{aligned}$$

Since  $M_x^1(0) = \left(\frac{dM_x(s)}{ds}\right)\Big|_{s=0}$  only involves parameters,  $G'(\theta)$  and  $G''(\theta)$  are functions of parameter, and they can be considered as constant for the terms involves expectation in equation 4.65, then

$$G'(\theta)E(\hat{\theta} - \theta) + G''(\theta)E\left\{\frac{1}{2}(\hat{\theta} - \theta)^2\right\} \approx 0$$

$$E(\hat{\theta} - \theta) \approx -\frac{E(\hat{\theta} - \theta)^2 G''(\theta)}{2G'(\theta)}$$

$$E(\hat{\theta} - \theta) \approx -\frac{\{\text{Var}(\hat{\theta}) + b^2(\hat{\theta})\}G''(\theta)}{2G'(\theta)} \quad (4.66)$$

By Lemma 7,  $\text{Var}(\hat{\theta}) = O\left(\frac{1}{n}\right)$ , then

$$E(\hat{\theta} - \theta) = O\left(\frac{1}{n}\right)$$

Subsequently

$$\begin{aligned} b^2(\hat{\theta}) &= \left\{E(\hat{\theta} - \theta)\right\}^2 \\ &= O\left(\frac{1}{n^2}\right) \end{aligned}$$

As we only want to approximate the estimated bias up to order  $n^{-1}$ , so we can remove  $b^2(\hat{\theta})$  in equation 4.66.

$$E(\hat{\theta} - \theta) = -\frac{\text{Var}(\hat{\theta})G''(\theta)}{2G'(\theta)} + O\left(\frac{1}{n^2}\right) \quad (4.67)$$

Furthermore,  $\text{Var}(\hat{\theta})$  can be approximated by using the result of the first order Taylor expansion of  $G(\hat{\theta})$  in  $\hat{\theta}$  at  $\theta$ .

$$\begin{aligned} G(\theta) + G'(\theta)(\hat{\theta} - \theta) &\approx 0 \\ \hat{\theta} - \theta &\approx -\frac{G(\theta)}{G'(\theta)} \end{aligned} \quad (4.68)$$

Since  $G'(\theta)$  is a functions of parameter, then  $\text{Var}(\hat{\theta})$  can be derived from equation 4.68,

$$\text{Var}(\hat{\theta}) = \frac{\text{Var}(G(\theta))}{(G'(\theta))^2} \quad (4.69)$$

Substitute equation 4.69 into equation 4.67 leads to

$$E(\hat{\theta} - \theta) \approx -\frac{\text{Var}(G(\theta))G''(\theta)}{2(G'(\theta))^3}$$

By equation 4.59, when  $r = 1$

$$\text{Var}(G(\theta)) = \text{Var}(M_x^1(0) - m_1) \quad (4.70)$$

As  $M_x^1(0) = \left. \frac{dM_x(s)}{ds} \right|_{s=0}$  is a function of parameter, and  $m_1 = \frac{1}{n} \sum_{i=1}^n x_i$ , then 4.70 becomes

$$\begin{aligned} \text{Var}(G(\theta)) &= \text{Var}(m_1) \\ &= \frac{\text{Var}(x)}{n} \\ &= \frac{E(x^2) - (E(x))^2}{n} \\ &= \frac{M_x^2(0) - (M_x^1(0))^2}{n} \end{aligned}$$

In one dimensional parameter case, the explicit form of bias  $b(\hat{\theta}) = E(\hat{\theta} - \theta)$  in terms of MGF is

$$E(\hat{\theta} - \theta) \approx -\frac{\text{Var}(G(\theta))G'''(\theta)}{2(G'(\theta))^3} + O\left(\frac{1}{n^2}\right) \quad (4.71)$$

where

$$\begin{aligned} G'(\theta) &= \frac{d}{d\theta}(M_x^1(0)) \\ G''(\theta) &= \frac{d^2}{d\theta^2}(M_x^1(0)) \\ \text{Var}(G(\theta)) &= \frac{M_x^2(0) - (M_x^1(0))^2}{n} \end{aligned}$$

Hence the estimated bias of  $\hat{\theta}$  can be determined by substituting the value of  $\hat{\theta}$  in equation 4.71,

$$\widehat{b(\hat{\theta})} = -\frac{\text{Var}(G(\hat{\theta}))G'''(\hat{\theta})}{2(G'(\hat{\theta}))^3} + O\left(\frac{1}{n^2}\right)$$

□

*Proof.* **Multivariate case**

When we have  $d$  dimensional parameters  $\theta = (\theta_1, \dots, \theta_d)$ , the first order Taylor expansion of  $G(\hat{\theta})$  for  $\hat{\theta}$  at  $\theta$  is

$$G(\hat{\theta}) \approx G(\theta) + G'(\theta)(\hat{\theta} - \theta)$$

where

$$G = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_d \end{bmatrix} = \begin{bmatrix} M_x^1(0) - m_1 \\ M_x^2(0) - m_2 \\ \vdots \\ M_x^d(0) - m_d \end{bmatrix}, \quad G' = \begin{bmatrix} \left( \frac{\partial G_1}{\partial \theta_1} \right) & \dots & \left( \frac{\partial G_1}{\partial \theta_d} \right) \\ \left( \frac{\partial G_2}{\partial \theta_1} \right) & \dots & \left( \frac{\partial G_2}{\partial \theta_d} \right) \\ \vdots & \dots & \vdots \\ \left( \frac{\partial G_d}{\partial \theta_1} \right) & \dots & \left( \frac{\partial G_d}{\partial \theta_d} \right) \end{bmatrix}$$

Since  $G(\hat{\theta}) = 0$  is given in equation 4.61, then

$$\begin{aligned} G(\theta) + G'(\theta)(\hat{\theta} - \theta) &\approx 0 \\ \hat{\theta} - \theta &\approx -(G'(\theta))^{-1}G(\theta) \end{aligned}$$

i.e.

$$\hat{\theta}_s - \theta_s = - \sum_{p=1}^d g_{sp} G_p(\theta) \quad (4.72)$$

for  $s = 1, \dots, d$ , where  $g_{sp}$  is the  $(s, p)$ -th entry of matrix  $[G'(\theta)]^{-1}$ . The second order multivariate Taylor expansion for each  $G_r(\hat{\theta})$ ,  $r = 1, \dots, d$ , of  $\hat{\theta}$  at  $\theta$  is

$$\begin{aligned} G_r(\hat{\theta}) &\approx G_r(\theta) + G'_r(\theta)(\hat{\theta} - \theta) + \frac{1}{2}(\hat{\theta} - \theta)^T G''_r(\theta)(\hat{\theta} - \theta) \\ &\approx G_r(\theta) + \sum_{s=1}^d (\hat{\theta}_s - \theta_s) \frac{\partial G_r(\theta)}{\partial \theta_s} + \frac{1}{2} \sum_{t=1}^d \sum_{u=1}^d (\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u) \frac{\partial^2 G_r(\theta)}{\partial \theta_t \partial \theta_u} \end{aligned} \quad (4.73)$$

Take expectation on both side of equation 4.73.

$$\begin{aligned} E(G_r(\hat{\theta})) &\approx E(G_r(\theta)) + \sum_{s=1}^d E \left\{ (\hat{\theta}_s - \theta_s) \frac{\partial G_r(\theta)}{\partial \theta_s} \right\} \\ &\quad + \frac{1}{2} \sum_{t=1}^d \sum_{u=1}^d E \left\{ (\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u) \frac{\partial^2 G_r(\theta)}{\partial \theta_t \partial \theta_u} \right\} \end{aligned} \quad (4.74)$$

By equation 4.60, the first term on the right hand side of equation 4.74 equals zeros, the second term can be expressed as

$$E\left\{(\hat{\theta}_s - \theta_s) \frac{\partial G_r(\theta)}{\partial \theta_s}\right\} = \text{cov}\left(\hat{\theta}_s - \theta_s, \frac{\partial G_r(\theta)}{\partial \theta_s}\right) + E(\hat{\theta}_s - \theta_s)E\left(\frac{\partial G_r(\theta)}{\partial \theta_s}\right) \quad (4.75)$$

By substituting equation 4.72 in  $\text{cov}\left(\hat{\theta}_s - \theta_s, \frac{\partial G_r(\theta)}{\partial \theta_s}\right)$ , then

$$\begin{aligned} \text{cov}\left(\hat{\theta}_s - \theta_s, \frac{\partial G_r(\theta)}{\partial \theta_s}\right) &= \text{cov}\left(-\sum_{p=1}^d g_{sp} G_p(\theta), \frac{\partial G_r(\theta)}{\partial \theta_s}\right) \\ &= -\sum_{p=1}^d g_{sp} \text{cov}\left(G_p(\theta), \frac{\partial G_r(\theta)}{\partial \theta_s}\right) \\ &= -\sum_{p=1}^d g_{sp} \left\{ E\left(G_p(\theta) \frac{\partial G_r(\theta)}{\partial \theta_s}\right) - E(G_p(\theta))E\left(\frac{\partial G_r(\theta)}{\partial \theta_s}\right) \right\} \end{aligned}$$

By equation 4.59,  $\frac{\partial G_r(\theta)}{\partial \theta_s}$  is a function of parameter, then

$$E\left(G_p(\theta) \frac{\partial G_r(\theta)}{\partial \theta_s}\right) = \frac{\partial G_r(\theta)}{\partial \theta_s} E(G_p(\theta))$$

By equation 4.60, we have  $E(G_p(\theta)) = 0$ , then

$$\text{cov}\left(\hat{\theta}_s - \theta_s, \frac{\partial G_r(\theta)}{\partial \theta_s}\right) = 0 \quad (4.76)$$

Substitute 4.76 into 4.75, we have

$$E\left\{(\hat{\theta}_s - \theta_s) \frac{\partial G_r(\theta)}{\partial \theta_s}\right\} = E(\hat{\theta}_s - \theta_s)E\left(\frac{\partial G_r(\theta)}{\partial \theta_s}\right) \quad (4.77)$$

The third terms on the right hand side of equation 4.74 can be expressed as

$$E\left\{(\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u)\frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right\} = \text{cov}\left\{(\hat{\theta}_t - \theta)(\hat{\theta}_u - \theta), \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right\} \quad (4.78)$$

$$+ E\left\{(\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u)\right\} E\left(\frac{\partial G_r(\theta)}{\partial\theta_t\partial\theta_u}\right)$$

Replace  $\hat{\theta}_t - \theta_t$  and  $\hat{\theta}_u - \theta_u$  by equation 4.72 then

$$\text{cov}\left\{(\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u), \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right\} = \text{cov}\left(\sum_{p=1}^d \sum_{q=1}^d g_{tp} G_p g_{uq} G_q, \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right)$$

$$= \sum_{p=1}^d \sum_{q=1}^d g_{tp} g_{uq} \text{cov}\left(G_p G_q, \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right)$$

where  $\text{cov}\left(G_p G_q, \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right) = 0$  because  $G_p G_q$  and  $\frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}$  are independent, then

$$\text{cov}\left\{(\hat{\theta}_t - \theta)(\hat{\theta}_u - \theta), \frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right\} = 0 \quad (4.79)$$

On the other hand

$$E\left\{(\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u)\right\} = \text{cov}(\hat{\theta}_t - \theta_t, \hat{\theta}_u - \theta_u) + E(\hat{\theta}_t - \theta_t)E(\hat{\theta}_u - \theta_u)$$

$$= \text{cov}(\hat{\theta}_t, \hat{\theta}_u) + O\left(\frac{1}{n^2}\right) \quad (4.80)$$

By substituting the result 4.79 and 4.80 in equation 4.78, we have

$$E\left\{(\hat{\theta}_t - \theta_t)(\hat{\theta}_u - \theta_u)\frac{\partial^2 G_r(\theta)}{\partial\theta_t\partial\theta_u}\right\} = \text{cov}(\hat{\theta}_t, \hat{\theta}_u) E\left(\frac{\partial G_r(\theta)}{\partial\theta_t\partial\theta_u}\right) \quad (4.81)$$

Since  $E(G(\hat{\theta})) = 0$  by 4.61, we substitute 4.77 and 4.81 into 4.74, then

$$\sum_{s=1}^d E(\hat{\theta}_s - \theta_s) E\left(\frac{\partial G_r(\theta)}{\partial\theta_s}\right) + \frac{1}{2} \sum_{t=1}^d \sum_{u=1}^d \text{cov}(\hat{\theta}_t, \hat{\theta}_u) E\left(\frac{\partial G_r(\theta)}{\partial\theta_t\partial\theta_u}\right) \approx 0$$



$$\sum_{s=1}^d E(\hat{\theta}_s - \theta_s) E\left(\frac{\partial G_r(\theta)}{\partial \theta_s}\right) = -\frac{1}{2} \sum_{t=1}^d \sum_{u=1}^d \text{cov}(\hat{\theta}_t, \hat{\theta}_u) E\left(\frac{\partial G_r(\theta)}{\partial \theta_t \partial \theta_u}\right) + O\left(\frac{1}{n^2}\right) \quad (4.82)$$

for  $r = 1, \dots, d$ .

We can write equation 4.82 in matrix form and solve the set of equations for the bias  $E(\hat{\theta}_s - \theta_s)$  of order  $n^{-1}$ .

$$\begin{aligned} BE &= A \\ E &= B^{-1}A \end{aligned}$$

where

$$E = \begin{bmatrix} E(\hat{\theta}_1 - \theta_1) \\ E(\hat{\theta}_2 - \theta_2) \\ \vdots \\ E(\hat{\theta}_d - \theta_d) \end{bmatrix}, \quad B = \begin{bmatrix} E\left(\frac{\partial G_1}{\partial \theta_1}\right) & \cdots & E\left(\frac{\partial G_1}{\partial \theta_d}\right) \\ E\left(\frac{\partial G_2}{\partial \theta_1}\right) & \cdots & E\left(\frac{\partial G_2}{\partial \theta_d}\right) \\ \vdots & \cdots & \vdots \\ E\left(\frac{\partial G_d}{\partial \theta_1}\right) & \cdots & E\left(\frac{\partial G_d}{\partial \theta_d}\right) \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}$$

and

$$A_i = -\frac{1}{2} CV_i \quad \text{for } i = 1, \dots, d.$$

$$V_i = [v^{(1)} \mid v^{(2)} \mid \dots \mid v^{(d)}]^T$$

$$v^{(j)} = \left[ E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_1}\right), E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_2}\right), \dots, E\left(\frac{\partial^2 G_i(\theta)}{\partial \theta_j \partial \theta_d}\right) \right]$$

and

$$C = [c^{(1)} \mid c^{(2)} \mid \dots \mid c^{(d)}]$$

where,

$$c^{(j)} = [\text{cov}(\hat{\theta}_j, \hat{\theta}_1), \text{cov}(\hat{\theta}_j, \hat{\theta}_2), \dots, \text{cov}(\hat{\theta}_j, \hat{\theta}_d)]$$

$$\text{cov}(\hat{\theta}_s, \hat{\theta}_t) = \begin{cases} \text{Var}(\hat{\theta}_s) & \text{if } s = t \\ 0 & \text{if } s \neq t \end{cases}$$

for  $j = 1, \dots, d$

□

Note that we can express  $\text{Var}(\hat{\theta})$  in terms of MGF. By equation 4.72 and 4.96, we have

$$\begin{aligned}
 \text{Var}(\hat{\theta}_s) &= \text{Var}\left[-\sum_{p=1}^d g_{sp} G_p(\theta)\right] \\
 &= \sum_{p=1}^d g_{sp}^2 \text{Var}[G_p(\theta)] \\
 &= \sum_{p=1}^d g_{sp}^2 \text{Var}[M_x^p(0) - m_p] \\
 &= \sum_{p=1}^d \frac{g_{sp}^2}{n} \text{Var}[x^p] \\
 &= \sum_{p=1}^d \frac{g_{sp}^2}{n} \left\{ E(x^{2p}) - [E(x^p)]^2 \right\} \\
 &= \sum_{p=1}^d \frac{g_{sp}^2}{n} \left\{ M_x^{2p}(0) - [M_x^p(0)]^2 \right\}
 \end{aligned} \tag{4.83}$$

$g_{sp}$  is the  $(s, p)$ -th entry of matrix  $[G'(\theta)]^{-1}$ .

## 4.5 Comparison of MLE and MM

### 4.5.1 Example 7: Series network with exponential waiting time

Figure 4.1 is a simple series network with exponential waiting time, we estimate the parameters  $\theta = (\lambda_1, \lambda_2)$  by maximum likelihood estimation, method of moments, and apply our formula to obtain the biased correction method of moment estimator. We will compare these estimators by their Mean Square Error (MSE).

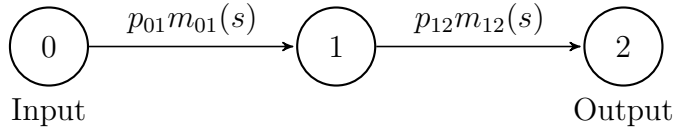


Figure 4.1: Series flowgraph model for hydraulic pump system

Flow direction	Distribution	MGF
0 $\rightarrow$ 1	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
1 $\rightarrow$ 2	Exponential( $\lambda_2$ )	$m_{12}(s) = \frac{\lambda_2}{\lambda_2 - s}$

The MGF of total waiting time from node 0 to node 2 is

$$M_{T_{02}}(s) = \left( \frac{\lambda_1}{\lambda_1 - s} \right) \left( \frac{\lambda_2}{\lambda_2 - s} \right)$$

The probability density function can be obtained by computing the inverse laplace transform of  $M_{T_{02}}(s)$

$$f_T(t) = \frac{\lambda_1 \lambda_2 (e^{-\lambda_2 t} - e^{-\lambda_1 t})}{\lambda_1 - \lambda_2}$$

Given data  $t_1, \dots, t_n$ , the log-likelihood function is

$$\begin{aligned} L(\lambda_1, \lambda_2) &= \log \left( \prod_{i=1}^n f_T(t_i) \right) \\ &= n \log \left( \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} \right) + \sum_{i=1}^n \log (e^{-\lambda_2 t_i} - e^{-\lambda_1 t_i}) \end{aligned}$$

We then need to solve the following equations for the MLE

$$\begin{aligned} \frac{\partial L}{\partial \lambda_1} &= -\frac{\lambda_2 n}{\lambda_1(\lambda_1 - \lambda_2)} + \sum_{i=1}^n \frac{t_i \exp(-\lambda_1 t_i)}{\exp(-\lambda_2 t_i) - \exp(-\lambda_1 t_i)} = 0 \\ \frac{\partial L}{\partial \lambda_2} &= \frac{\lambda_1 n}{\lambda_2(\lambda_1 - \lambda_2)} + \sum_{i=1}^n \frac{t_i \exp(-\lambda_2 t_i)}{\exp(-\lambda_1 t_i) - \exp(-\lambda_2 t_i)} = 0 \end{aligned} \quad (4.84)$$

The MLE are computed by using the Newton-Raphson method to numerically solve above equations for  $\lambda_1$  and  $\lambda_2$ .

In order to apply the method of moments, we need to solve two equations for estimating two unknown parameter  $\lambda_1$  and  $\lambda_2$ .

$$\begin{aligned}\frac{1}{\lambda_1} + \frac{1}{\lambda_2} - m_1 &= 0 \\ 2 \left( \frac{1}{\lambda_1^2} + \frac{1}{\lambda_1 \lambda_2} + \frac{1}{\lambda_2^2} \right) - m_2 &= 0\end{aligned}$$

where  $m_1 = \frac{1}{n} \sum_{i=1}^n t_i$ ,  $m_2 = \frac{1}{n} \sum_{i=1}^n t_i^2$ .

Define  $a = \frac{1}{\lambda_1}$  and  $b = \frac{1}{\lambda_2}$ , to reparameterise the above equations as

$$\begin{aligned}a + b - m_1 &= 0 \\ a^2 + ab + b^2 - \frac{1}{2}m_2 &= 0\end{aligned}$$

which can be easily solved and gives closed form expression for parameter estimation

$$\begin{aligned}\hat{\lambda}_{01} &= \frac{2}{m_1 + \sqrt{2m_2 - 3(m_1)^2}} \\ \hat{\lambda}_{12} &= \frac{2}{m_1 - \sqrt{2m_2 - 3(m_1)^2}}\end{aligned}\tag{4.85}$$

For the bias corrected method of moments estimator, we need to first estimate the bias  $b(\theta)$  that is given in Theorem 8. Let  $G$  be a  $2 \times 1$  vector

$$\begin{aligned}G &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \\ &= \begin{bmatrix} M_x^1(0) - m_1 \\ M_x^2(0) - m_2 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - m_1 \\ 2 \left( \frac{1}{\lambda_1^2} + \frac{1}{\lambda_1 \lambda_2} + \frac{1}{\lambda_2^2} \right) - m_2 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}
 G' &= \begin{bmatrix} \frac{\partial G_1}{\partial \lambda_1} & \frac{\partial G_1}{\partial \lambda_2} \\ \frac{\partial G_2}{\partial \lambda_1} & \frac{\partial G_2}{\partial \lambda_2} \end{bmatrix} \\
 &= \begin{bmatrix} -\lambda_1^{-2} & -\lambda_2^{-2} \\ -4\lambda_1^{-3} - 2\lambda_1^{-2}\lambda_2^{-1} & -2\lambda_1^{-1}\lambda_2^{-2} - 4\lambda_2^{-3} \end{bmatrix}
 \end{aligned}$$

The inverse of  $G'$  is

$$\begin{aligned}
 (G')^{-1} &= \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \\
 &= \begin{bmatrix} -\frac{(\lambda_2+2\lambda_1)\lambda_1^2}{\lambda_1-\lambda_2} & \frac{\lambda_2\lambda_1^3}{2(\lambda_1-\lambda_2)} \\ \frac{(2\lambda_2+\lambda_1)\lambda_2^2}{\lambda_1-\lambda_2} & -\frac{\lambda_1\lambda_2^3}{2(\lambda_1-\lambda_2)} \end{bmatrix} \tag{4.86}
 \end{aligned}$$

and

$$\begin{aligned}
 V_1 &= \left[ E\left(\frac{\partial^2 G_1}{\partial \lambda_1^2}\right), E\left(\frac{\partial^2 G_1}{\partial \lambda_1 \lambda_2}\right), E\left(\frac{\partial^2 G_1}{\partial \lambda_2 \lambda_1}\right), E\left(\frac{\partial^2 G_1}{\partial \lambda_2^2}\right) \right]^T \\
 &= \left[ \frac{2}{\lambda_1^3}, 0, 0, \frac{2}{\lambda_2^3} \right]^T
 \end{aligned}$$

$$\begin{aligned}
 V_2 &= \left[ E\left(\frac{\partial^2 G_2}{\partial \lambda_1^2}\right), E\left(\frac{\partial^2 G_2}{\partial \lambda_1 \lambda_2}\right), E\left(\frac{\partial^2 G_2}{\partial \lambda_2 \lambda_1}\right), E\left(\frac{\partial^2 G_2}{\partial \lambda_2^2}\right) \right]^T \\
 &= \left[ \frac{12}{\lambda_1^4} + \frac{4}{\lambda_1^3 \lambda_2}, \frac{2}{\lambda_1^2 \lambda_2^2}, \frac{2}{\lambda_1^2 \lambda_2^2}, \frac{4}{\lambda_1 \lambda_2^3} + \frac{12}{\lambda_2^4} \right]^T
 \end{aligned}$$

$$C = [c^{(1)}, 0, 0, c^{(4)}]$$

where  $c_1$  and  $c_4$  can be determined by formula given in equation 4.83 such that

$$\begin{aligned}
 c^{(1)} &= g_{11}^2 \frac{\text{Var}(x)}{n} + g_{12}^2 \frac{\text{Var}(x^2)}{n} \\
 c^{(4)} &= g_{21}^2 \frac{\text{Var}(x)}{n} + g_{22}^2 \frac{\text{Var}(x^2)}{n}
 \end{aligned}$$

Also

$$\begin{aligned}\text{Var}(x) &= E(x^2) - (E(x))^2 \\ &= M_x^2(0) - (M_x^1(0))^2 \\ &= \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2}\end{aligned}$$

$$\begin{aligned}\text{Var}(x^2) &= E(x^4) - (E(x^2))^2 \\ &= M_x^4(0) - (M_x^2(0))^2 \\ &= \frac{4}{\lambda_1^4 \lambda_2^4} (5\lambda_2^4 + 4\lambda_1 \lambda_2^3 + 3\lambda_1^2 \lambda_2^2 + 4\lambda_1^3 \lambda_2 + 5\lambda_1^4)\end{aligned}$$

then

$$\begin{aligned}A_1 &= -\frac{1}{2}CV_1 \\ A_2 &= -\frac{1}{2}CV_2\end{aligned}$$

Hence the expression for the bias in Method of Moment estimator are

$$\begin{aligned}b_1(\lambda_1) &= E(\hat{\lambda}_1 - \lambda_1) \\ &= g_{11}A_1 + g_{12}A_2 \\ &= \frac{\lambda_1(-21\lambda_2^5 - 16\lambda_1^2\lambda_2^3 - 18\lambda_1\lambda_2^4 - 16\lambda_1^3\lambda_2^2 - 16\lambda_1^4\lambda_2 + 9\lambda_1^5)}{(\lambda_1 - \lambda_2)^3\lambda_2^2n} \quad (4.87)\end{aligned}$$

$$\begin{aligned}b_2(\lambda_2) &= E(\hat{\lambda}_2 - \lambda_2) \\ &= g_{21}A_1 + g_{22}A_2 \\ &= \frac{\lambda_2(16\lambda_1\lambda_2^4 + 16\lambda_1^3\lambda_2^2 + 16\lambda_1^2\lambda_2^3 + 18\lambda_1^4\lambda_2 + 21\lambda_1^5\lambda_2 - 9\lambda_2^5)}{(\lambda_1 - \lambda_2)^3\lambda_1^2n} \quad (4.88)\end{aligned}$$

where  $g_{ij}$  is defined in equation 4.86.

Recall that  $T_1$  is the random waiting time in node 0 until node 1 is reached, and  $T_2$  is the random waiting time in node 1 until node 2 is reached. To simulate  $n$  total waiting time data from node 0 to node 2, we simulate each internode waiting time independently,  $t_1 \sim \text{Exp}(\lambda_1)$  and  $t_2 \sim \text{Exp}(\lambda_2)$ , and then take the sum of these two internode waiting time,  $t = t_1 + t_2$ . Given the total waiting time data, we compute the maximum likelihood estimator (MLE)  $\hat{\theta}_{mle}$  by solving equations 4.84, method of moment (MM) estimator  $\hat{\theta}_m$  by equation 4.85, then the estimated bias in MM estimator can be determined by substituting the value of  $\hat{\theta}_m = (\hat{\lambda}_1, \hat{\lambda}_2)$  in equation 4.87 and 4.88, then the bias corrected MM estimator can be obtained as  $\hat{\theta}_{bc} = \hat{\theta}_m - b(\hat{\theta})$ .

We set the initial value at  $\lambda_1^{(0)} = 1.12$  and  $\lambda_2^{(0)} = 2.82$  for the numerical method (i.e. Newton-Raphson method) that determines the MLE and MM estimator. Sample size of 1000, 10000 and 100000 were simulated with  $\lambda_1 = 1$  and  $\lambda_2 = 3$ . The simulation is repeated for 1000 times, each time gives MLE  $\hat{\theta}_{mle}$ , MM estimator  $\hat{\theta}_m$ , and the biased corrected MM estimator  $\hat{\theta}_{bc}$ . The comparison of these estimators is given by examining their mean square error (MSE).

$$\text{MSE}(\hat{\theta}) = \text{E} \left\{ (\hat{\theta} - \theta)^2 \right\}$$

Table 4.22: Summary of Mean Square Error in  $\lambda_1$ 

Sample size	MM	BCMM	MLE
10000	$3.5385 \times 10^{-4}$	$1.4663 \times 10^{-4}$	$2.79912 \times 10^{-4}$
100000	$3.4553 \times 10^{-5}$	$2.6165 \times 10^{-5}$	$2.61894 \times 10^{-5}$
1000000	$3.22 \times 10^{-6}$	$2.45 \times 10^{-6}$	$2.22 \times 10^{-6}$

Table 4.23: Summary of Mean Square Error in  $\lambda_2$ 

Sample size	MM	BCMM	MLE
10000	$2.1393 \times 10^{-2}$	$1.8941 \times 10^{-2}$	$1.4810 \times 10^{-2}$
100000	$1.9756 \times 10^{-3}$	$1.9124 \times 10^{-3}$	$1.3470 \times 10^{-3}$
1000000	$1.92 \times 10^{-4}$	$1.49 \times 10^{-4}$	$1.32 \times 10^{-4}$

Table 4.22 and Table 4.23 show mean square error of three different estimators for  $\hat{\lambda}_1$  and  $\hat{\lambda}_2$ . In terms of accuracy, the maximum likelihood estimators give the best estimation, since they have the smallest mean square error in all different sample sizes. Although the method of moments (MM) estimator has the largest mean square error, the corresponding bias corrected method of moment (BCMM) estimators, after subtracting the estimated bias in equation 4.87 and 4.88, are effective in reducing mean square error. It can be seen that the bias corrected method of moments provides better result, where the mean square error difference between MLE and BCMM estimator is getting smaller than in standard MM estimator.

### 4.5.2 Computational time

To justify the computational efficiencies of the method of moment and maximum likelihood method, we present a direct comparison of times for computing MM estimator and MLE in three different flowgraph examples of this chapter. In each example, we use the same simulated total waiting time data, and set the corresponding numerical method to start from the same initial value of target parameters.

Table 4.24: Comparison of computational time (seconds)

Example	7	6.1.1	1.4.1
MLE	0.15	21.77	93.07
MM	0.07	0.27	8.22

Table 4.24 illustrates the actual computational time in seconds for both methods in estimating different number of parameters. It is clearly shown that the method of moments substantially reduces the time in parameter estimation over the maximum likelihood method by an order of magnitude.



# Chapter 5

## Tail area probability approximations

In order to determine the probability that a system will survive beyond a specified time, we are often interested in approximating the survival function of the total waiting time between two nodes of interest in a flowgraph, rather than its probability density function. The tail area probability of survival function is defined as  $P(T > t_p) = 1 - p$ , where  $t_p$  is the  $p$ -th percentile such that  $P(T < t_p) = p$ .

This chapter focuses on the approximations for tail probabilities of the total waiting time distribution, particularly the shape of the survival function  $P(T > t)$  when  $t$  is large, based on two different approaches, Padé approximation and saddlepoint approximation. In the first approach, we approximate the behavior of total waiting time survival function  $S(t) = P(T > t)$  at large  $t$  by a simple exponential function in the form of  $ve^{-at}$ , where the rate  $a$  can be obtained based on the analysis of singularities in the Padé approximation of MGF. The second approach is to estimate survival function by the use of Lugannani-Rice saddlepoint formula, which requires the original MGF of total waiting time. The major contributions we present is to show the connection between the two apparently unrelated methods in estimating survival function, and propose a bias corrected Padé-type saddlepoint approximation, which significantly simplifies the calculations of original saddlepoint approximation method.

This chapter is organised as follow. In Section 1 we review theorems that relate to asymptotic behavior of inverse Laplace transforms, which is the foundation of exponential function approximation for tail area probabilities. We then apply the the final value theorem to propose a new formula to determine the asymptotic constant and the asymptotic rate in this exponential function. Numerical comparisons are made with the true survival probabilities at different percentiles  $t_p$ . In section 2 we illustrate flowgraph examples for the error analysis in tail area approximation by the inversion of Padé approximation of MGF. Section 3 demonstrates a detailed procedure for applying Lugannani-Rice formula for survival function approximation. Section 4 extends the application of Lugannani-Rice formula by using the Padé approximation of a given MGF as baseline function, and propose a general Padé-type saddlepoint approximation method. Section 5 discusses the limiting behavior of error in saddlepoint approximation for tail area probability, and present a new bias correction method for Padé-type saddlepoint approximation of both PDF and survival function.

## **5.1 Tail area approximation by exponential function**

The main purpose in this section is to discuss the use of exponential approximations to describe the limiting behavior of survival function of total waiting time. First, we derive the form of exponential approximations based on the asymptotic property of Laplace transform. Secondly, we demonstrate through examples that tail area survival probabilities approximations provided by exponential function is exceptionally good. The comparison of approximation for tail probabilities based on exponential functions that constructed from different order of Padé approximation is also presented.

Lew (1973) discusses the asymptotic behavior of the inversion of Laplace transform, that is, a complex-valued locally integrable function  $f(t)$  on  $[0, +\infty)$  from  $L[f](s)$ . By equation (1.3) of Lew (1973), it is shown that the behavior of  $f(t)$  near  $t \rightarrow +\infty$  depends on the rightmost singularities of  $L[f](s)$ .

**Theorem 9.** *Lew's theorem*

Let  $f(t)$  be a function on  $[0, +\infty)$  with Laplace transform  $L[f](s)$  such that

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} \exp(ts) L[f](s) ds \quad (5.1)$$

where the integration is done along the vertical line  $\text{Re}(s) = \gamma$  in the complex plane such that  $\gamma$  is greater than the real part of all singularities of  $L[f](s)$ . If  $L[f](s)$  can be continued analytically to the left, and if the contour of (5.1) can be moved sufficiently in that direction, then the behavior of  $f(t)$  as  $t \rightarrow \infty$  is determined by that of  $L[f](s)$  near its rightmost singularities.

$$f(t) = O(e^{\nu t}) \text{ as } t \rightarrow \infty$$

where  $\nu$  is the rightmost singularities of  $L[f](s)$

Suppose we let  $T$  be the total waiting time random variable with density function  $f(t)$  and MGF  $M_T(s)$ . Based on a Padé approach, we can approximate the MGF of total waiting time by a rational function, which can then be applied in the Heaviside formula to obtain true probability density function  $f(t)$  in the form of sum of exponential functions, with the rates are the poles in the denominator of Padé approximation ( see Lemma 4 on page 40). By Theorem 9, the behavior of  $f(t)$  near  $t \rightarrow +\infty$  is dominated by the corresponding Laplace transform near its rightmost singularities, then we can approximate the density function by an exponential function as

$$f(t) \approx ce^{-at} \quad \text{as } t \rightarrow \infty \quad (5.2)$$

where  $a$  is the absolute value of the rightmost singularities of  $M_T(-s)$ , and  $c$  is constant. It is important to note that 5.2 only hold for suitably large  $t$ , but does not hold for all  $t$ . We clarify on this point in Section 5.1.1 and Section 5.2.

**Definition 12.** Let  $T$  be a continuous random variable with CDF  $F(t)$ , PDF  $f(t)$  and MGF  $M_T(s)$  on the interval  $[0, +\infty)$ , the survival function of  $T$  is defined as follows:

$$\begin{aligned} S(t) &= P(T > t) \\ &= 1 - F(t) \\ &= \int_t^{\infty} f(x)dx \end{aligned}$$

In our case,  $T$  denotes the total waiting time between two nodes of interest in a flowgraph. By Definition 12 and equation 5.2, we can obtain an exponential function to estimate the tail area probabilities of survival function as follow

$$P(T > t) \approx ve^{-at} \quad \text{as } t \rightarrow \infty \quad (5.3)$$

where the asymptotic constant is

$$v = \frac{c}{a}$$

and the asymptotic decay rate  $a$  is the absolute value of the rightmost singularities of  $M_T(-s)$ . Furthermore, equation 5.3 is equivalent to

$$\lim_{t \rightarrow \infty} e^{at}P(T > t) = v$$

### 5.1.1 Example 1.3.1

We first consider the flowgraph model for a reversible illness-death system, which has exponential distribution for all internodes waiting time (see Table 3.1 on page 41), where  $\lambda_1 = 1$ ,  $\lambda_2 = 1.2$ ,  $\lambda_3 = 0.5$ ,  $\lambda_4 = 2$ , and  $p_{01} = p_{02} = p_{10} = p_{12} = \frac{1}{2}$ . The MGF of the total waiting time between node 0 and node 2 is

$$\begin{aligned} M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\ &= \frac{(2-s)(1.8 - 3.4s + s^2)}{(6 - 12s + 4s^2)(0.5 - s)(1.2 - s)} \end{aligned} \quad (5.4)$$

In Section 3.1 of Chapter 3, it was shown that the closed form expression of probability density function can be obtained by the exact inversion of MGF  $M_{T_{02}}(-s)$ . For a MGF with  $k$  singularities, the resulting total waiting time density function is in the following form

$$f(t) = c_1 e^{-a_1 t} + c_2 e^{-a_2 t} + \dots + c_k e^{-a_k t}$$

where  $a_1, \dots, a_k$  are the absolute value of singularities of  $M_{T_{02}}(-s)$ , and  $c_1, \dots, c_k$  are the corresponding constant terms.

By applying the Heaviside formula to invert equation 5.4, the true PDF is

$$f(t) = 0.75e^{-0.5t} - 0.3636e^{-1.2t} - 0.1207e^{-0.6340t} - 0.0157e^{-2.366t}$$

and the corresponding true survival function is

$$\begin{aligned} S(t) &= P(T > t) \\ &= 1.5e^{-0.5t} - 0.303e^{-1.2t} + 0.1903e^{-0.6340t} + 0.0066e^{-2.366t} \end{aligned} \quad (5.5)$$

For large  $t$ , the limit behavior of  $S(t)$  is dominated by the exponential function  $1.5e^{-0.5t}$ , which gives good approximation to the tail area probabilities of total waiting time survival function. Given numerator order  $p$  and denominator  $q$ , we establish Padé approximation of original MGF from 5.4, and compare the approximations for tail probabilities that constructed from each  $PA_{[p,q]}(s)$  with the true survival function given in 5.5. The detail expression of estimated survival functions in each case are given below. The Padé approximation of order  $p = 1$  and  $q = 2$  is

$$PA_{[1,2]}(-s) = \frac{1 + 0.4543s}{1 + 2.8988s + 1.771s^2} \quad (5.6)$$

where

$$f_{PA_{[1,2]}}(t) = 0.6753e^{-0.4942t} - 0.4188e^{-1.1426t}$$

and

$$S_{PA_{[1,2]}}(t) = 1.3665e^{-0.4942t} - 0.3665e^{-1.1426t}$$

The Padé approximation of order  $p = 1$  and  $q = 3$  is

$$\text{PA}_{[1,3]}(-s) = \frac{1 + 0.5272s}{1 + 2.9716s + 1.9491s^2 + 0.04813s^3} \quad (5.7)$$

where

$$f_{PA_{[1,3]}}(t) = 0.6874e^{-0.4957t} - 0.4086e^{-1.0767t} - 0.2788e^{-38.9275t}$$

and

$$S_{PA_{[1,3]}}(t) = 1.3866e^{-0.4957t} - 0.3795e^{-1.0767t} - 0.0072e^{-38.9275t}$$

The Padé approximation of order  $p = 2$  and  $q = 3$  is

$$\text{PA}_{[2,3]}(-s) = \frac{1 + 1.8019s + 0.5791s^2}{1 + 4.2463s + 5.6440s^2 + 2.3056s^3} \quad (5.8)$$

where

$$f_{PA_{[2,3]}}(t) = 0.7301e^{-0.4990t} - 0.1262e^{-0.6908t} - 0.3528e^{-1.2582t}$$

and

$$S_{PA_{[2,3]}}(t) = 1.4630e^{-0.4990t} - 0.1827e^{-0.6908t} - 0.2804e^{-1.2582t}$$

The Padé approximation of order  $p = 2$  and  $q = 4$  is

$$\text{PA}_{[2,4]}(-s) = \frac{1 + 1.9092s + 0.6357s^2}{1 + 4.3536s + 5.9631s^2 + 2.5149s^3 + 0.0052s^4} \quad (5.9)$$

where

$$f_{PA_{[2,4]}}(t) = 0.7434e^{-0.4997t} - 0.1262e^{-0.6508t} - 0.3672e^{-1.2287t} - 0.2537e^{-484.2466t}$$

and

$$S_{PA_{[2,4]}}(t) = 1.4877e^{-0.4997t} - 0.1883e^{-0.6508t} - 0.2988e^{-1.2287t} - 0.0005e^{-484.2466t}$$

The Padé approximation of order  $p = 3$  and  $q = 4$  is

$$\text{PA}_{[3,4]}(-s) = \frac{1 + 2.3890s + 1.5002s^2 + 0.2778s^3}{1 + 4.8334s + 8.0004s^2 + 5.2227s^3 + 1.1113s^4} \quad (5.10)$$

$$f_{PA_{[3,4]}}(t) = 0.75e^{-0.5t} - 0.3636e^{-1.12t} - 0.1207e^{-0.6340t} - 0.0157e^{-2.3656t}$$

$$S_{PA_{[3,4]}}(t) = 1.500000844e^{-0.5t} - 0.3030e^{-1.12t} - 0.1903e^{-0.6340t} - 0.006e^{-2.3656t}$$

Table 5.1: Summary of pole analysis

Method	asymptotic constant $v$	the right-most singularities $-a^*$
PA <sub>[1,2]</sub> ( $s$ )	1.366534374	-0.4941829741
PA <sub>[1,3]</sub> ( $s$ )	1.386621578	-0.4957436281
PA <sub>[2,3]</sub> ( $s$ )	1.463030169	-0.4990385780
PA <sub>[2,4]</sub> ( $s$ )	1.487693424	-0.4997200087
PA <sub>[3,4]</sub> ( $s$ )	1.500000844	-0.5
$M_{T_{02}}(s)$	1.5	-0.5

Table 5.1 summarise the value of parameters for the dominated exponential function, in the form of equation 5.3, to describe the limiting behavior of true survival function  $P(T > t)$ . Note that the last row of Table 5.1 shows the true value of the right-most singularities and the asymptotic constant that are computed based on the original MGF  $M_{T_{02}}(s)$ . Clearly, as the order of Padé approximation increases, the value of asymptotic constant  $v$  and the right-most singularities of Padé approximation of MGF,  $-a$ , converges to 1.5 and -0.5 respectively. These result imply that a higher order Padé approximation leads to an exponential function that gives more precise description for the limiting behavior of the total waiting time tail probabilities than a lower order Padé approximation.

Table 5.2 compares the explicit value of estimate survival probabilities  $P(T > t)$  based on the dominated exponential function that specified in Table 5.1. The comparison is given in estimating probability at each percentile  $t_p$ , where  $p = 50\%$ ,  $75\%$ ,  $90\%$ ,  $95\%$ ,  $99\%$ ,  $99.9\%$ , and  $99.9\%$ . The true survival probabilities are illustrated in the last column of Table 5.2. Numerical result shows that

Table 5.2: Summary of exponential approximation for survival function: The unit of probability for  $t^*$ ,  $t^{**}$  and  $t^{***}$  are  $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$  respectively

percentile p	Time $t_p$	PA <sub>[1,2]</sub>	PA <sub>[1,3]</sub>	PA <sub>[2,3]</sub>	PA <sub>[2,4]</sub>	PA <sub>[3,4]</sub>	True
50%	1.86	0.5442	0.5506	0.5774	0.5863	0.5909	0.5
75%	3.37	0.2581	0.2604	0.2718	0.2757	0.2777	0.25
90%	5.27	0.1155	0.1163	0.1207	0.1223	0.1231	0.1
95%	6.69	0.0501	0.0502	0.0519	0.0525	0.0528	0.05
99%*	9.95	0.9989	0.9980	1.0190	1.0292	1.0348	1
99.9%**	14.59	1.0099	1.0016	1.0072	1.0141	1.0183	1
99.99%***	19.21	1.02868	1.0129	1.0032	1.0069	1.0098	1

all the exponential functions give reasonable good approximation for the tail area probability of  $P(T > t)$ , particularly, we see that the exponential function based on  $PA_{[1,2]}(s)$  provides relatively the best estimation for the survival probabilities up to 99% percentile at time  $t_{0.99} = 9.95$ , whereas the optimal approximation of survival probabilities at time 14.59 and 19.21 are given by the exponential functions that derived from  $PA_{[1,3]}(s)$  and  $PA_{[2,3]}(s)$  respectively.

Since the value of survival probability becomes significantly small and approaches to zeros as  $t$  gets increasingly large, in order to give a better graphical comparison, we plot  $-\log(P(T > t))$  against  $t$ , where  $-\log(P(T > t)) = -\log(v) - \log(e^{-a^*t}) = -\log(v) + a^*t$  is a straight line with interception  $-\log(v)$  and positive slop  $a^*$ . Figure 5.1 compares approximation for distribution in term of  $-\log$  transform. It appears that the quality of approximation is generally very good, even there are relatively large difference between the estimated and the true value when  $t$  is smaller than the 50% percentile  $t_{0.5}$ , i.e.  $t < 1.86$  (see Figure 5.2). However, once  $t$  is greater than  $t_{0.9} = 5.27$ , the 90% percentile, we see a clear evidence of convergence to the true survival function.

Figure 5.2 indicates that the exponential approximation from  $PA_{[1,2]}(s)$  yields the best estimation for the true survival function when  $t \in [0, 1.86]$ , whereas Figure 5.3 illustrates the survival function approximation for  $t \in [19, 19.21]$  is optimal for exponential function that results from  $PA_{[2,3]}(s)$ . Furthermore, Figure 5.4 demonstrates the quality of these exponential functions in estimating tail area



probabilities of survival function for  $t \in [29.99, 30]$ , where the plot of exponential function from  $PA_{[3,4]}(s)$  is the closest to the true survival probabilities, follow by the estimation based on  $PA_{[2,4]}(s)$ , and the relatively poor approximation is observed in the exponential function derived from  $PA_{[2,3]}(s)$ .

We also make comparison for the estimation of hazard function  $\phi(t) = \frac{f(t)}{s(t)}$  based on the PDF and survival function that derived from different order of Padé approximation of MGF. The results of hazard function estimation are compared under different time intervals, see Figure 5.5, 5.6, 5.7, and 5.8. All of these results conclude that the closer to the order of original MGF, the greater accuracy we will have for estimating survival probability at large  $t$ .

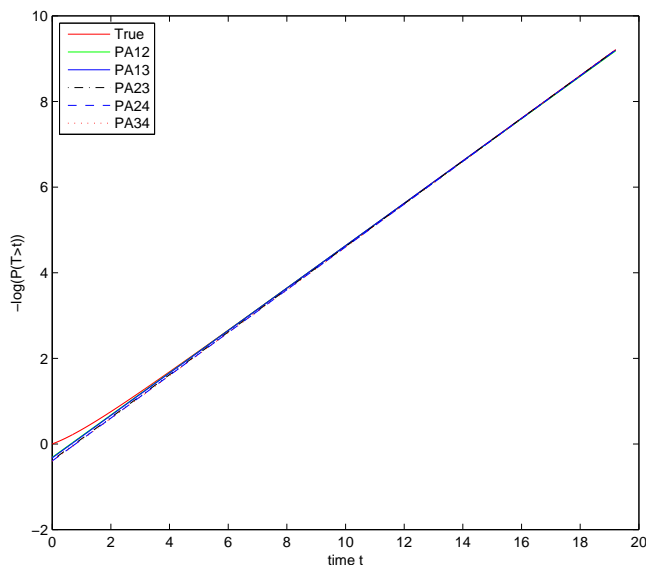


Figure 5.1: Plot of  $-\log (P(T > t))$  against  $t \in [0, 19.21]$

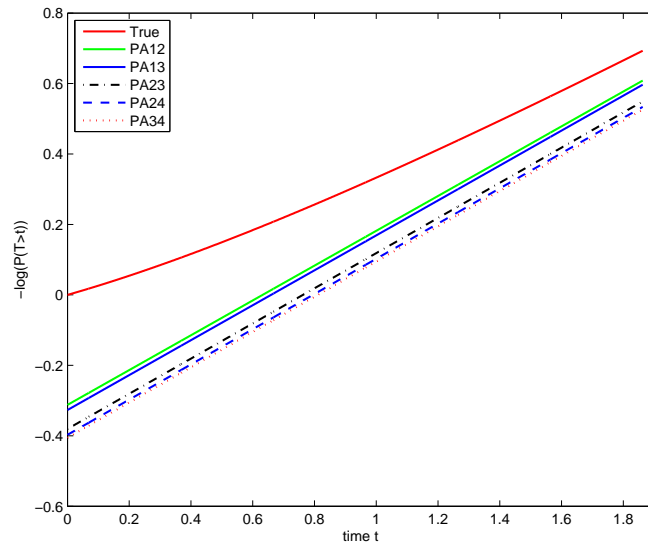


Figure 5.2: Plot of  $-\log(P(T > t))$  against  $t \in [0, 1.86]$

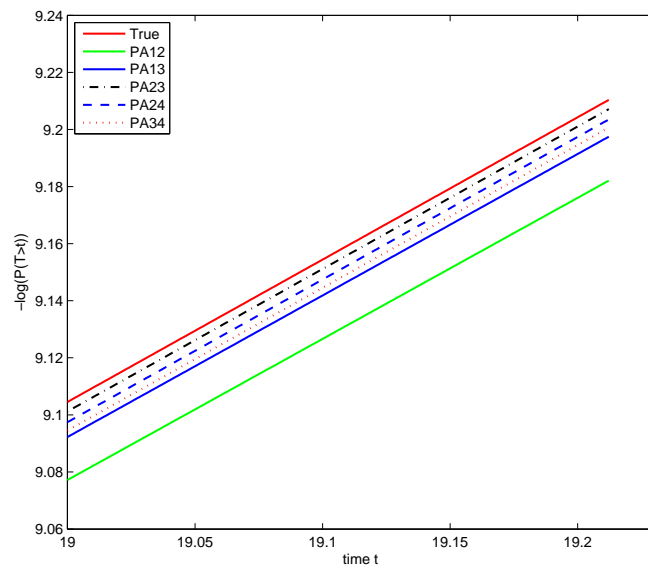


Figure 5.3: Plot of  $-\log(P(T > t))$  against  $t \in [19, 19.21]$

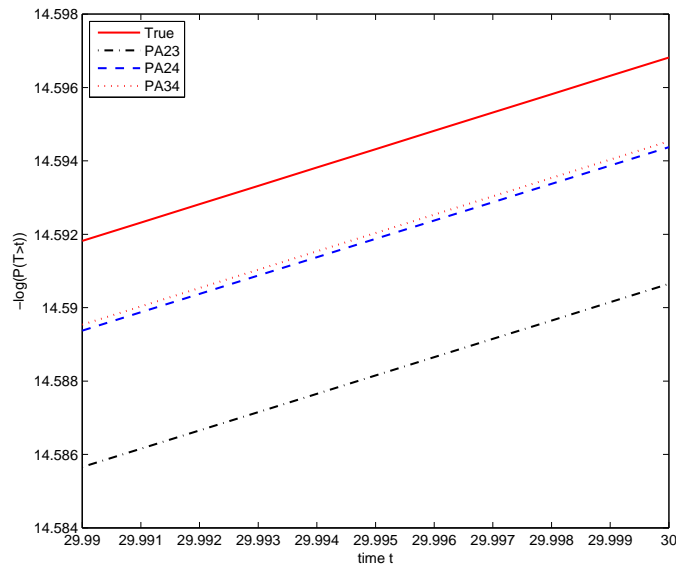


Figure 5.4: Plot of  $-\log(P(T > t))$  against  $t \in [29.99, 30]$

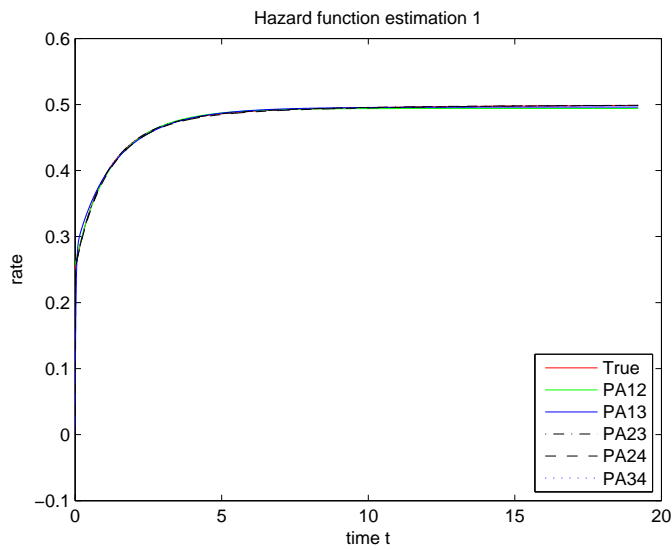


Figure 5.5: Plot of hazard function estimation against  $t \in [0, 19.21]$

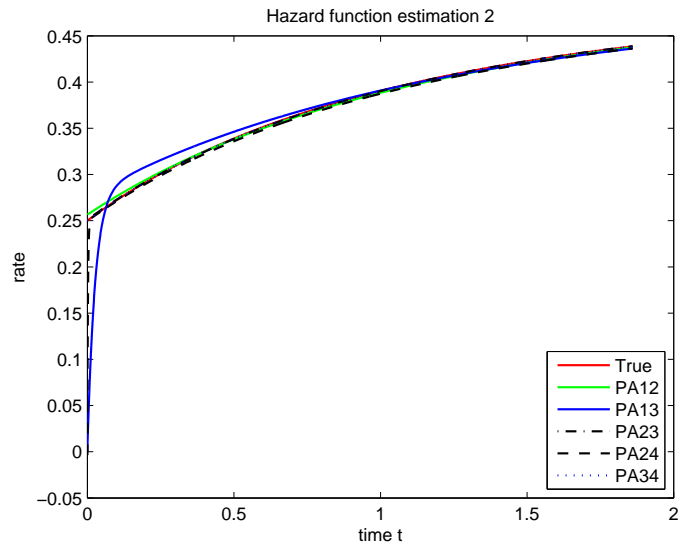


Figure 5.6: Plot of hazard function estimation against  $t \in [0, 1.86]$

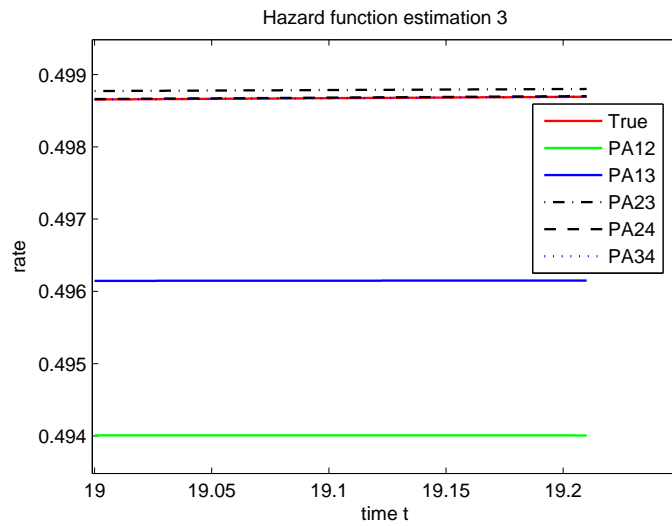


Figure 5.7: Plot of hazard function estimation against  $t \in [19, 19.21]$

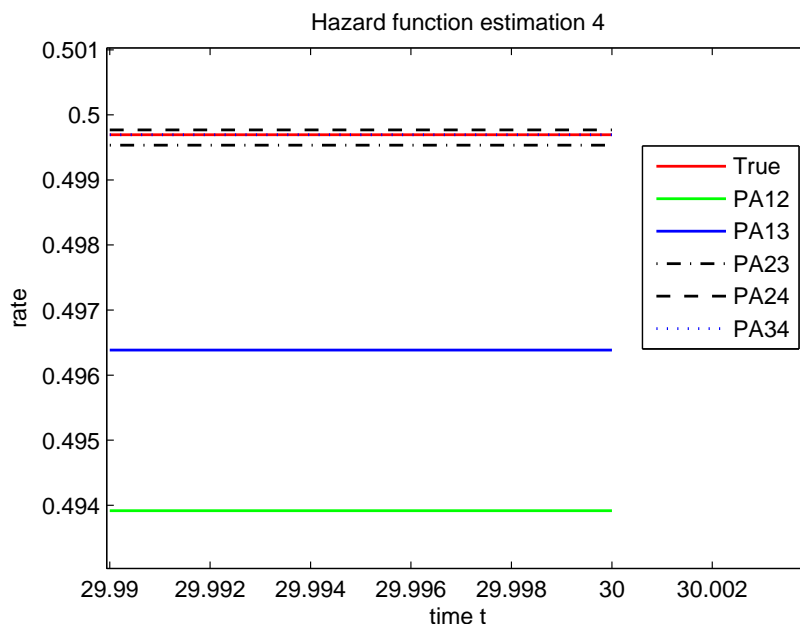


Figure 5.8: Plot of hazard function estimation against  $t \in [29.99, 30]$

### 5.1.2 Example 1.3.2

We now consider some internode waiting time to follow non-exponential distribution in a flowgraph, the PDF of total waiting time  $f(t)$  in this case usually doesn't have closed form expression, and the exponential function  $ce^{-at}$  can't be determined directly by inspecting  $f(t)$ . In general, the value of rate  $a$  can always be determined by finding the rightmost singularities of the Laplace transform of  $f(t)$ ,  $L[f](s)$  (i.e. solve the denominator of MGF  $M_T(-s)$  for  $s$ ), however, we still need to compute the constant term  $c$  in order to describe the limiting behavior of total waiting time distribution. By using the final value theorem, we suggest a simple method to determine the asymptotic constant  $c$  based on the MGF of total waiting time and the asymptotic decay rate  $a$ .

**Proposition 1.** Let  $T$  be a positive random variable with PDF  $f(t)$ , MGF  $M_T(s)$ . Suppose  $f(t)$  can be approximated as

$$f(t) \approx ce^{-at} \quad \text{as } t \rightarrow \infty$$

Then  $a$  is the absolute value of the right-most singularity of  $M_T(-s)$ , and

$$c = \lim_{s \rightarrow 0} sM_T(a - s) \quad (5.11)$$

The asymptotic constant  $c$  and asymptotic decay rate  $a$  are defined by

$$\lim_{t \rightarrow \infty} e^{at} f(t) = c \quad (5.12)$$

Let  $G(t) = e^{at} f(t)$ . By the final value theorem, we have

$$\lim_{t \rightarrow \infty} G(t) = \lim_{s \rightarrow 0} sL[G](s) \quad (5.13)$$

where  $L[G](s)$  is the Laplace transform of function  $G(t)$ .

Since

$$\begin{aligned} L[G](s) &= \int_0^{\infty} e^{-st} G(t) dt \\ &= \int_0^{\infty} e^{-st} e^{at} f(t) dt \\ &= \int_0^{\infty} e^{(a-s)t} f(t) dt \\ &= M_T(a - s) \end{aligned} \quad (5.14)$$

where  $M_T(s)$  is the MGF of total waiting time  $T$ .

Substituting 5.14 into 5.13. By equation 5.12, we therefore have

$$c = \lim_{s \rightarrow 0} sM_T(a - s)$$

where  $a$  is the absolute value of the right-most singularity of  $M_T(-s)$ .

The formula 5.11 is particularly useful for evaluating the asymptotic constant when the closed form expression of total waiting density is not available, because it only depends on the MGF of total waiting time, which can be easily obtain by the use of Mason's rule in Chapter 2. Since the Laplace transform of total waiting time density is just the MGF with argument  $-s$  instead of  $s$ , and the MGF is usually a rational function, we will always have the Laplace transform in rational form, which guarantees its singularities exists.

To show that our formula 5.11 in Proposition 1 works in the situation of non-closed form density function, we let the waiting time between node 1 and node 0 in Example 1.3.2 to follow Gamma distribution.

Table 5.3: Summary of waiting time distribution

Flow direction	Distribution	MGF
$0 \rightarrow 1$	Exponential( $\lambda_1$ )	$m_{01}(s) = \frac{\lambda_1}{\lambda_1 - s}$
$1 \rightarrow 0$	Gamma( $\alpha, \beta$ )	$m_{10}(s) = \left(\frac{\beta}{\beta - s}\right)^\alpha$
$0 \rightarrow 2$	Exponential( $\lambda_3$ )	$m_{02}(s) = \frac{\lambda_3}{\lambda_3 - s}$
$1 \rightarrow 2$	Exponential( $\lambda_2$ )	$m_{12}(s) = \frac{\lambda_2}{\lambda_2 - s}$

We assume  $\lambda_1 = 0.5$ ,  $\lambda_2 = 2$ ,  $\lambda_3 = 0.2$ ,  $\alpha = 1.5$ ,  $\beta = 2$ , and  $p_{01} = p_{02} = p_{10} = p_{12} = \frac{1}{2}$ . The MGF of the total waiting time between node 0 and node 2 is

$$\begin{aligned}
 M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\
 &= \frac{\frac{1}{4}\left(\frac{0.5}{0.5-s}\right)\left(\frac{2}{2-s}\right) + \frac{1}{2}\left(\frac{0.2}{0.2-s}\right)}{1 - \frac{1}{4}\left(\frac{0.5}{0.5-s}\right)\left(\frac{2}{2-s}\right)^{1.5}} \\
 &= \frac{0.6 - 2s - 0.4s^2}{(s-2)(s-0.2)\left\{2 - 4s - 1.4142\left(\frac{1}{2-s}\right)^{1.5}\right\}} \tag{5.15}
 \end{aligned}$$

From equation 5.15, we see that the denominator of MGF  $M_{T_{02}}(s)$  is not a polynomial function, because the value of  $\alpha$  is non-integer (i.e.  $\alpha = 1.5$ ), which does not allow us to decompose  $M_{T_{02}}(s)$  by partial fraction, and it leads to a non-closed form probability density function, which it can only be determined by the numerical inversion of Laplace transforms. Also note that the Laplace transform of  $f(t)$  is  $M_{T_{02}}(-s)$ , and the singularities can be obtained easily by solving the denominator of  $M_{T_{02}}(-s)$  for  $s$  such that

$$(s + 2)(s + 0.2) \left\{ 2 + 4s - 1.4142 \left( \frac{1}{2 + s} \right)^{1.5} \right\} = 0$$

The solutions are  $s_1 = -2$ ,  $s_2 = -0.2$ ,  $s_3 = -0.3354$ ,  $s_4 = -2.2051 - 0.2801i$ ,  $s_5 = -2.2051 + 0.2801i$

As the rightmost singularity is  $s_2$ , the asymptotic decay rate is  $a = |s_2| = 0.2$ , then we have

$$M_{T_{02}}(0.2 - s) = \frac{0.2 + 2s + 0.4(0.2 - s)^2}{s(1.8 + s) \left\{ 1.2 + 4s - 1.4142 \left( \frac{1}{1.8+s} \right)^{1.5} \right\}}$$

By equation 5.11, we have

$$c = \lim_{s \rightarrow 0} sM_{T_{02}}(0.2 - s) = 0.1953$$

Next, we simulated a sample size of 10000 total waiting time data between node 0 and node 2, and computed the estimated density function  $\hat{f}(t)$  by using De Hoog's method (1982) to numerically invert  $M_{T_{02}}(-s)$ . To verify the value of asymptotic constant, function  $G(t) = \hat{f}(t)/\exp(-0.2t)$  is plotted against  $t$  in Figure 5.9. As can be seen that the evidence of convergence of function  $G(t)$  is remarkable for large  $t$ , where the solid line describes the behavior of function  $G(t)$ , and the horizontal dash line indicate the value of limit 0.1953. Therefore, the PDF can be approximated by an exponential function in the following form

$$f(t) \approx 0.1953e^{-0.2t} \quad \text{as } t \rightarrow \infty \quad (5.16)$$



To further investigate the quality of approximation for tail area probabilities based on equation 5.16, we take log on both side of equation 5.16

$$\log(f(t)) \approx \log(0.1953) - 0.2t$$

If our calculation for parameter  $c$  and  $a$  are correct, we expect that the plot of  $\log(0.1953) - 0.2t$  is similar to the shape of  $\log(f(t))$  when  $t$  is large. Figure 5.10 illustrates that there is a remarkably good agreement between  $\log(f(t))$  and  $\log(0.1953) - 0.2t$  in almost the entire range of  $t$ . Furthermore, we can obtain the approximation for survival function based on equation 5.16,

$$P(T > t) \approx 0.9765e^{-0.2t} \quad \text{as } t \rightarrow \infty \quad (5.17)$$

Similarly, the plot of  $-\log(P(T > t))$  against  $-\log(0.9765) + 0.2t$  is presented in Figure 5.11 to check the asymptotic constant 0.9765.

Table 5.4: Summary of exponential approximation for PDF: The unit of probability for  $t^*$ ,  $t^{**}$  are  $10^{-4}$ ,  $10^{-5}$  respectively

Time	$f(t)$	$0.1953 \exp(-0.2t)$
3.57	0.1030	0.0955
6.96	0.0509	0.0486
11.47	0.0202	0.0197
14.91	0.0101	0.0099
22.89	0.0020	0.0021
34.14*	2.1199	2.1169
44.49**	2.6709	2.6697

The comparison of approximation for  $f(t)$  and  $P(T > t)$  with estimated value based on  $0.1953 \exp(-0.2t)$  and  $0.9765 \exp(-0.2t)$  is presented in Table 5.4 and Table 5.5 separately. The numerical results show that the quality of exponential function approximation is exceptionally good. All these results indicate that equation 5.16 and 5.17 can correctly provide the description of the characteristic tail behavior of the true PDF and survival function.

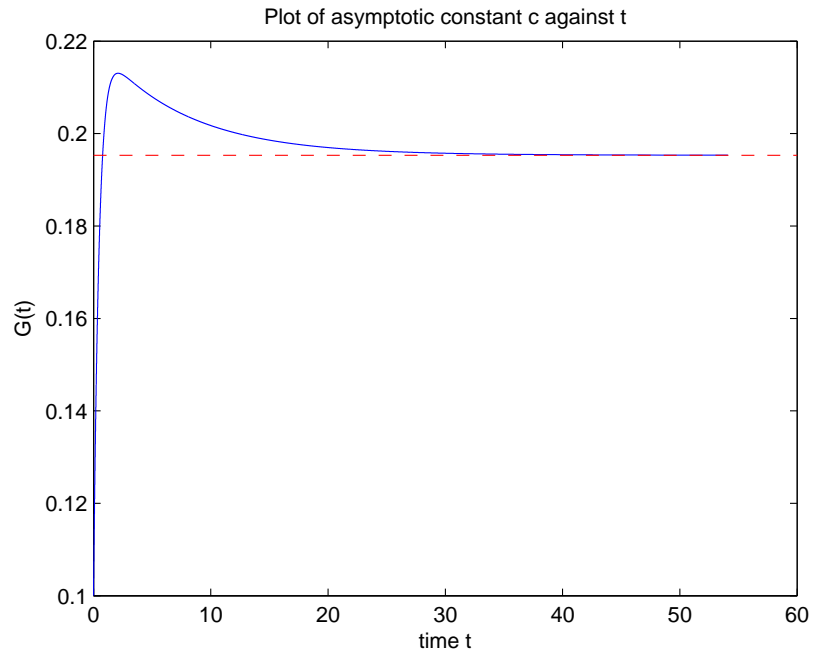


Figure 5.9: Plot of the function  $G(t) = e^{at}f(t)$  against  $t$

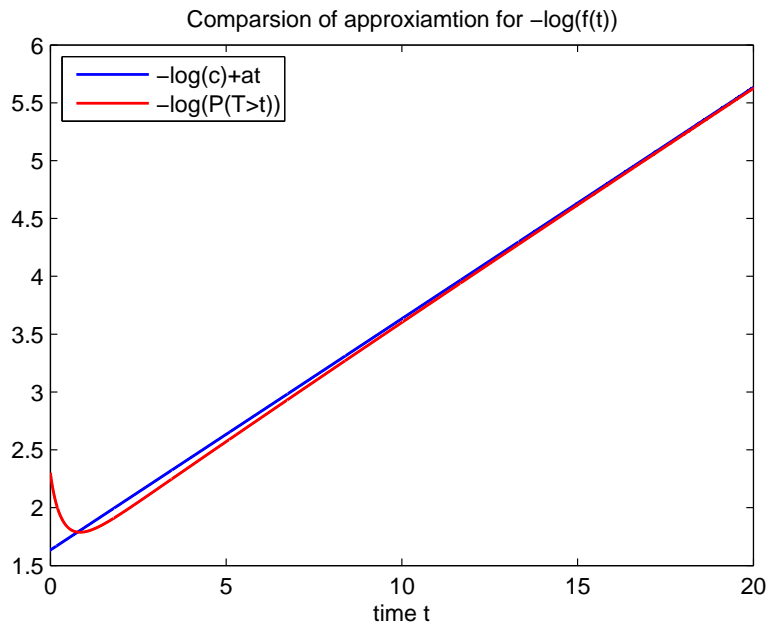
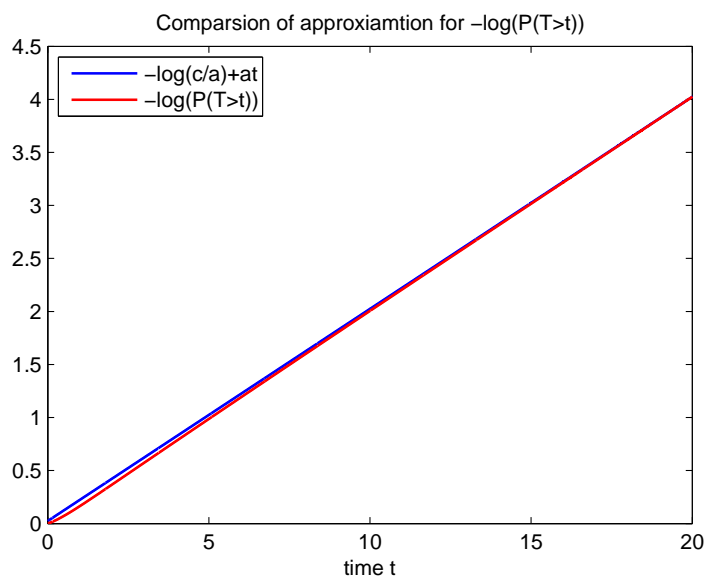


Figure 5.10: Plot of  $-\log(f(t))$  and  $-\log(0.1953) + 0.2t$  against  $t$

Table 5.5: Summary of exponential approximation for tail distribution.

Percentile	Time	$P(T > t)$	$0.9765 \exp(-0.2t)$
50%	3.57	0.5	0.4777
75%	6.96	0.25	0.2429
90%	11.47	0.1	0.0985
95%	14.91	0.05	0.0496
99%	22.89	0.01	0.0101
99.9%	34.14	0.001	0.0011
99.99%	44.49	0.0001	0.00013

Figure 5.11: Plot of  $-\log(P(T > t))$  and  $-\log(0.9765) + 0.2t$  against  $t$ 

## 5.2 Inversion of Padé approximations

As we have seen in Section 5.1.1, Chapter 3, Padé approximation is the only method that can provide density estimation in the form of a mixture of exponential densities, which is closely related to the exponential approximation in Section 5.1. This distinctive feature allows us to apply the formula we have proposed in

Proposition 1 to estimate tail area probabilities. For application, it is very important to examine the size of approximation error, and the quality of approximation for survival function must be assessed by evaluating the size of difference between the estimated value and true value of  $P(T > t)$ . In this section, we focus on estimating the error in survival function approximation in Example 1.3.2, and develop an closed form expression for modeling the behavior of error in the estimation of tail area probabilities of survival function.

Let  $M(s) = \frac{U(s)}{R(s)}$  be the MGF of total waiting time between input and output of a flowgraph, where the degree of polynomial  $U(s)$  and  $R(s)$  are  $p$  and  $q$  respectively. By Definition 3.21, Chapter 3, Let  $\text{PA}_{[\hat{p}, \hat{q}]}(s)$  denote the Padé approximation of  $M(s)$  with numerator order  $\hat{p}$  and denominator order  $\hat{q}$ , where  $\hat{p} < \hat{q}$  so that the inversion of  $\text{PA}_{[\hat{p}, \hat{q}]}(s)$  leads to a valid probability density function (see Amindavar and Ritcey (1994)).

**Lemma 8.** *Let  $f_1(t)$  and  $f_2(t)$  be the inverse Laplace transforms of  $F_1(s)$  and  $F_2(s)$ , respectively, and let  $c_1$  and  $c_2$  be constant. The linearity property of inverse transform states*

$$\begin{aligned} \text{L}^{-1}\{c_1 F_1(s) + c_2 F_2(s)\} &= \text{L}^{-1}[c_1 F_1(s)] + \text{L}^{-1}[c_2 F_2(s)] \\ &= c_1 f_1(t) + c_2 f_2(t) \end{aligned}$$

Since  $\text{PA}_{[\hat{p}, \hat{q}]}(s)$  is a rational function with the order of its numerator less than denominator, the exact inversion of  $\text{PA}_{[\hat{p}, \hat{q}]}(s)$  will always lead to a PDF in the form of sum of exponential function. By analogy with equation 5.2 and 5.3, we can describe the limiting behavior of  $\hat{f}(t)$  and  $\hat{P}(T > t)$  in terms of exponential function.

$$\hat{f}(t) \approx c^* e^{-a^* t} \quad \text{as } t \rightarrow \infty \quad (5.18)$$

$$\hat{P}(T > t) \approx v^* e^{-a^* t} \quad \text{as } t \rightarrow \infty \quad (5.19)$$

where  $a^*$  is the absolute value of the right-most singularities of  $\text{PA}_{[\hat{p},\hat{q}](-s)}$ ,  $c^*$  can be determined by  $c^* = \lim_{s \rightarrow 0} s \text{PA}_{[\hat{p},\hat{q}]}(a - s)$ , and  $v^* = \frac{c^*}{a^*}$ .

To quantify the error in tail area approximation by exponential functions that derived from different order of Padé approximations, we consider examining their relative error in term of asymptotic constant and asymptotic decay rate

$$\begin{aligned} \eta(t) &= \frac{P(T > t) - \hat{P}(T > t)}{P(T > t)} \\ &= \frac{ve^{-at} - v^*e^{-a^*t}}{ve^{-at}} \\ &= 1 - \frac{v^*}{v}e^{(a-a^*)t} \end{aligned}$$

where  $v^* = \frac{c^*}{a^*}$ , and  $v = \frac{c}{a}$ . We then have

$$v^* \rightarrow v \quad \text{as } c^* \rightarrow c, a^* \rightarrow a \quad (5.20)$$

and

$$e^{(a-a^*)t} \rightarrow 1 \quad \text{as } a^* \rightarrow a \quad (5.21)$$

Then equation 5.20 and 5.21 imply that

$$\eta(t) \rightarrow 0 \quad \text{as } c^* \rightarrow c, a^* \rightarrow a$$

Figure 5.12 depicts the plot of relative error in approximating survival probabilities at  $t \in [0, 20]$  by 5 different order of Padé approximations. Apparently, the exponential functions constructed from  $\text{PA}_{[1,2]}$  and  $\text{PA}_{[1,3]}$  have smaller relative error among these functions in time interval  $[0, 16]$ . However, Figure 5.13 reveals evidence that  $\text{PA}_{[2,3]}$ ,  $\text{PA}_{[2,4]}$  and  $\text{PA}_{[2,4]}$  perform better for large  $t$ , particularly the exponential approximation of survival function obtained by the inversion of  $\text{PA}_{[2,3]}$  has the smallest relative error for  $t \in [16.32, 20]$ . Thus, it appears that the choice of the order of Padé approximation of MGF depends on the size of survival probability that we want to estimate, the smaller survival probability (i.e.  $t$  is large) the higher order of Padé approximations is required to obtain accurate estimation, which is consistent with the results given by Table 5.2 in Section 5.1.1.

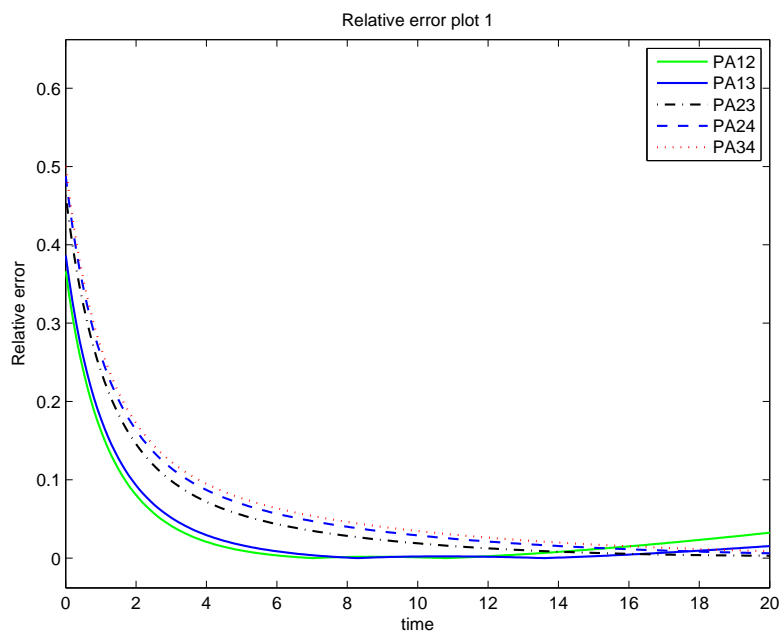
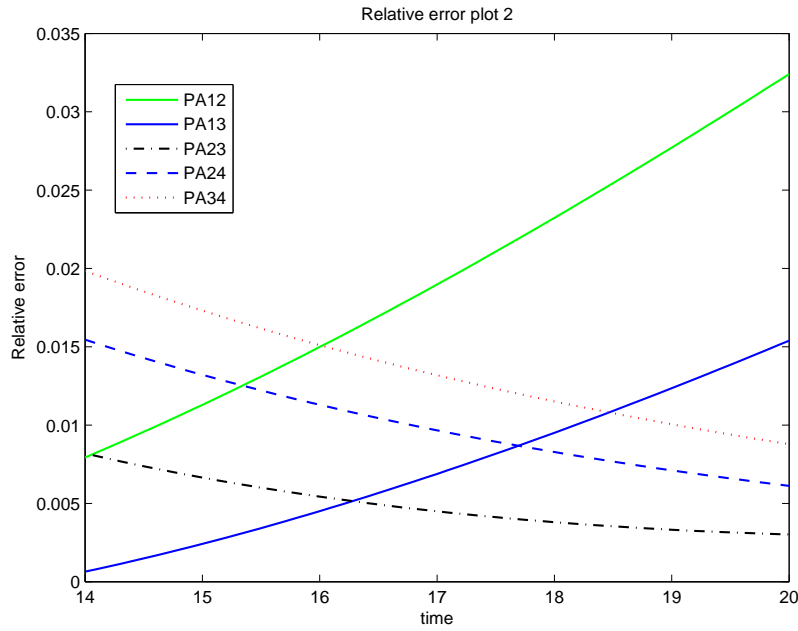


Figure 5.12: Plot of relative error in  $P(T > t)$  against  $t \in [0, 20]$

We now consider describing the behavior of error in tail area approximation. Let  $L[\varepsilon_{[\hat{p}, \hat{q}] }](s) = \text{PA}_{[\hat{p}, \hat{q}]}(s) - M_T(s)$  be the difference between the original MGF and its Padé approximation with numerator order  $\hat{p}$  and denominator order  $\hat{q}$ . By Lemma 8, the inverse Laplace transform of  $L[\varepsilon_{[\hat{p}, \hat{q}] }](s)$ ,  $\varepsilon_{[\hat{p}, \hat{q}]}(t)$ , can be expressed as

$$\begin{aligned} \varepsilon_{[\hat{p}, \hat{q}]}(t) &= L^{-1}\left\{\text{PA}_{[\hat{p}, \hat{q}]}(s) - M(s)\right\} \\ &= \hat{f}(t) - f(t) \end{aligned} \tag{5.22}$$

Define  $\delta(t) = \hat{P}(T > t) - P(T > t)$ . To capture the behavior of error in estimating tail area probabilities  $\delta(t)$  for large  $t$ , we substitute equation 5.2 and 5.18 in equation 5.22, then


 Figure 5.13: Plot relative error in  $P(T > t)$  against  $t \in [14, 20]$ 

$$\begin{aligned}
 \delta(t) &= \hat{P}(T > t) - P(T > t) \\
 &= \int_t^\infty \hat{f}(u) du - \int_t^\infty f(u) du \\
 &\approx \int_t^\infty (c^* e^{-a^* u} - c e^{-a u}) du \\
 &\approx \frac{c^*}{a^*} e^{-a^* t} - \frac{c}{a} e^{-a t} \\
 &\approx v^* e^{-a^* t} - v e^{-a t} \quad \text{as } t \rightarrow \infty
 \end{aligned}$$

where  $v^* = \frac{c^*}{a^*}$ , and  $v = \frac{c}{a}$ . Hence the error in tail area approximation  $\delta(t)$  can be approximated by a closed form expression, denoted by  $\hat{\delta}(t)$ , which consists of two exponential function.

$$\begin{aligned}
 \delta(t) &\approx \hat{\delta}(t) \\
 &= v^* e^{-a^* t} - v e^{-a t} \quad \text{as } t \rightarrow \infty
 \end{aligned} \tag{5.23}$$

In the situation when the true survival function does not have closed form, it becomes difficult to directly identify the dominating exponential function. The motivation to construct 5.23 is to analyse the error of tail area approximation  $\delta(t)$  based on the singularities of MGF and its Padé approximation. The simple form of 5.23 enables us to easily obtain a closed form estimation for  $\delta(t)$ , even the closed forms of original tail distribution is not available. The calculation in 5.23 is straightforward, because it only requires to find the singularities of MGF and Padé approximation, and the asymptotic constant that can be determined by Proposition 1.

### 5.2.1 Example 8

We return to the reversible illness-death system example in Section 5.1.1, and investigate the behavior of error in tail area approximation based on the Padé approximations for MGF given by equation 5.4 on page 158.

$$\begin{aligned}
 M_{T_{02}}(-s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\
 &= \frac{(2+s)(1.8+3.4s+s^2)}{(6+12s+4s^2)(0.5+s)(1.2+s)} \\
 &= \frac{1 + 2.3889s + 1.5s^2 + 0.27778s^3}{1 + 4.8333s + 8s^2 + 5.2222s^3 + 1.1111s^4}
 \end{aligned}$$

Having found the Padé approximation of  $M_{T_{02}}(-s)$  subjected to different orders, which are presented in equation 5.6, 5.7, 5.8, 5.9, and 5.10, we can obtain exponential functions to approximate the estimated density function  $\hat{f}(t)$ , and subsequently derive the function in the form of equation 5.19 to describe the limiting behavior of survival function  $\hat{P}(T > t)$ . In each case, we apply 5.23 to compute the estimated error  $\hat{\delta}(t)$ , and compare it with the value of true error  $\delta(t)$  by plotting them in the same graph.



As can be observed in Figure 5.14, 5.15, 5.16, 5.17 and 5.18, all the estimated error are gradually converging to the true error in the approximation of survival function as  $t$  increases, and there is clearly a good match between  $\hat{\delta}(t)$  and  $\delta(t)$  for  $t > 10$ . These results indicate that the the form of function in equation 5.23 is effective and provides a useful frame of reference to help us understand the behavior of error in the approximation of tail area probabilities.

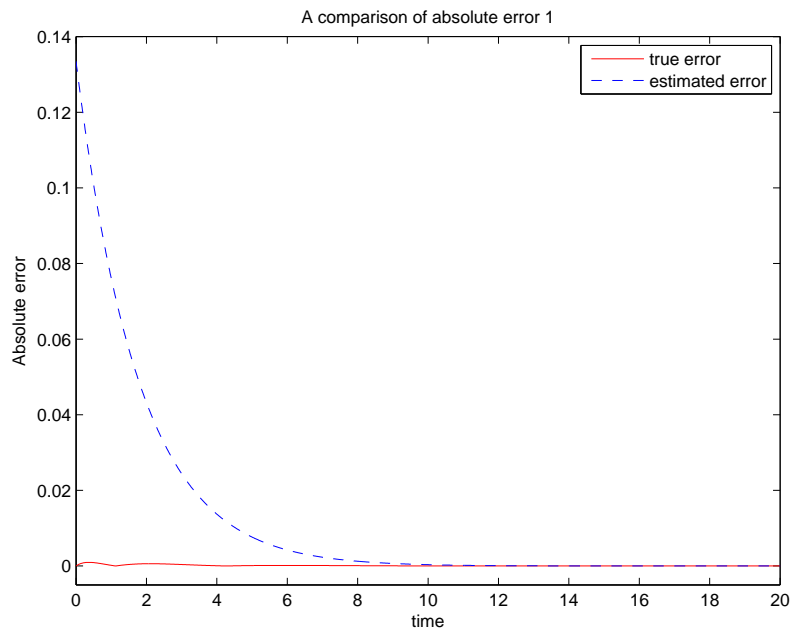


Figure 5.14: Estimated error based on  $PA_{[1,2]}(s)$

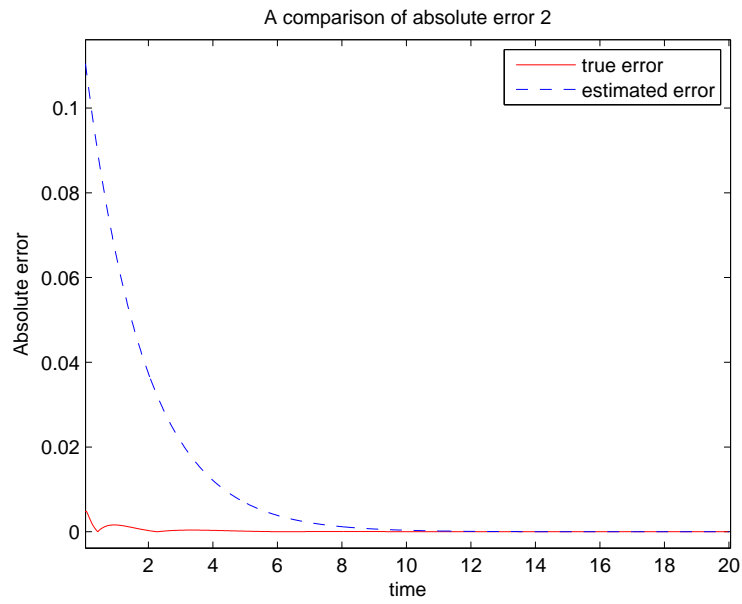


Figure 5.15: Estimated error based on  $PA_{[1,3]}(s)$

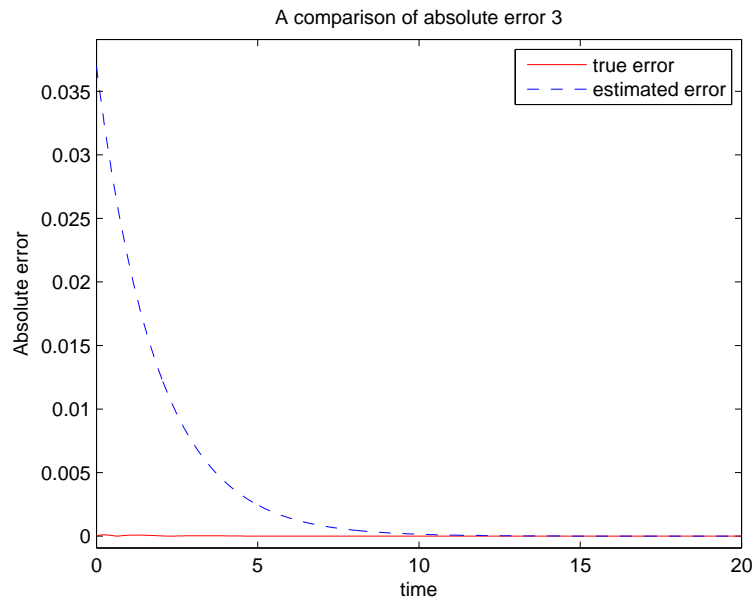


Figure 5.16: Estimated error based on  $PA_{[2,3]}(s)$

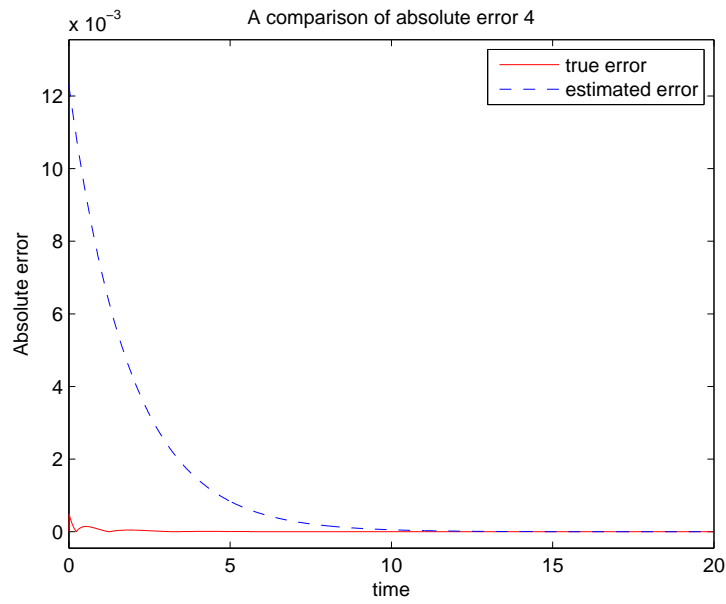


Figure 5.17: Estimated error based on  $PA_{[2,4]}(s)$

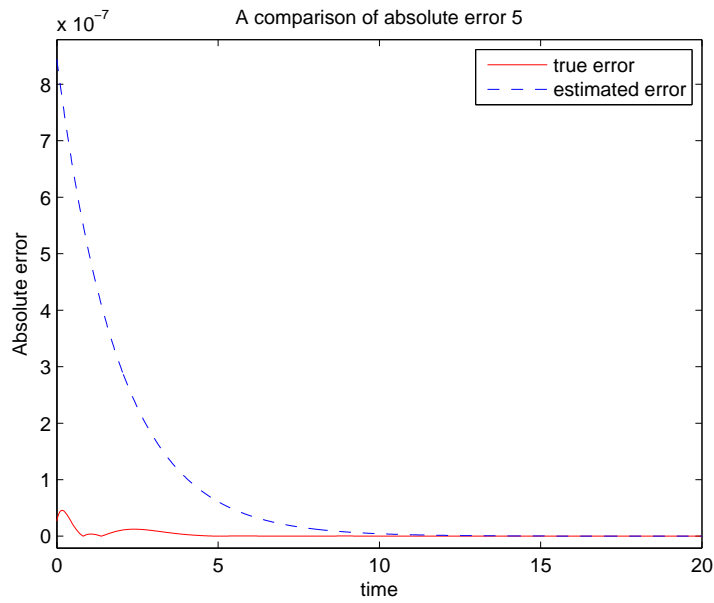


Figure 5.18: Estimated error based on  $PA_{[3,4]}(s)$

### 5.3 Saddlepoint approximation for survival function

Huzurbazar (1999) shows evidence that the saddlepoint method can provide extremely accurate approximation for tail probabilities, and the accuracy holds even in the small sample case. Traditionally, the saddlepoint method is often applied for statistical inference in small sample size problems by approximating  $p$ -value when the exact distribution is intractable. Further examples that illustrate the accuracy of the saddlepoint approximation can be found in Huzurbazar. S (1999, Section 2.1). In probability theory, the saddlepoint method is more generally known as the large deviation theory (see Hall(1992), Dembo and Zeitouni (1998)), which concerns the asymptotic behaviour of the tails probabilities of extreme events. More recently, the saddlepoint approximation is considered as a suitable technique in the context of portfolio credit loss. Glasserman (2004) applies this method to estimate the tail of the distribution of large portfolio losses, while Yang, Hurd and Zhang (2006) shows that Saddlepoint approximation has superior performance for CDO (collateralized debt obligations) pricing problem. As we are usually interested in determining the distribution of total waiting time between input node and output node in flowgraph model, but not the mean of total waiting time, in this section, we specifically focus on the implementation of saddlepoint approximation for the survival function of single random variable based on the Lugannani-Rice formula 3.75 ( i.e.  $n = 1$  case).

Let  $T$  be the random variable that represents the total waiting time, and let  $t_{max}$  be the largest value at which the survival function is estimated. Suppose there are totally  $\ell$  data points in interval  $[0, t_{max}]$  for approximation, that is, we want to calculate the estimated value of  $P(T > t_i)$  for  $t_i \in [0, t_{max}]$ ,  $i = 1 \dots \ell$ . In order to apply the Lugannani-Rice formula, we need to first establish the expression of first and second derivatives of cumulant generating function (CGF)  $K_T(s)$  with respect to  $s$ . For each given value  $t_i$  in interval  $[0, t_{max}]$ ,  $i = 1 \dots \ell$ , the approximation for  $P(T > t)$  is computed by the following procedures.

1. Calculate  $\hat{s}_i$  by solving equation  $K'_T(s_i) = t_i$  for  $s_i$ .

2. Given  $\hat{s}_i$  at each value of  $t_i$ , evaluate the value of  $K_T(\hat{s}_i)$  and  $K_T''(\hat{s}_i)$ .
3. Substitute  $K_T''(\hat{s}_i)$  and  $\hat{s}_i$  in  $\hat{s}\sqrt{K_T''(\hat{s})}$  for  $z_1$ , and  $t_i, \hat{s}_i, K_T(\hat{s}_i)$  in  $\sqrt{2(\hat{s}t - K_T(\hat{s}))}$  for  $z_2$  (see Theorem 5, page 77).
4. Compute the estimated value of  $P(T > t_i)$  by substituting the value of  $z_1$  and  $z_2$  in equation 3.75 accordingly.

For most problems involving complicated flograph,  $\hat{s}$  can be obtained by using numerical method to solve  $K_T'(s) = t$  for  $s$ , while in simple flowgarph case, it can be explicitly determined in closed form. We will discuss further on the method to simplify the calculation of  $\hat{s}$  in Section 5.5.

### 5.3.1 Example 1.3.3

We now demonstrate the Lugannani-Rice formula for survival function approximations in Section 5.1.1. The algebraic expressions of the MGF given by equation 5.4 is

$$M_{T_{02}}(s) = \frac{(\lambda_4 - s)(3\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2s - 2\lambda_3\lambda_1s - 2\lambda_3\lambda_2s + 2\lambda_3s^2)}{(3\lambda_1\lambda_4 - 4\lambda_1s - 4\lambda_4s + 4s^4)(\lambda_3 - s)(\lambda_2 - s)}$$

The CGF,  $K_{T_{02}}(s) = \log(M_{T_{02}}(s))$ , can be easily calculated by using symbolic algebra package such as MAPLE.

$$\begin{aligned} K_{T_{02}}(s) &= \log(3\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2s - 2\lambda_3\lambda_1s - 2\lambda_3\lambda_2s + 2\lambda_3s^2) \\ &\quad \log(\lambda_4 - s) - \log(3\lambda_1\lambda_4 - 4\lambda_1s - 4\lambda_4s + 4s^4) \\ &\quad - \log(\lambda_3 - s) - \log(\lambda_2 - s) \end{aligned} \tag{5.24}$$

Then

$$\begin{aligned} K'_{T_{02}}(s) &= \frac{4\lambda_3s - \lambda_1\lambda_2 - 2\lambda_3\lambda_1 - 2\lambda_3\lambda_2}{3\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2s - 2\lambda_3\lambda_1s - 2\lambda_3\lambda_2s + 2\lambda_3s^2} + \frac{1}{s - \lambda_4} \\ &\quad + \frac{4\lambda_1 + 4\lambda_4 - 8s}{3\lambda_1\lambda_4 - 4\lambda_1s - 4\lambda_4s + 4s^2} - \frac{1}{s - \lambda_2} - \frac{1}{s - \lambda_3} \end{aligned}$$

$$\begin{aligned}
 K''_{T_{02}}(s) = & \frac{1}{(s - \lambda_2)^2} + \frac{1}{(s - \lambda_3)^2} + \frac{8}{3\lambda_1\lambda_4 - 4\lambda_1s - 4\lambda_4s + 4s^2} \\
 & - \frac{1}{(s - \lambda_4)^2} + \frac{16\lambda_4^2 - 16\lambda_1\lambda_4 + 16\lambda_1^2}{(3\lambda_1\lambda_4 - 4\lambda_1s - 4\lambda_4s + 4s^2)^2} \\
 & - \frac{4\lambda_3}{3\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2s - 2\lambda_3\lambda_1s - 2\lambda_3\lambda_2s + 2\lambda_3s^2} \\
 & + \frac{16\lambda_1\lambda_2\lambda_3^2 - 4\lambda_1^2\lambda_2\lambda_3 - 4\lambda_3^2\lambda_1^2 - \lambda_1^2\lambda_2^2 - 4\lambda_1\lambda_2^2\lambda_3 - 4\lambda_2^2\lambda_3^2}{(3\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2s - 2\lambda_3\lambda_1s - 2\lambda_3\lambda_2s + 2\lambda_3s^2)^2}
 \end{aligned}$$

By substituting parameter  $\lambda_1 = 1$ ,  $\lambda_2 = 1.2$ ,  $\lambda_3 = 0.5$ ,  $\lambda_4 = 2$ , and  $p_{01} = p_{02} = p_{10} = p_{12} = \frac{1}{2}$  into equation 5.25, and equate it to each  $t_i \in [0, t_{max}]$ ,  $i = 1 \dots \ell$ , and we solve

$$K'_{T_{02}}(s) = t_i \quad (5.25)$$

for  $\hat{s}_i$  at each given  $t_i$ . Next, we evaluate the terms in equation 5.24 and 5.25 at  $\hat{s}$ , and follow the step 3 of our procedure to compute  $z_1, z_2$  respectively. Once the explicit values for the required terms in Lugannani-Rice formula 3.86 are obtained, the approximation for the survival function  $P(T > t)$  can then be easily computed. For illustration, we consider applying Lugannani-Rice saddlepoint approximation to estimate  $P(T > t)$  in time interval  $[0, 20]$ .

By using the  $-\log$  transform, Figure 5.19 presents the plot of true survival function along with its saddlepoint approximation by Lugannani-Rice formula for the survival function of total waiting time  $T_{02}$ . In general, we can see that the Lugannani-Rice saddlepoint approximation is extremely close to the true survival function, and there is virtually no graphical difference between them in the whole range of  $t$ . To further investigate how the error in survival function estimation is distributed, Figure 5.20 illustrates the plot of relative error for  $t \in [0, 20]$ . It clearly reveals that most of error occurs in estimating  $P(T > t)$  when  $t < 5.27$ , that is, up to 90% percentile, where the relative error rises steadily until it peaks at 3.34% when  $t = 1.34$ , then it drops sharply as  $t$  increases. All of these plots indicates that the saddlepoint approximations using the Lugannani-Rice formula are highly suited for estimating the tail area probabilities of  $P(T > t)$  for large  $t$ .

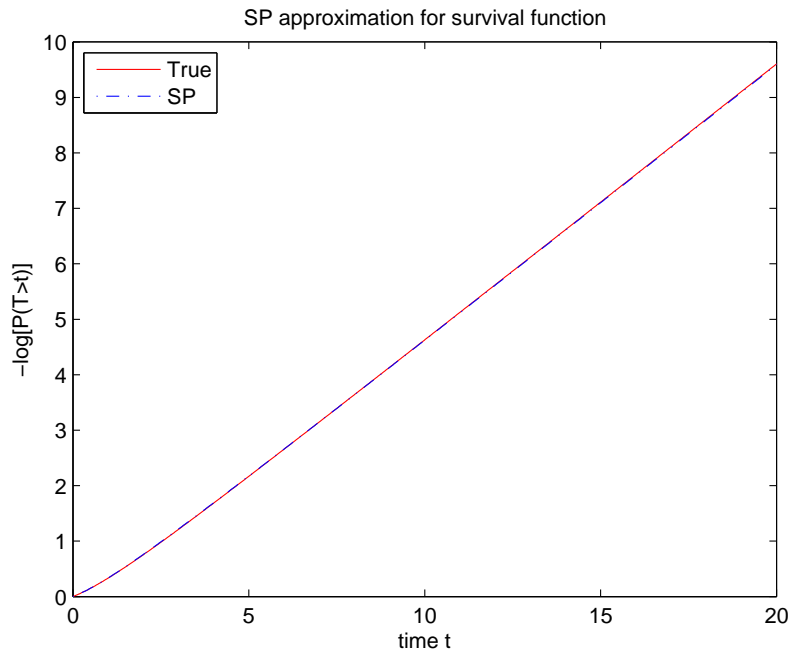


Figure 5.19: Survival function approximation by Lugannani-Rice formula

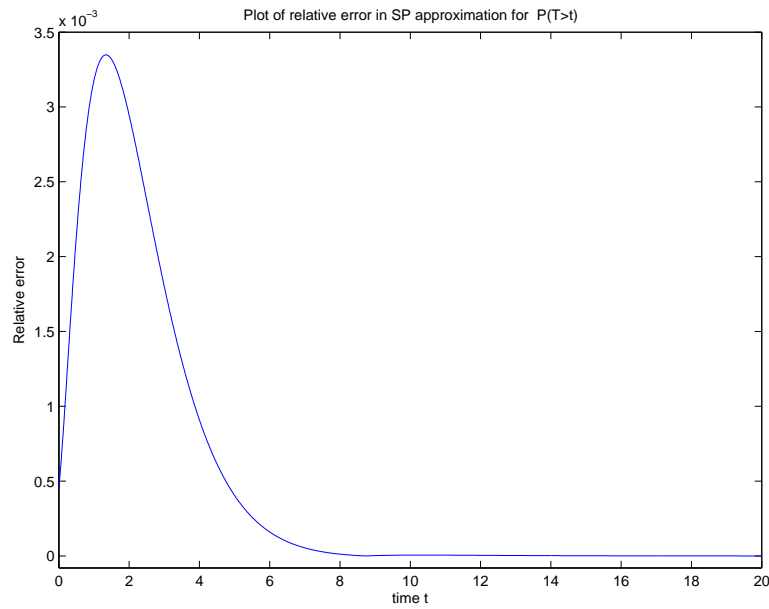


Figure 5.20: Relative error in estimating  $P(T > t)$  by Lugannani-Rice formula

## 5.4 Padé-type Saddlepoint approximation for survival function

For flowgraphs consists of large number of nodes and several feedback loops, such as Figure 2.5, Chapter 2, the resulting MGF of total waiting time is often a complicated rational function, leads to even more complicated expression for CGF  $K_T(s)$ , as well as its derivative  $K_T'(s)$  and  $K_T''(s)$ , which makes subsequent calculation very computationally demanding. The challenge is effectively to solve equation  $K_T'(s) = t$  for  $\hat{s}$  when the first derivative of CGF,  $K'(s)$ , is in complicated form, and numerical method is often required to determine the value of  $\hat{s}$ .

In this section, we introduce an alternative approach to apply saddlepoint approximation based on the Padé approximation of MGF instead of original MGF, that is, replace the “based line MGF”  $M_T(s)$  with its Padé approximation  $PA_{[p,q]}(s)$  in Definition 7 and continue with the procedures we outlined in Section 5.3 to estimate survival function. Next, we reconsider the reversible illness-death system of Example 1.3.2, and present a comparison of Lugannani-Rice saddlepoint approximation based on the original MGF given in equation 5.4, its 5 different order of Padé approximations, as well as the true survival function given by equation 5.5. For each Padé approximations given in equations 5.6, 5.7, 5.8, 5.9, 5.10, we follow the procedures in Section 5.3 to compute the exponential function approximation of survival function.

The resulting saddlepoint approximations for survival function are plotted in Figure 5.21. It is shown that the Padé-type saddlepoint approximations are really close to each other and there is no obvious difference between them over the entire range of  $t$ . By further comparing the plot of saddlepoint approximation for survival function nears the 50% percentile,  $t = 1.86$ , and 99.99% percentile,  $t = 19.21$ , in Figure 5.22 and 5.23 separately, we can observe that the approximation based on high order of Padé approximation tend to have a smaller deviation to the approximation based on the original MGF than the low order one, particularly, the plots of approximation based on Padé approximation  $PA_{[2,3]}(s)$ ,  $PA_{[2,4]}(s)$ , and  $PA_{[3,4]}(s)$  follow closer to the plot of approximation based MGF  $M_T(s)$ .



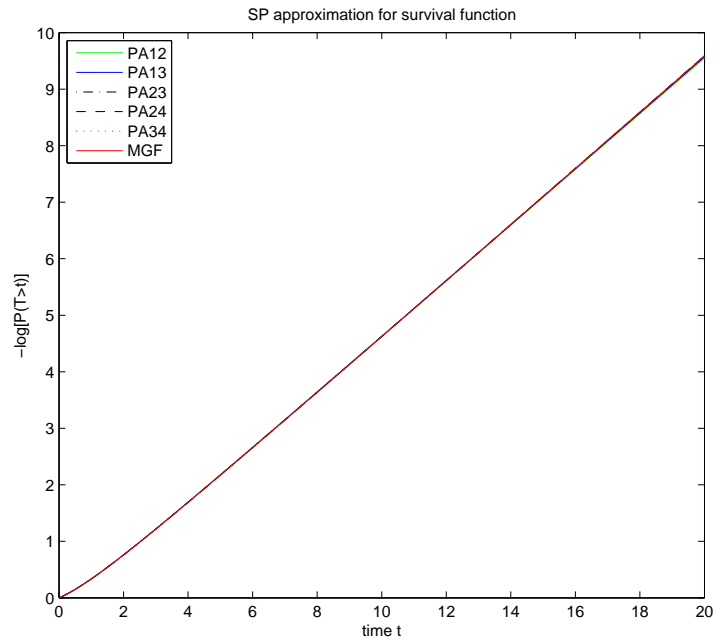


Figure 5.21: Plot of  $-\log\hat{P}((T > t))$  against  $t \in [0, 20]$ .

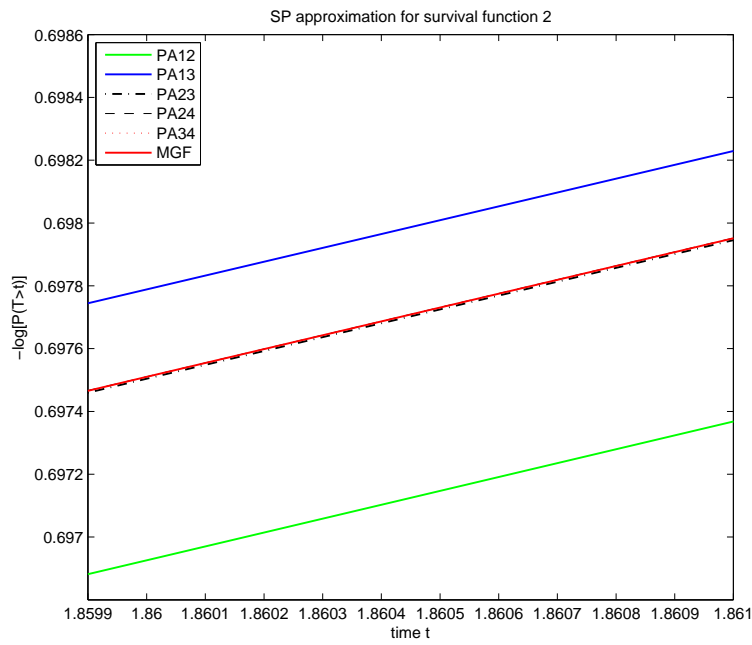


Figure 5.22: Plot of  $-\log\hat{P}((T > t))$  against  $t \in [1.86, 1.861]$ .

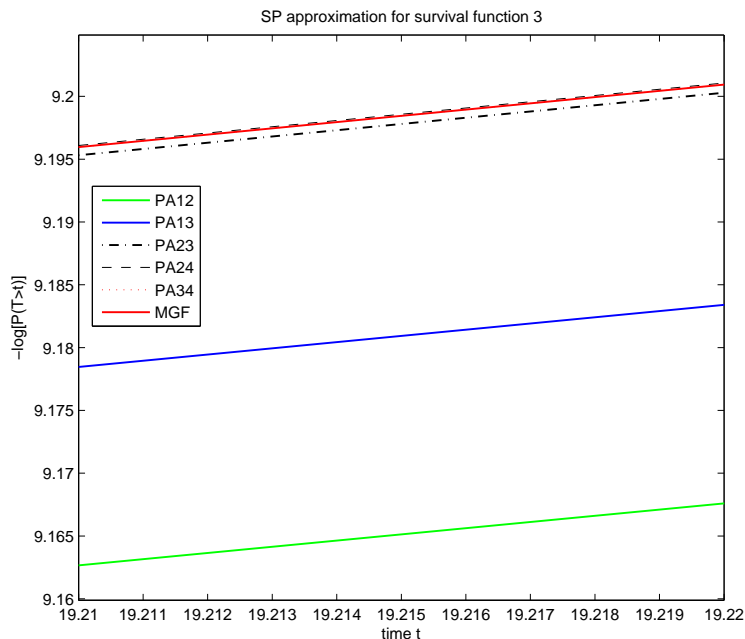


Figure 5.23: Plot of  $-\log\hat{P}((T > t))$  against  $t \in [19.21, 19.22]$ .

In addition, Figure 5.24 plots the relative error in estimating the saddlepoint approximation of  $M_T(s)$  for  $t \in [14, 19.5]$  after applying Lugannani-Rice formula under different order of Padé approximations of MGF. Consistent with the result from Figure 5.22 and Figure 5.23, the closer the order of Padé approximation to the order of original MGF we use in saddlepoint approximation, the smaller relative error it will result. The upward trend in both the relative error plot of  $SP_{PA_{[1,2]}}(t)$  and  $SP_{PA_{[1,3]}}(t)$  implies that high order of Padé approximations should be chosen as baseline line function, so that we could have an accurate estimation for the saddlepoint approximation of original MGF at large  $t$ .

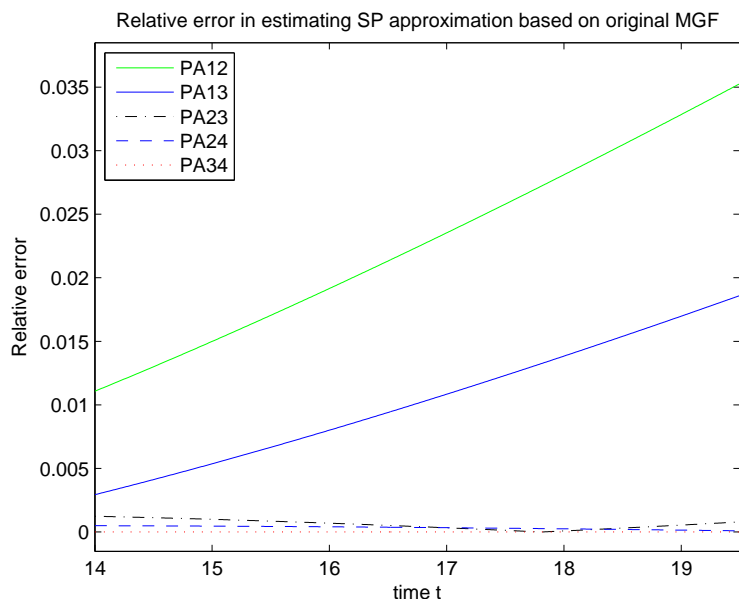


Figure 5.24: Comparison of relative error in estimating  $SP_{M(s)}$ .

Table 5.6, summarise the value of Lugannani-Rice saddlepoint approximations for  $P(T > t)$  at different percentile  $t_p$ , and makes comparisons with the true survival probability. Again, it becomes clear that higher order of Padé approximation gives relatively more accurate estimation, particularly for the survival probability at large  $t$ .

After computing the saddlepoint approximation under a range of order of Padé approximations of MGF for  $t \in [0, 20]$ , Table 5.7 presents the sum of absolute error in estimating the result of saddlepoint approximation of survival function-based on the original MGF. Evidently, there is a decreasing trend in the sum of absolute error as the order of Padé approximations get closer to the order of original MGF. By contrast, the estimation based on  $PA_{[3,4]}(s)$ , among the others, has the smallest sum of absolute error  $1.3498 \times 10^{-6}$ , while the Padé approximation with numerator order 1 and denominator 2 has the biggest sum of absolute error 0.3041 in the same time interval.

Table 5.6: Summary of Lugannani-Rice saddlepoint approximation for survival function  $P(T > t)$ : The unite of survival probability at time  $t^*$   $10^{-3}$ ,  $t^{**}$  is  $10^{-4}$

Time	PA <sub>[1,2]</sub>	PA <sub>[1,3]</sub>	PA <sub>[2,3]</sub>	PA <sub>[2,4]</sub>	PA <sub>[3,4]</sub>	$M_T(s)$	True
1.86	0.498114	0.497685	0.497826	0.497822	0.497823	0.4978234	0.5
3.37	0.248220	0.248498	0.248638	0.248642	0.248643	0.2486428	0.25
5.28	0.099487	0.099648	0.099935	0.099963	0.099969	0.0999699	0.1
6.69	0.049775	0.049818	0.050012	0.050041	0.050050	0.0500502	0.05
9.95	0.012419	0.012394	0.012431	0.012444	0.012449	0.0124485	0.01
14.59*	1.023370	1.014268	1.008766	1.009399	1.009882	1.0098819	1
19.21**	1.048834	1.032398	1.015143	1.014358	1.014485	1.0144851	1

Table 5.7: Sum of absolute error in estimating the saddlepoint approximation of survival function based on MGF for  $t \in [0, 20]$

PA <sub>[1,2]</sub>	PA <sub>[1,3]</sub>	PA <sub>[2,3]</sub>	PA <sub>[2,4]</sub>	PA <sub>[3,4]</sub>
0.3041	0.0284	0.0268	0.0143	$1.3498 \times 10^{-6}$

To conclude, all of the above numerical and graphical results suggest that the accuracy of the Lugannani-Rice saddlepoint approximation for survival function crucially depends on the structure of “based line MGF”, it is therefore more convenient to use Padé approximation that both the order of both numerator and denominator is similar to the original MGF.

## 5.5 Bias correction for the Padé-type Saddlepoint approximation

As the saddlepoint method based on the Padé approximation of MGF would usually generates error, and the the magnitude of error depends on whether the baseline Padé approximation has similar structure to the original MGF, that is, the result of saddlepoint approximation would be more accurate if both the numerator and denominator order of baseline Padé approximation is close to the MGF of random variable that we are interested. This raises our interest of how to effectively characterise the error of in such cases, and investigate the order of error in the Padé-type saddlepoint approximation.

In this section, we examine the case of applying saddlepoint approximation with Padé approximation as baseline function for the estimation of both PDF and survival function. We show that the behavior of error between saddlepoint approximation with original MGF and its Padé approximation can be described by the error between saddlepoint approximation constructed with their corresponding dominated exponential function. Next, we propose a biased corrected method for the Padé-type saddlepoint approximation for both PDF and tail distribution estimation. Numerical examples are also presented in the end to illustrate our method.

Let  $T$  be a positive random variable with MGF  $M_T(s)$ . The derivation of our method use the fact that the probability density function of  $T$ ,  $f(t)$ , can be approximated by an exponential function as follow

$$f(t) \approx ce^{-at} \quad \text{as } t \rightarrow \infty \quad (5.26)$$

where  $a$  is the absolute value of the right-most singularity of  $M_T(-s)$ , and  $c$  is a constant that can be obtained by Proposition 1 (see 5.11 on page 168).

The Laplace transform of this dominated exponential function is

$$\begin{aligned} L[ce^{-at}](s) &= cL[e^{-at}](s) \\ &= \frac{c}{a+s} \end{aligned}$$

Since the MGF is just the Laplace transform with argument  $-s$  instead of  $s$ , so the corresponding baseline function in saddlepoint approximation for estimating function  $ce^{-at}$  is

$$\hat{M}(s) = \frac{c}{a-s} \quad (5.27)$$

Suppose we estimate functions on both side of equation 5.26 by the saddlepoint approximation for density estimation (see Theorem 4, Section 3.2.3, Chapter 3), then

$$SP_M(t) \approx SP_{\hat{M}}(t) \quad \text{as } t \rightarrow \infty \quad (5.28)$$

where  $SP_M(t)$  is the saddlepoint approximation for  $f(t)$  based on the MGF of  $T$ ,  $M_T(s)$  and  $SP_{\hat{M}}(t)$  is the saddlepoint approximation for the dominated exponential function with baseline function in the form of equation 5.27. On the other hand, let  $\hat{f}(t)$  be the density function obtained by the exact inversion of Padé approximation of MGF. As it is shown in 5.18 on page 174, we have

$$\hat{f}(t) \approx c^* e^{-a^* t} \quad \text{as } t \rightarrow \infty \quad (5.29)$$

Analogy to equation 5.28, we apply saddlepoint approximation to both side of equation 5.29, then

$$SP_{pa}(t) \approx SP_{\hat{pa}}(t) \quad \text{as } t \rightarrow \infty \quad (5.30)$$

where the saddlepoint approximation for the dominated exponential function of  $\hat{f}(t)$  is  $SP_{\hat{pa}}(t)$ , with baseline function

$$\hat{pa}(s) = \frac{c^*}{a^* - s}$$

By equation 5.28 and 5.30, the order of error between  $SP_M(t)$  and  $SP_{pa}(t)$  can be approximated as

$$SP_M(t) - SP_{pa}(t) = O(SP_{\hat{M}}(t) - SP_{\hat{pa}}(t)) \quad \text{as } t \rightarrow \infty \quad (5.31)$$

Similarly to the idea of deriving equation 5.23 in Section 5.5, equation 5.31 implies that, for large  $t$ , the behavior of error between the saddlepoint approximation with original MGF and its Padé approximation can be described by the saddlepoint approximation with their exponential dominated term. Hence, we propose a general formula to correct the bias in the saddlepoint approximation that having Padé approximation of MGF as baseline function.

**Theorem 10.** *Let  $T$  be a random variable with MGF,  $M_T(s)$  such that*

$$M_T(s) \approx \frac{c}{a-s} \quad \text{as } s \rightarrow 0$$

*where  $a$  is the absolute value of the right-most singularity of  $M_T(-s)$ , and*

$$c = \lim_{s \rightarrow 0} sM_T(a-s)$$

*Also let  $PA_{[p,q]}(s)$  be the Padé approximation of  $M_T(s)$  with numerator order  $p$  and denominator order  $q$  such that*

$$PA_{[p,q]}(s) \approx \frac{c^*}{a^*-s} \quad \text{as } s \rightarrow 0$$

*where  $a^*$  is the absolute value of the right-most singularity of  $PA_{[p,q]}(-s)$ , and*

$$c^* = \lim_{s \rightarrow 0} sPA_{[p,q]}(a-s)$$

*Denote the saddlepoint approximation for the PDF with baseline function (\*) by  $SP_*(t)$ , define  $\eta(t) = SP_M(t) - SP_{pa}(t)$  as the bias in Padé-type saddlepoint approximation, and  $\hat{\eta}(t) = SP_{\hat{M}}(t) - SP_{\hat{pa}}(t)$  be the estimated bias. Then for  $t \rightarrow \infty$ , the bias corrected Padé-type saddlepoint approximation for the PDF is*

$$SP_{pa^*}(t) \approx SP_{pa}(t) + c_1 \hat{\eta}(t) \quad \text{as } t \rightarrow \infty \quad (5.32)$$

*where  $c_1$  is a constant that satisfies*

$$c_1 = \lim_{t \rightarrow \infty} \frac{\eta(t)}{\hat{\eta}(t)}$$

**Corollary 2.** *Suppose we define the Lugannani-Rice formula with baseline function (\*) by  $\text{LR}_*(t)$ , then for  $t \rightarrow \infty$ , the bias corrected Lugannani-Rice Padé-type saddlepoint approximation for survival function based on  $\text{PA}_{[p,q]}(s)$  can be expressed as*

$$\text{LR}_{\text{pa}^*}(t) \approx \text{LR}_{\text{pa}}(t) + c_2 \hat{\omega}(t) \quad \text{as } t \rightarrow \infty$$

and  $c_2$  is a constant that satisfies

$$c_2 = \lim_{t \rightarrow \infty} \frac{\omega(t)}{\hat{\omega}(t)}$$

where  $\omega(t) = \text{LR}_{\text{M}}(t) - \text{LR}_{\text{pa}}(t)$ ,  $\hat{\omega}(t) = \text{LR}_{\hat{\text{M}}}(t) - \text{LR}_{\hat{\text{pa}}}(t)$ .

As we have shown in Example 1.3.3, the drawback of saddlepoint approximation is the difficulty in effectively solving  $K'_T(s) = t$  for the saddlepoint  $\hat{s}$ , particularly when the MGF of total waiting time becomes very complicated in a large flowgraph that has feedback loops, which constantly occurs in practice. In that case, the step of determining  $\hat{s}$  often involves with high computational costs since the solution of  $K'_T(s) = t$  usually doesn't have closed form expression, and numerical method such as Newton-Raphson method is required.

However, the calculation of saddlepoint approximation is much easier if our bias correction formula 5.32 is applied. Firstly, the structure of Padé approximation is relatively simpler than the original MGF, especially for a lower order Padé approximation that has simple CGF, which could significantly reduce the computational cost in the subsequent calculation of saddlepoint  $\hat{s}$ . Secondly, it is worth pointing out that we can always have a closed form solution of  $\hat{s}$  for computing saddlepoint approximation  $\text{SP}_{\hat{\text{M}}}(t)$  and  $\text{SP}_{\hat{\text{pa}}}(t)$ .



For example, given equation 5.27, we have

$$K(s) = \log(\hat{M}(s)) = \log\left(\frac{c}{a-s}\right)$$

and

$$K'(s) = \frac{dK(s)}{ds} = \frac{1}{a-s}$$

Then equation  $K'(s) = t$  become

$$\frac{1}{a-s} = t$$

which immediately gives the closed form solution of  $s$  as follow

$$\hat{s} = a - \frac{1}{t} \tag{5.33}$$

where  $a$  is the absolute value of the rightmost original MGF. Since  $\hat{p}a(s)$  is in the same form of  $\hat{M}(s)$ , we can also find a closed form solution of  $\hat{s}^*$  for saddlepoint approximation  $SP_{pa}(t)$  as

$$\hat{s}^* = a^* - \frac{1}{t} \tag{5.34}$$

where  $a^*$  is the absolute value of the rightmost singularity of Padé approximation  $PA_{[p,q]}(-s)$ . With our method, the calculation of saddlepoint is less computationally demanding because it is directly computed from the original data  $t$ , which avoids using numerical method. Despite there is a problem in equation 5.33 and 5.34 if  $t$  equals zeros, we can still apply them in practise because total waiting time will never occur to be zero in real life problem. In the following two sections we demonstrate our bias corrected method in the Padé-type approximation for both probability density function and survival function estimation.

### 5.5.1 Probability density function estimation

To illustrate our method, we apply saddlepoint approximation to estimate probability density function of total waiting from node 0 to node 2 in Section 5.1.1 on page 158. The MGF of total waiting time  $T_{02}$  is

$$\begin{aligned} M_{T_{02}}(s) &= \frac{p_{01}p_{12}m_{01}(s)m_{12}(s) + p_{02}m_{02}(s)}{1 - p_{01}p_{10}m_{01}(s)m_{10}(s)} \\ &= \frac{(2-s)(1.8 - 3.4s + s^2)}{(6 - 12s + 4s^2)(0.5 - s)(1.2 - s)} \end{aligned} \quad (5.35)$$

The true density function is

$$f(t) = 0.75e^{-0.5t} - 0.3636e^{-1.2t} - 0.1207e^{-0.6340t} - 0.0157e^{-2.366t}$$

Since the absolute value of the right-most singularity of  $M_{T_{02}}(-s)$  is 0.5, and the true density function  $f(t)$  can be approximated by  $0.75e^{-0.5t}$  for large  $t$ , which has the corresponding baseline function for saddlepoint approximation

$$\hat{M}(s) = \frac{0.75}{0.5 - s} \quad (5.36)$$

Given a sample size of 2000 total waiting time data in interval  $[0, 20]$ , the saddlepoint point  $\hat{s}$  can be easily obtained based upon 5.35 and 5.36, and we can compute the saddlepoint approximation,  $SP_M(t)$  and  $SP_{\hat{M}}(t)$ , for density function  $f(t)$  and its exponential approximation function  $0.75e^{-0.5t}$  respectively. On the other hand, as explained in Section 5.5, saddlepoint approximation with lower order of Padé approximation typically generates larger error than that of using higher order of Padé approximation. For the purpose of illustration, we choose the Padé approximation with numerator order 1 and denominator 2,  $PA_{[1,2]}(s)$  given by 5.6 on page 159, as the baseline function in saddlepoint approximation, so that it will allow us to better examine our formula 5.32 in a large error case.

$$PA_{[1,2]}(s) = \frac{1 + 0.4543s}{1 + 2.8988s + 1.771s^2}$$

which leads to a density as  $f_{PA_{[1,2]}}(t) = 0.6753e^{-0.4942t} - 0.4188e^{-1.1426t}$ .

Since

$$f_{PA_{[1,2]}}(t) \approx 0.6753e^{-0.4942t} \quad \text{as } t \rightarrow \infty$$

The Laplace transform of  $0.6753e^{-0.4942t}$  after replacing argument  $s$  with  $-s$  is

$$\hat{PA}_{[1,2]}(s) = \frac{0.6753}{0.4942 - s}$$

Similarly, the saddlepoint approximation for density  $f_{PA_{[1,2]}}(t)$  and function  $0.6753e^{-0.4942t}$  can be determined, denoted them separately as  $SP_{pa}(t)$  and  $SP_{\hat{pa}}(t)$ . Define  $\eta(t) = SP_M(t) - SP_{pa}(t)$ , and  $\hat{\eta}(t) = SP_{\hat{M}}(t) - SP_{\hat{pa}}(t)$ . In order to determine the constant term  $c_1$  in Theorem 10, we first plot  $\log(\eta(t)) - \log(\hat{\eta}(t))$  against  $t$  in Figure 5.25, which indicates a clear evidence of convergence in  $\log\left(\frac{\eta(t)}{\hat{\eta}(t)}\right)$  for large  $t$ . Further calculation shows that

$$\log\left(\frac{\eta(t)}{\hat{\eta}(t)}\right) \rightarrow 2.8367 \quad \text{as } t \rightarrow \infty \quad (5.37)$$

Taking exponential on both side of equation 5.37,

$$\frac{\eta(t)}{\hat{\eta}(t)} \rightarrow \exp(2.8367) = 17.0597$$

Hence

$$\eta(t) \approx 17.0597\hat{\eta}(t) \quad \text{as } t \rightarrow \infty \quad (5.38)$$

To verify our result, we need to compare the plot of  $\eta(t)$  with  $\hat{\eta}(t)$  and  $17.0597\hat{\eta}(t)$  respectively. Figure 5.26 presents both the plot of  $\eta(t)$  and  $\hat{\eta}(t)$  against  $t$  in interval  $[19, 20]$ , where there is a relatively big difference between the sp error and the estimated sp error. After multiplying 17.0597 to  $\hat{\eta}(t)$ , Figure 5.27 shows that the plot of  $17.0597\hat{\eta}(t)$  denoted by “Estimated Sp error after correction”, is close to the plot of “Sp error  $\eta(t)$ ” which provides a solid evidence that our calculation for constant  $c_1 = 17.0597$  is correct, and equation 5.38 holds.

Furthermore, the agreement between the error  $SP_M(t) - SP_{pa}(t)$  and the estimated error  $17.0597(SP_{\hat{M}}(t) - SP_{\hat{pa}}(t))$  suggests that we can improve the result of saddlepoint approximation that derived from Padé approximation of MGF by adding the estimated error as follow

$$SP_{pa^*}(t) \approx SP_{pa}(t) + 17.0597(SP_{\hat{M}}(t) - SP_{\hat{pa}}(t)) \quad (5.39)$$

as  $t \rightarrow \infty$ .

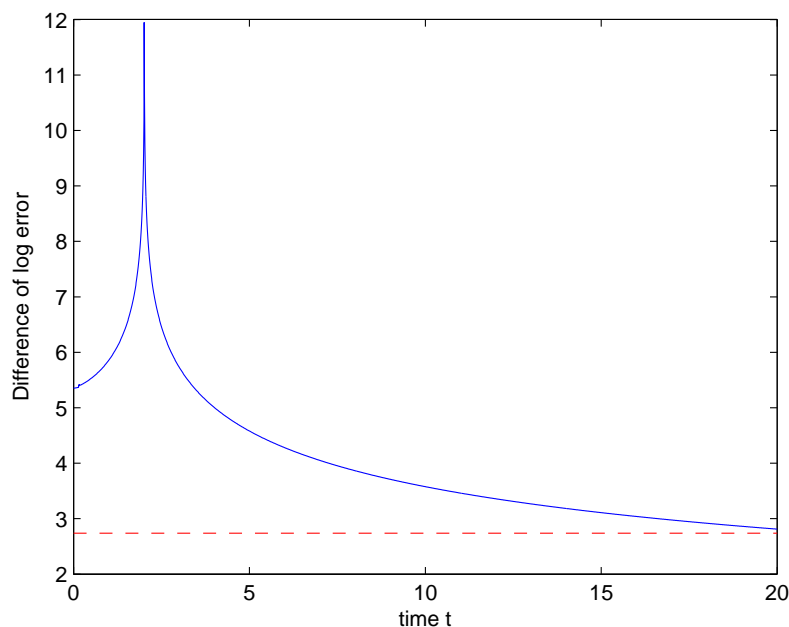


Figure 5.25: Plot of convergence test

Figure 5.28 shows the plot of PDF estimation by  $SP_{pa}(t)$  without correction and  $SP_M(t)$ , whereas Figure 5.29 shows the plot of  $SP_M(t)$  with  $SP_{pa^*}(t)$  that defined in formula 5.39 (denoted as “New SP-PADE12”). As can be seen in Figure 5.29, the bias corrected saddlepoint approximation for PDF based on  $PA_{[1,2]}(s)$  significantly improves the quality of density estimation, and there is a good match to the saddlepoint approximation based on original MGF  $M_{T_{02}}(s)$ , compared with previous result in Figure 5.28.

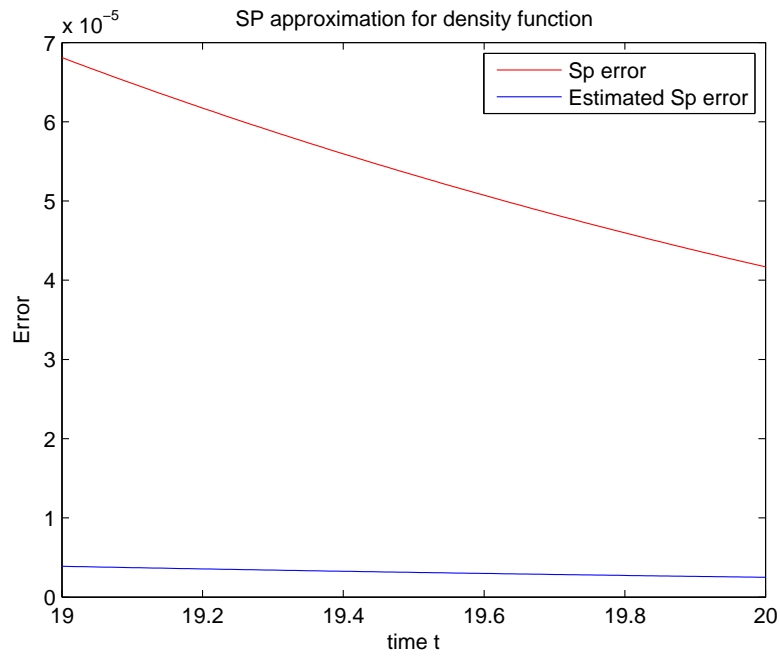


Figure 5.26: A comparison of error in SP approximation 1

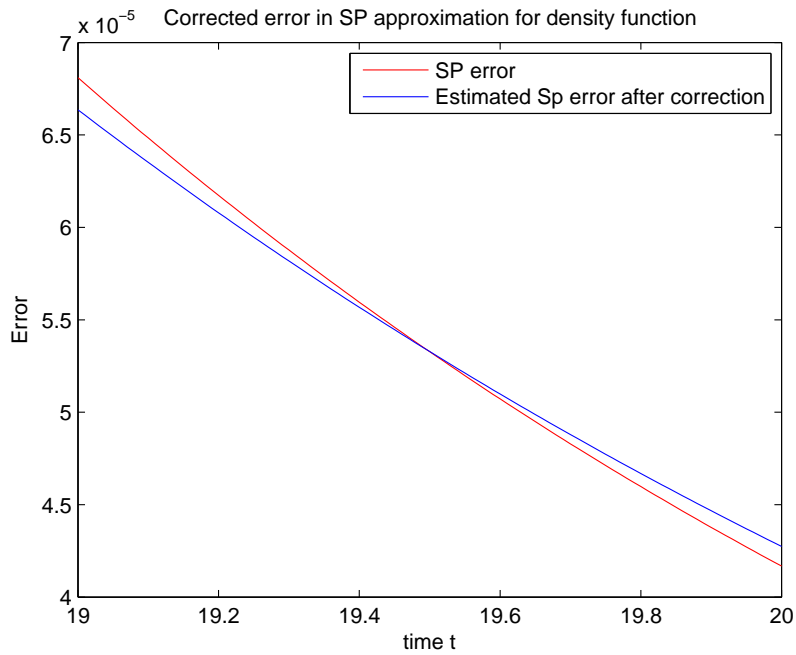


Figure 5.27: A comparison of error in SP approximation 2

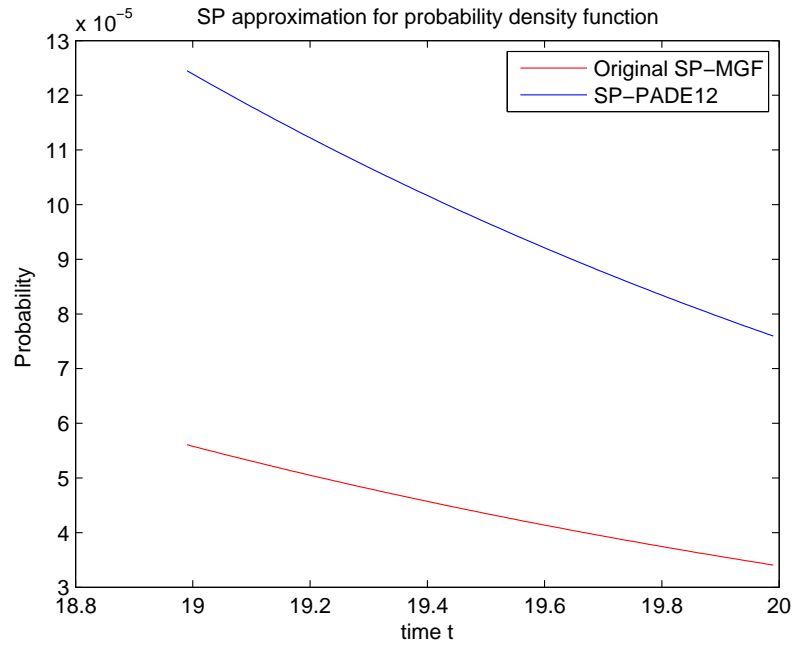


Figure 5.28: SP approximation for probability density function 1

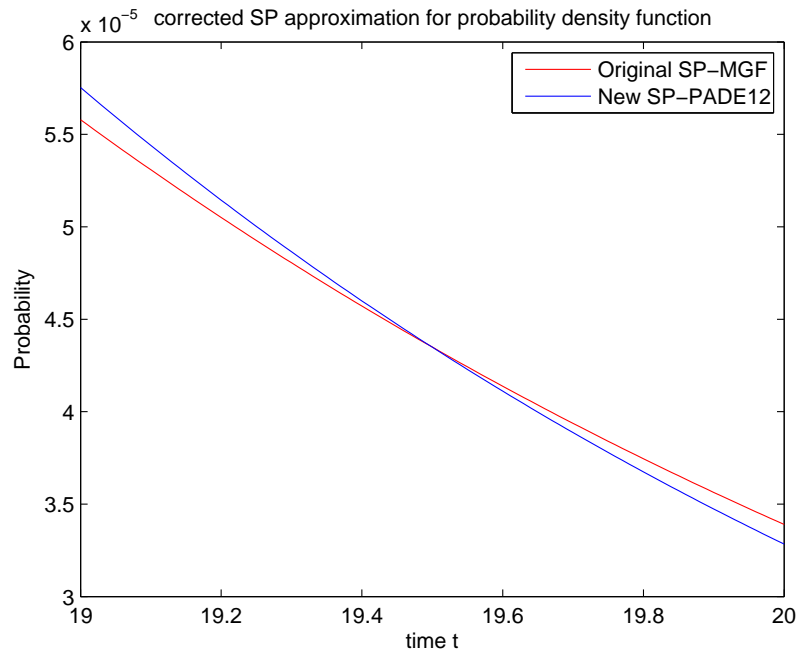


Figure 5.29: SP approximation for probability density function 2

Table 5.8: Saddlepoint approximation for probability density function estimation.

Time	3.37	5.27	6.69	9.9	14.59	19.21
$SP_{pa}(t)$	0.2803	0.1096	0.0543	0.0134	$10.95 \times 10^{-4}$	$11.168 \times 10^{-5}$
$SP^*_{pa}(t)$	0.2677	0.0979	0.0460	0.0102	$6.5535 \times 10^{-4}$	$5.1154 \times 10^{-5}$
$SP_M(t)$	0.1167	0.0484	0.0245	0.0061	$5.0020 \times 10^{-4}$	$5.0251 \times 10^{-5}$

Table 5.9: Absolute error in saddlepoint approximation for PDF.

Time	3.37	5.27	6.69	9.9	14.59	19.21
$SP_{pa}(t)$	0.1636	0.0612	0.0298	0.0073	$5.9480 \times 10^{-4}$	$6.1429 \times 10^{-5}$
$SP^*_{pa}(t)$	0.1510	0.0495	0.0215	0.0041	$1.5515 \times 10^{-4}$	$0.0903 \times 10^{-5}$

Table 5.8 summarise the explicit value of probability obtained from saddlepoint approximation based on  $PA_{[1,2]}(s)$ ,  $SP_{pa}(t)$ , original MGF  $M_{T_{02}}(s)$ ,  $SP_M(t)$ , and the biased corrected saddlepoint approximation given by Theorem 10,  $SP^*_{pa}(t)$ . By comparing to the value of  $SP_M(t)$ , we find that  $SP^*_{pa}(t)$  has substantially improved the accuracy after we corrected the bias in  $SP_{pa}(t)$ , and the bias corrected Padé-type saddlepoint approximation is better in the sense that it is close to the value obtained by saddlepoint approximation based on original MGF. Further evidence is given in Table 5.9, where it illustrates that the absolute error in density estimation decreases dramatically in  $SP^*_{pa}(t)$ , particularly the absolute error becomes smaller as  $t$  increases.

### 5.5.2 Survival function approximation

We now illustrate the performance of the bias corrected Padé-type saddlepoint approximation for survival function estimation in Section 5.1.1. For implementation, we consider estimating survival function  $P(T > t)$  in the time interval  $[0, 20]$ , where a sample size of 2000 total waiting time data is uniformly discretised with sub-interval 0.01. The Lugannani-Rice formula given in Theorem 5 is applied for the calculation of tail area probability of survival function. Recall equation 5.5 in Section 5.1.1, the true survival function of total waiting time from node 0 to node 2 is

$$P(T > t) = 1.5e^{-0.5t} - 0.303e^{-1.2t} + 0.1903e^{-0.6340t} + 0.0066e^{-2.366t}$$

By equation 5.3 in Section 5.1, we have

$$P(T > t) \approx 1.5e^{-0.5t} \quad \text{as } t \rightarrow \infty$$

which leads to

$$\hat{M}(s) = \frac{1.5}{0.5 - s}$$

The survival function derived from the Padé approximation  $PA_{[1,2]}(s)$  is

$$S_{PA_{[1,2]}}(t) = 1.3665e^{-0.4942t} - 0.3665e^{-1.1426t}$$

giving

$$S_{PA_{[1,2]}}(t) \approx 1.3665e^{-0.4942t} \quad \text{as } t \rightarrow \infty$$

then

$$\hat{P}A(s) = \frac{1.3665}{0.4942 - s}$$

The estimation of survival function from Lugannani-Rice formula with original MGF  $M_{T_{02}}(s)$ , Padé approximation  $PA_{[1,2]}(s)$ ,  $\hat{M}(s)$ , and  $\hat{P}A(s)$  are denoted by  $LR_M(t)$ ,  $LR_{pa}(t)$ ,  $LR_{\hat{M}}(t)$ ,  $LR_{\hat{pa}}(t)$  respectively.

Let  $\omega(t) = LR_M(t) - LR_{pa}(t)$ ,  $\hat{\omega}(t) = LR_{\hat{M}}(t) - LR_{\hat{pa}}(t)$ . We plot  $\log\left(\frac{\omega(t)}{\hat{\omega}(t)}\right)$  against  $t$  in Figure 5.30. Evidently, it converges to 0.2143 after as  $t$  increases, and



we find the corresponding constant  $c_2$  is  $\exp(0.2143) = 1.2389$ , such that

$$\omega(t) \approx 1.2389\hat{\omega}(t) \quad \text{as } t \rightarrow \infty$$

Having see the relatively large difference between the two type of errors in Figure 5.31, the plot given by Figure 5.32 shows that the estimated error in tail area probability,  $\hat{\omega}(t)$ , is now closely matching the true error  $\omega(t)$  after we multiply 1.2389 to  $\omega(t)$ . According to Corollary 2, the bias corrected Lugannani-Rice Padé-type saddlepoint approximation for tail area survival probability can be computed by

$$\text{LR}_{\text{pa}^*}(t) \approx \text{LR}_{\text{pa}}(t) + 1.2389 (\text{LR}_{\hat{M}}(t) - \text{LR}_{\hat{\text{pa}}}(t)) \quad (5.40)$$

as  $t \rightarrow \infty$ .

Next, for  $t \in [19, 20]$ , we compare the estimated tail area survival probabilities from the Lugannani-Rice Padé-type approximation  $\text{LR}_{\text{pa}}(t)$ ,  $\text{LR}_{\text{pa}^*}(t)$  with the result from approximation based on original MGF,  $\text{LR}_{\text{M}}(t)$  in Figure 5.33 and Figure 5.34. Again, we see that the bias correction method given by equation 5.40 dramatically improves the quality of Padé-type approximation, and the plot of  $\text{LR}_{\text{pa}^*}(t)$  and  $\text{LR}_{\text{M}}(t)$  are almost indistinguishable in Figure 5.34. Moreover, Table 5.10 provides more details of the survival probabilities estimation made by the Lugannani-Rice approximation. By looking the explicit numerical result, we find that the accuracy of tail approximation is highly satisfactory after bias correction, and the deviation to the approximation based on original MGF,  $\text{SP}_{\text{M}}(t)$ , is typically smaller in  $\text{LR}_{\text{pa}^*}(t)$  than  $\text{LR}_{\text{pa}}(t)$ .

Table 5.10: Saddlepoint approximation for survival probabilities estimation. Unit of probability is in  $10^{-3}$

Time	14.59	15	16	17	18	19	19.21
$\text{LR}_{\text{pa}}(t)$	1.0234	0.8361	0.5107	0.3119	0.1905	0.1163	0.1049
$\text{LR}_{\text{pa}^*}^*(t)$	0.9914	0.8092	0.4931	0.3004	0.1830	0.1115	0.1005
$\text{LR}_{\text{M}}(t)$	1.0034	0.8183	0.4976	0.3024	0.1837	0.1117	0.1006

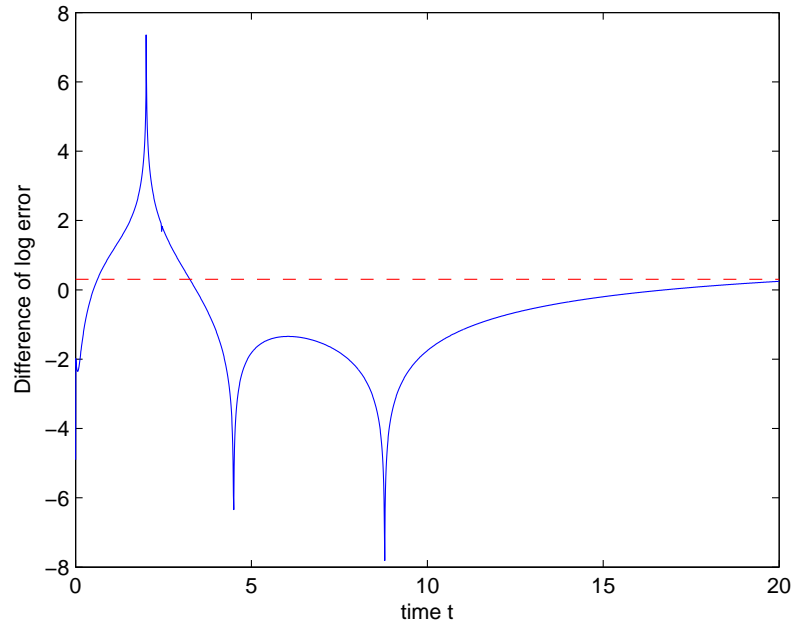


Figure 5.30: Plot of convergence test

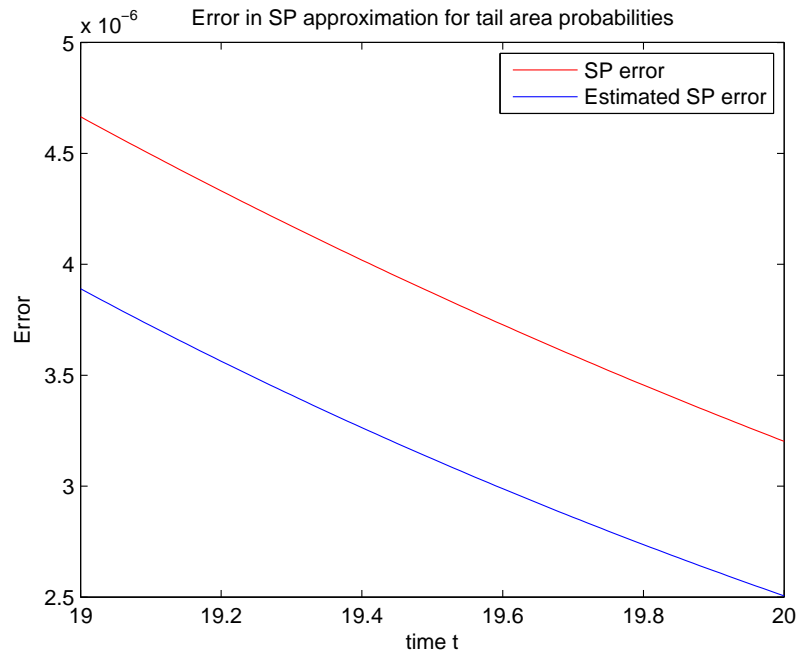


Figure 5.31: A comparison of error in tail area approximation 1

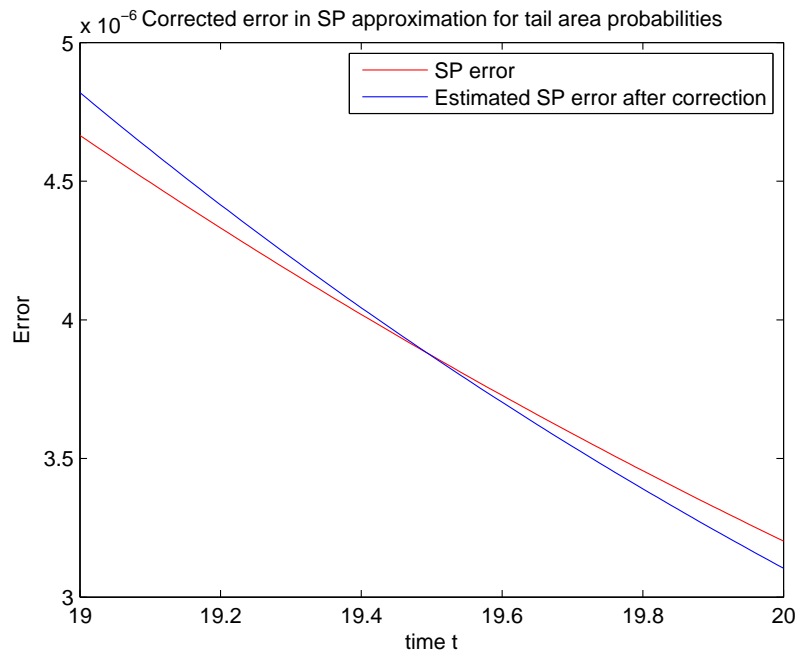


Figure 5.32: A comparison of error in tail area approximation 2

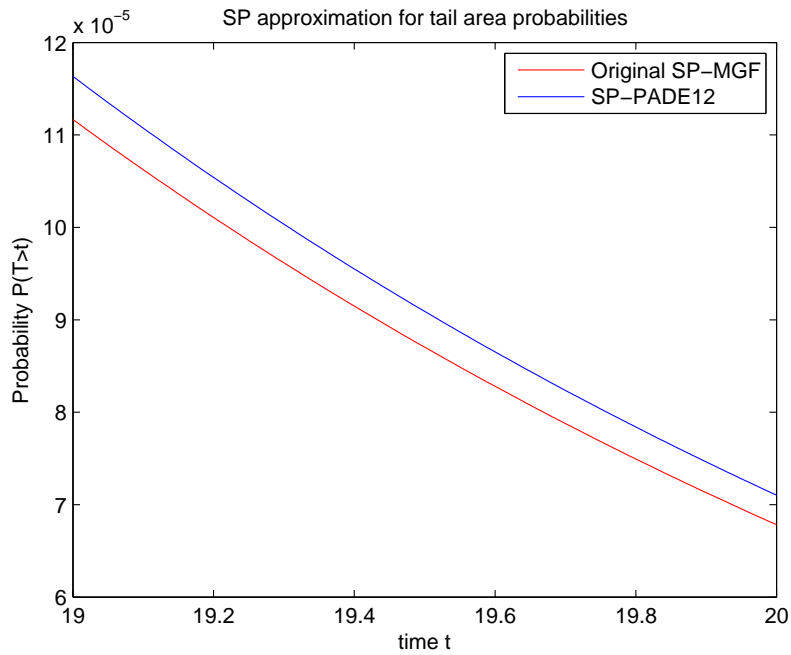


Figure 5.33: SP approximation for tail area probability 1

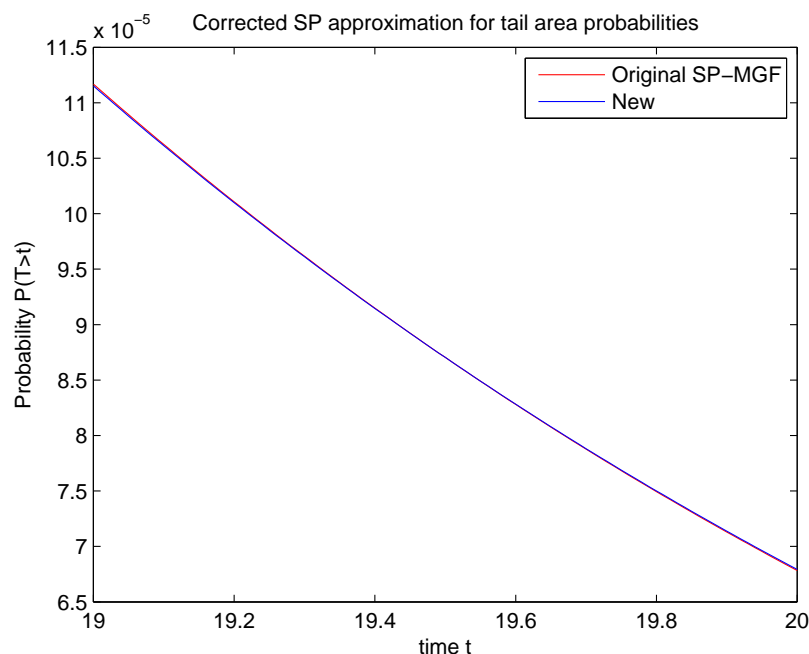


Figure 5.34: SP approximation for tail area probability 2

Table 5.11: A comparison of sum of absolute error of Padé-type saddlepoint approximation of PDF and survival function for  $t \in [14, 20]$ .

	original	bias corrected	reduce%
PDF	0.1537	0.0309	79.87%
Survival function	0.0065	0.0024	62.22%

It is also interesting to explore the performance of our bias corrected method in Padé-type saddlepoint approximation for PDF and survival function. By setting the saddlepoint approximation based on original MGF as benchmark result, we calculate the sum of absolute error in each Padé-type approximation for  $t \in [14, 20]$ . Table 5.11 gives strong evidence that the bias correction method does very well and it significantly improves the accuracy in the tail area of both function, particularly we have a 62.22% decrease in the sum of absolute error in the survival function estimation, while there is 79.87% reduction in the density function approximation.

# Chapter 6

## Application

In this chapter, we introduce our Matlab package, Method of Moments in Flowgraph (MMF), to analyse flowgraph data and estimate parameters. The MMF package supports a wide range of tasks, from calculating the branch transition matrix to derive the total waiting time MGF between two nodes of interests. It also includes functions and interactive tools for simulating flowgraph data between user defined input and output node, as well as the calculation of the Method of Moments estimator for the transition probability and the parameters of internode distribution.

The purpose of this chapter is to consolidate the methods we have developed for modeling flowgraph data and show the details of implementing our package in Matlab for flowgraph analysis. In the first part, we describe the main operations of MMF package and explain the theorems associated with the built-in functions. In the second part, we demonstrate the implementation of our package in a large complicated flowgraph example.

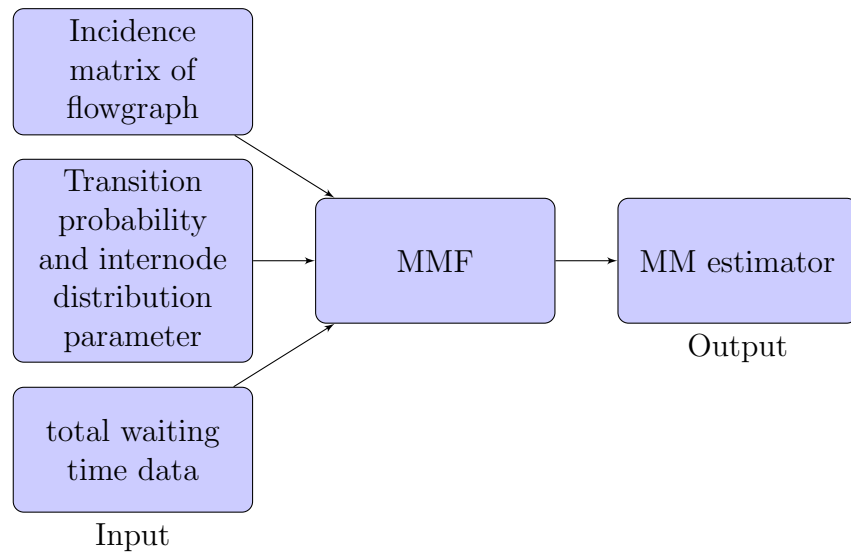


Figure 6.1: Illustration of MMF package

## 6.1 Overview

Figure 6.1 represents the operations of **MMF** package to compute the MM estimator. The **MMF** package only requires three inputs as they are illustrated in Figure 6.1. The incidence matrix initialise the structure of flowgraph by defining the direction of each edge, we can then assign the corresponding transition probability, specify the type of internode distribution and the corresponding parameters value, the total waiting time data can either be the real sample data or the data generated by simulation. The typical steps of our **MMF** package are given below.

**Step 1:** Derive the branch transition matrix .

**Step 2:** Compute the MGF of total waiting time between input and output.

**Step 3:** Calculate the Method of Moment estimators.

The following is the Matlab code of MMF package

```
1 function est=MMF(IM,iniv,data)
2 %Input:
3 %1)IM: Incidence matrix
4 %2)iniv: Value of parameters
5 %3)data: Total waiting time sample data
6
7 %Output:
8 %est: The Method of Moment estimator
9
10
11 % Derive the transition matrix Q
12 infor=tranm(IM);
13
14 % Calculate the sample moments
15 datm=zeros(1,length(iniv));
16 for i=1:length(iniv)
17 datm(i)=mean(data.^i);
18 end
19
20 %Compute MM estimator by the Nelder-Mead simplex algorithm
21 pad=infor.dist;
22 options=optimset('Display','iter');
23 est=fminsearch(@(theta) momeq(IM,pad,theta,datm),iniv,options);
24
25 end
```

The MMF package contains two functions that we have programmed in Matlab, namely, `tranm` and `momeq`. The description of each function is provided in the script of MMF package. To clearly illustrate what computations are being performed, we will give further explanation for each function, demonstrate the details of implementing each steps of MMF package in the next section.

## 6.2 Implementation in MATLAB

### 6.2.1 Function to compute the branch transition matrix

As the branch transition matrix plays a key role in our algebraic approach to compute the total waiting MGF, we create a user defined Matlab function files, `tranm.m`, to compute the branch transition matrix of a flowgraph model. This function requires the incidence matrix of flowgraph as input, then it will ask us to manually enter the transition probability for each directed edge, choose the type of internode distribution and the corresponding parameters value.

**Definition 13.** *The incidence matrix of a flowgraph is a  $l \times k$  matrix  $H = \{h_{ij}\}$ , where  $l$  and  $k$  are the total number of vertices and edges respectively in a flowgraph, such that  $h_{ij} = -1$  if the edge  $e_j$  (the  $j$ -th column) leaves vertex  $v_i$  (the  $i$ -th row),  $h_{ij} = 1$  if the edge  $e_j$  enters vertex  $v_i$ , and 0 otherwise.*

For example the incidence matrix for the flowgraph in Figure 2.5, page 17 is a matrix consists of 3 rows (corresponding to the three vertices) and 4 columns (corresponding to the four directed edges).

$$H = \begin{pmatrix} -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Next, we need to transform the incidence matrix  $H$  into the matrix  $B$  such that

$$B = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

where  $b_{ij} = -1$  if there is a potential transition from node  $i$  (the  $i$ -th row) to node  $j$  (the  $j$ -th column), and  $b_{ij} = 0$  for no transition.

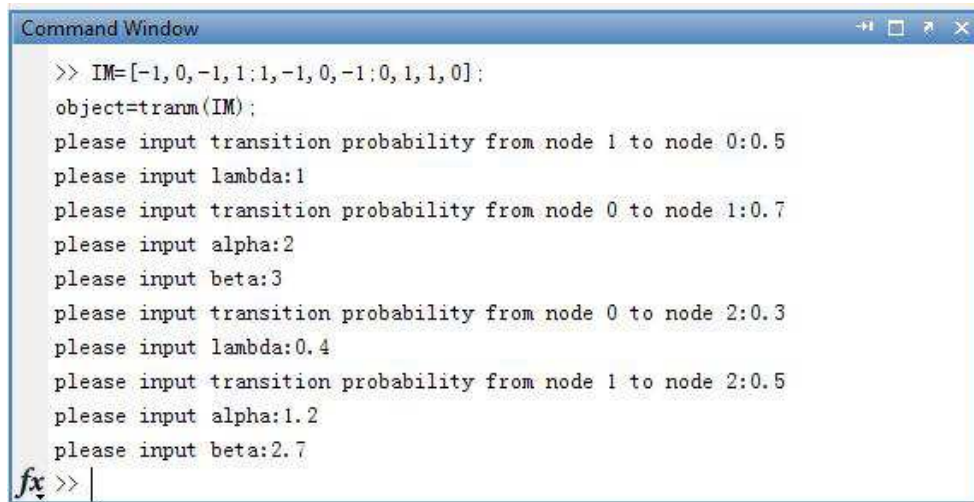


Given the transition probability  $p_{ij}$  and the MGF of waiting time between node  $i$  and node  $j$ ,  $m_{ij}(s)$ , the branch transition matrix  $Q$  is determined as

$$Q(s) = \begin{pmatrix} 0 & q_{01} & q_{02} \\ q_{10} & 0 & q_{12} \\ 0 & 0 & 0 \end{pmatrix}$$

where  $q_{ij} = q_{ij}(s) = p_{ij}m_{ij}(s)$ . Therefore, we can apply our Theorem 3 in Chapter 2 to compute the MGF of total waiting time between input at node 0 and output at node 2.

The implementation of `tranm.m` in Matlab are displayed as below.



```

Command Window
>> IM=[-1, 0, -1, 1:1, -1, 0, -1:0, 1, 1, 0];
object=tranm(IM);
please input transition probability from node 1 to node 0:0.5
please input lambda:1
please input transition probability from node 0 to node 1:0.7
please input alpha:2
please input beta:3
please input transition probability from node 0 to node 2:0.3
please input lambda:0.4
please input transition probability from node 1 to node 2:0.5
please input alpha:1.2
please input beta:2.7
fx >> |

```

Figure 6.2: Illustration of `tranm.m` function

Figure 6.2 shows the first output of implementing the `tranm.m` function for the flowgraph model on page 17. After substituting the incidence matrix `IM` into `tranm.m` function (i.e. `tranm(IM)`), we will be asked to enter the value of the transition probability for each edge, then a popup menu contains the choice of internode distribution, Exponential and Gamma will appear (see Figure 6.3). We can then simply enter the corresponding value of parameters for the selected distribution.

The output of `tranm.m` function consists of three items: the type of internode distribution for each edge (1 for exponential and 2 for Gamma) in `dist`, the set of parameter value in `theta`, and the branch transition matrix  $Q$  (see Figure 6.4). In practice, the `tranm.m` function allows us to easily determine the branch transition matrix  $Q$  of a large flowgraph by providing an interactive way to assign the information for each directed edge.

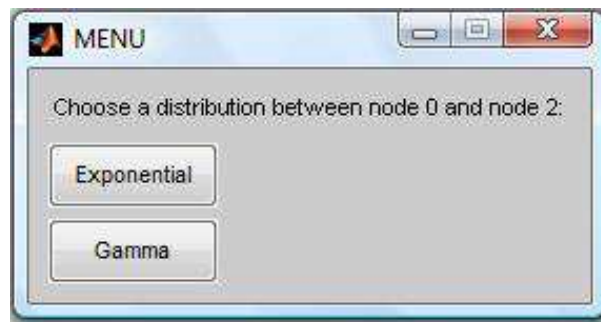


Figure 6.3: Menu for the choice of internode distribution

```
>> object

object =

    dist: [1 2 1 2]
    theta: [0.5000 0.7000 0.3000 0.5000 1 2 3 0.4000 1.2000 2.7000]
    Q: [3x3 sym]

>> object.Q

ans =

[      0, -7/(5*(s - 3)), -3/(25*(s - 2/5))]
[ -1/(2*(s - 1)),      0, -3/(5*(s - 27/10))]
[      0,      0,      0]
```

Figure 6.4: Output of `tranm.m` function

### 6.2.2 Function to compute the MGF

After we obtain the branch transition matrix  $Q$ , we program a Matlab function called `MV.m` to derive the total waiting time MGF based on Theorem 3, as well as the population moment (i.e.  $\mu_r(\theta)$  in Definition 11 Chapter 4).

The `MV.m` function requires the branch transition matrix  $Q$ , which is one of the output of `tranm.m` function, to compute the MGF evaluated at the value of parameters, while it needs the total number of parameter to be estimated in a flowgraph for calculating the population moments. The following is the Matlab code of `MV.m` function.

```

1 function object2=MV(Q,theta)
2 %Input:  1) Q: The branch transition matrix
3 %        2) theta: The set of parameter
4 %Output: 1) object2.MGF: The MGF of total waiting time
5 %        2) object2.mv: The population moment
6
7 % Construct an identity function
8 syms s
9 dim=size(Q);
10 im=eye(length(Q),length(Q));
11
12 % Apply Theorem 3 Chapter 2 to compute MGF
13 A=im-transpose(Q);
14 M=inv(A);
15 object2.MGF=M(dim(1),1);
16
17 % Calculate the population moment
18 np=length(theta);
19 ME=[zeros(np-1,1);s];
20 object2.mv=zeros(1,np);
21 for i=1:np;
22     ME(i)=diff(object2.MGF,s,i);
23     object2.mv(i)=subs(ME(i),s,0);
24 end

```

### 6.2.3 Sampling Methods

As the accuracy of parameter estimation depends on the sample size of total waiting time data, and it is common to have difficulty to obtain large enough real flowgraph data, this motivates us to apply the simulation method to generate total waiting time data between the input and output of a flowgraph.

For a simple flowgraphs with small number of nodes and feedback loops, we can simulate the path of each particle from input node to output node and then obtain the total waiting time according to the internode distribution. However, in a large complicated flowgraph case, it is not convenient and time consuming to write Matlab codes for simulation in a case by case manner. Therefore, instead of following the simulation method based on paths as we did in Section 4.2.2.2 Chapter 4, we creat a Matlab function call `simdat.m` (see Appendix) based on the inverse transform sampling method to obtain total waiting time data.

The inverse transform sampling is a method to generate sample of random variable  $X$  given its cumulative distribution function  $F(x)$ , it is based upon the following standard theorem.

**Theorem 11.** *Let  $F$  be a continuous cumulative distribution function on  $\mathbb{R}$  with inverse  $F^{-1}$  defined by*

$$F^{-1}(u) = \inf \{x : F(x) = u, 0 < u < 1\}$$

*If  $U$  is a uniform random variable from  $Uniform(0,1)$ , then  $F^{-1}(U)$  has distribution function  $F$ .*

The inverse transform sampling method is particularly useful for simulating total waiting time data in a large flowgraph model. Apart from the traditional path simulation approach, it allows us to obtain sample data based on its CDF that is derived from the MGF, which can be easily determined by using the algebraic approach formula in Theorem 3 Chapter 2. Given the MGF of total waiting time  $M(s)$ , we can determine the PDF  $f(x)$  by either the exact inversion method in Section 3.1 or the numerical method in Section 3.2, Chapter 3, then

we can apply the procedure of inverse transform sampling as follow

1. Derive the inverse CDF  $F^{-1}(x)$  from PDF  $f(x)$
2. Generate  $y \sim Uniform(0, 1)$
3. Compute  $x = F^{-1}(y)$

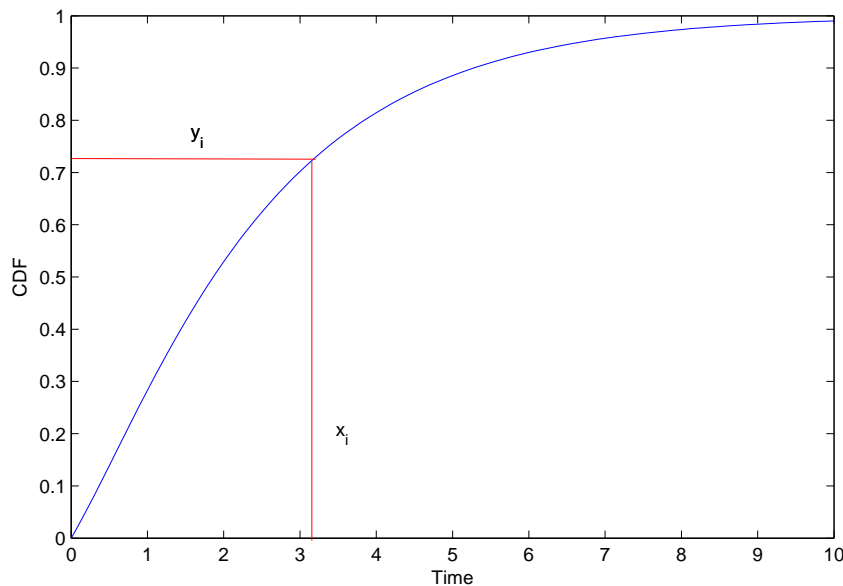


Figure 6.5: Illustration of inverse transform sampling

Figure 6.5 demonstrates how the random sample  $x_i$  is generated given uniform random number  $y_i$ . The inversion method is exact when  $F^{-1}(x)$  is in closed form. However, the CDF  $F(t)$  of total waiting time between two nodes of interests in a flowgraph model is usually nonlinear function, and the analytical expression of  $F^{-1}(t)$  is not available. In this case, we need to apply numerical method to solve the equation  $F(x) = u$  numerically for  $x$ . Note that we could apply the built-in Matlab function called “fsolve” to find the solution of nonlinear equations.

### 6.2.4 Method to compute the MM estimator

As the MM estimator  $\hat{\theta}$  for parameter  $\theta = (\theta_1, \dots, \theta_d)$  is the solution to the system of equations determined by Definition 11, Chapter 4, we can construct a function  $R(\theta)$  as

$$R(\theta) = \sum_{r=1}^d (\mu_r(\theta) - m_r)^2 \quad \text{for } r = 1, \dots, d \quad (6.1)$$

then  $R(\hat{\theta}) = 0$  will be satisfied if  $\hat{\theta}$  is the MM estimator. Hence the computation of MM estimator can be considered as a problem for finding the zeros of a scalar function  $R(\theta)$  of several variables  $\theta = (\theta_1, \dots, \theta_d)$ .

The Nelder-Mead algorithm is one of the well known algorithms for multidimensional optimization problem. It is developed by Nelder and Mead (1965) and it has been extensively used to solve parameter estimation. The basic idea is to use the term simplex (a generalized triangle in  $d$  dimensions) and find the minimum of a function of  $d$  variables. For reference, see Powell (1973) and McKinnon (1999). Byatt, Coope and Price (2003) summary the development of this method.

Since it is computationally demanding to construct the Jacobian matrix of the systems of moment equations, this algorithm belongs to a general class of direct search methods which do not require to use any derivatives, therefore it is relatively easy to implement than other optimisation method for computing the MM estimators. We use Matlab's built-in function `fminsearch` (see Lagarius, Reeds, Wright, and Wright (1998)) to find the minimum of  $R(\theta)$  function (i.e.  $R(\theta) = 0$ ) and calculate the MM estimator.

To demonstrate the use of `fminsearch` function, we consider the series flow-graph model example in Figure 2.1 Chapter 2 again. The incidence matrix is defined as

$$H = \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix}$$

Suppose we assume  $t_{01} \sim \text{Exp}(\lambda_1)$  and  $t_{12} \sim \text{Exp}(\lambda_2)$ , A sample size of  $n$  total waiting time between node 0 and node 2 is generated by the inverse sampling method (i.e. `simdat` function), compute the sample moments and derive function  $R(\theta)$  for  $d = 4$  case by equation 6.1. Given the sample moments, we create a function handle called `momeq.m` (see Appendix) to compute the value of  $R(\theta)$  at each value of parameters  $\theta$ , so that the Nelder-Mead algorithm can be applied iteratively to minimise  $R(\theta)$  for calculating the MM estimator.

```

1 %Define the incidence matrix
2 IM=[-1,0;1,-1;0,1];
3
4 %Define the initial value of parameters
5 iniv=[0.9,1.1,0.6,1.4];
6
7 %simulate 10000 waiting time data
8 n=10000;
9 datv=simdat(out.MGF,n);
10
11 %Call function momeq and calculate the MM estimator
12 options=optimset('Display','iter');
13 [est,fval]=fminsearch(@(theta) ...
    momeq(IM,pad,theta,datm),iniv,options);

```

Note that we only use the total waiting time data generated by `simdat.m` function in Section 6.2.3, in practice, we usually use the sample total waiting time data for parameter estimation. The computation proceeds by applying Nelder-Mead algorithm (i.e. `fminsearch` function) to find a minimum of `momeq` function (i.e. the value of  $R(\theta)$ ), starting at an initial value of parameter specified in `iniv`. The `options` function allows us to see the output at each iteration of calculation, such as the value of  $R(\theta)$  and the operation of Nelder-Mead algorithm. The computation will be terminated if the change in the function value is less than  $10^{-4}$  (i.e.  $|R(\theta^{l+1}) - R(\theta^l)| \leq 10^{-4}$ ), the result of MM estimator  $\hat{\theta}$  and the corresponding value of  $R(\theta)$  at  $\hat{\theta}$  will be stored in variable `est` and `fval` respectively.

## 6.3 Example 9

In this section, we implement our MMF package to estimate the MM estimator for a total of 10 parameters in a flowgraph, which has a similar structure to the flowgraph given in Figure 2.5, Section 2.2.1, Chapter 2, but contains some internode waiting time that follow Gamma distribution. We summary the type of the internode distribution and the corresponding value of parameters in Table 6.1.

Table 6.1: Summary of waiting time distribution

Direction	transition probability	Distribution
$1 \rightarrow 0$	$p_{10} = 0.4$	Exponential(2)
$0 \rightarrow 1$	$p_{01} = 0.7$	Gamma(1.5, 3)
$0 \rightarrow 2$	$p_{02} = 0.3$	Exponential(1)
$1 \rightarrow 2$	$p_{12} = 0.6$	Gamma(9, 0.5)

**Step 1:** Define the incidence matrix of flowgraph in Figure 2.5 on page 17.

$$H = \begin{pmatrix} -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Recall that the column and row of matrix  $H$  represents the edge and vertex of flowgraph, such that  $h_{ij} = -1$  if edge  $e_j$  (the  $j$ -th column) leaves vertex  $v_i$  (the  $i$ -th row),  $h_{ij} = 1$  if edge  $e_j$  enters vertex  $v_i$ , and 0 otherwise.

**Step 2:** Simulate a sample size of  $n = 10^6$  total waiting time data between node 0 and node 2 by the use of `simdat` function. Note that we usually input the real sample data rather than the simulated data for practical application.

**Step 3:** Execute the MMF package (see Figure 6.6), input the incidence matrix `IM`, the sample data `datv`, and the initial value `iniv` for the transition probability and the parameters of each internode distribution by the menu in Figure 6.3.



```

est=MMF(IM, iniv, datv):
please input transition probability from node 1 to node 0:0.4
please input lambda:2
please input transition probability from node 0 to node 1:0.7
please input alpha:1.5
please input beta:3
please input transition probability from node 0 to node 2:0.3
please input lambda:1
please input transition probability from node 1 to node 2:0.6
please input alpha:9
please input beta:0.5

```

Figure 6.6: Implementation of MMF package in Matlab

The package takes 596.94 seconds to complete the computation, where it requires 507 iterations of the Nelder-Mead algorithm. The calculated results obtained from the MMF package are displayed in the Table 6.2 and Table 6.3. By comparing with the true value of parameters, we can see that reasonably good estimation results are obtained for both transition probability and internode distribution parameters.

Table 6.2: Summary of MM estimators for transition probabilities

Parameter	$p_{10}$	$p_{01}$	$p_{02}$	$p_{12}$
True value	0.4	0.7	0.3	0.6
Estimated value	0.4216	0.7163	0.3256	0.6131

Table 6.3: Summary of MM estimators for internode distribution parameters

Parameter	$\lambda_{10}$	$\lambda_{02}$	$\alpha_{01}$	$\beta_{01}$	$\alpha_{12}$	$\beta_{12}$
True value	2	1	1.5	3	9	0.5
Estimated value	2.1947	1.1735	1.2335	2.7618	9.6072	0.5131

To conclude, the MMF package not only provides a user-friendly way to input the information from flowgraph but also it unifies the processes of constructing branch transition matrix, deriving the total waiting time MGF, and computing the MM estimators. For practical application, we can easily adapt the MMF package in MATLAB as needed for the parameter estimation problem in any type of flowgraph models.



# Appendix

1. We summary the system of 6 moment equations for MM in Section 4.3.4.1

$$\begin{aligned} & \mu_1(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\ = & \frac{-\lambda_4 p_{01} p_{12} \lambda_1 \lambda_3 - \lambda_4 \lambda_1 \lambda_2 + \lambda_4 \lambda_1 \lambda_2 p_{01} - p_{01} \lambda_2 \lambda_3 \lambda_4 - p_{01} \lambda_2 \lambda_3 \lambda_1 + p_{01} \lambda_2 \lambda_3 p_{12} \lambda_1}{\lambda_4 \lambda_1 \lambda_2 \lambda_3 (p_{01} - p_{01} p_{12} - 1)} \end{aligned}$$

$$\begin{aligned} & \mu_2(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\ = & \frac{2}{\lambda_4^2 (p_{01} - p_{01} p_{12} - 1)^2 \lambda_1^2 \lambda_2^2 \lambda_3^2} \cdot \\ & \left( -\lambda_4^2 p_{01} p_{12} \lambda_1 \lambda_2 \lambda_3^2 + p_{01}^2 \lambda_2^2 \lambda_3^2 \lambda_4 p_{12} \lambda_1 - p_{01}^2 \lambda_2 \lambda_3^2 \lambda_4 p_{12} \lambda_1^2 + p_{01}^2 \lambda_2 \lambda_3^2 \lambda_4 p_{12}^2 \lambda_1^2 \right. \\ & + p_{01} \lambda_2^2 \lambda_3 \lambda_4 \lambda_1^2 p_{12} - p_{01}^2 \lambda_2^2 \lambda_3 \lambda_4^2 \lambda_1 p_{12} - p_{01}^2 \lambda_2^2 \lambda_3 \lambda_4 \lambda_1^2 p_{12} + p_{01} \lambda_2^2 \lambda_3^2 p_{12} \lambda_1 \lambda_4 - p_{01} \lambda_2^2 \lambda_3^2 \lambda_4^2 \\ & - p_{01} \lambda_2^2 \lambda_3^2 \lambda_1^2 + 2\lambda_4^2 \lambda_1^2 \lambda_2^2 p_{01} - \lambda_4^2 \lambda_1^2 \lambda_2^2 p_{01}^2 - \lambda_4^2 \lambda_1^2 \lambda_2^2 - \lambda_4^2 p_{01}^2 p_{12}^2 \lambda_1^2 \lambda_3^2 - \lambda_4^2 p_{01} p_{12} \lambda_1^2 \lambda_3^2 \\ & + \lambda_4^2 p_{01}^2 p_{12} \lambda_1^2 \lambda_3^2 - \lambda_4^2 \lambda_1^2 \lambda_2^2 p_{01} p_{12} + \lambda_4^2 \lambda_1^2 \lambda_2^2 p_{01}^2 p_{12} - p_{01} \lambda_2^2 \lambda_3 \lambda_4^2 \lambda_1 - p_{01} \lambda_2^2 \lambda_3 \lambda_4 \lambda_1^2 \\ & \left. + p_{01}^2 \lambda_2^2 \lambda_3 \lambda_4^2 \lambda_1 + p_{01}^2 \lambda_2^2 \lambda_3 \lambda_4 \lambda_1^2 - p_{01}^2 \lambda_2^2 \lambda_3^2 \lambda_1 \lambda_4 - p_{01} \lambda_2^2 \lambda_3^2 \lambda_1 \lambda_4 + p_{01} \lambda_2^2 \lambda_3^2 p_{12} \lambda_1^2 \right. \\ & \left. + p_{01} \lambda_2^2 \lambda_3 \lambda_4^2 \lambda_1 p_{12} \right) \end{aligned}$$

$$\begin{aligned}
& \mu_3(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
= & \frac{6}{\lambda_4^3(p_{01} - p_{01}p_{12} - 1)^3 \lambda_1^3 \lambda_2^3 \lambda_3^3} \cdot \\
& \left( 3\lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01} - 3\lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^2 + \lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^3 - \lambda_4^3 p_{01}^2 p_{12}^2 \lambda_1^2 \lambda_2 \lambda_3^3 + \lambda_4^3 p_{01}^2 p_{12} \lambda_1^2 \lambda_2 \lambda_3^3 \right. \\
& - \lambda_4^3 p_{01} p_{12} \lambda_1 \lambda_2^2 \lambda_3^3 - \lambda_4^3 p_{01} p_{12} \lambda_1^2 \lambda_2 \lambda_3^3 \lambda_4^3 p_{01} p_{12} \lambda_1^2 \lambda_2 \lambda_3^3 - p_{01}^2 \lambda_2 \lambda_3^3 \lambda_4^2 p_{12} \lambda_1^3 \\
& + p_{01}^2 \lambda_2 \lambda_3^3 \lambda_4^2 p_{12}^2 \lambda_1^3 + p_{01} \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 p_{12} - 3p_{01}^2 \lambda_2^3 \lambda_3 \lambda_4^2 \lambda_1^3 p_{12} - 2p_{01}^3 \lambda_2 \lambda_3^3 \lambda_4^2 p_{12}^2 \lambda_1^3 \\
& + p_{01}^3 \lambda_2 \lambda_3^3 \lambda_4^2 p_{12} \lambda_3^3 \lambda_4^2 p_{12} \lambda_1^2 + 2p_{01}^2 \lambda_2^2 \lambda_3^3 \lambda_4^2 p_{12}^2 \lambda_1^2 - 3p_{01}^2 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 p_{12} \\
& + p_{01}^2 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 p_{12}^2 + 2p_{01}^3 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 p_{12} - p_{01}^3 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 p_{12}^2 - p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4^3 \lambda_1 p_{12} \\
& + p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 p_{12} - p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 p_{12}^2 - p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4 \lambda_1^3 p_{12} - 2p_{01}^3 \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 p_{12} \\
& + p_{01}^3 \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1 p_{12}^2 - p_{01}^2 \lambda_2^2 \lambda_3^3 \lambda_4 p_{12} \lambda_1^3 + p_{01}^2 \lambda_2^2 \lambda_3^3 \lambda_4 p_{12}^2 \lambda_1^3 + p_{01} \lambda_2^3 \lambda_3^2 \lambda_4^3 \lambda_1 p_{12} \\
& + p_{01} \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 p_{12} + p_{01} \lambda_2^3 \lambda_3^2 \lambda_4 \lambda_1^3 p_{12} + 3p_{01}^2 \lambda_2^3 \lambda_3^3 \lambda_4 p_{12} \lambda_1^2 - p_{01}^2 \lambda_2^3 \lambda_3^3 \lambda_4 p_{12}^2 \lambda_1^2 \\
& + p_{01} \lambda_2^3 \lambda_3^3 p_{12} \lambda_1 \lambda_4^2 + p_{01} \lambda_2^3 \lambda_3^3 p_{12} \lambda_1^2 \lambda_4 - \lambda_4^3 \lambda_1^3 \lambda_2^3 - \lambda_4^3 p_{01}^3 p_{12}^3 \lambda_1^3 \lambda_3^3 - 2\lambda_4^3 p_{01}^2 p_{12}^2 \lambda_1^3 \lambda_3^3 \\
& + 2\lambda_4^3 p_{01}^3 p_{12}^2 \lambda_1^3 \lambda_3^3 + 2\lambda_4^3 p_{01}^2 p_{12} \lambda_1^3 \lambda_3^3 - \lambda_4^3 p_{01}^3 p_{12} \lambda_1^3 \lambda_3^3 - \lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^2 p_{12}^2 - 2\lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01} p_{12} \\
& + 4\lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^2 p_{12} + \lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^3 p_{12}^2 - 2\lambda_4^3 \lambda_1^3 \lambda_2^3 p_{01}^3 p_{12} - \lambda_4^3 p_{01} p_{12} \lambda_1^3 \lambda_3^3 + 2p_{01}^2 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 \\
& - p_{01}^3 \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 - p_{01} \lambda_2^3 \lambda_3 \lambda_4^2 \lambda_1^3 - p_{01} \lambda_2^3 \lambda_3 \lambda_4^3 \lambda_1^2 + 2p_{01}^2 \lambda_2^3 \lambda_3 \lambda_4^2 \lambda_1^3 - p_{01}^3 \lambda_2^3 \lambda_3 \lambda_4^2 \lambda_1^3 \\
& + p_{01}^3 \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 + p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4 \lambda_1^3 + p_{01}^2 \lambda_2^3 \lambda_3^2 \lambda_4^3 \lambda_1 - p_{01} \lambda_2^3 \lambda_3^2 \lambda_4 \lambda_1^3 - p_{01} \lambda_2^3 \lambda_3^2 \lambda_4^3 \lambda_1 \\
& - p_{01} \lambda_2^3 \lambda_3^2 \lambda_4^2 \lambda_1^2 - 2p_{01}^2 \lambda_2^3 \lambda_3^3 \lambda_1^2 \lambda_4 - 2p_{01}^2 \lambda_2^3 \lambda_3^3 \lambda_1 \lambda_4^2 - p_{01} \lambda_2^3 \lambda_3^3 \lambda_1^2 \lambda_4 - p_{01} \lambda_2^3 \lambda_3^3 \lambda_1 \lambda_4^2 \\
& \left. + p_{01} \lambda_2^3 \lambda_3^3 p_{12} \lambda_1^3 - p_{01} \lambda_2^3 \lambda_3^3 \lambda_1^3 - p_{01} \lambda_2^3 \lambda_3^3 \lambda_4^3 \right)
\end{aligned}$$

$$\begin{aligned}
& \mu_4(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
= & \frac{24}{\lambda_4^4(p_{01} - p_{01}p_{12} - 1)^4 \lambda_1^4 \lambda_2^4 \lambda_3^4} \cdot \\
& \left( -2p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4^3 \lambda_1^2 p_{12}^2 + p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^4 + p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4^4 \lambda_1 - p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 - p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^4 p_{12} \right. \\
& \lambda_1^4 p_{12} + p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 + 2p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 5p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 + 7p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} \\
& + 2p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 p_{12}^2 - 5p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 p_{12}^2 + 2p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 p_{12}^2 - 4p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1 p_{12} \\
& - 4p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 p_{12} - p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^2 \lambda_1^4 + 6p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^2 - p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 - 2p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12}^2 \\
& \lambda_1^2 + 2p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^3 - 2p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^3 - p_{01} \lambda_2^4 \lambda_3^4 \lambda_1^4 - p_{01} \lambda_2^4 \lambda_3^4 \lambda_4 - p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 \\
& + \lambda_4^4 p_{01}^2 p_{12} \lambda_1^2 \lambda_2^2 \lambda_3^4 - \lambda_4^4 p_{01}^3 p_{12}^3 \lambda_1^3 \lambda_2 \lambda_3^4 - 2\lambda_4^4 p_{01}^2 p_{12}^2 \lambda_1^3 \lambda_2 \lambda_3^4 - \lambda_4^4 p_{01}^4 p_{12}^4 \lambda_1^4 \lambda_3^4 - \lambda_4^4 p_{01} p_{12} \lambda_1 \lambda_2^3 \lambda_3^4 \\
& - \lambda_4^4 p_{01} p_{12} \lambda_1^2 \lambda_2^2 \lambda_3^4 - \lambda_4^4 p_{01} p_{12} \lambda_1^3 \lambda_2 \lambda_3^4 + 2\lambda_4^4 p_{01}^3 p_{12}^3 \lambda_1^3 \lambda_2 \lambda_3^4 - 4p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_1^2 \lambda_4^2 - 3p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_1^3 \lambda_4 \\
& - 3p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_1 \lambda_4^3 + p_{01} \lambda_2^4 \lambda_3^4 p_{12} \lambda_1^4 - p_{01} \lambda_2^4 \lambda_3^4 \lambda_1^2 \lambda_4^2 - p_{01} \lambda_2^4 \lambda_3^4 \lambda_1^3 \lambda_4 - p_{01} \lambda_2^4 \lambda_3^4 \lambda_1 \lambda_4^3 \\
& + 3p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 3p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} - p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} + 3p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 3p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} \\
& + p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 + 2p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^3 + p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 p_{12}^3 - 5p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} + 2p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^3 p_{12}^3 \lambda_1^3 \\
& - 4p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^3 - p_{01}^3 \lambda_2^4 \lambda_3^4 \lambda_1^2 \lambda_4^2 + p_{01} \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 p_{12} - 5p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 p_{12} + 3p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3 \\
& + p_{01}^4 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^4 \lambda_1^4 + 2p_{01}^3 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^3 \lambda_1^4 - 3p_{01}^4 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^3 \lambda_1^4 - 4p_{01}^3 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^4 + 3p_{01}^4 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^4 \\
& + 2p_{01}^3 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^4 - p_{01}^4 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^4 - 2p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^3 - p_{01} \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^4 + p_{01} \lambda_2^4 \lambda_3 p_{12} \lambda_1 \lambda_3^4 \\
& + p_{01} \lambda_2^4 \lambda_3 p_{12} \lambda_1^2 \lambda_4^2 + p_{01} \lambda_2^4 \lambda_3 p_{12} \lambda_1^3 \lambda_4 - p_{01}^4 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 - p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 - p_{01}^2 \lambda_2^2 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^4 \\
& + p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 p_{12} + p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12} + p_{01}^2 \lambda_2^2 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^4 + p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1 p_{12} - p_{01} \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 \\
& - \lambda_4^4 p_{01}^2 p_{12}^2 \lambda_1^2 \lambda_2^2 \lambda_3^4 - p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12}^3 + 3p_{01}^4 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12} + 2p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 p_{12} + p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^2 p_{12}^3 \lambda_1^4 \\
& - 2p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^4 + p_{01}^3 \lambda_2^2 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^4 + p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12}^2 - 3p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12} + p_{01}^3 \lambda_2^4 \lambda_3^4 \lambda_4^3 \lambda_1^3 \\
& - p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12} - p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^2 \lambda_1^4 p_{12}^2 + 2p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^2 \lambda_1^4 p_{12} + 4\lambda_4^4 \lambda_1^4 \lambda_2 p_{01} - \lambda_4^4 \lambda_1^4 \lambda_2 p_{01} - 6\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^2 \\
& + 4\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^3 + p_{01}^4 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12}^3 - 3p_{01}^4 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12}^2 + p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1 p_{12}^2 - 3p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1 p_{12} \\
& + p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 - p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^2 \lambda_1^4 + 2p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^2 \lambda_1^4 + 2p_{01}^3 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^2 + 2p_{01}^2 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 \\
& - p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^4 \lambda_1^2 p_{12}^2 + 3p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12}^2 - 3p_{01}^3 \lambda_2^4 \lambda_3^2 \lambda_4^3 \lambda_1^3 p_{12} - p_{01}^3 \lambda_2^4 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^2 - p_{01}^3 \lambda_2^3 \lambda_3^4 \lambda_4^2 p_{12}^3 \lambda_1^3 \\
& + 2p_{01}^3 \lambda_2^3 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^3 - p_{01}^3 \lambda_2^3 \lambda_3^4 \lambda_4^2 p_{12} \lambda_1^3 + p_{01} \lambda_2^4 \lambda_3^3 \lambda_4^4 \lambda_1 p_{12} + p_{01} \lambda_2^4 \lambda_3^3 \lambda_4^4 \lambda_1^2 p_{12} + p_{01} \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 p_{12} \\
& + 3\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^4 p_{12} + 2\lambda_4^4 p_{01}^2 p_{12} \lambda_1^3 \lambda_2 \lambda_3^4 - \lambda_4^4 p_{01}^3 p_{12} \lambda_1^3 \lambda_2 \lambda_3^4 - 3\lambda_4^4 p_{01}^3 p_{12}^3 \lambda_1^4 \lambda_3^4 + 3\lambda_4^4 p_{01}^4 p_{12}^3 \lambda_1^4 \lambda_3^4 \\
& - 3\lambda_4^4 p_{01}^2 p_{12}^2 \lambda_1^4 \lambda_3^4 + 6\lambda_4^4 p_{01}^3 p_{12}^2 \lambda_1^4 \lambda_3^4 - 3\lambda_4^4 p_{01}^4 p_{12}^2 \lambda_1^4 \lambda_3^4 + 3\lambda_4^4 p_{01}^2 p_{12} \lambda_1^4 \lambda_3^4 - 3\lambda_4^4 p_{01}^3 p_{12} \lambda_1^4 \lambda_3^4 \\
& + \lambda_4^4 p_{01}^4 p_{12} \lambda_1^4 \lambda_3^4 - \lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^3 p_{12}^3 - 3\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^2 p_{12}^2 + 6\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^3 p_{12}^2 - 3\lambda_4^4 \lambda_1^4 \lambda_2 p_{01} p_{12} \\
& + 9\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^2 p_{12} - 9\lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^3 p_{12} + \lambda_4^4 \lambda_1^4 \lambda_2 p_{01}^4 p_{12}^3 + 7p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 p_{12} - 3p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^4 \lambda_1^3
\end{aligned}$$

$$\begin{aligned}
& +3p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1 + 3p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^2 - 3p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^2 + 2p_{01}^2 \lambda_2^2 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^3 \\
& +3p_{01}^2 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 - 3p_{01}^3 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 + p_{01}^4 \lambda_2^4 \lambda_3 \lambda_4^3 \lambda_1^4 + p_{01}^2 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12}^2 \lambda_1^4 - p_{01}^2 \lambda_2 \lambda_3^4 \lambda_4^3 p_{12} \lambda_1^4 \\
& +p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4 p_{12}^2 \lambda_1^4 - p_{01}^2 \lambda_2^3 \lambda_3^4 \lambda_4 p_{12} \lambda_1^4 - 3\lambda_4^4 \lambda_1^4 \lambda_2^4 p_{01}^2 p_{12}^2 - p_{01} \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^4 - p_{01} \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1 \\
& -p_{01} \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 + p_{01} \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^4 p_{12} - p_{01} \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 - \lambda_4^4 p_{01} p_{12} \lambda_1^4 \lambda_3^2 - 2p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4 p_{12}^2 \lambda_1^3 \\
& +5p_{01}^2 \lambda_2^4 \lambda_3^4 \lambda_4 p_{12} \lambda_1^3 - p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 - 2p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 p_{12}^2 - p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1 p_{12} \\
& +3p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4^2 \lambda_1^3 p_{12} - \lambda_4^4 \lambda_1^4 \lambda_2^4 + 2p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 p_{12}^2 + 2p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^3 + 2p_{01}^3 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 \\
& +p_{01} \lambda_2^4 \lambda_3 \lambda_4^3 - p_{01} \lambda_2^4 \lambda_3 \lambda_4 \lambda_1^3 + 3p_{01}^2 \lambda_2^4 \lambda_3^3 \lambda_4 \lambda_1^2 p_{12} \Big)
\end{aligned}$$

$$\begin{aligned}
& \mu_5(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
= & \frac{120}{\lambda_4^5(p_{01} - p_{01}p_{12} - 1)^5 \lambda_1^5 \lambda_2^5 \lambda_3^5} \cdot \\
& \left( 3p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 - 2p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 + 3p_{01}^3 \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 + p_{01}^2 \lambda_2^2 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^5 - 6\lambda_4^5 p_{01}^3 p_{12} \lambda_1^5 \lambda_3^5 \right. \\
& - \lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^4 p_{12}^4 + 8\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^4 p_{12}^3 - 18\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^4 p_{12}^2 + 16\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^4 p_{12} - 24\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^3 p_{12} \\
& + 18\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^3 p_{12}^2 - 4\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^3 p_{12}^3 + 16\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^2 p_{12} - 6\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^2 p_{12}^2 - 4\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01} p_{12} \\
& + \lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^5 p_{12}^4 - 4\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^5 p_{12}^3 + 6\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^5 p_{12}^2 - p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 - p_{01}^2 \lambda_2^4 \lambda_3^5 \lambda_4 p_{12} \lambda_1^5 \\
& + p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^5 \lambda_1 p_{12} + 6p_{01}^3 \lambda_2^4 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^3 - 3p_{01}^2 \lambda_2^4 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^3 - 2p_{01}^2 \lambda_2^4 \lambda_3^5 \lambda_4^2 p_{12} \lambda_1^4 \\
& - p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^5 \lambda_1 p_{12} + 5p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 p_{12} + 7p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 p_{12} + 5p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^2 \lambda_1^4 p_{12} \\
& + p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^5 \lambda_1 + p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^2 \lambda_1^5 p_{12} + 3p_{01}^3 \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 p_{12}^2 - 10\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^2 - 5\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^4 \\
& + 10\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^3 + 5\lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01} + \lambda_4^5 \lambda_1^5 \lambda_2^5 p_{01}^5 + 4p_{01}^5 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 p_{12} + 3p_{01}^4 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^4 \lambda_1^5 \\
& + p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 p_{12}^2 - p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 p_{12}^2 + 2p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1 p_{12} + 2p_{01}^2 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12}^2 + 3p_{01}^2 \lambda_2^5 \lambda_3^2 \lambda_4^3 \lambda_1^5 \\
& + p_{01} \lambda_2^5 \lambda_3^5 p_{12} \lambda_1 \lambda_4^4 + p_{01} \lambda_2^5 \lambda_3^5 p_{12} \lambda_1^3 \lambda_4^2 + p_{01} \lambda_2^5 \lambda_3^5 p_{12} \lambda_1^5 - p_{01} \lambda_2^5 \lambda_3^5 \lambda_1^5 - p_{01} \lambda_2^5 \lambda_3^5 \lambda_4^5 + 6p_{01}^5 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^4 p_{12}^2 \\
& - 3p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12} + 7p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^4 p_{12} + 5p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^3 p_{12} - 4p_{01}^5 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^3 p_{12} - 13p_{01}^4 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 p_{12} \\
& + p_{01} \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 p_{12} - p_{01} \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 + 4p_{01}^2 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 - 6p_{01}^3 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 + 4p_{01}^4 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 - p_{01}^5 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 \\
& + 15p_{01}^3 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 p_{12} + 3p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^4 p_{12} \lambda_1^3 - p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^4 p_{12} \lambda_1^3 - 6p_{01}^5 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^5 p_{12}^2 - 2p_{01}^2 \lambda_2^5 \lambda_3 \lambda_4^4 \lambda_1^2 \\
& - 3p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 - 2p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^2 \lambda_1^4 + p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4 \lambda_1^5 + 4p_{01}^2 \lambda_2^5 \lambda_3^5 \lambda_4^4 p_{12} \lambda_1 + 9p_{01}^2 \lambda_2^5 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^2 \\
& + 10p_{01}^2 \lambda_2^5 \lambda_3^5 \lambda_4^2 p_{12} \lambda_1^3 + 4p_{01}^2 \lambda_2^4 \lambda_3^5 \lambda_4^4 p_{12}^2 \lambda_1^2 - p_{01} \lambda_2^5 \lambda_3^2 \lambda_4^3 \lambda_1^5 - 2p_{01}^3 \lambda_2^4 \lambda_3^5 \lambda_4^2 p_{12} \lambda_1^4 - p_{01}^4 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12}^4 \lambda_1^4 \\
& - p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 2p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12}^3 + 2p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12}^3 + p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12}^3 \lambda_1^4 + 3p_{01}^4 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12}^3 \lambda_1^4 \\
& + 15p_{01}^3 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 p_{12} - 13p_{01}^4 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 p_{12} - 3p_{01}^3 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{01} \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12} - p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12}^2 \\
& - 4p_{01}^5 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^4 \lambda_1^5 + 3p_{01}^3 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^3 \lambda_1^5 - 9p_{01}^4 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^3 \lambda_1^5 + 6p_{01}^5 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^3 \lambda_1^5 - 6p_{01}^3 \lambda_2 \lambda_3^5 \lambda_4^4 p_{12}^2 \lambda_1^5 \\
& + 2p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^3 p_{12}^3 \lambda_1^5 - 4p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^5 + 2p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^5 - 6\lambda_4^5 p_{01}^3 p_{12}^2 \lambda_1^5 \lambda_3^5 - 4\lambda_4^5 p_{01}^2 p_{12}^2 \lambda_1^5 \lambda_3^5 \\
& + 4\lambda_4^5 p_{01}^4 p_{12} \lambda_1^5 \lambda_3^5 + 4\lambda_4^5 p_{01}^2 p_{12} \lambda_1^5 \lambda_3^5 - \lambda_4^5 p_{01}^5 p_{12} \lambda_1^5 \lambda_3^5 + p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 - 2p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^4 \\
& - 3p_{01}^4 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^4 - p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12}^2 + 7p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12}^2 - 6p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12}^2 \\
& - 8p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12} + 6p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12} - \lambda_4^5 p_{01}^5 p_{12} \lambda_1^5 \lambda_3^5 + 4\lambda_4^5 p_{01}^5 p_{12} \lambda_1^5 \lambda_3^5 - 6\lambda_4^5 p_{01}^5 p_{12}^3 \lambda_1^5 \lambda_3^5 \\
& + 4\lambda_4^5 p_{01}^5 p_{12}^2 \lambda_1^5 \lambda_3^5 - 12\lambda_4^5 p_{01}^4 p_{12}^2 \lambda_1^5 \lambda_3^5 + 12\lambda_4^5 p_{01}^4 p_{12}^2 \lambda_1^5 \lambda_3^5 - 4\lambda_4^5 p_{01}^4 p_{12}^2 \lambda_1^5 \lambda_3^5 - 3p_{01}^3 \lambda_2^4 \lambda_3^5 \lambda_4^3 p_{12}^3 \lambda_1^3 \\
& - 2p_{01}^3 \lambda_2^4 \lambda_3^5 \lambda_4^2 p_{12}^3 \lambda_1^4 - 3p_{01}^2 \lambda_2^5 \lambda_3^5 \lambda_4^2 p_{12}^2 \lambda_1^2 + p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 p_{12} - 4p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^2 p_{12}^2 \lambda_1^3 + 7p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4 p_{12} \lambda_1^4 \\
& + p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^5 p_{12} - p_{01} \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 - p_{01} \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^4 - p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^2 \lambda_1^5 p_{12} + p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^4
\end{aligned}$$

$$\begin{aligned}
& -p_{01}\lambda_2^5\lambda_3^3\lambda_4^2\lambda_1^5 - p_{01}^3\lambda_2^5\lambda_3^3\lambda_4^5\lambda_1^2 + p_{01}^5\lambda_2\lambda_3^5\lambda_4^4p_{12}^5\lambda_1^5 - p_{01}^4\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12}^3 - 4p_{01}^2\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12}^2 \\
& + p_{01}^3\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12}^2 + 3p_{01}^4\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12}^2 - 5p_{01}^3\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12} - 3p_{01}^4\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3p_{12} \\
& + 3p_{01}^2\lambda_2^4\lambda_3^5\lambda_4^3p_{12}^2\lambda_1^3 + 2p_{01}^2\lambda_2^4\lambda_3^5\lambda_4^2p_{12}^2\lambda_1^4 - 4p_{01}^2\lambda_2^4\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^2 + p_{01}\lambda_2^5\lambda_3^4\lambda_4^2\lambda_1^2p_{12} \\
& - p_{01}\lambda_2^5\lambda_3^4\lambda_4^3\lambda_1^3 + 7p_{01}^3\lambda_2^5\lambda_3^5\lambda_4^2p_{12}^2\lambda_1^3 + p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12}^2 - 5p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{12} \\
& - 3p_{01}^2\lambda_2^5\lambda_3^4\lambda_4^2\lambda_1^4p_{12}^2 - 4p_{01}^2\lambda_2^5\lambda_3^5\lambda_1\lambda_4^4 - 3p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12} - 5p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12}^2 \\
& + 7p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12} - 3p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12}^3 - p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12}^3 \\
& + 3p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12}^2 + 3p_{01}^3\lambda_2^3\lambda_3^5\lambda_4^4p_{12}^3\lambda_1^3 - 6p_{01}^3\lambda_2^3\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^3 - 4p_{01}^5\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12} \\
& - 3p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12} - 2p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4 + p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^3p_{12}^4\lambda_1^5 - p_{01}\lambda_2^5\lambda_3^5\lambda_4^4\lambda_4 - p_{01}\lambda_2^5\lambda_3^5\lambda_1^3\lambda_4^2 \\
& - p_{01}\lambda_2^5\lambda_3^5\lambda_1^2\lambda_4^3 - p_{01}\lambda_2^5\lambda_3^5\lambda_1\lambda_4^4 + p_{01}\lambda_2^5\lambda_3^5p_{12}^2\lambda_1^2\lambda_4^3 - 6p_{01}^2\lambda_2^5\lambda_3^5\lambda_1^2\lambda_4^3 - 6p_{01}^2\lambda_2^5\lambda_3^5\lambda_1^3\lambda_4^2 \\
& - 4p_{01}^2\lambda_2^5\lambda_3^5\lambda_1^4\lambda_4 - 3p_{01}^3\lambda_2^5\lambda_3^5\lambda_1^2\lambda_4^3 - 3p_{01}^3\lambda_2^5\lambda_3^5\lambda_1^3\lambda_4^2 + 2p_{01}^2\lambda_2^3\lambda_3^5\lambda_4^3p_{12}^2\lambda_1^4 - p_{01}^3\lambda_2^5\lambda_3^3\lambda_4^2\lambda_1^5 \\
& - 2p_{01}^3\lambda_2^5\lambda_3^3\lambda_4^4\lambda_1^3p_{12}^3 - p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12}^4 - p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^3p_{12}^3 + 2p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12}^2 \\
& - 5p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5p_{12} + 3p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3 + 2p_{01}^2\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4 - 3p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^3\lambda_1^5 + 2p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^4\lambda_1^4 \\
& + 4p_{01}^3\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^3\lambda_1^4 - 6p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^3\lambda_1^4 - 8p_{01}^3\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^4 + 6p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^4 \\
& + 4p_{01}^3\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^4 - p_{01}\lambda_2^5\lambda_3^3\lambda_4^3\lambda_1^4 + p_{01}\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12} - 7p_{01}^2\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12} \\
& - 7p_{01}^4\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^3 - 12p_{01}^3\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^2 - 3p_{01}^3\lambda_2\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^4 + p_{01}^4\lambda_2\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^4 \\
& - p_{01}^5\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^4 - 7p_{01}^4\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^3 - 5p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{12}^2 + 3p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^3p_{12}^2\lambda_1^5 \\
& - p_{01}^4\lambda_2^2\lambda_3^5\lambda_4^3p_{12}^2\lambda_1^5 + 2p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12}^2 + 7p_{01}^3\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{12} + 3p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{12}^2 \\
& + p_{01}^5\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12}^4 - 9p_{01}^4\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12}^2 + 2p_{01}^4\lambda_2^5\lambda_3^3\lambda_4^4\lambda_1^3p_{12}^3 + 9p_{01}^4\lambda_2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^5 \\
& - 4p_{01}^5\lambda_2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^5 + 4p_{01}^5\lambda_2\lambda_3\lambda_4^4\lambda_1^5p_{12} - 6p_{01}^3\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4 + 4p_{01}^4\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4 \\
& - p_{01}^5\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4 + p_{01}^4\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12} + 3p_{01}^3\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^3 + 3p_{01}^2\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^2 \\
& - 3p_{01}^3\lambda_2^4\lambda_3^5\lambda_4^3p_{12}^3\lambda_1^3 + 3p_{01}^3\lambda_2^5\lambda_3^4\lambda_4^2\lambda_1^4p_{12}^2 - 6p_{01}^3\lambda_2^5\lambda_3^4\lambda_4^2\lambda_1^4p_{12} + 3p_{01}^3\lambda_2^5\lambda_3^4\lambda_4^2\lambda_1^4 \\
& + p_{01}\lambda_2^5\lambda_3^5p_{12}^2\lambda_1^4\lambda_4 - p_{01}^2\lambda_2^5\lambda_3^3\lambda_4^3p_{12}^2\lambda_1^5 + 3p_{01}^3\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^3 + 3p_{01}^2\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^2 \\
& + 3p_{01}^2\lambda_2^3\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^3 + p_{01}^2\lambda_2^3\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^2 - 3p_{01}^2\lambda_2^3\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^3 - 2p_{01}^2\lambda_2^2\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^3 \\
& + 2p_{01}^2\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^4 - p_{01}\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5 + 2p_{01}^2\lambda_2^2\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^3 - p_{01}^2\lambda_2^3\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^2 \\
& - 3p_{01}^4\lambda_2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^5 + p_{01}^5\lambda_2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^5 - p_{01}^5\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^4 + 4p_{01}^5\lambda_2^5\lambda_3\lambda_4^4\lambda_1^5p_{12}^3 \\
& - 4\lambda_4^5\lambda_1^5\lambda_2p_{01}^5p_{12} + 12\lambda_4^5p_{01}^3p_{12}^2\lambda_1^5\lambda_3^5 + 6p_{01}^3\lambda_2^5\lambda_3^5\lambda_4^3p_{12}^2\lambda_1^2 + 4p_{01}^3\lambda_2^4\lambda_3^5\lambda_4^2p_{12}^2\lambda_1^4 \\
& + p_{01}\lambda_2^5\lambda_3^2\lambda_4^5\lambda_1^3p_{12} + p_{01}\lambda_2^5\lambda_3^2\lambda_4^4\lambda_1^4p_{12} + 4p_{01}^5\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^3 - 12p_{01}^3\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^2 \\
& + 15p_{01}^4\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^2 - 6p_{01}^5\lambda_2^5\lambda_3\lambda_4^5\lambda_1^4p_{12}^2 - 2p_{01}^2\lambda_2^2\lambda_3^5\lambda_4^4p_{12}^2\lambda_1^4 - 3p_{01}^2\lambda_2\lambda_3^5\lambda_4^5p_{12}^2\lambda_1^4
\end{aligned}$$



---


$$\begin{aligned}
& +3p_{01}^2 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12} \lambda_1^4 - 7p_{01}^2 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 p_{12} + 4p_{01}^2 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 + p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^2 \lambda_1^5 p_{12}^2 \\
& + p_{01}^2 \lambda_2^3 \lambda_3^5 \lambda_4^2 p_{12}^2 \lambda_1^5 - \lambda_4^5 p_{01} p_{12} \lambda_1 \lambda_2^4 \lambda_3^5 - \lambda_4^5 p_{01} p_{12} \lambda_1^2 \lambda_2^3 \lambda_3^5 - \lambda_4^5 p_{01} p_{12} \lambda_1^3 \lambda_2^2 \lambda_3^5 \\
& - \lambda_4^5 p_{01} p_{12} \lambda_1^4 \lambda_2 \lambda_3^5 - \lambda_4^5 p_{01} p_{12} \lambda_1^5 \lambda_3^5 + p_{01}^3 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12}^3 - p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^5 p_{12}^3 \lambda_1^3 \\
& + 2p_{01}^3 \lambda_2^2 \lambda_3^5 \lambda_4^5 p_{12}^2 \lambda_1^3 - p_{01}^2 \lambda_2^3 \lambda_3^5 \lambda_4^2 p_{12} \lambda_1^5 + p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 p_{12} + p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12} \\
& + p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^3 \lambda_1^4 p_{12} - 3p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^2 \lambda_1^5 p_{12} + 2p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 + p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^3 \lambda_1^5 \\
& + p_{01}^5 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^4 - 3p_{01}^4 \lambda_2^2 \lambda_3^5 \lambda_4^3 p_{12}^3 \lambda_1^5 + p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12} + p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^4 p_{12} \\
& + 3p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 - 2p_{01}^4 \lambda_2^2 \lambda_3^5 \lambda_4^4 p_{12} \lambda_1^4 - p_{01}^4 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12}^4 \lambda_1^4 - 3p_{01}^3 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12}^3 \lambda_1^4 \\
& + 3p_{01}^4 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12}^3 \lambda_1^4 + 6p_{01}^3 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12}^2 \lambda_1^4 + p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^2 \lambda_1^4 p_{12} - 3p_{01}^2 \lambda_2^5 \lambda_3^5 \lambda_4 p_{12}^2 \lambda_1^4 \\
& + p_{01}^3 \lambda_2^5 \lambda_3^5 \lambda_4^2 p_{12}^3 \lambda_1^3 - 2p_{01}^2 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^4 - 3p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 p_{12} - p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^5 \lambda_1^2 \\
& + p_{01}^3 \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 p_{12}^3 - 6p_{01}^3 \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 p_{12} + p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^5 p_{12} - p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^5 \lambda_1 \\
& + 3p_{01}^3 \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 + p_{01}^4 \lambda_2^5 \lambda_3^4 \lambda_4^3 \lambda_1^3 - 3p_{01}^4 \lambda_2 \lambda_3^5 \lambda_4^5 p_{12}^2 \lambda_1^4 - p_{01} \lambda_2^5 \lambda_3^4 \lambda_4^2 \lambda_1^4 \\
& - p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^5 p_{12} + p_{01}^2 \lambda_2^4 \lambda_3^5 \lambda_4 p_{12}^2 \lambda_1^5 - 2p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^2 p_{12}^2 \lambda_1^5 + p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^2 p_{12} \lambda_1^5 \\
& + p_{01}^4 \lambda_2^3 \lambda_3^5 \lambda_4^3 p_{12} \lambda_1^4 + 7p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 6p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12}^2 - 8p_{01}^3 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12} \\
& + p_{01}^3 \lambda_2^3 \lambda_3^5 \lambda_4^2 p_{12}^3 \lambda_1^5 + 2p_{01}^2 \lambda_2^5 \lambda_3^3 \lambda_4^2 \lambda_1^5 - p_{01} \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 - p_{01}^3 \lambda_2^5 \lambda_3^2 \lambda_4^4 \lambda_1^4 p_{12} \\
& + 3p_{01}^3 \lambda_2 \lambda_3^5 \lambda_4 p_{12} \lambda_1^5 + p_{01}^4 \lambda_2^5 \lambda_3 \lambda_4^5 \lambda_1^4 p_{12}^4 - p_{01} \lambda_2^5 \lambda_3^4 \lambda_4 \lambda_1^5 - \lambda_4^5 \lambda_1^5 \lambda_2^5 - 5p_{01}^3 \lambda_2^5 \lambda_3^5 \lambda_4^2 p_{12}^2 \lambda_1^3 \\
& - 3p_{01}^2 \lambda_2^5 \lambda_3^4 \lambda_4^4 \lambda_1^2 p_{12}^2 - 3p_{01}^3 \lambda_2^5 \lambda_3^5 \lambda_4^3 p_{12}^2 \lambda_1^2 + 6p_{01}^4 \lambda_2^5 \lambda_3^3 \lambda_4^4 \lambda_1^3 p_{12} - p_{01}^4 \lambda_2^5 \lambda_3^2 \lambda_4^5 \lambda_1^3 p_{12}^3 )
\end{aligned}$$

$$\begin{aligned}
 & \mu_6(p_{01}, p_{12}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
 = & \frac{720}{\lambda_4^6(p_{01} - p_{01}p_{12} - 1)^6 \lambda_1^6 \lambda_2^6 \lambda_3^6} \cdot \\
 & \left( p_{01} \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 p_{12} + p_{01}^2 \lambda_4 \lambda_2^5 \lambda_3^6 p_{12}^2 \lambda_1^6 - p_{01}^2 \lambda_4 \lambda_2^5 \lambda_3^6 p_{12} \lambda_1^6 - p_{01} \lambda_4 \lambda_2^6 \lambda_3^5 \lambda_1^6 \right. \\
 & + p_{01}^3 \lambda_2^6 \lambda_3^3 \lambda_4^3 \lambda_1^6 p_{12}^3 + p_{01}^4 \lambda_4^6 \lambda_2^6 \lambda_3^2 \lambda_1^4 p_{12}^4 + 3p_{01}^2 \lambda_4^6 \lambda_2^6 \lambda_3^2 \lambda_1^4 p_{12}^2 + 3p_{01}^2 \lambda_4^3 \lambda_2^6 \lambda_3^4 \lambda_1^5 p_{12} \\
 & + 3p_{01}^3 \lambda_4^6 \lambda_2^6 \lambda_3^2 \lambda_1^4 p_{12}^3 - p_{01}^5 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^5 - 2p_{01}^4 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^4 + 2p_{01}^2 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^2 \\
 & + p_{01}^5 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^6 - 3p_{01}^2 \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 + 3p_{01}^3 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^6 - 3p_{01}^4 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^6 \\
 & - p_{01}^6 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 + 4p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 p_{12} - p_{01}^2 \lambda_4 \lambda_2^6 \lambda_3^5 \lambda_1^6 p_{12} + 4p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 p_{12}^3 \\
 & + 4p_{01}^4 \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 + p_{01} \lambda_2^6 \lambda_3^6 p_{12} \lambda_1^6 - 3p_{01}^4 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_1^3 \lambda_4^3 - 6p_{01}^3 \lambda_2^6 \lambda_3^6 \lambda_1^4 \lambda_4^2 \\
 & - 6p_{01}^3 \lambda_2^6 \lambda_3^6 \lambda_1^2 \lambda_4^4 + 21p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12} \lambda_1^3 \lambda_4^3 + 15p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12} \lambda_1^4 \lambda_4^2 + 12p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12} \lambda_4^4 \lambda_1^2 \\
 & - 3p_{01}^3 \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12} - 15p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_1^3 \lambda_4^3 + 3p_{01}^3 \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12}^3 + 3p_{01}^3 \lambda_4^3 \lambda_2^6 \lambda_3^5 \lambda_1^4 p_{12}^3 \\
 & + 4p_{01}^3 \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 p_{12}^2 + 2p_{01}^2 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^2 \lambda_1^5 - 3p_{01}^3 \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12}^2 - 3p_{01}^3 \lambda_4^3 \lambda_2^6 \lambda_3^5 \lambda_1^4 p_{12}^2 \\
 & + 4p_{01}^3 \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 p_{12}^2 - 8p_{01}^3 \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 p_{12} - 6p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_4^4 \lambda_1^2 - 7p_{01}^2 \lambda_4^6 \lambda_2^6 \lambda_3^2 \lambda_1^4 p_{12} \\
 & - 5p_{01}^2 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12} + 3p_{01}^2 \lambda_4^4 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^4 - 3p_{01}^2 \lambda_4^4 \lambda_2^4 \lambda_3^6 p_{12} \lambda_1^4 - p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 \\
 & - 2p_{01}^2 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12} \lambda_1^5 + 9p_{01}^4 \lambda_4^5 \lambda_2^6 \lambda_3^4 \lambda_1^3 p_{12} + 5p_{01}^4 \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 p_{12} + 9p_{01}^4 \lambda_4^3 \lambda_2^6 \lambda_3^4 \lambda_1^5 p_{12} \\
 & + p_{01} \lambda_2^6 \lambda_3^3 \lambda_4^3 \lambda_1^3 p_{12} - 6p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 p_{12}^2 + 4p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 p_{12} - 6p_{01}^3 \lambda_4^3 \lambda_2^5 \lambda_3^6 p_{12}^2 \lambda_1^4 \\
 & - 12p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_1^4 \lambda_4^2 - 30\lambda_4^6 p_{01}^5 p_{12}^3 \lambda_1^6 \lambda_3^6 + 3p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12}^3 \lambda_4^3 \lambda_1^3 + 3p_{01}^3 \lambda_2^6 \lambda_3^6 p_{12}^3 \lambda_4^4 \lambda_1^2 \\
 & - p_{01}^4 \lambda_2^6 \lambda_3^6 \lambda_1^3 \lambda_4^3 - p_{01}^2 \lambda_2^3 \lambda_3^6 \lambda_4^3 p_{12} \lambda_1^6 - 5\lambda_4^6 p_{01}^5 p_{12}^5 \lambda_1^6 \lambda_3^6 + p_{01} \lambda_2^6 \lambda_3^3 \lambda_4^5 \lambda_1^4 p_{12} \\
 & - 5p_{01}^2 \lambda_2^6 \lambda_3^3 \lambda_4^3 \lambda_1^6 p_{12} + p_{01}^2 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 - 30\lambda_4^6 p_{01}^4 p_{12}^2 \lambda_1^6 \lambda_3^6 + 20\lambda_4^6 p_{01}^5 p_{12}^4 \lambda_1^6 \lambda_3^6 \\
 & - 10\lambda_4^6 p_{01}^6 p_{12}^4 \lambda_1^6 \lambda_3^6 - 10\lambda_4^6 p_{01}^4 p_{12}^4 \lambda_1^6 \lambda_3^6 + 30\lambda_4^6 p_{01}^4 p_{12}^3 \lambda_1^6 \lambda_3^6 - 5\lambda_4^6 p_{01}^2 p_{12}^2 \lambda_1^6 \lambda_3^6 \\
 & + 5\lambda_4^6 p_{01}^2 p_{12} \lambda_1^6 \lambda_3^6 + 20\lambda_4^6 p_{01}^3 p_{12}^2 \lambda_1^6 \lambda_3^6 - 10\lambda_4^6 p_{01}^3 p_{12}^3 \lambda_1^6 \lambda_3^6 - 10\lambda_4^6 p_{01}^3 p_{12} \lambda_1^6 \lambda_3^6 \\
 & - \lambda_4^6 \lambda_1^6 \lambda_2^6 p_{01}^5 p_{12}^5 - 5\lambda_4^6 \lambda_1^6 \lambda_2^6 p_{01}^4 p_{12}^4 + 7p_{01}^2 \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 p_{12} + 10\lambda_4^6 p_{01}^6 p_{12}^3 \lambda_1^6 \lambda_3^6 \\
 & - p_{01} \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 - p_{01}^5 \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 + 4p_{01}^2 \lambda_4^4 \lambda_2^6 \lambda_3^2 \lambda_1^6 - p_{01}^5 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^5 \lambda_1^5 \\
 & - 6p_{01}^5 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^3 \lambda_1^5 + 3p_{01}^3 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^3 \lambda_1^5 + 4p_{01}^5 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^2 \lambda_1^5 - 6p_{01}^3 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12}^2 \lambda_1^5 \\
 & - 10p_{01}^6 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^2 - 5p_{01}^6 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^4 + 10p_{01}^6 \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_1^5 p_{12}^3 - 3p_{01}^4 \lambda_4^3 \lambda_2^5 \lambda_3^6 p_{12}^3 \lambda_1^4 \\
 & + 3p_{01}^4 \lambda_4^3 \lambda_2^5 \lambda_3^6 p_{12}^2 \lambda_1^4 + 2p_{01}^5 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^4 + 12p_{01}^5 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^2 - 2p_{01}^4 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^4 \\
 & + 11p_{01}^4 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^3 - 21p_{01}^4 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^2 + 17p_{01}^4 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12} + 9p_{01}^3 \lambda_2^6 \lambda_3^3 \lambda_4^4 \lambda_1^5 p_{12}^2
 \end{aligned}$$



$$\begin{aligned}
& -3p_{01}^4 \lambda_4^5 \lambda_2^6 \lambda_3^4 \lambda_1^3 - 2p_{01}^4 \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 - 3p_{01}^4 \lambda_4^3 \lambda_2^6 \lambda_3^4 \lambda_1^5 + p_{01} \lambda_4^6 \lambda_2^6 \lambda_3^5 \lambda_1 p_{12} - p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^5 \lambda_1^5 \\
& + 4p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^4 \lambda_1^5 - 6p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^5 - 6p_{01}^3 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^5 + 4p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^5 \\
& + 12p_{01}^3 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^5 + 12p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^5 - 4p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^4 \lambda_1^5 - 12p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^5 \\
& + 10p_{01}^6 \lambda_4^5 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_1^6 + 26p_{01}^3 \lambda_4^5 \lambda_2^6 \lambda_3^6 p_{12} \lambda_1^6 - 34p_{01}^4 \lambda_4^5 \lambda_2^6 \lambda_3^6 p_{12} \lambda_1^6 + 42p_{01}^4 \lambda_4^5 \lambda_2^6 \lambda_3^6 p_{12}^2 \lambda_1^6 \\
& + 5p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^6 + 10p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^4 \lambda_1^6 + p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^6 \lambda_1^6 - 5p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^5 \lambda_1^6 \\
& - 10p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^6 + 4p_{01}^5 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^5 \lambda_1^6 - 16p_{01}^5 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^4 \lambda_1^6 + 24p_{01}^5 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^6 \\
& + 4p_{01}^3 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^6 - 16p_{01}^5 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^6 - 8p_{01}^3 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^6 - 18p_{01}^4 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^3 \lambda_1^6 \\
& + 6p_{01}^4 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^4 \lambda_1^6 + 18p_{01}^4 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12}^2 \lambda_1^6 - p_{01}^6 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12} \lambda_1^6 + 4p_{01}^5 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12} \lambda_1^6 \\
& - 4p_{01}^3 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12} \lambda_1^6 - p_{01} \lambda_4^6 \lambda_2^5 \lambda_3^5 \lambda_1 - p_{01} \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 - p_{01} \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 - p_{01} \lambda_4^3 \lambda_2^6 \lambda_3^5 \lambda_1^4 \\
& - p_{01} \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 + p_{01} \lambda_4 \lambda_2^6 \lambda_3^5 \lambda_1^6 p_{12} + p_{01} \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 p_{12} + p_{01} \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12} - 4p_{01}^2 \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 p_{12}^2 \\
& - 6p_{01}^2 \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12}^2 - 6p_{01}^2 \lambda_4^3 \lambda_2^6 \lambda_3^5 \lambda_1^4 p_{12}^2 - 4p_{01}^2 \lambda_4^2 \lambda_2^6 \lambda_3^5 \lambda_1^5 p_{12}^2 + 7p_{01}^2 \lambda_4^5 \lambda_2^6 \lambda_3^5 \lambda_1^2 p_{12} \\
& + 11p_{01}^2 \lambda_4^4 \lambda_2^6 \lambda_3^5 \lambda_1^3 p_{12} + 6p_{01}^4 \lambda_4^3 \lambda_2^4 \lambda_3^6 p_{12}^3 \lambda_1^5 - 2p_{01}^4 \lambda_4^3 \lambda_2^4 \lambda_3^6 p_{12}^4 \lambda_1^5 - 6p_{01}^4 \lambda_4^3 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^5 \\
& - 6p_{01}^4 \lambda_4^5 \lambda_2 \lambda_3^6 p_{12} \lambda_1^6 + 2p_{01}^2 \lambda_4^3 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^5 - p_{01}^2 \lambda_4^2 \lambda_2^4 \lambda_3^6 p_{12} \lambda_1^6 + p_{01}^2 \lambda_4^2 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^6 \\
& + p_{01} \lambda_4^2 \lambda_2^6 \lambda_3^4 \lambda_1^6 p_{12} - p_{01} \lambda_4 \lambda_2^6 \lambda_3^4 \lambda_1^2 - p_{01} \lambda_4^5 \lambda_2^6 \lambda_3^4 \lambda_1^3 - p_{01} \lambda_4^4 \lambda_2^6 \lambda_3^4 \lambda_1^4 - p_{01} \lambda_4^3 \lambda_2^6 \lambda_3^4 \lambda_1^5 \\
& + 2p_{01}^2 \lambda_4^2 \lambda_2^6 \lambda_3^4 \lambda_1^6 - p_{01}^3 \lambda_4^2 \lambda_2^6 \lambda_3^4 \lambda_1^6 - 2p_{01}^2 \lambda_4^2 \lambda_2^5 \lambda_3^6 p_{12} \lambda_1^5 + p_{01}^2 \lambda_2^3 \lambda_3^6 \lambda_4^3 p_{12}^2 \lambda_1^6 - p_{01} \lambda_2^6 \lambda_3^6 \lambda_1^5 \lambda_4 \\
& - p_{01} \lambda_2^6 \lambda_3^6 \lambda_1^4 \lambda_4^2 - p_{01} \lambda_2^6 \lambda_3^6 \lambda_1^2 \lambda_4^4 - p_{01} \lambda_2^6 \lambda_3^6 \lambda_1 \lambda_4^5 + 9p_{01}^4 \lambda_4^4 \lambda_2^4 \lambda_3^6 p_{12}^3 \lambda_1^4 - 3p_{01}^4 \lambda_4^4 \lambda_2^4 \lambda_3^6 p_{12}^4 \lambda_1^4 \\
& - 9p_{01}^4 \lambda_4^4 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^4 - p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12} \lambda_1^5 - 6p_{01}^3 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12} \lambda_1^5 + 4p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 p_{12} \lambda_1^5 \\
& - 9p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^4 + 26p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^3 + 21p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12} - 22p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^3 \\
& - 34p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12} + 26p_{01}^3 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12} - p_{01}^3 \lambda_4^6 \lambda_2^4 \lambda_3^2 \lambda_1^2 + 26p_{01}^5 \lambda_4^5 \lambda_2^6 \lambda_3^6 \lambda_1 p_{12}^3 \\
& + 21p_{01}^5 \lambda_4^5 \lambda_2^6 \lambda_3^6 \lambda_1 p_{12} + 2p_{01}^5 \lambda_4^5 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^5 + 6p_{01}^3 \lambda_4^5 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^5 - 6p_{01}^4 \lambda_4^5 \lambda_2^2 \lambda_3^6 p_{12} \lambda_1^5 \\
& - p_{01}^3 \lambda_4^2 \lambda_2^6 \lambda_3^4 \lambda_1^6 p_{12}^2 + 2p_{01}^3 \lambda_4^2 \lambda_2^6 \lambda_3^4 \lambda_1^6 p_{12} - 22p_{01}^3 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^2 + 5p_{01}^3 \lambda_4^5 \lambda_2^6 \lambda_3^4 \lambda_1^3 \\
& + 5p_{01}^3 \lambda_4^3 \lambda_2^6 \lambda_3^4 \lambda_1^5 - 2p_{01}^3 \lambda_4^2 \lambda_2^4 \lambda_3^6 p_{12}^2 \lambda_1^6 + p_{01}^3 \lambda_4^2 \lambda_2^4 \lambda_3^6 p_{12} \lambda_1^6 + p_{01}^3 \lambda_4^2 \lambda_2^4 \lambda_3^6 p_{12}^3 \lambda_1^6 - p_{01} \lambda_2^6 \lambda_3^6 \lambda_1^3 \lambda_4^3 \\
& + p_{01} \lambda_4^6 \lambda_2^6 \lambda_3^6 \lambda_1 p_{12} - p_{01} \lambda_4 \lambda_2^6 \lambda_3^2 \lambda_4^4 - p_{01} \lambda_4^5 \lambda_2^6 \lambda_3^2 \lambda_4^5 + 6p_{01}^5 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^3 \lambda_1^6 + p_{01} \lambda_2^6 \lambda_3^6 \lambda_4^3 \lambda_1^6 p_{12} \\
& - p_{01} \lambda_2^6 \lambda_3^6 \lambda_4^5 \lambda_1^4 - p_{01} \lambda_2^6 \lambda_3^6 \lambda_4^4 \lambda_1^5 + 3p_{01}^2 \lambda_2^6 \lambda_3^6 \lambda_4^3 \lambda_1^6 + p_{01}^4 \lambda_2^6 \lambda_3^6 \lambda_4^3 \lambda_1^6 + 3p_{01}^3 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^3 \lambda_1^6 \\
& - 3p_{01}^3 \lambda_2^6 \lambda_3^6 \lambda_4^3 \lambda_1^6 - p_{01}^5 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12} \lambda_1^5 + 3p_{01}^3 \lambda_4^4 \lambda_2^3 \lambda_3^6 p_{12} \lambda_1^5 + p_{01}^5 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^5 \lambda_1^6 \\
& - 4p_{01}^5 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^2 \lambda_1^6 - 6p_{01}^3 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^2 \lambda_1^6 - 9p_{01}^4 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^3 \lambda_1^6 + 3p_{01}^4 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^4 \lambda_1^6 \\
& + 9p_{01}^4 \lambda_4^4 \lambda_2^2 \lambda_3^6 p_{12}^2 \lambda_1^6 - 34p_{01}^5 \lambda_4^5 \lambda_2^6 \lambda_3^6 \lambda_1 p_{12}^2 + 42p_{01}^4 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^2 - 34p_{01}^5 \lambda_4^6 \lambda_2 \lambda_3^6 \lambda_1^5 p_{12}^2
\end{aligned}$$



$$\begin{aligned}
& +p_{01}\lambda_2^6\lambda_3^6p_{12}\lambda_1^3\lambda_4^3 + p_{01}\lambda_2^6\lambda_3^6p_{12}\lambda_1^4\lambda_4^2 + p_{01}\lambda_2^6\lambda_3^6p_{12}\lambda_1^5\lambda_4 + p_{01}\lambda_2^6\lambda_3^6p_{12}\lambda_4^5\lambda_1 \\
& +9p_{01}^4\lambda_4^4\lambda_2^6\lambda_3^5\lambda_1^3p_{12}^2 + 9p_{01}^4\lambda_4^3\lambda_2^6\lambda_3^5\lambda_1^4p_{12}^2 - 9p_{01}^4\lambda_4^4\lambda_2^6\lambda_3^5\lambda_1^3p_{12} - 9p_{01}^4\lambda_4^3\lambda_2^6\lambda_3^5\lambda_1^4p_{12} \\
& +3p_{01}^2\lambda_4^5\lambda_2^3\lambda_3^6p_{12}^2\lambda_1^4 + 2p_{01}^2\lambda_4^6\lambda_2^3\lambda_3^6p_{12}\lambda_1^3 - 3p_{01}^2\lambda_4^5\lambda_2^3\lambda_3^6p_{12}\lambda_1^4 - 3p_{01}^2\lambda_4^6\lambda_2^2\lambda_3^6p_{12}^2\lambda_1^4 \\
& +2p_{01}^2\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^2\lambda_1^5 + 3p_{01}^2\lambda_4^6\lambda_2^2\lambda_3^6p_{12}\lambda_1^4 - 2p_{01}^2\lambda_4^5\lambda_2^2\lambda_3^6p_{12}\lambda_1^5 - 4p_{01}^2\lambda_4^6\lambda_2\lambda_3^6p_{12}^2\lambda_1^5 \\
& +4p_{01}^2\lambda_4^6\lambda_2\lambda_3^6p_{12}\lambda_1^5 - p_{01}\lambda_4^6\lambda_2\lambda_3\lambda_1^5 + 5p_{01}^2\lambda_4^5\lambda_2^6\lambda_3\lambda_1^6 + p_{01}^6\lambda_4^5\lambda_2^6\lambda_3\lambda_1^6 - 5p_{01}^5\lambda_4^5\lambda_2^6\lambda_3\lambda_1^6 \\
& -10p_{01}^3\lambda_4^5\lambda_2^6\lambda_3\lambda_1^6 + 10p_{01}^4\lambda_4^5\lambda_2^6\lambda_3\lambda_1^6 - 9p_{01}^2\lambda_4^6\lambda_2^6\lambda_3\lambda_1^5p_{12} + 9p_{01}^4\lambda_4^5\lambda_2^3\lambda_3^6p_{12}^2\lambda_1^4 \\
& +2p_{01}^5\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^5\lambda_1^5 - 8p_{01}^5\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^4\lambda_1^5 + 12p_{01}^5\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^3\lambda_1^5 + 6p_{01}^3\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^3\lambda_1^5 \\
& -8p_{01}^5\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^2\lambda_1^5 - 12p_{01}^3\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^2\lambda_1^5 - 18p_{01}^4\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^3\lambda_1^5 + 6p_{01}^4\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^4\lambda_1^5 \\
& +18p_{01}^4\lambda_4^5\lambda_2^2\lambda_3^6p_{12}^2\lambda_1^5 - 3p_{01}^3\lambda_4^6\lambda_2^2\lambda_3^6p_{12}\lambda_1^4 + p_{01}^4\lambda_4^6\lambda_2^2\lambda_3^6p_{12}\lambda_1^4 + 4p_{01}^3\lambda_4^5\lambda_2^6\lambda_3^5\lambda_1^2 \\
& +3p_{01}^3\lambda_4^3\lambda_2^6\lambda_3^5\lambda_1^4 + 4p_{01}^3\lambda_4^2\lambda_2^6\lambda_3^5\lambda_1^5 - 3p_{01}^2\lambda_4^2\lambda_2^6\lambda_3^4\lambda_1^6p_{12} + p_{01}\lambda_4^6\lambda_2^6\lambda_3^4\lambda_1^2p_{12} + p_{01}\lambda_4^5\lambda_2^6\lambda_3^4\lambda_1^3p_{12} \\
& +p_{01}\lambda_4^4\lambda_2^6\lambda_3^4\lambda_1^4p_{12} + p_{01}\lambda_4^3\lambda_2^6\lambda_3^4\lambda_1^5p_{12} + 2p_{01}^2\lambda_4^6\lambda_2\lambda_3^4\lambda_1^2 - p_{01}^2\lambda_4^5\lambda_2^6\lambda_3^4\lambda_1^3 - 2p_{01}^2\lambda_4^4\lambda_2^6\lambda_3^4\lambda_1^4 \\
& -p_{01}^2\lambda_4^3\lambda_2^6\lambda_3^4\lambda_1^5 + 3p_{01}^2\lambda_4^5\lambda_2^6\lambda_3^4\lambda_1^3p_{12} + p_{01}^2\lambda_4^6\lambda_2^6\lambda_3^4\lambda_1^2p_{12} - 2p_{01}^2\lambda_4^5\lambda_2^6\lambda_3^4\lambda_1^3p_{12}^2 \\
& -2p_{01}^2\lambda_4^3\lambda_2^6\lambda_3^4\lambda_1^5p_{12}^2 - 3p_{01}^2\lambda_4^6\lambda_2^6\lambda_3^4\lambda_1^2p_{12} + 5p_{01}^2\lambda_4^4\lambda_2^6\lambda_3^4\lambda_1^4p_{12} + 2p_{01}^4\lambda_4^3\lambda_2^4\lambda_3^6p_{12}\lambda_1^5 \\
& +17p_{01}^4\lambda_4^6\lambda_2^3\lambda_3^5\lambda_1^4p_{12} - 6p_{01}^5\lambda_4^6\lambda_2^2\lambda_3^2\lambda_1^4p_{12}^2 + p_{01}\lambda_2^6\lambda_3^6p_{12}\lambda_4^4\lambda_1^2 - p_{01}^3\lambda_4^6\lambda_2^3\lambda_3^6p_{12}\lambda_1^3 \\
& -3p_{01}^2\lambda_4^4\lambda_2^6\lambda_3^4\lambda_1^4p_{12}^2 + 3p_{01}^3\lambda_4^4\lambda_2^6\lambda_3^5\lambda_1^3 - 21p_{01}^4\lambda_2^6\lambda_3^3\lambda_4^5\lambda_1^4p_{12}^2 + 30\lambda_4^6\lambda_1^6\lambda_2^6p_{01}^4p_{12}^3 \\
& -4p_{01}^5\lambda_4^4\lambda_2^2\lambda_3^6p_{12}^4\lambda_1^6 + 6p_{01}^3\lambda_4^4\lambda_2^6\lambda_3^4\lambda_1^4 + 22p_{01}^5\lambda_4^5\lambda_2^6\lambda_3^2\lambda_1^5p_{12}^2 + 11p_{01}^3\lambda_4^3\lambda_2^6\lambda_3^4\lambda_1^5p_{12}^2 \\
& +11p_{01}^2\lambda_4^3\lambda_2^6\lambda_3^5\lambda_1^4p_{12} - 5\lambda_4^6p_{01}^6p_{12}^2\lambda_1^6\lambda_3^6 + 10\lambda_4^6p_{01}^4p_{12}\lambda_1^6\lambda_3^6 + 20\lambda_4^6p_{01}^5p_{12}^2\lambda_1^6\lambda_3^6 \\
& +2p_{01}^2\lambda_2^6\lambda_3^3\lambda_4^3\lambda_1^6p_{12}^2 + p_{01}\lambda_2^6\lambda_3^3\lambda_4^4\lambda_1^5p_{12} + 4p_{01}^5\lambda_4^4\lambda_2^3\lambda_3^6p_{12}^4\lambda_1^5)
\end{aligned}$$

---

2. Jacobian determinant for Example 5.2 continued, in Section 4.3.2.

$$\det(\mathbf{H}) = \frac{2p_{01}^3\alpha(p_{01}-1)(\beta-\lambda_1)}{\lambda_1^8\lambda_2^{10}\beta^{14}}G(p_{01}, \lambda_1, \lambda_2, \alpha, \beta)$$

where

$$\begin{aligned} & G(p_{01}, \lambda_1, \lambda_2, \alpha, \beta) \\ = & \left( -6180\alpha^4\lambda_1^4\beta^3\lambda_2^6 + 1720\alpha^4\lambda_1^4\beta^2\lambda_2^7 + 21\alpha^4\lambda_1^4\lambda_2^8\beta - 1920\alpha^4\lambda_1^5\lambda_2^4\beta^4 + 2280\alpha^4\lambda_1^5\beta^3\lambda_2^5 \right. \\ & - 596\alpha^4\lambda_1^5\lambda_2^6\beta^2 - 24\alpha^4\lambda_1^5\beta\lambda_2^7 + 8640\lambda_1^4\beta^9 + 960\alpha^4\lambda_1^4\lambda_2^5\beta^4 + 5760\alpha^4\lambda_1^4\lambda_2^4\beta^5 \\ & - 12000\alpha^4\lambda_1^2\beta^4\lambda_2^7 + 300\alpha^4\lambda_1^2\beta^3\lambda_2^8 - 18720\alpha^4\lambda_1^3\lambda_2^5\beta^5 + 9840\alpha^4\lambda_1^3\lambda_2^6\beta^4 + 3600\alpha^4\lambda_1^3\beta^3\lambda_2^7 \\ & - 1124\alpha^4\lambda_1^3\beta^2\lambda_2^8 - 34560\lambda_1\beta^9\lambda_2^3 + 51840\lambda_1^2\beta^9\lambda_2^2 - 34560\lambda_1^3\beta^9\lambda_2 - 72\alpha^6\lambda_1^4\lambda_2^8\beta \\ & + 64\alpha^6\lambda_1^4\beta^2\lambda_2^7 + 384\alpha^6\lambda_1^4\beta^3\lambda_2^6 + 64\alpha^6\lambda_1^3\beta^2\lambda_2^8 - 672\alpha^6\lambda_1^3\beta^3\lambda_2^7 + 16\alpha^7\lambda_1^5\beta\lambda_2^7 \\ & + 3\alpha^8\lambda_1^4\lambda_2^8\beta - 7200\alpha^4\lambda_1\beta^5\lambda_2^7 + 3120\alpha^4\lambda_1\beta^4\lambda_2^8 + 19440\alpha^4\lambda_1^2\lambda_2^6\beta^5 - 8\alpha^7\lambda_1^4\lambda_2^8\beta \\ & - 48\alpha^7\lambda_1^4\beta^2\lambda_2^7 + 48\alpha^7\lambda_1^3\beta^2\lambda_2^8 - 128\alpha^6\lambda_1^5\lambda_2^6\beta^2 + 72\alpha^6\lambda_1^5\beta\lambda_2^7 - 98\alpha^5\lambda_1^4\lambda_2^8\beta \\ & + 624\alpha^5\lambda_1^5\beta^3\lambda_2^5 - 552\alpha^5\lambda_1^5\lambda_2^6\beta^2 + 88\alpha^5\lambda_1^5\beta\lambda_2^7 + 288\alpha^6\lambda_1^2\beta^3\lambda_2^8 + 984\alpha^5\lambda_1^4\beta^2\lambda_2^7 \\ & - 9792\lambda_1^2\lambda_2^6\beta^5 + 29376\lambda_1^4\beta^6\lambda_2^3 - 5184\lambda_1^3\lambda_2^6\beta^4 + 3456\lambda_1^4\lambda_2^5\beta^4 - 864\lambda_1^5\lambda_2^4\beta^4 \\ & - 3456\alpha^5\lambda_1^2\beta^4\lambda_2^7 + 984\alpha^5\lambda_1^2\beta^3\lambda_2^8 + 4608\alpha^5\lambda_1^3\lambda_2^6\beta^4 - 1344\alpha^5\lambda_1^3\beta^3\lambda_2^7 - 432\alpha^5\lambda_1^3\beta^2\lambda_2^8 \\ & - 1872\alpha^5\lambda_1^4\lambda_2^5\beta^4 + 720\alpha^5\lambda_1\beta^4\lambda_2^8 - 11520\alpha^3\lambda_1^4\beta^6\lambda_2^3 + 58560\alpha^3\lambda_1^2\lambda_2^6\beta^5 + 744\alpha^3\lambda_1^4\beta^2\lambda_2^7 \\ & - 104\alpha^3\lambda_1^5\beta\lambda_2^7 - 27840\alpha^3\lambda_1\beta^5\lambda_2^7 - 1920\alpha^3\lambda_1^4\lambda_2^4\beta^5 - 36480\alpha^3\lambda_1^3\lambda_2^5\beta^5 + 106\alpha^3\lambda_1^4\lambda_2^8\beta \\ & - 5760\alpha^3\lambda_1^5\lambda_2^4\beta^4 + 20880\alpha^3\lambda_1^4\lambda_2^5\beta^4 - 11520\alpha^3\lambda_1^3\lambda_2^6\beta^4 + 8640\beta^9\lambda_2^4 - 23040\beta^8\lambda_2^5 \\ & + 144\alpha^3\lambda_1^5\lambda_2^6\beta^2 - 7200\alpha^3\lambda_1^2\beta^4\lambda_2^7 + 21888\lambda_1^3\lambda_2^5\beta^5 - 16992\lambda_1^4\lambda_2^4\beta^5 - 1152\lambda_1\beta^5\lambda_2^7 \\ & - 12960\lambda_1^4\beta^7\lambda_2^2 + 3456\lambda_1^2\beta^4\lambda_2^7 - 10608\alpha^3\lambda_1^4\beta^3\lambda_2^6 + 10272\alpha^3\lambda_1^3\beta^3\lambda_2^7 + 2112\alpha^3\lambda_1^5\beta^3\lambda_2^5 \\ & + 3840\alpha^3\lambda_2^3\beta^5\lambda_1^5 - 1776\alpha^3\lambda_1^2\beta^3\lambda_2^8 + 3600\alpha^3\lambda_1\beta^4\lambda_2^8 - 888\alpha^3\lambda_1^3\beta^2\lambda_2^8 + 46080\alpha^3\lambda_2^4\beta^6\lambda_1^3 \\ & - 60480\alpha^3\lambda_1^2\beta^6\lambda_2^5 + 31680\alpha^3\lambda_1\beta^6\lambda_2^6 - 7008\alpha^2\lambda_1^5\lambda_2^4\beta^4 + 21600\beta^7\lambda_2^6 + 28992\alpha^2\lambda_1^4\lambda_2^5\beta^4 \\ & - 31008\alpha^2\lambda_1^3\lambda_2^6\beta^4 + 292\alpha^2\lambda_1^5\lambda_2^6\beta^2 - 5760\alpha^2\lambda_1^4\beta^6\lambda_2^3 + 36000\alpha^2\lambda_1^2\lambda_2^6\beta^5 - 56\alpha^2\lambda_1^4\beta^2\lambda_2^7 \\ & - 48\alpha^2\lambda_1^5\beta\lambda_2^7 - 33120\alpha^2\lambda_1^4\lambda_2^4\beta^5 + 8640\alpha^2\lambda_1^3\lambda_2^5\beta^5 - 8640\beta^6\lambda_2^7 + 48\alpha^2\lambda_1^4\lambda_2^8\beta \\ & - 30240\alpha^2\lambda_1\beta^5\lambda_2^7 + 17280\alpha^2\lambda_1^4\beta^7\lambda_2^2 + 9696\alpha^2\lambda_1^2\beta^4\lambda_2^7 - 6156\alpha^2\lambda_1^4\beta^3\lambda_2^6 \\ & \left. - 34560\lambda_2^3\lambda_1^3\beta^7 + 4608\lambda_2^3\beta^5\lambda_1^5 + 74880\lambda_2^2\beta^8\lambda_1^3 - 126720\lambda_1^2\beta^8\lambda_2^3 + 89280\lambda_1\beta^8\lambda_2^4 \right) \end{aligned}$$

$$\begin{aligned}
&+888\alpha^2\lambda_1^5\beta^3\lambda_2^5 - 11520\beta^8\lambda_1^4\lambda_2 - 69120\alpha^2\lambda_2^3\lambda_1^3\beta^7 - 77760\lambda_1\beta^7\lambda_2^5 + 25056\lambda_1\beta^6\lambda_2^6 \\
&-13824\lambda_1^2\beta^6\lambda_2^5 + 95040\lambda_1^2\beta^7\lambda_2^4 + 8640\lambda_1^5\beta^7\lambda_2 - 9504\lambda_1^5\beta^6\lambda_2^2 + 11520\alpha^2\lambda_2^3\beta^5\lambda_1^5 \\
&+80640\alpha^2\lambda_2^4\beta^6\lambda_1^3 - 5760\alpha^2\lambda_1^5\beta^6\lambda_2^2 + 103680\alpha^2\lambda_1^2\beta^7\lambda_2^4 - 141120\alpha^2\lambda_1^2\beta^6\lambda_2^5 \\
&+95040\alpha^2\lambda_1\beta^6\lambda_2^6 - 69120\alpha^2\lambda_1\beta^7\lambda_2^5 - 22464\lambda_2^4\beta^6\lambda_1^3 - 236\alpha^2\lambda_1^3\beta^2\lambda_2^8 - 12912\alpha\lambda_1^2\lambda_2^6\beta^5 \\
&+48\alpha\lambda_1^4\beta^2\lambda_2^7 - 1596\alpha^2\lambda_1^2\beta^3\lambda_2^8 - 672\alpha^2\lambda_1\beta^4\lambda_2^8 + 31680\alpha\lambda_1^4\beta^6\lambda_2^3 - 19440\alpha\lambda_1^3\lambda_2^6\beta^4 \\
&-24\alpha\lambda_1^5\lambda_2^6\beta^2 + 14400\alpha\lambda_1^4\lambda_2^5\beta^4 + 11808\alpha\lambda_1^2\beta^4\lambda_2^7 + 14400\alpha\lambda_1^4\beta^7\lambda_2^2 - 10752\alpha\lambda_1\beta^5\lambda_2^7 \\
&-42432\alpha\lambda_1^4\lambda_2^4\beta^5 + 48288\alpha\lambda_1^3\lambda_2^5\beta^5 - 4032\alpha\lambda_1^5\lambda_2^4\beta^4 + 12288\alpha\lambda_2^3\beta^5\lambda_1^5 - 115200\alpha\lambda_2^3\lambda_1^3\beta^7 \\
&+432\alpha\lambda_1^5\beta^3\lambda_2^5 - 1080\alpha\lambda_1^4\beta^3\lambda_2^6 + 864\alpha\lambda_1^3\beta^3\lambda_2^7 - 144000\alpha\lambda_1\beta^7\lambda_2^5 - 4\alpha^7\lambda_2^8\lambda_1^5 \\
&-\alpha^8\lambda_2^8\lambda_1^5 + 720\alpha^4\beta^5\lambda_2^8 - 17280\alpha\beta^8\lambda_1^4\lambda_2 + 69120\alpha\lambda_1\beta^8\lambda_2^4 + 5\alpha^4\lambda_2^8\lambda_1^5 - 103680\alpha\lambda_1^2\beta^8\lambda_2^3 \\
&+69120\alpha\lambda_2^2\beta^8\lambda_1^3 - 23040\alpha^2\beta^6\lambda_2^7 + 17280\alpha^2\beta^7\lambda_2^6 + 2\alpha^3\lambda_2^8\lambda_1^5 + 3840\alpha^3\beta^5\lambda_2^8 - 5760\alpha^3\beta^6\lambda_2^7 \\
&+2\alpha^5\lambda_2^8\lambda_1^5 - 4\alpha^6\lambda_2^8\lambda_1^5 + 89280\alpha\lambda_1\beta^6\lambda_2^6 + 7200\alpha^2\beta^5\lambda_2^8 + 37440\alpha\beta^7\lambda_2^6 + 5760\alpha\lambda_1^5\beta^7\lambda_2 \\
&+201600\alpha\lambda_1^2\beta^7\lambda_2^4 - 17280\alpha\beta^8\lambda_2^5 - 97920\alpha\lambda_1^2\beta^6\lambda_2^5 - 216\alpha\lambda_1^2\beta^3\lambda_2^8 - 2736\alpha\lambda_1\beta^4\lambda_2^8 \\
&-24\alpha\lambda_1^3\beta^2\lambda_2^8 + 17280\alpha\lambda_2^4\beta^6\lambda_1^3 - 864\lambda_1\beta^4\lambda_2^8 + 5520\alpha\beta^5\lambda_2^8 + 1440\beta^5\lambda_2^8 - 2880\beta^8\lambda_1^5 \\
&-14400\alpha\lambda_1^5\beta^6\lambda_2^2 - 25920\alpha\beta^6\lambda_2^7 - 264\alpha^5\lambda_1^4\beta^3\lambda_2^6 + 6864\alpha^2\lambda_1^3\beta^3\lambda_2^7)
\end{aligned}$$



3. Jacobian determinant for Example 6.1.2, Section 4.3.3.2

$$\det(H) = \frac{2(p_{02} - 1)^3 \alpha (\beta - \lambda_2) (\alpha + 1)}{\lambda_1^6 \lambda_2^8 p_{02}^6 \beta^{14}} G(p_{01}, \lambda_1, \lambda_2, \alpha, \beta) \quad (2)$$

where

$$\begin{aligned} & G(\lambda_1, \lambda_2, \alpha, \beta) \\ = & 2880\lambda_1^2\beta^7 - 2880\beta^7\lambda_1\lambda_2 - 1440\beta^6\lambda_2^3p_{02} + 2880\beta^7\lambda_2^2p_{02} + 3120\alpha^2\beta^5\lambda_1^4 + 720\alpha^3\beta^5\lambda_1^4 \\ & - 3\alpha^6\lambda_1^4\lambda_2^5 - \alpha^7\lambda_1^4\lambda_2^5 - 240\alpha^2\beta^3\lambda_1\lambda_2^5p_{02} + 2712\alpha^2\beta^4\lambda_1\lambda_2^4p_{02} - 3960\alpha^2\beta^5\lambda_1\lambda_2^3p_{02} \\ & + 3600\alpha^2\beta^5\lambda_1^2\lambda_2^2p_{02} + 1920\alpha^2\beta^5\lambda_1^3\lambda_2p_{02} - 1440\alpha^2\beta^6\lambda_1\lambda_2^2p_{02} - 1440\alpha^2\beta^6\lambda_1^2\lambda_2p_{02} \\ & + 264\alpha^3\beta^3\lambda_1\lambda_2^5p_{02} - 912\alpha^3\beta^4\lambda_1\lambda_2^4p_{02} + 72\alpha^3\beta^4\lambda_1^2\lambda_2^3p_{02} + 960\alpha^3\beta^4\lambda_1^3\lambda_2^2p_{02} \\ & - 720\alpha^3\beta^5\lambda_1^2\lambda_2^2p_{02} + 36\alpha^4\beta^2\lambda_1^2\lambda_2^5p_{02} - 24\alpha^4\beta^3\lambda_1\lambda_2^5p_{02} - 72\alpha^4\beta^3\lambda_1^2\lambda_2^4p_{02} \\ & + 72\alpha^4\beta^4\lambda_1\lambda_2^4p_{02} - 72\alpha^4\beta^4\lambda_1^2\lambda_2^3p_{02} + 1680\alpha^2\beta^4\lambda_1^4\lambda_2 + 36p_{02}\lambda_1^3\alpha^3\lambda_2^5\beta + 12\lambda_1^4p_{02}\alpha^4\lambda_2^4\beta \\ & + 204\lambda_1^4p_{02}\alpha^3\lambda_2^3\beta^2 + 480\lambda_1^4p_{02}\alpha^2\lambda_2^2\beta^3 - 240\lambda_1^4p_{02}\alpha\lambda_2\beta^4 - 49\lambda_1^4\alpha^4\lambda_2^4\beta - 880\lambda_1^4\alpha^3\lambda_2^3\beta^2 \\ & - 1860\lambda_1^4\alpha^2\lambda_2^2\beta^3 - 1632\lambda_1^4\alpha\lambda_2\beta^4 - \lambda_1^4\alpha^5\lambda_2^5 - 1440\lambda_1^3p_{02}\beta^5\lambda_2 + 864p_{02}\lambda_1^3\beta^4\lambda_2^2 \\ & - 12\lambda_1^3p_{02}\beta\lambda_2^5\alpha^4 - 24\lambda_1^3p_{02}\beta^2\lambda_2^4\alpha^3 + 948\lambda_1^3p_{02}\beta^3\lambda_2^3\alpha^2 + 528\lambda_1^3p_{02}\beta^4\lambda_2^2\alpha - 36p_{02}\lambda_1^4\alpha^3\lambda_2^4\beta \\ & - 156p_{02}\lambda_1^4\alpha^2\lambda_2^3\beta^2 + 192p_{02}\lambda_1^4\alpha\lambda_2^2\beta^3 + 2304p_{02}\lambda_1\beta^5\lambda_2^3 - 424\lambda_1^4\alpha^4\lambda_2^3\beta^2 - 156\lambda_1^4\alpha^3\lambda_2^2\beta^3 \\ & - 24\lambda_1^4p_{02}\alpha^4\lambda_2^3\beta^2 - 240\lambda_1^4p_{02}\alpha^3\lambda_2^2\beta^3 - 61\lambda_1^4\alpha^5\lambda_2^4\beta - 12p_{02}\lambda_1^3\beta^2\lambda_2^4\alpha^4 - 636p_{02}\lambda_1^3\beta^3\lambda_2^3\alpha^3 \\ & - 2640p_{02}\lambda_1^3\beta^4\lambda_2^2\alpha^2 - 1200p_{02}\lambda_1^3\beta^5\lambda_2\alpha + 448\lambda_1^3\beta^2\lambda_2^4\alpha^4 + 2772\lambda_1^3\beta^3\lambda_2^3\alpha^3 \\ & - 3600\lambda_1^3\beta^5\lambda_2\alpha + 612p_{02}\lambda_1^2\beta^3\lambda_2^4\alpha^3 + 288p_{02}\lambda_1^2\beta^4\lambda_2^3\alpha^2 - 720p_{02}\lambda_1^2\beta^5\lambda_2^2\alpha - 4320\lambda_1^3\beta^6 \\ & + 448\lambda_1^3\alpha^3\lambda_2^4\beta^2 + 2052\lambda_1^3\alpha^2\lambda_2^3\beta^3 + 3648\lambda_1^3\alpha\lambda_2^2\beta^4 + 28\lambda_1^3\alpha^4\lambda_2^5\beta + 12p_{02}\lambda_1^3\alpha^2\lambda_2^4\beta^2 \\ & + 24p_{02}\lambda_1^3\alpha\lambda_2^3\beta^3 - 180p_{02}\lambda_1^2\beta^2\lambda_2^5\alpha^3 - 720p_{02}\lambda_1^2\beta^4\lambda_2^3\alpha - 1188p_{02}\lambda_1^2\beta^3\lambda_2^4\alpha^2 + 1440\lambda_1^4\beta^5 \\ & + 1440\lambda_2^2\lambda_1\beta^6 - 8640\lambda_1^3\alpha\beta^6 + 4320\lambda_1^2\alpha\beta^7 - 2880\lambda_1^3\alpha^2\beta^6 + 4080\lambda_1^4\beta^5\alpha \\ & + 120\lambda_2^5\lambda_1\alpha^4\beta^3 - 144\lambda_2^4\lambda_1^2\alpha^4\beta^3 - 360\lambda_2^4\lambda_1\alpha^4\beta^4 + 1368\lambda_2^3\lambda_1^2\alpha^4\beta^4 \\ & + 120\lambda_2^4\lambda_1\alpha^2\beta^4 + 28\lambda_2^5\lambda_1^3\alpha^5\beta - 144\lambda_2^5\lambda_1^2\alpha^4\beta^2 + 10080\lambda_2\lambda_1^2\alpha\beta^6 - 4320\lambda_2\lambda_1\alpha\beta^7 \\ & + 10080\lambda_2\lambda_1^2\alpha^2\beta^6 - 432\lambda_2^3\lambda_1^3\alpha^4\beta^3 + 56\lambda_2^4\lambda_1^3\alpha^5\beta^2 - 336\lambda_2^3\lambda_1^3\alpha^5\beta^3 + 3312\lambda_1^3\beta^4\lambda_2^2\alpha^2 \\ & - 5232\lambda_2^2\lambda_1^3\alpha^3\beta^4 - 1728\lambda_2^2\lambda_1^3\alpha^4\beta^4 + 3000\lambda_2^3\lambda_1\alpha^2\beta^5 + 6120\lambda_2^2\lambda_1^2\alpha^2\beta^5 - 7200\lambda_2^2\lambda_1\alpha^2\beta^6 \\ & - 120\lambda_2^5\lambda_1\alpha^3\beta^3 + 1680\lambda_2^4\lambda_1\alpha^3\beta^4 + 1152\lambda_2^3\lambda_1^2\alpha^3\beta^4 - 3240\lambda_2^3\lambda_1\alpha^3\beta^5 + 6120\lambda_2^2\lambda_1^2\alpha^3\beta^5 \\ & - 12240\lambda_2\lambda_1^3\beta^5\alpha^2 - 3600\lambda_2\lambda_1^3\beta^5\alpha^3 + 1680\lambda_2^3\lambda_1\alpha\beta^5 - 1440\lambda_2^2\lambda_1\alpha\beta^6 - 2880\beta^6\lambda_1\lambda_2^2p_{02} \end{aligned}$$

$$\begin{aligned}
&+4320\beta^6\lambda_1^2\lambda_2p_{02} - 2880\beta^7\lambda_1\lambda_2p_{02} - 240\alpha\beta^5\lambda_2^4p_{02} - 1440\alpha\beta^6\lambda_2^3p_{02} + 4320\alpha\beta^7\lambda_2^2p_{02} \\
&+120\alpha^2\beta^4\lambda_2^5p_{02} - 1560\alpha^2\beta^5\lambda_2^4p_{02} + 2880\alpha^2\beta^6\lambda_2^3p_{02} + 2400\alpha^3\beta^4\lambda_1^4\lambda_2 - 120\alpha^3\beta^4\lambda_2^5p_{02} \\
&+360\alpha^3\beta^5\lambda_2^4p_{02} + 696\alpha^4\beta^3\lambda_1^4\lambda_2^2 + 720\alpha^4\beta^4\lambda_1^4\lambda_2 + 16\alpha^5\beta^2\lambda_1^4\lambda_2^3 + 288\alpha^5\beta^3\lambda_1^4\lambda_2^2 \\
&-11\alpha^6\beta\lambda_1^4\lambda_2^4 + 48\alpha^6\beta^2\lambda_1^4\lambda_2^3 + 3\alpha^7\beta\lambda_1^4\lambda_2^4 + 2\lambda_1^4\alpha^3\lambda_2^5 + 2880\lambda_1^2\beta^6\lambda_2 - 864\lambda_2^2p_{02}\lambda_1^2\beta^5 \\
&-2304\lambda_2^2\lambda_1^2\beta^5 + 864\lambda_2\lambda_1^3\beta^5 + 864\lambda_2^2\lambda_1^3\beta^4 - 864\lambda_2\lambda_1^4\beta^4 - 864\lambda_2^3p_{02}\beta^4\lambda_1^2 - 2160\lambda_1^2\beta^5\lambda_2^2\alpha \\
&-5112\lambda_1^2\beta^4\lambda_2^3\alpha^2 - 2016\lambda_1^2\beta^4\lambda_2^3\alpha - 2196\beta^3\lambda_2^4\lambda_1^2\alpha^3 - 72\beta^3\lambda_2^4\lambda_1^2\alpha^2 + 180\beta^2\lambda_2^5\lambda_1^2\alpha^3 \\
&+360\alpha^3\beta^5\lambda_1\lambda_2^3p_{02} + 96\alpha^4\beta^3\lambda_1^3\lambda_2^3p_{02} + 3\alpha^4\lambda_2^5\lambda_1^4 - 480\lambda_1^4p_{02}\beta^4\alpha^2\lambda_2 - 36\lambda_2^5\lambda_1^2\alpha^5\beta^2 \\
&+144p_{02}\beta^2\lambda_2^5\lambda_1^2\alpha^2 + 408\lambda_1^3\alpha\lambda_2^3\beta^3 - 216p_{02}\lambda_1^2\alpha\lambda_2^4\beta^3 - 64\lambda_1^3\alpha^2\lambda_2^4\beta^2 - 64\lambda_1^3\alpha^3\lambda_2^5\beta \\
&-24p_{02}\lambda_1^3\alpha^2\lambda_2^5\beta - 408\alpha\beta^3\lambda_1^4\lambda_2^2 - 56\alpha^2\beta^2\lambda_1^4\lambda_2^3 + 94\alpha^3\beta\lambda_1^4\lambda_2^4 + 24\alpha^2\beta\lambda_1^4\lambda_2^4 \\
&-24p_{02}\lambda_1^4\alpha\lambda_2^3\beta^2 + 24p_{02}\lambda_1^3\alpha\lambda_2^4\beta^2 + 432\alpha\beta^4\lambda_1\lambda_2^4p_{02} + 2160\alpha\beta^5\lambda_1\lambda_2^3p_{02} \\
&+5760\alpha\beta^6\lambda_1^2\lambda_2p_{02} - 4320\alpha\beta^7\lambda_1\lambda_2p_{02} + 8\lambda_2^5\lambda_1^3\alpha^6\beta - 24\lambda_2^4\lambda_1^3\alpha^6\beta^2 \\
&-4320\alpha\beta^6\lambda_1\lambda_2^2p_{02} + 24\alpha^2\beta\lambda_1^4\lambda_2^4p_{02} + 108\lambda_2^4\lambda_1^2\alpha^5\beta^3
\end{aligned}$$

---

4. Jacobian determinant for Example 5.1 continued, Section 4.3.1

$$\begin{aligned} \det(H) = & \frac{288p_{01}^2(1-p_{01})}{\lambda_1^7\lambda_2^7\lambda_3^8}(\lambda_2 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_1 - \lambda_2)(-3\lambda_1^3\lambda_2^2\lambda_3 + 3\lambda_1^3\lambda_2\lambda_3^2 \\ & + \lambda_1^3\lambda_2^3 - \lambda_1^3\lambda_3^3 + 9\lambda_1^2\lambda_2^2\lambda_3^2 - 9\lambda_1^2\lambda_2\lambda_3^3 + 3\lambda_3^4\lambda_1^2 - 3\lambda_1^2\lambda_2^3\lambda_3 \\ & + 3\lambda_1\lambda_2^3\lambda_3^2 + 4\lambda_1\lambda_2\lambda_3^4 - 9\lambda_1\lambda_2^2\lambda_3^3 + 3\lambda_3^4\lambda_2^2 - \lambda_2^3\lambda_3^3) \end{aligned}$$

5. Jacobian determinant for Example 6.1.1, Section 4.3.3.1.

$$\begin{aligned} \det(H) = & \frac{288(1-p_{02})^2(\lambda_3 - \lambda_1)}{\lambda_2^5\lambda_3^7\lambda_1^7p_{02}^5} \cdot \left( 2p_{02}\lambda_3^2\lambda_1^3 - 2\lambda_3\lambda_1^3\lambda_2p_{02} - 2\lambda_3\lambda_1^3\lambda_2 \right. \\ & + 2\lambda_2^2\lambda_1^3 - 3p_{02}\lambda_2\lambda_3^2\lambda_1^2 + p_{02}\lambda_2^2\lambda_3\lambda_1^2 - 4\lambda_1^2\lambda_2\lambda_3^2 + 7\lambda_2^2\lambda_3\lambda_1^2 - 3\lambda_2^3\lambda_1^2 \\ & + 2p_{02}\lambda_3^3\lambda_1^2 - 2p_{02}\lambda_2\lambda_1\lambda_3^3 + \lambda_3\lambda_1\lambda_2^3p_{02} - 2\lambda_3^3\lambda_1\lambda_2 \\ & \left. - 5\lambda_2^3\lambda_3\lambda_1 + p_{02}\lambda_2^2\lambda_1\lambda_3^2 + 7\lambda_2^2\lambda_3^2\lambda_1 - 3\lambda_2^3\lambda_3^2 + 2\lambda_2^2\lambda_3^3 \right) \end{aligned}$$

6. The following is the Matlab code of `simdat.m` function

```
1 function datv=simdat(MGF,n)
2 %Input: 1) MGF—The MGF of total waiting time
3 %       2) n—The sample size
4
5 %Output: datv—total waiting time data
6
7 % Derive the PDF by the inverse Laplace transform
8 syms s t x
9 M=subs(MGF,s,-s);
10 f=ilaplace(M,s,t);
11
12 % Compute the CDF
13 matlabFunction(f,'file','pdfs')
14 F=int(f,t,0,x);
15 matlabFunction(F,'file','cdfs')
16
17 % Simulate dat2 from Uniform(0,1)
18 dat2=rand(1,n);
19 initv=2;
20 sn=length(dat2);
21 tol=10^-6;
22
23 %Solve F(x)=dat2 for x
24 options=optimset('TolFun',tol,'Display','off');
25 sdat=zeros(1,sn);
26 infov=zeros(1,sn);
27 fvalv=zeros(1,sn);
28 for i=1:sn
29     [est, fval, info]=fsolve(@(x) cdfu(x,dat2(i)),initv,options);
30     sdat(i)=est;
31     fvalv(i)=fval;
32     infov(i)=info;
33 end
34 datv=sdat;
35 end
```

---

7. The following is the Matlab code of `momeq.m` function

```
1 function fv=momeq(IM,pad,theta,datm)
2 %Input
3 %1)IM——Incidence matrix
4 %2)pad——The type of internode distribution, 1 for ...
   Exponential, 2 for Gamma
5 %3)theta——The set of parameters
6 %4)datm——The total waiting time sample data
7
8
9 %Output
10 %fv——The value of R function defined in equation 6.1
11
12
13 % Derive the transition matrix
14 syms s
15 G=CIM(IM);
16 y=[zeros(1,length(G)-1),s];
17 x=[G;y];
18 x(end-1,:)=[];
19 dim=size(IM);
20 nz=find(x== -1);
21
22 pv=theta(1:dim(2));
23 indx=dim(2);
24
25 for i=1:dim(2)
26     p=pv(i);
27     pnum=pad(i);
28     if pnum==1
29         lambda=theta(indx+1);
30         x(nz(i))=p*lambda/(lambda-s);
31         indx=indx+1;
32     end
33
34     if pnum==2
35         alpha=theta(indx+1);
36         beta=theta(indx+2);
```

```
37         x(nz(i))=p*alpha/(beta-s);
38         indx=indx+2;
39     end
40
41 end
42
43 x(end,end)=0;
44
45 %compute the MGF
46 K=MV(x,theta);
47 MGF=K.MGF;
48
49 %Determine the moment from the coefficient of taylor series of...
50     MGF.
51 np=length(theta);
52 f=MGF;
53 T=taylor(f,np+1);
54 TC=sym2poly(T);
55 c=factorial(1:np);
56
57 c2=sort(c,'descend');
58 PM=c2.*TC(1:np);
59 meq=sort(PM,'ascend');
60
61 %Construct the R function
62 fv=sum((meq-datm).^2);
63 end
```

# Bibliography

- [1] Aalen, O.O. (1995). Phase type distributions in survival analysis. *Scandinavian Journal of Statistics*, 22, 447-463.
- [2] Amindavar, H., and Ritcey, J.A. (1994). Padé Approximation of Probability Density Function. *Aerospace and Electronic Systems, IEEE Transactions*, 30, 416-424.
- [3] Andersen, P.K., and Keiding, N. (2002). Multi-state models for event history analysis. *Statistical Methods in Medical Research*, 11, 91-115.
- [4] Asmussen, S. (1987). *Applied probability and queues*. Wiley, New York.
- [5] Baker, G.A., and Graves-Morris, P.R. (1996). *Padé Approximants*. Cambridge University Press, Cambridge.
- [6] Bartholomew, D.J. (1969). Sufficient conditions for a mixture of exponentials to be a probability density function. *The Annals of Mathematical Statistics*, 40, 2183-2188.
- [7] Barndorff-Nielsen, O.E., and Cox, D.R. (1997). Edgeworth and saddlepoint approximations with statistical application. *Journal of Royal Statistical Society Series, Series B*, 42, 279-312.
- [8] Billingsley, P. (1986). *Probability and Measure*. 2nd edition. Wiley, New York.
- [9] Butler, R.W. (1997a). System reliability, flowgraphs, and Saddlepoint approximation. Technical report, Dep. Statistics, Colorado State Univ.
- [10] Butler, R.W. (1997b). First passage distributions in semi-Markov processes and their saddlepoint approximation. In *Data Analysis from Statistical Foundations* (2001), page 347, Nova Science Publishers, Huntington, NY.

- [11] Butler, R.W. (2000). Reliabilities for Feedback Systems and Their Saddlepoint Approximation. *Statistical Science*, 15, 279-298.
- [12] Butler, R.W., and Huzurbazar, A.V. (1997). Stochastic Network Models for Survival Analysis. *Journal of the American Statistical Association*, 92, 246-257.
- [13] Byatt, D., Coope, I., and Price, C. (2003). 40 Years of the Nelder-Mead Algorithm. University of Canterbury, New Zealand.
- [14] Carr, P., and Madan, D. (2009). Saddlepoint methods for option pricing. *The Journal of Computational Finance*, 13, 49C61.
- [15] Casella, G., and Goutis, C. (1999). Explaining the Saddlepoint Approximation. *American Statistical Association*, 53, 216-224.
- [16] Casella, G., and Berger, R.L. (2002). *Statistical inference*. Thomson Learning, Pacific Grove, CA.
- [17] Cordeiro, G.M., and McCullagh, P. (1991). Bias reduction in generalized linear models. *Journal of the Royal Statistical Society, Series B*, 53, 629-643.
- [18] Cox, D.R., and Snell, E.J. (1968). A general distribution of residuals (with discussion). *Journal of the Royal Statistical Society, Series B*, 30, 248-275.
- [19] Daniels, H. (1954). Saddlepoint approximations in statistics. *The Annals of Mathematical Statistics*, 25, 631-650.
- [20] Daniels, H. (1987). Tail probability approximations. *International Statistical Review* 55, 37-48.
- [21] Dalla Valle, J.M. (1931). Note on the Heaviside Expansion Formula. *Proceedings of the National Academy of Sciences of the United States of America*, 17, 678-684.
- [22] Dawid, A.P., and Grünwald, P.D. (2004). Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics*, 32, 1367-1433.



- [23] De Hoog, F.R., Knight, J.H., and Stokes, A.N (1982). An improved method for numerical inversion of Laplace transforms. *SIAM Journal on Scientific and Statistical Computing*, 3, 357-366.
- [24] Dennis, J.E., and Schnabel, R.B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ.
- [25] Dembo, A., and Zeitouni, O. (1998). *Large deviations techniques and applications*. vol. 38 of *Applications of Mathematics* (New York), Springer-Verlag, New York.
- [26] Dickson, D.C.M., Waters, H.R., and Hardy, M.R. (2009). *Actuarial Mathematics for Life Contingent Risks*. Cambridge University Press, New York.
- [27] Dempster, A.P., Laird, N.M., and Rubin, D.R. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1-38.
- [28] Feller, W. (1971). *An Introduction to Probability Theory and Its Applications*, Vol.2, 2nd edn, John Wiley, New York.
- [29] Field, C., and Ronchetti, E. (1990). *Small Sample Asymptotics*. Hayward, CA: Institute of Mathematical Statistics.
- [30] Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80, 27-38.
- [31] Fix, E., and Neyman, J. (1951). A simple stochastic model of recovery, relapse, and loss of patients. *Human Biology*, 23, 205-241.
- [32] Gander, W., and Gautschi, W. (2000). Adaptive Quadrature revisited. *BIT*, 40, 84-101.
- [33] Gatto, R. (2008). A saddlepoint approximation to the probability of ruin in the compound Poisson process with diffusion. *Statistics and Probability Letters*, 78, 1948-1954.
- [34] Giffin, A., and Caticha, A. (2007). *Updating Probabilities with Data and Moments*. Presented at the 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Saratoga Springs, NY, July 8-13.

- [35] Glasserman, P. (2004). Tail approximations for portfolio credit risk. *The Journal of Derivatives*, 12, 24-42.
- [36] Gordy, M.B. (2002). Saddlepoint approximation of CreditRisk. *Journal of Banking and Finance*, 26, 1335-1353.
- [37] Haberman, S., and Pitacco, E. (1999). *Actuarial Models for Disability Insurance*, Chapman and Hall, Boca Raton.
- [38] Harris, C.M, and Sykes, E.A. (1987). Likelihood estimation for generalized mixed exponential distributions, *Naval Research Logistics (NRL)*, 34, 251-279.
- [39] Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. Springer-Verlag.
- [40] Heyde, C.C. (1963a). On a property of the log-normal distribution. *Journal of the Royal Statistical Society, Series B*, 25, 392-393.
- [41] Heyde, C.C. (1963b). Some remarks on the moment problem. *Quarterly Journal of Mathematics*, 14, 91-96.
- [47] Hougaard, P. (1999). Multi-state models: A review. *Lifetime Data Analysis*, 5, 239-64.
- [43] Howard, R.A. (1964). System analysis of semi-Markov processes. *IEEE Trans. Military Electronics*, 8, 114-124.
- [44] Howard, R.A. (1971). *Dynamic Probabilistic Systems 2: Semi-Markov and Decision Processes*. Wiley, New York.
- [45] Hoem, J.M. (1969). Markov chain models in life insurance. *Blätter der DGVMF*, 9, 91-107.
- [46] Hoem, J.M. (1972). Inhomogeneous semi-Markov processes, select actuarial tables, and duration dependence in demography. *Population Dynamics*, Academic Press New York, 251-296.
- [47] Hougaard, P. (2000). *Analysis of multivariate survival data*. Springer, New York.

- [48] Huzurbazar, S. (1999). Practical Saddlepoint Approximations. *The American Statistician*, 53, 225-232.
- [49] Huzurbazar, A.V. (2002). Flowgraph models for generalized phase type distributions with non-exponential waiting times. *Scandinavian Journal of Statistics*, 26, 145-157.
- [50] Huzurbazar, A.V. (2005). Flowgraph models for multistate time-to-event data. Wiley, New Jersey.
- [51] Huggins, W. (1957). Signal-flow graphs and random signals. *Proceedings of the Institute of Radio Engineers*, January, 74-86.
- [52] Jaynes, E.T. (1957). Information Theory and Statistical Mechancis. *Physics Review*, 106, 620-630.
- [53] Johanssen, F. (1907). Waiting times and number of calls. Reprinted in *Post Office Electrical Engineers Journal (London)*, October 1910 and January 1911, as cited by Kendall (1951).
- [54] Jensen, J.L. (1995). Saddlepoint Approximation. Clarendon Press, Oxford.
- [55] Kagan, A.M., Linnik, Y.V., and Rao, C.R. (1973). Characterization Problems in Mathematical Statistics. John Wiley, New York.
- [56] Kelly, F.P., Zachary, S., and Ziedins, I.B. (1996). Stochastic Networks: Theory and Applications. Royal Statistical Society Lecture Notes Series, 4, Oxford University Press.
- [57] Kendall, D.G. (1951). Some problems in the theory of queues. *Journal of the Royal Statistical Society, Series B*, 13, 151-185.
- [58] Kolassa, J.E. (1994). Series Approximation Methods in Statistics. Springer-Verlag, New York.
- [59] Koornwinder, T.H., Wong, R.S.C, Koekoek, R., and René, F. (2010). Orthogonal Polynomials. *NIST Handbook of Mathematical Functions*, Cambridge University Press.

- [60] Lagarias, J.C., Reeds, J.A., Wright, M.H., and Wright, P.E. (1998). Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, 9, 112-147.
- [61] Lew, J.S. (1973). Asymptotic Inversion of Laplace Transforms: A class of counterexamples. *Proceedings of the American Mathematical Society*, 39, 329-336.
- [62] Lorens, C.S. (1964). *Flowgraphs for the Modeling and Analysis of Linear Systems*. McGraw-Hill, New York.
- [63] Lugannani, R., and Rice, S. (1980). Saddlepoint approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability*, 12, 475-490.
- [64] Mason, S.J. (1953). Feedback theory: Some properties of signal flow graphs. *Proceedings of the Institute of Radio Engineers*, 41, 1144-1156.
- [65] Mason, S.J. (1956). Feedback theory: Further properties of signal flow graphs. *Proceedings of the Institute of Radio Engineers*, 44, 920-926.
- [66] McKinnon, K.I.M. (1999). Convergence of the Nelder-Mead simplex method to a non-stationary point. *SIAM J Optimization*, 9, 148-158.
- [67] McLachlan, G.J., and Krishnan, T. (1997). *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, New York, Wiley.
- [68] Neuts, M.F. (1981). *Matrix-geometric solutions in stochastic models: An Algorithmic Approach*. Johns Hopkins University Press, Baltimore.
- [69] Nelder, J.A., and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7, 308-313.
- [70] Norberg, R. (2008). Multistate Models for Life Insurance Mathematics. *Encyclopedia of Quantitative Risk Analysis and Assessment*, 3, 1128-1138.
- [71] Phillips, C.L., and Harbor, R.D. (1996). *Feedback Control Systems*. Prentice Hall, Englewood-Cliffs, New Jersey.

- [72] Pitacco, E. (1995). Actuarial models for pricing disability benefits: Towards a unifying approach. *Insurance Mathematics and Economics*, 16, 39-62.
- [73] Powell, M.J.D. (1973). On Search Directions for Minimization Algorithms. *Mathematical Programming*, 4, 193-201.
- [74] Pritsker, A.A.B., and Happ, W.W. (1966). GERT: Graphical evaluation and review technique, Part I : fundamentals. *Journal of Industrial Engineering*, 17, 267-274.
- [75] Pyke, R. (1961). Markov renewal processes with finitely many states. *The Annals of Mathematical Statistics*, 32, 1243-1259.
- [76] Reid, N. (1988). Saddlepoint methods and statistical inference. *Statistical science*, 3, 213-238.
- [77] Reid, N. (1991). "Approximations and asymptotics. *Statistical science*" in *statistical theory and models, Essays in honor of D.R. Cox*, Chapman and Hall, 287-334.
- [78] Sen, R. (2008). A multi-state Vasicek model for correlated default rate and loss severity. *Risk Magazine*, 21(6), 94-100.
- [79] Sittler, R. (1956). Systems analysis of discrete Markov processes. *Institute of Radio Engineers Transactions on Circuit Theory*, 3, 257-266.
- [80] Sjödin, T. (1987). A note on the Carleman condition for determinacy of moment problems. *Arkiv för Matematik*, 25, 289-294.
- [81] Spiegel, M. (1965). *Schaum's Outline of Laplace Transforms*. McGraw-Hill, New York.
- [82] Spierdijk, L., and Koning, R.H. (2011). Calculating Loss Reserves in a Multistate Model for Income Insurance. SSRN, <http://dx.doi.org/10.2139/ssrn.1871229>.
- [83] Stoyanov, J. (1997). *Counterexamples in Probability*. 2nd edn. John Wiley, Chichester.
- [84] Terrell, G.R. (2003). A Stabilized Lugannani-Rice Formula, *Proceedings of the 35th Symposium on the Interface*. [Interfacesymposia.org](http://Interfacesymposia.org).

- [85] Wagner, U., and Geyer, A.L.J. (1995). A Maximum Entropy Method for Inverting Laplace Transforms of Probability Density Functions. *Biometrika*, 82, 887-892.
- [86] Whitehouse, G.E. (1970). GERT, a useful technique for analyzing reliability problems. *Technometrics*, 12, 33-48.
- [88] Wu, C.F. Jeff. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11, 95-103.
- [88] Wu, X (2003). Calculation of maximum entropy densities with application to income distribution. *Journal of Econometrics*, 115, 347-354.
- [89] Yang, J, Hurd, T.R., and Zhang, X. (2006). Saddlepoint approximation method for pricing CDOs. *Journal of Computational Finance*, 10, 1-20.