ESTIMATION AND INFERENCE IN PROBLEMS

FROM IMAGING AND BIOPHYSICS

by

Yingxiang Zhou

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied Mathematics

Summer 2018

ESTIMATION AND INFERENCE IN PROBLEMS

FROM IMAGING AND BIOPHYSICS

by

Yingxiang Zhou

Approved: _____
Louis F. Rossi, Ph.D.
Chair of the Department of Mathematical Sciences

Approved: _____
George H. Watson, Ph.D.
Dean of the College of Arts and Sciences

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Pak-Wing Fok, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Richard Braun, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

David A. Edwards, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Richard G. Spencer, M.D., Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

Chapter

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Many physical processes and phenomena have solutions containing sums of exponential functions. These exponential functions generally describe the decay of processes and signals they emit. In order to better understand them, one needs to determine the parameters that underly these processes, given measured data. The estimation and inference of these parameters from data is an inverse problem, and is usually ill-posed in that multiple sets of parameters may generate the same, or similar data.

In this dissertation, we study exponential analysis in three different applications. The first application is inferring the transition rates of a birth-death process (BDP) from its extinction time (ET) distribution. We first investigated the analytical solution with noise-free data as the sum of exponential functions, and then solved small BDP problems from exact ETs. Then with maximum sites as additional information, we proposed a new numerical scheme to infer the transition rates from a BDP of length $N+1$, in the context of protein folding with atomic force microscopy (AFM) data. This method focuses on the coefficients of the characteristic polynomial of the underlying ODE, and establishes recurrence relations between them. Transition rates are recovered sequentially with initial errors propagating exponentially in the BDP.

The second problem arises in nuclear magnetic resonance (NMR) and medical imaging. In an effort to determine the type of tissue in NMR experiments, we need to apply the inverse Laplace transform (ILT) on NMR signals which are functions of two relaxation times T_1 and T_2. Since ILT is ill-conditioned, we use Tikhonov regularization to recover the distribution of relaxation times. The inversion can be done either in a single step by resizing the solution matrix as a large vector with one parameter, or in two steps by sequential inversion of T_1 and T_2 that involves two parameters. We show

that the one-parameter approach performs well, and adding extra parameters does not improve the result.

The third problem is to fit a two-term exponential probability distribution given measured data. When the exponents are close to each other, many classic methods fail. We propose the moment constraint method, which is revised from the modified Prony method, that takes into account moments of data, regularization, and expectation-maximization (EM) techniques to overcome the difficulties. This method outperforms many other methods, such as maximum-likelihood, when two exponents are very close. The moment constraint method is also applied to four-term exponential fitting problems whose exponents can be separated into two groups by magnitude. It breaks down to two-term exponential fitting subproblems, after a preprocessing step involving Tikhonov regularization and EM sorting, and yields results with reasonable accuracy.

Chapter 1

INTRODUCTION

1.1 Motivation

Estimation and inference problems are ubiquitous in understanding physical phenomena. In reality, we have direct access to data being generated by certain underlying processes whose mathematical models and equations are well defined. In order to better interpret these processes, one often finds it important to figure out the parameters or intrinsic relationship hidden in the models.

Usually, these estimation and inference problems classify into parametric and nonparametric models. Parametric models are ones that are explicitly determined by a fixed set of parameters, whereas nonparametric models have variable parameter sets that may depend on the data collected. A typical example of a parametric model is the prediction of house prices using regression, where one needs to assign weights to a set of features, such as square footage and number of beds/baths. Once the weight parameters are calculated, this model can be used to predict the price of new houses. Another example of a parametric model is the application of the Black-Scholes equation to find the implied volatility of an option given the current option price, underlying stock price and other related information. Parametric models are also used in spam filtering, fraud detection, and many other fields. On the other hand, nonparametric models are becoming popular nowadays. They are used in decision tree models, neural networks, and other machine learning techniques, which can be used for image classification, text retrieval, speech recognition, and other technologies.

In this dissertation, however, we only consider parametric models with the problem of estimating the *parameters* that are defined in a fixed but unknown distribution.

The challenge is that learning from data to infer parameters is an inverse problem, which is usually subject to ill-conditioning. That is, several different parameter sets can potentially generate the same data that we observe.

We will mainly study exponential analysis in this dissertation, in three different contexts. The first problem is inferring the transition rates in a birth-death process given its extinction time distribution. The second application is in medical imaging, where we try to infer the type of tissues from the magnetic resonance signal. The last one is a straightforward exponential fitting problem. These problems appear to be distinct, but they can be analyzed with the same mathematical framework.

1.2 Outline of Dissertation

This dissertation mainly consists of the following chapters.

Chapter 1 herein motivates all the work done in my PhD studies, introduces the physical and biological background of these problems in the real world, and summarizes the contents to be presented.

Chapter 2 is concerned with the birth-death process (BDP), where our main interest is to infer the transition rates from relevant quantities in a birth-death process, such as the extinction times. We start with reviewing basic concepts of the stochastic process and birth-death process in particular, and proceed to analytical inference in a noise-free environment. Next, numerical schemes are proposed in the case of noisy data of birth-death chains with various lengths, where we mainly consider the reflection problems in a birth-death chain. We also apply this methodology to protein folding problems and landscape theories.

Chapter 3 discusses the inference problem in a biomedical setting – Nuclear Magnetic Resonance (NMR) and Magnetic Resonance Imaging (MRI). The backgrounds and concepts of NMR are introduced along with the mathematical formulation of the inverse Laplace transform. The inverse Laplace transform is well-known to be ill-posed, and Tikhonov regularization is introduced to resolve this issue in practice. Theoretical relaxometry problems are then derived in both 1D and 2D, and numerical examples of

relaxometry inverse imaging comes after that. We finally propose a Tikhonov regularization scheme using directional total variation in order to deal with a special type of NMR signal.

Chapter 4 presents the work related to exponential analysis. Since this is a problem with a long history, we first perform a brief survey of major classic methods that solve exponential fitting problems. Finally, we propose a new method for estimating the parameters in a two-exponential function which potentially overcomes the limitation of similar methods.

<center>**Chapter 2**</center>

<center>**INFERENCE OF THE BIRTH-DEATH PROCESSES FROM
EXTINCTION TIMES**</center>

In this chapter, we will first review some basic concepts from stochastic processes in the context of the birth-death process in Section 2.1, and then discuss our contributions to the analytical and numerical inference problems in different scenarios in Sections 2.2-2.4.

2.1 Introduction to the Birth-Death Process

2.1.1 General examples

Birth-death processes (BDP) are widely used in modeling many physical, chemical and biological processes and phenomena [102]. These are special Markov processes defined on a lattice $\mathcal{S} = \{0, 1, \ldots, N\}$, where the largest state N could be finite (which is our focus) or infinite. There are two sets of parameters that underlie the process, namely the birth and death rates (see Definition 2.2)

$$
\begin{aligned}
\lambda_n: &\quad n \to n + 1, \\
\mu_n: &\quad n \to n - 1.
\end{aligned}
\tag{2.1.1}
$$

In this dissertation, we only consider continuous-time BDPs. It is essentially a random walk on $\mathcal{S}$ in which the random walker may only move forward or backward by one step to its neighboring state. The probability of the process moving forward in a small time interval dt is $\lambda_n \, dt$ if it is currently in state n, and the probability of moving backward in that time period is $\mu_n \, dt$. Since a population of organisms stops evolving once it becomes extinct, we define the state $n = 0$ to be an absorbing state at which the BDP terminates, which is also called "extinction", so that $\lambda_0 = 0$. In the

<center>4</center>

following sections, we will be mainly concerned with the times to extinction, also called extinction times. To further clarify, the process has to go extinct because state 0 is the only absorption state. The expected time to absorption is finite in finite absorbing Markov chains: see section 4.6 in [92] for details.

To get started, let's consider some simple examples of BDP from epidemiology and population biology [28]. The Susceptible-Infected-Susceptible (SIS) model of epidemiology [55] describes a common situation where n individuals from a larger population of size N suffer from an infection such as influenza, and the remaining $N - n$ individuals are susceptible. Each recovered individual becomes susceptible again, hence the name "SIS". Suppose the infection rate per contact is Λ/N, and the recovery rate is unity. Since there are $n(N - n)$ possible contacts, we can describe this model as a BDP with birth rates $\lambda_n = \Lambda n(1 - n/N)$, and death rates $\mu_n = n$. It has been shown that this model has a threshold of $\Lambda = 1$, above which the BDP remains near a quasi-stationary state for quite a long time before extinction [73]. If we apply the inference problem here, we would record the extinction time in multiple populations, get a distribution of extinction times, and try to infer the underlying infection rate in this situation.

Another example is the M/M/1 queue in queueing theory [58]. An M/M/1 queue represents the queue length in a system with a single server, where customer arrivals follow a Poisson process with rates λ and service times are exponentially distributed with mean $1/\mu$. Each queue length corresponds to one state in the BDP, i.e., state i represents there are i customers currently in the queue. In fact, the name M/M/1 follows Kendall's notation [57], and it actually stands for a queue with "Markovian arrival time/Markovian service time/one server". Since a single server queue follows a first-come first-serve discipline, the BDP can be used to model the queueing process, where birth rates $\lambda_n = \lambda$, and death rates $\mu_n = \mu$ are constant throughout the birth-death chain. The process starts when the first customer arrives, and ends when the last customer leaves the queue. It can be further extended to an M/M/k queue where k servers are available with other assumptions unchanged. In this case, birth rates are

still $\lambda_n = \lambda$ from the Poisson process, while the death rates are $\mu_n = n\mu$ if $n < k$ and $\mu_n = k\mu$ if $n \geq k$, when the number of customers is beyond capacity.

In the field of chemistry, BDP is used for modeling chemical kinetics. Suppose there are two chemical species A and B with overall population of N, and a reaction between them

$$A \underset{k_2}{\overset{k_1}{\rightleftarrows}} B,$$
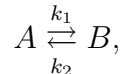
where the reaction rates are proportional to species counts. Then both species counts $A(t), B(t)$ are BDPs, with birth and death rates for $A(t)$ as $\lambda_n = k_2(N - n)$ and $\mu_n = k_1 n$. By recording and analyzing the time it takes for species A to vanish, we can infer the coefficients k_1, k_2 that determine the process.

2.1.2 Application in Protein Folding

One important application of this inference problem is in the area of single-molecule biophysics, specifically the folding of proteins and nucleic acids [21, 27, 108]. The folding of these macromolecules depends on the sequence of amino acids or nucleotides that make up the protein or nucleic acid (primary structure) [2, 13]. The prediction of the native conformation of a protein, given its amino acid sequence, is one of the great open problems in structural biology [7]. Some of the main experimental techniques to study this problem are Atomic Force Spectroscopy [76] and Förster Resonance Energy Transfer [67, 95], which have allowed experimentalists to explore the relationship between macromolecular structure and folding/unfolding rates. A computational approach to protein folding is usually implemented by large-scale all-atom molecular dynamics (MD) simulations [94].

Energy landscape theory [78] provides the fundamental biophysical model for the structural conformation of these macromolecules. Proteins may fold into many possible conformational microstates, and the free-energy landscape is a hyper-dimensional surface that spans all of these configurations. Each point on this hypersurface represents the free energy in a specific structural conformation. In view of this, the folding process may be considered as a diffusion process over the free-energy hypersurface, and

it naturally tends to arrive at the configuration with minimal energy. This configuration is similar to the "absorption state" in a BDP. Another approach to understand protein folding is to introduce a *reaction coordinate* [9] which effectively maps the high-dimensional space onto a single scalar. In practice, this scalar is often an observable metric that measures the progress of the folding, such as bond angle, bond length, etc. In the reaction coordinate, this simple landscape may exhibit multiple minima, corresponding to multiple metastable configurations. In addition, the rates of transition among these configurations differ, in that many macromolecules have evolved to fold rapidly towards the native configuration, while energy barriers may exist among the microstates that slow the process. Inferring the shape of these landscapes from quantities such as first passage times [37], rupture forces [99] or time-displacement trajectories [19] remains a challenging theoretical problem.

One of the most common ways of probing the energy landscape is through Atomic Force Microscopy (AFM) [91]. In AFM experiments, one end of the molecule is tethered to a movable platform and the other is attached to a cantilever tip: see Fig. 2.1(a). Small deflections of the cantiliever are detected using a laser-photodiode setup. The AFM can operate in several ways. One protocol is "force-ramp" mode where the platform lowers at a constant speed. As a protein domain is coercively unfolded, the cantilever deflection increases until a critical platform position is reached. Beyond this point, the cantilever quickly relaxes, corresponding to domain rupture. The resulting force-extension curve allows quantification of the "entropic elasticity" of that particular domain. The procedure can also be performed sequentially if multiple domains are present in a large protein [75, 89]. The discrete event corresponding to rupture is interesting both physically and mathematically. Experiments show that rupture does not always occur at the same force. Furthermore, the rupture force distribution shifts towards higher values when the platform speed is larger. Both of these observations are in stark contrast to mechanical bonds that break at a single yield stress. They point towards a model of bond-breaking that is based on "thermally activated escape", i.e., a theory of random walks.
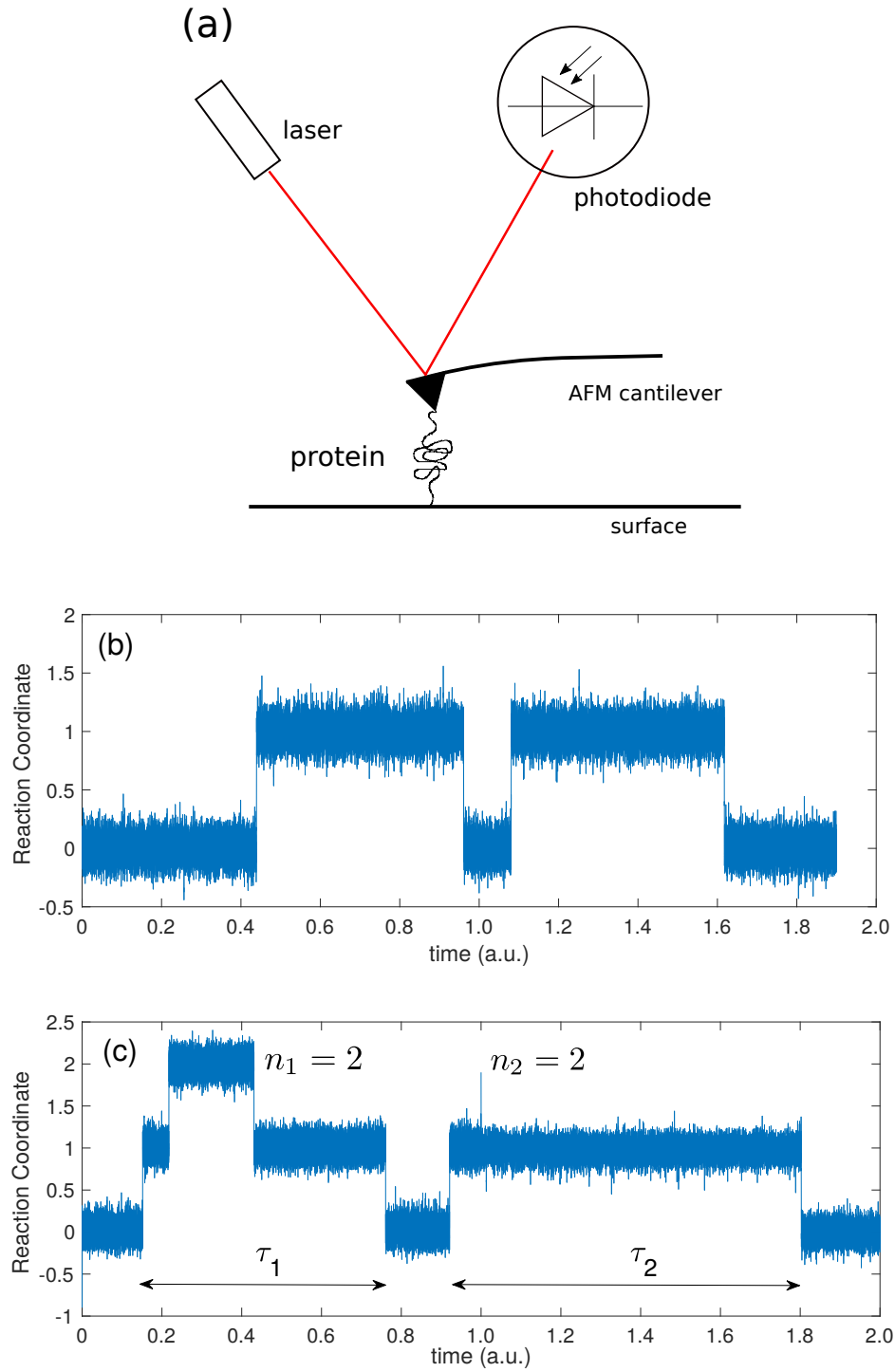
Figure 2.1: (a) Experimental schematic for Atomic Force Microscopy of proteins. Deflection of a soft cantilever is detected using a laser-photodiode setup. (b) Possible time trace of reaction coordinate for a two-state protein. (c) Possible trace of reaction coordinate for a three-state protein. Extinction times τ_1 and τ_2 along with maximal sites $n_1 = 2$ and $n_2 = 2$ respectively are used for inference. a.u. = arbitrary units.

Besides force-ramp mode, another protocol is to keep the AFM platform stationary and operate in "force-clamp" mode. Under this mode of operation, one focuses on deflections of the cantilever which essentially provide the reaction coordinate as a function of time. Some possible time traces are shown in Fig. 2.1(b) and (c). The protein spends most of its time in metastable configurations (when the reaction coordinate is an integer) and very little time in-between these states. Figure 2.1(b) shows the trace for a simple protein with a single folding domain that can be in one of two states: folded or unfolded. The kinetics in this case are well-described by two exponential distributions [90]; one for the $1 \to 0$ transition and the other for the $0 \to 1$ transition. The half-lives or rate constants associated with each exponential distribution can easily be inferred from the time trace in Fig. 2.1(b). For proteins with multiple domains the traces can be more complex and could resemble Fig. 2.1(c). If one assumes exponential kinetics as before (i.e. the transition between states always follows an exponential distribution, but the parameters of the distribution could be state-dependent) the resulting stochastic process is a birth-death chain. These chains correspond to energy landscapes with multiple metastable states: see Fig. 2.2. Small proteins typically follow single-exponential kinetics, which is well-described by a two-state chain see Fig. 2.2(a). However, larger proteins may exhibit more complicated kinetics corresponding to one or more intermediate states: see Fig. 2.2(b). The practical inference of transition rates in these longer birth-death chains is the focus of this chapter.

An extinction time is the time taken for the reaction coordinate to start at 1 and reach 0 for the first time, corresponding to the state where all domains are folded. In Fig. 2.1(c), the measurement of τ_j starts when the reaction coordinate reaches 1 for the first time. For each excursion, one can also define the maximal site n_j that is reached before extinction occurs. After suitable processing of the signal, a single time trace generates many pairs (τ_j, n_j), $j = 1, \ldots, M$ and $M \gg 1$. In Section 2.4, we show how to recover all the transition rates of the birth-death chain from $\{\tau_j, n_j\}$.

We now briefly describe some of the existing methods used to interpret AFM
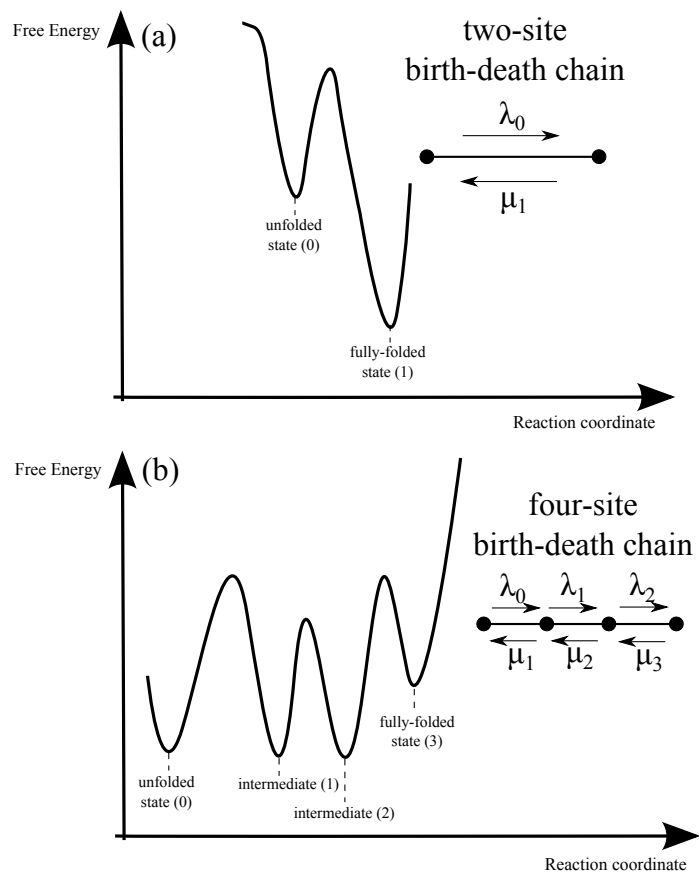
Figure 2.2: (a) Two-state and (b) Four-state Markov models for protein folding dynamics. Small proteins are usually described by a two-state model. Larger proteins may have multiple metastable states giving rise to longer birth-death chains.

data. One of the earliest methods is Bell's method [6] which assumes a force-dependent rate ansatz (of Arrhenius form) for the state of a domain or chemical bond. Providing that the force on the protein depends on the instantaneous deflection in the cantilever, Bell's method predicts rupture force distributions and survival probabilities for a chemical bond under the action of a force ramp. The method is used to solve the "forward" problem in the sense that rupture distributions are predicted from a potential well whose shape is known. In contrast, Dudko and co-authors [29] and Freund [40] essentially solved the inverse problem by inferring rate constants, features of the potential well and other related parameters from rupture force distributions.

Chang and co-authors [19] utlilized a path integral method that takes trajectory data (time traces) as input rather than force distributions. Based on non-parametric Bayesian estimation, the method makes very few assumptions about the underlying energy landscape and is able to simultaneously infer the energy landscape and an effective spatially-dependent diffusivity for the reaction coordinate.

Most of the above methods are mainly concerned with the AFM operating in force-ramp mode. However, in works such as [70, 37], the authors develop methodology to extract energy landscapes from data generated by AFMs in force-clamp mode. They treat the reaction coordinate using a Smoluchowksi framework to infer features of the energy landscape; this type of analysis dates back to Kramers's classic transition state theory [61] for chemical reactions. Finally, it is worth mentioning that if the reaction coordinate is treated as a Brownian random walker on an energy landscape, estimating the parameters of the resulting stochastic process from sample paths is a classic problem in statistics and control theory [36, 74].

The method described in this chapter is different because from the outset, we assume that the underlying stochastic process is a birth-death chain (and subsequently, transitions are always exponentially distributed). Estimation of parameters from sample paths of a birth-death chain is a classic problem [107, 56]. However, our goal is to estimate transition rates "mainly" from extinction times. Unfortunately, as discussed below, as well as in Chapter 4, using only extinction times results in a severely ill-posed

problem. Having access to maximal site data turns out to render the inference problem much better-posed.

The inclusion of maximal sites is important. The calculation of transition rates in a birth-death chain purely from extinction times essentially reduces to finding the best-fit coefficients and exponents to a given extinction time distribution. Such problems are highly ill-posed: a small amount of noise added to the curve can lead to a large change in the best-fit coefficients/exponents. Nevertheless, because fitting exponential modes to given data is one of the most commonly-arising inverse problems, it has a long history and has been investigated by many researchers: see for example [33, 62, 80, 81] and references within.

2.1.3 Continuous-Time Markov Chains (CTMC)

A Markov chain is a stochastic process that possesses the memoryless property, such that the conditional probability distribution of future states of the process depends only on the present state. We only consider the continuous-time (discrete-space) Markov chains in this context: see [92] for backgroud. The Poisson process, birth-death process, and chemical reaction networks are examples of such CTMC, in that these physical processes may only be in certain discrete states while the transition between states could occur at any nonnegative time, and they do not have "memory" of the past. Suppose that $\mathcal{X} = \{X(t) : t \geq 0\}$ is a continuous time process with X taking values in a discrete state space $\mathcal{S}$. Then $\mathcal{X}$ is a continuous time Markov chain if

$$\mathbb{P}[X(s+t) = j | X(u) = x(u), u \in [0, s), X(s) = i] = \mathbb{P}[X(s+t) = j | X(s) = i], \quad (2.1.2)$$

where $x = x(u)$ is the state at time u. In addition, we call $\mathcal{X}$ a time-homogeneous CTMC if

$$\mathbb{P}[X(s+t) = j | X(s) = i] = \mathbb{P}[X(t) = j | X(0) = i], \quad (2.1.3)$$

meaning that the probability distribution only depends on the time elapsed rather than the absolute time, and we will always assume time-homogeneity for a CTMC in this

12

dissertation. Transition from state i to state j within time t defines the transition probabilities,

$$p_{ij}(t) = \mathbb{P}[X(s+t) = j|X(s) = i]. \tag{2.1.4}$$

The transition times $\{T_i\}$, or the inter-event times, in a CTMC are exponentially distributed with parameter being the positive transition rates $\{\nu_i\}_{i \in \mathcal{S}}$ associated with specific states:

$$T_i \sim \exp(\nu_i), \text{ for } i \in \mathcal{S}. \tag{2.1.5}$$

Let's suppose $X(0) = i \in \mathcal{S}$, and T_i is the time for $\mathcal{X}$ to transition into some other state, then

$$\mathbb{P}[T_i > s + t|T_i > s]$$

$$= \mathbb{P}[X(0, s+t) = i|X(0, s) = i]$$

$$= \mathbb{P}[X(s, s+t) = i|X(s) = i] \qquad \text{(Markov property)}$$

$$= \mathbb{P}[X(0, t) = i|X(0) = i] \qquad \text{(Time-homogeneity)}$$

$$= \mathbb{P}[T_i > t]$$

Thus T_i is a memoryless random variable. Since the only memoryless continuous probability distribution is the exponential distribution, T_i must be exponentially distributed. The connection of a continuous time Markov chain with its discrete time counterpart is established through conditioning on the transition times. Suppose that X is at state i now, then the probabilities for the next state $j \in \mathcal{S}$ are given by

$$p_{ij}(T_i) = \mathbb{P}[X(T_i) = j|X(0) = i]. \tag{2.1.6}$$

This is exactly the transition probability in the discrete chain, and all elements from different i, j make up the transition matrix $\mathbf{P} = [p_{ij}]_{i,j \in \mathcal{S}}$ of that discrete-time Markov chain (DTMC). For this reason, such DTMC is called the "embedded chain" of the given CTMC. It follows that the embedded chain transition probabilities together with trainsition rates $\{\nu_i\}_{i \in \mathcal{S}}$ determine the distribution of the CTMC $\mathcal{X}$. For any pair of states i and j, let

$$q_{ij} = \nu_i p_{ij}. \tag{2.1.7}$$

13

Since ν_i is the rate at which the process makes a transition from state i and p_{ij} is the probability that this transition is into state j, then q_{ij} is the rate of transition of the process from i to j. These quantities q_{ij} are usually called the *instantaneous transition rates*. Since all transition probabilities at state i sum to one,

$$\nu_i = \sum_j \nu_i p_{ij} = \sum_j q_{ij} \qquad (2.1.8)$$

and

$$p_{ij} = \frac{q_{ij}}{\nu_i} = \frac{q_{ij}}{\sum_j q_{ij}}. \qquad (2.1.9)$$

We have the following lemma for the transition probabilities in a small time interval h: see [92] for more details.

Lemma 2.1. *For a CTMC $\mathcal{X} = \{X(t) : t \geq 0\}$ with instantaneous transition rates q_{ij},*

$$p_{ij}(h) = \mathbb{P}[X(t+h) = j | X(t) = i] = q_{ij}h + o(h), \qquad (2.1.10)$$

$$p_{ii}(h) = \mathbb{P}[X(t+h) = i | X(t) = i] = 1 - \nu_i h + o(h). \qquad (2.1.11)$$

Proof. Note that since the inter-event time T_i is exponentially distributed with parameter ν_i, we use Taylor expansion to get

$$
\begin{aligned}
p_{ij}(h) &= \mathbb{P}[X(t+h) = j | X(t) = i] \\
&= \mathbb{P}[T_i < h, X(T_i) = j | X(0) = i] \\
&= (1 - e^{-\nu_i h})p_{ij} \\
&= \nu_i h p_{ij} + o(h) \\
&= q_{ij}h + o(h).
\end{aligned}
$$

and

$$
\begin{aligned}
p_{ii}(h) &= \mathbb{P}[X(t+h) = i | X(t) = i] \\
&= \mathbb{P}[T_i > h | X(0) = i] \\
&= e^{-\nu_i h} \\
&= 1 - \nu_i h + o(h).
\end{aligned}
$$

$\square$

Given that $p_{ij}(0) = \delta_{ij}$, when the time interval h approaches zero,

$$\lim_{h \to 0^+} \frac{p_{ij}(h) - p_{ij}(0)}{h} = \begin{cases} q_{ij}, & \text{if } i \neq j \\ -\nu_i, & \text{if } i = j. \end{cases} \qquad (2.1.12)$$

Definition 2.1. *The infinitesimal generator of a CTMC $\mathcal{X} = \{X(t) : t \geq 0\}$ is a matrix Q with entries*

$$Q_{ij} = \begin{cases} q_{ij}, & \text{if } i \neq j \\ -\nu_i, & \text{if } i = j. \end{cases} \qquad (2.1.13)$$

Or,

$$Q = \begin{pmatrix} -\nu_1 & q_{12} & q_{13} & \cdots \\ q_{21} & -\nu_2 & q_{23} & \cdots \\ q_{31} & q_{32} & -\nu_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \qquad (2.1.14)$$

with rows of Q summing up to 0.

So far we have reviewed main concepts of a CTMC. We now introduce the birth-death process as a special kind of CTMC.

2.1.4 Birth-Death Process (BDP)

Definition 2.2. *A birth and death process is a continuous-time Markov chain with non-negative integer states $\mathcal{S} = \{0, 1, 2, 3, \ldots\}$ for which transitions from state n may go only to either state $n - 1$ or state $n + 1$. The rate at which the process moves from state n to state $n + 1$ is called the birth rate, denoted by λ_n; and the rate of moving from state n to state $n - 1$ is called the death rate, denoted by μ_n.*

From the definitions of the CTMC, we may directly write the infinitesimal generator matrix of a birth-death process. Notice that the death rate at state 0 is always zero; thus the process never enters a negative state.

Lemma 2.2. *The infinitesimal generator of a birth-death process on a semi-infinite domain $\mathcal{S} = \{0, 1, 2, 3, \ldots\}$ with birth rates $\{\lambda_n\}_{n=0}^{\infty}$ and death rates $\{\mu_n\}_{n=0}^{\infty}$ is a tri-diagonal matrix*

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & & & \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & & \\ & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}. \qquad (2.1.15)$$

For the rest part of this chapter, we consider the state space to be finite with $\mathcal{S} = \{0, 1, \ldots, N\}$ and assume that state 0 is an "absorption" state at which the birth-death process terminates because both birth and death rates here are zeros. In addition, the birth rate at site N is also zero, preventing the process from evolving further. Figure 2.3 shows an representation of such a birth-death process.
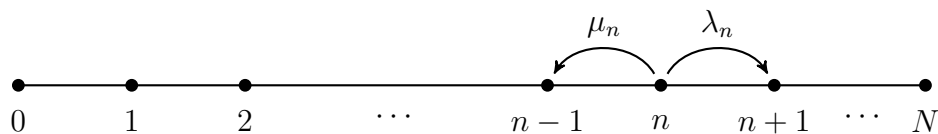
Figure 2.3: Finite birth-death chain with $N + 1$ sites, with $\lambda_N = 0$.

We study the "reflection" problem on this chain. Suppose that a birth-death process starts at site 1 in this chain of length $N+1$. Since state 0 is the only absorption state of this finite chain, the process will terminate at state 0 with probability one, possibly after visiting some other states, and the total time spent at these other states is called the *extinction time* (ET), or *exit time*. If this process is repeated with all birth and death rates unchanged (i.e. in the same underlying chain), we obtain a collection of extinction times which follows an intrinsic distribution. This distribution is the given data to our problem, and the goal is to determine the underlying birth and death rates $\{\lambda_n, \mu_n\}_{n=1}^{N}$ from it.

2.2 Analytical Inference of Birth-Death Rates in a Reflection Problem

In this section, let's first relax the assumption of the BDP starting at state 1 to a general state n. Suppose that $W_n(t)$ is the cumulative distribution function (c.d.f.) for the exit times of the particle starting at position n, and $S(n,t) = 1 - W_n(t)$ is the survival probability function corresponding to the c.d.f.

Suppose the c.d.f. $W_n(t)$ can be obtained exactly, so that there is no numerical error involved. We first derive the analytical inference of the birth and death rates under this assumption. Generally, we can obtain the exact solution to the c.d.f. if all transition rates λ_n and μ_n are known.

2.2.1 Survival Probabilities

Let $P(m,t|n)$ be the probability that the BDP is in state m at time t given that it was in state n at time 0. For $t < 0$, $P(m,t|n) = 0$; for $t > 0$, since

$$P(m, t + \delta t|n) = \lambda_n \delta t P(m,t|n+1) + \mu_n \delta t P(m,t|n-1) + (1 - \lambda_n \delta t - \mu_n \delta t) P(m,t|n),$$

we have the following ODEs:

$$P(m,t|n=0) = 0, \tag{2.2.1}$$

$$\frac{d}{dt} P(m,t|n) = \lambda_n P(m,t|n+1) - (\lambda_n + \mu_n) P(m,t|n) + \mu_n P(m,t|n-1),$$

$$\text{for } 1 \leq n \leq N - 1$$

$$\tag{2.2.2}$$

$$\frac{d}{dt} P(m,t|n=N) = -\mu_N P(m,t|N) + \mu_N P(m,t|N-1). \tag{2.2.3}$$

The survival probability $S(n,t)$ is the probability that the particle has not become extinct at time t:

$$S(n,t) = \sum_{m=1}^{N} P(m,t|n) \tag{2.2.4}$$

For $t < 0$, the process has not started yet, thus

$$S(n,t) = 1 \tag{2.2.5}$$

For $t > 0$:

$$S(0, t) = 0, \tag{2.2.6}$$

$$\frac{d}{dt}S(n, t) = \lambda_n S(n-1, t) - (\lambda_n + \mu_n)S(n, t) + \mu_n S(n+1, t), \; 1 \le n \le N - 1 \tag{2.2.7}$$

$$\frac{d}{dt}S(N, t) = -\mu_N S(N, t) + \mu_N S(N-1, t) \tag{2.2.8}$$

if we sum both sides of (2.2.1)–(2.2.3).

2.2.2 Equation for the Exit Time Distribution

Since the c.d.f. is related to the survival probability by $W_n(t) = 1 - S(n, t)$, we have the following equations for the c.d.f.:

$$W_0(t) = H(t), \tag{2.2.9}$$

$$\frac{dW_n}{dt} = \lambda_n W_{n+1} - (\lambda_n + \mu_n)W_n + \mu_n W_{n-1}, \quad 1 \le n \le N - 1 \tag{2.2.10}$$

$$\frac{dW_N}{dt} = -\mu_N W_N + \mu_N W_{N-1}. \tag{2.2.11}$$

where $H(t)$ is the Heaviside function. Furthermore, by the nature of a c.d.f.,

$$W_n(0) = 0, \quad n = 1, 2, 3, \ldots \tag{2.2.12}$$

and

$$W_n(t) = 0, \quad t < 0. \tag{2.2.13}$$

Equation (2.2.10) can be arranged so that

$$W_{n+1} = \left(1 + \frac{\mu_n}{\lambda_n}\right) W_n + \frac{1}{\lambda_n}\frac{dW_n}{dt} - \frac{\mu_n}{\lambda_n}W_{n-1}. \tag{2.2.14}$$

2.2.3 Exact Solution to the Birth-Death Process

For the birth-death process, let $\boldsymbol{W}(t) = (W_1(t), W_2(t), \ldots, W_N(t))^T$ be the c.d.f.s corresponding to all possible starting points, then the infinitesimal generator

18

matrix of the birth-death process is given by

$$Q = \begin{pmatrix} -(\lambda_1 + \mu_1) & \lambda_1 & & & & \\ \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & & & \\ & \mu_3 & -(\lambda_3 + \mu_3) & \lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{N-1} & -(\lambda_{N-1} + \mu_{N-1}) & \lambda_{N-1} \\ & & & & \mu_N & -\mu_N \end{pmatrix}.$$

$$(2.2.15)$$

Then the equations that govern the BDP are

$$\begin{cases} \dfrac{d}{dt} \boldsymbol{W}(t) = Q\boldsymbol{W}(t) + \mu_1 \boldsymbol{e}_1, \\ \boldsymbol{W}(t) = 0, \quad t \leq 0 \end{cases} \qquad (2.2.16)$$

where $\boldsymbol{e}_1 = (1, 0, \dots, 0)^T$. A particular solution to this system is $\boldsymbol{W}(t) = -\mu_1 Q^{-1} \boldsymbol{e}_1$. Let σ_k and $\boldsymbol{v}_k$ be the eigenvalues and corresponding eigenvectors of the matrix Q. Then the general solution to (2.2.16) is

$$\boldsymbol{W}(t) = \sum_{k=1}^{N} c_k \boldsymbol{v}_k e^{\sigma_k t} - \mu_1 Q^{-1} \boldsymbol{e}_1 = V \Sigma(t) \boldsymbol{c} - \mu_1 Q^{-1} \boldsymbol{e}_1, \qquad (2.2.17)$$

where $V = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_N]$ contains all eigenvectors of Q and $\Sigma(t)$ is a diagonal matrix satisfying $\Sigma_{kk} = e^{\sigma_k t}$, and $\boldsymbol{c} = (c_1, \dots, c_N)^T$ for some constants c_k. Applying the initial condition, we have $\boldsymbol{c} = \mu_1 V^{-1} Q^{-1} \boldsymbol{e}_1$, and hence $\boldsymbol{W}(t) = [V\Sigma(t)V^{-1} - I] \mu_1 Q^{-1} \boldsymbol{e}_1$. Expressing (2.2.17) element-wise, we have

$$W_n(t) = 1 + \sum_{k=1}^{N} \alpha_k^{(n)} e^{\sigma_k t}, \qquad (2.2.18)$$

where $\alpha_k^{(n)} = c_k v_k^{(n)}$, and $v_k^{(n)}$ is the n-th element in vector $\boldsymbol{v}_k$. This expression is a sum of exponential terms, and we call it a *hyperexponential* function [10]. Specifically, given the starting state is $n = 1$, we have

$$W_1(t) = 1 + \sum_{k=1}^{N} \alpha_k^{(1)} e^{\sigma_k t}. \qquad (2.2.19)$$

19

The coefficients $\{\alpha_k^{(n)}\}_{n=1}^N$ satisfy a recurrence relation which can be derived from (2.2.14),

$$1 + \sum_{k=1}^N \alpha_k^{(n+1)} e^{\sigma_k t} = \left(1 + \frac{\mu_n}{\lambda_n}\right)\left(1 + \sum_{k=1}^N \alpha_k^{(n)} e^{\sigma_k t}\right) + \frac{1}{\lambda_n}\sum_{k=1}^N \alpha_k^{(n)} \sigma_k e^{\sigma_k t}$$
$$- \frac{\mu_n}{\lambda_n}\left(1 + \sum_{k=1}^N \alpha_k^{(n-1)} e^{\sigma_k t}\right).$$

Matching the coefficients for the exponential terms, we get the recurrence relation:

$$\alpha_k^{(n+1)} = \frac{(\lambda_n + \mu_n + \sigma_k)\alpha_k^{(n)} - \mu_n \alpha_k^{(n-1)}}{\lambda_n}, \quad n \geq 1, \tag{2.2.20}$$

with $\alpha_k^{(0)} = 0$ for all k.

Going back to the problem where the BDP starts at state 1, our goal is to find $\{\lambda_k, \mu_k\}$ from $W_1(t)$. If the actual function $W_1(t)$ is known, λ_k and μ_k can be found from the Maclaurin series expansion of $W_k(t)$ in a recursive way given by (2.2.20). On the other hand, if $W_1(t)$ is only given at specific time nodes, then we can use the exponential fitting method from Chapter 4.

Theorem 2.1. *The c.d.f.s have the following Taylor series expansion,*

$$W_n(t) = \sum_{j=n}^{\infty} c_{n,j} t^j, \quad t \geq 0 \tag{2.2.21}$$

for some coefficients $c_{n,j}$.

Proof. Let's integrate the derivative of the master equation (2.2.10) on $[-\epsilon, \epsilon]$, for any $\epsilon > 0$. Note that $W_n(t) = 0$ for all $t < 0$, $W_n'(t)$ is bounded for $n > 0$ via (2.2.18), and $W_n'(t) = 0$ for $t < 0$. The jump of W_1' at $t = 0$ is

$$
\begin{aligned}
[W_1']_{t=0} &= \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} W_1''(t)\,dt \\
&= \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{0} \underbrace{[\lambda_1 W_2' - (\lambda_1 + \mu_1)W_1']}_{=0}\,dt + \lim_{\epsilon \to 0^+} \int_{0}^{\epsilon} [\lambda_1 W_2' - (\lambda_1 + \mu_1)W_1']\,dt \\
&\quad + \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} \mu_1 W_0'(t)\,dt \\
&= \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} \mu_1 \delta(t)\,dt = \mu_1. \tag{2.2.22}
\end{aligned}
$$

In addition, since $W_n(0) = 0$ for $n = 1, \ldots, N$, it follows from (2.2.10) that

$$W_j'(0) = 0, \text{ for all } j > 1. \tag{2.2.23}$$

Similarly, let's consider the second derivatives. The jump of W_2'' at $t = 0$ is

$$
\begin{aligned}
[W_2'']_{t=0} &= \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} W_2'''(t) \, dt \\
&= \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{0} \underbrace{[\lambda_2 W_3'' - (\lambda_2 + \mu_2) W_2'']}_{=0} \, dt + \lim_{\epsilon \to 0^+} \int_{0}^{\epsilon} [\lambda_2 W_3'' - (\lambda_2 + \mu_2) W_2''] \, dt \\
&\quad + \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} \mu_2 W_1''(t) \, dt \\
&= \mu_2 \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\epsilon} W_1''(t) \, dt \\
&= \mu_2 [W_1']_{t=0} = \mu_1 \mu_2, \tag{2.2.24}
\end{aligned}
$$

and

$$W_j''(0) = 0, \text{ for all } j > 2. \tag{2.2.25}$$

Inductively, one can prove that the jump in the n-th derivative of W_n at $t = 0$ is

$$[W_n^{(n)}]_{t=0} = \prod_{i=1}^{n} \mu_i, \tag{2.2.26}$$

and

$$W_j^{(n)}(0) = 0, \text{ for all } j > n. \tag{2.2.27}$$

Motivated by these equations under the assumption that W_n is differentiable, the c.d.f.s thus have the following Taylor series expansion:

$$W_n(t) = \sum_{j=n}^{\infty} c_{n,j} t^j \tag{2.2.28}$$

$\square$

Plugging the expansion (2.2.21) into (2.2.10), we have

$$
\begin{aligned}
&n c_{n,n} t^{n-1} + c_{n,n+1}(n+1) t^n + c_{n,n+2}(n+2) t^{n+1} + \ldots \\
&= \lambda_n \left[c_{n+1,n+1} t^{n+1} + c_{n+1,n+2} t^{n+2} + c_{n+1,n+3} t^{n+3} + \ldots \right] \\
&\quad - (\lambda_n + \mu_n) \left[c_{n,n} t^n + c_{n,n+1} t^{n+1} + c_{n,n+2} t^{n+2} + \ldots \right] \\
&\quad + \mu_n \left[c_{n-1,n-1} t^{n-1} + c_{n-1,n} t^n + c_{n-1,n+1} t^{n+1} + \ldots \right]. \tag{2.2.29}
\end{aligned}
$$

Equating coefficients for t^{n-1}:

$$nc_{n,n} = \mu_n c_{n-1,n-1}, \quad 1 \leq n \leq N. \tag{2.2.30}$$

Equating coefficients for $t^{n+m}(m \geq 1)$:

$$c_{n+1,n+m} = \frac{(n+m+1)c_{n,n+m+1} + (\lambda_n + \mu_n)c_{n,n+m} - \mu_n c_{n-1,n+m}}{\lambda_n} \tag{2.2.31}$$

Specifically, in each step we need to compute the following two entries:

$$c_{n,n} = \frac{(n+1)c_{n-1,n+1} + (\lambda_{n-1} + \mu_{n-1})c_{n-1,n} - \mu_{n-1}c_{n-2,n}}{\lambda_{n-1}}, \tag{2.2.32}$$

$$c_{n,n+1} = \frac{(n+2)c_{n-1,n+2} + (\lambda_{n-1} + \mu_{n-1})c_{n-1,n+1} - \mu_{n-1}c_{n-2,n+1}}{\lambda_{n-1}}, \tag{2.2.33}$$

where $c_{0,0} = 1$ and $c_{0,i} = 0, i \geq 1$.

The relations of these coefficients are shown in the matrix (2.2.34) with the first two rows are given. The tableaux is updated using the triangular stencil [14] given above, and each blue entry is obtained from three red entries above it:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ & c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,n-1} & c_{1,n} & c_{1,n+1} & \dots \\ & & c_{2,2} & c_{2,3} & \dots & c_{2,n-1} & c_{2,n} & c_{2,n+1} & \dots \\ & & & \ddots & & \vdots & \vdots & & \\ & & & & & c_{n-2,n-1} & c_{n-2,n} & c_{n-2,n+1} & \dots \\ & & & & & c_{n-1,n-1} & c_{n-1,n} & c_{n-1,n+1} & \dots \\ & & & & & & c_{n,n} & c_{n,n+1} & \dots \\ & & & & & & & \ddots & \ddots \end{bmatrix} \tag{2.2.34}$$

Using (2.2.30), we have

$$\mu_1 = c_{1,1}. \tag{2.2.35}$$

It is easy to show that

$$c_{n,n} = \frac{[W_n^{(n)}]_{t=0}}{n!} = \frac{1}{n!} \prod_{k=1}^{n} \mu_k. \tag{2.2.36}$$

Comparing the coefficient of t^n in eq. (2.2.29), we find that

$$c_{n,n+1} = \frac{-(\lambda_n + \mu_n)c_{n,n} + \mu_n c_{n-1,n}}{n+1} \tag{2.2.37}$$

In general, it can be proved by induction that

$$M_n \equiv \prod_{k=1}^{n} \mu_k. \tag{2.2.38}$$

$$c_{n,n+1} = -\frac{M_n}{(n+1)!}\left(\sum_{k=1}^{n}\lambda_k + \sum_{k=1}^{n}\mu_k\right), \tag{2.2.39}$$

2.2.4 Detailed inference steps

We use a layer stripping method to infer the original birth and death rates. Data is assumed to be perfect, so that the functional form of $W_1(t)$ is known.

1. Initiate $M_0 = 1$;

2. Find the coefficient $\alpha_k^{(n)}$ from $W_n(t) = 1 + \sum_{k=1}^{N}\alpha_k^{(n)}e^{\sigma_k t}$. If $n = 1$, these coefficients are known since the function $W_1(t)$ is known; if $n \geq 2$, they are obtained through the recurrence relations in equation (2.2.20).

3. Find $c_{n,n}$ and $c_{n,n+1}$ by the exact c.d.f.. Assuming the form in (2.2.21), we will use Maclaurin series expansion to match the coefficients of t^n with $c_{n,n}$, and match the coefficients of t^{n+1} with $c_{n,n+1}$.

4. Compute μ_n and λ_n:

$$\mu_n = \frac{n!\, c_{n,n}}{M_{n-1}}, \tag{2.2.40}$$

$$\lambda_n = -\frac{(n+1)!\, c_{n,n+1}}{M_n} - \left(\sum_{k=1}^{n-1}\lambda_k + \sum_{m=1}^{n}\mu_m\right) \tag{2.2.41}$$

In summary, given the c.d.f. $W_1(t)$ exactly, we may infer the birth and death rates exactly. In Algorithm 1, we described the procedure of inference when the coefficients of Taylor expansion are explicitly given, while in Algorithm 2, we infer the transition rates from $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ in $W_1(t)$.

Algorithm 1 Inference of birth and death rates given Taylor coefficients of $W_1(t)$

Input: $\{c_{1,j}\}_{j=1}^{\infty}$.

Initialize: $c_{0,1} = 0$ and $c_{1,1} = \mu_1$; $M_0 \leftarrow 1, \mu_k \leftarrow 0, \lambda_k \leftarrow 0, k = 1, \ldots, N$; $n = 1$

1: $\mu_1 = c_{1,1}, \ M_1 = \mu_1, \ \lambda_1 = -\dfrac{2! \, c_{1,2}}{M_1} - \mu_1$

2: **while** $n < N$ **do**

3: $\quad c_{n,n} = \dfrac{(n+1)c_{n-1,n+1} + (\lambda_{n-1} + \mu_{n-1})c_{n-1,n} - \mu_{n-1}c_{n-2,n}}{\lambda_{n-1}}$

4: $\quad c_{n,n+1} = \dfrac{(n+2)c_{n-1,n+2} + (\lambda_{n-1} + \mu_{n-1})c_{n-1,n+1} - \mu_{n-1}c_{n-2,n+1}}{\lambda_{n-1}}$

5: $\quad \mu_n = \dfrac{n! \, c_{n,n}}{M_{n-1}}$

6: $\quad M_n = \mu_n M_{n-1}$

7: $\quad \lambda_n = -\dfrac{(n+1)! \, c_{n,n+1}}{M_n} - \left(\sum_{k=1}^{n-1} \lambda_k + \sum_{m=1}^{n} \mu_m \right)$

8: $\quad n \leftarrow n + 1$

9: **end while**

Output: $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$

Algorithm 2 Inference of birth and death rates given functional form of c.d.f. $W_1(t)$
___
Input: $\boldsymbol{\alpha} = \{\alpha_k\}_{k=1}^N$ and $\boldsymbol{\sigma} = \{\sigma_k\}_{k=1}^N$

Initialize: $M_0 \leftarrow 1, \mu_k \leftarrow 0, \lambda_k \leftarrow 0, \alpha_k^{(0)} \leftarrow 0, \alpha_k^{(1)} \leftarrow \alpha_k, k = 1, \ldots, N; n = 1$

1: $c_{1,1} = \sum_{k=1}^N \alpha_k^{(1)} \sigma_k$ and $c_{1,2} = \sum_{k=1}^N \dfrac{\alpha_k^{(1)} \sigma_k^2}{2!}$

2: $\mu_1 = \dfrac{c_{1,1}}{M_0}, \ M_1 = M_0 \mu_1, \ \lambda_1 = -\dfrac{2! \, c_{1,2}}{M_1} - \mu_1$

3: **while** $n < N$ **do**

4: $\qquad \alpha_k^{(n+1)} = \dfrac{(\lambda_n + \mu_n + \sigma_k)\alpha_k^{(n)} - \mu_n \alpha_k^{(n-1)}}{\lambda_n}, \ k = 1, \ldots, N$

5: $\qquad c_{n,n} = \sum_{k=1}^N \dfrac{\alpha_k^{(n)} \sigma_k^n}{n!}$ and $c_{n,n+1} = \sum_{k=1}^N \dfrac{\alpha_k^{(n)} \sigma_k^{n+1}}{(n+1)!}$

6: $\qquad \mu_n = \dfrac{n! \, c_{n,n}}{M_{n-1}}$

7: $\qquad M_n = \mu_n M_{n-1}$

8: $\qquad \lambda_n = -\dfrac{(n+1)! \, c_{n,n+1}}{M_n} - \left(\sum_{k=1}^{n-1} \lambda_k + \sum_{m=1}^n \mu_m\right)$

9: $\qquad n \leftarrow n + 1$

10: **end while**
___
Output: $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$
___

This is the end of analytical inference of the BDP. In the next two sections, we make a closer connection to actual data, where the c.d.f. $W_1(t)$ is obtained numerically and is contaminated with noise. In this case, Algorithms 1 and 2 do not work in practice, for fitting the hyper-exponential function to noisy data is an ill-posed problem, and the Taylor coefficients cannot be accurately computed in the noisy setting. This motivates the need for more data, and we will explore numerical schemes to tackle such problems.

2.3   Numerical Estimation of Transition Rates from Extinction Times

Starting from here, the problem becomes more realistic in that the input data is not an exact c.d.f. anymore. Instead, the data is a collection of extinction times obtained from reiterated numerical simulation of the BDP. We will introduce the basic steps of generating the BDP first, and discuss the inference later in the section.

2.3.1 Monte-Carlo Simulation of the BDP

In order to simulate the BDP, we use the idea of the embedded-chain. The simulation reduces to two separate steps: sampling the exponentially distributed inter-event times, and determining the next jump according to birth and death rates. We generate the exponential random variable by inverse sampling [25].

Lemma 2.3 (Inverse Sampling). *If a U has a uniform distribution on $[0,1]$ and if X has a cumulative distribution F_X, then the random variable $F_X^{-1}(U)$ has the same distribution as X.*

Proof. By definition of c.d.f. and uniform distribution,

$$\mathbb{P}[F_X^{-1}(U) \le x] = \mathbb{P}[U \le F_X(x)] = F_X(x). \tag{2.3.1}$$

we have that the random variable $F_X^{-1}(U)$ has F_X as its c.d.f.. □

Since the inter-event time (with rates λ) has c.d.f. $F(t) = 1 - e^{-\lambda t}$, its inverse function is $F^{-1}(u) = -\frac{1}{\lambda}\ln(1-u)$, and we can easily sample them in this manner. The details of generating an extinction time of a BDP of $N+1$ states is in Algorithm 3. In practice, repeating this experiment a large amount of times yields a collection of extinction times, and we compute the numerical c.d.f. with them.

We consider the same problem discussed in Section 2.2, i.e., to recover the birth and death rate of a chain of length $N+1$. Instead of working with analytical expressions, we now generate a large dataset of extinction times $\{\tau_i\}_{i=1}^M$ by Algorithm 3. As the number of ETs in the sample approaches infinity, $\{\tau_i\}$ should have distribution of $W_1(t)$ in (2.2.19). Considering the noise in the data generation process, and the limited number of ETs from simulation, we obtain an approximate distribution of $W_1(t)$. The goal is to find $\{\lambda_n, \mu_n\}$ from these ETs. We first derive the method from 4-site chain ($N = 3$), and later generalize to $(N+1)$-site chain.

The following subsections are closely related to exponential fitting techniques which are discussed in details in Chapter 4. We will first introduce the Variable projection method and the modified Prony method here.

Algorithm 3 Generating the Extinction Time of a Birth-Death Process of $N + 1$ states.

Input: Birth rates $\{\lambda_n\}_{n=1}^N$, death rates $\{\mu_n\}_{n=1}^N$.

Initialize: Initial state $i = 1$, time $t = 0$, iteration $n = 0$, maximum iteration $n_{\max}$.

1: **while** $n < n_{\max}$ **do**
2:      Generate a standard uniform random number $u \sim \text{uniform}(0, 1)$.
3:      Sample exponential inter-event time $T = -\dfrac{\ln(1 - u)}{\lambda_i + \mu_i}$.
4:      Generate another standard uniform random number $v \sim \text{uniform}(0, 1)$.
5:      **if** $v < \dfrac{\lambda_i}{\lambda_i + \mu_i}$ **then**
6:         $i = i + 1$
7:      **else**
8:         $i = i - 1$
9:      **end if**
10:     Update time $t = t + T$
11:     **if** $i == 0$ **then** break
12:     **end if**
13: **end while**
14: **if** $i \neq 0$ **then** discard current result
15:     Return
16: **end if**

Output: Extinction time t.

2.3.2 Variable Projection

The Variable projection (Varpro) method is used to solve separable nonlinear least squares problems [43], in which the data is a linear combination of nonlinear functions of multiple parameters. In general, such a separable nonlinear least squares problem seeks to minimize the residual vector

$$r_i(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = y_i - \sum_{j=1}^{n} \alpha_j \phi_j(\boldsymbol{\sigma}; t_i), \text{ for } i = 1, \dots, n. \qquad (2.3.2)$$

In matrix notation we have,

$$r(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \|\boldsymbol{y} - \Phi(\boldsymbol{\sigma})\boldsymbol{\alpha}\|_2^2, \qquad (2.3.3)$$

where $\boldsymbol{y}$ is the vector of measurements, $\boldsymbol{\alpha}$ is the vector of linear coefficients, $\boldsymbol{\sigma}$ is the vector of nonlinear coefficients, and the j-th column of the matrix $\Phi(\boldsymbol{\sigma})$ is the vector $(\phi_j(\boldsymbol{\sigma}, t_1), ..., \phi_j(\boldsymbol{\sigma}, t_N))^T$, i.e., the nonlinear function $\phi(\boldsymbol{\sigma})$ evaluated at all time values $\{t_i\}_{i=1}^N$. Note that in the special case of exponential fitting, we have the nonlinear function as $\phi_j(\boldsymbol{\sigma}; t_i) = e^{-\sigma_j t_i}$.

If the nonlinear parameters $\boldsymbol{\sigma}$ are known, then the linear parameters can be found by

$$\boldsymbol{\alpha} = \Phi(\boldsymbol{\sigma})^\dagger \boldsymbol{y}, \qquad (2.3.4)$$

where $\Phi(\boldsymbol{\sigma})^\dagger$ is the pseudo-inverse of the matrix $\Phi(\boldsymbol{\sigma})$. Substituting (2.3.4) into (2.3.3) gives

$$\min_{\boldsymbol{\sigma}} \|(\mathbf{I} - \Phi(\boldsymbol{\sigma})\Phi(\boldsymbol{\sigma})^\dagger)\boldsymbol{y}\|_2^2, \qquad (2.3.5)$$

where the linear parameters α are eliminated from the optimization. The method first minimizes (2.3.5) to get nonlinear parameters $\boldsymbol{\sigma}$, and then solves (2.3.4) for linear parameters $\boldsymbol{\alpha}$. We then define

$$\psi(\boldsymbol{\sigma}) = \|(\mathbf{I} - \Phi(\boldsymbol{\sigma})\Phi(\boldsymbol{\sigma})^\dagger)\boldsymbol{y}\|_2^2, \qquad (2.3.6)$$

as the *Variable Projection functional*, since the first term $\mathbf{I} - \Phi(\boldsymbol{\sigma})\Phi(\boldsymbol{\sigma})^\dagger$ is the projector onto the orthogonal complement of the column space of $\Phi(\boldsymbol{\sigma})$, or $\mathbf{P}_{\Phi(\boldsymbol{\sigma})}^\perp$.

A naive approach to solve the nonlinear least squares problem (2.3.3) is minimization of the residual r over both $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ simultaneously. The variable projection method is an improvement over algorithms that aim to minimize (2.3.3) because it breaks down the minimization problem into two smaller subproblems that solve $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ sequentially. The Varpro algorithm will converge in fewer iterations compared to minimizing the original functional (2.3.3), and also guarantees that the set of stationary points of the two approaches are the same. Algorithm 4 summarizes the steps for the variable projection method.

Algorithm 4 Variable Projection for separable nonlinear least squares problems
---
Input: $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ at regular time nodes $t_k = k \times \delta t$.

1: Build the matrix $\Phi(\boldsymbol{\sigma})$ whose (i, j) entry is $\phi_j(\boldsymbol{\sigma}; t_i)$. Specifically, $\phi_j(\boldsymbol{\sigma}; t_i) = e^{-\sigma_j t_i}$ in exponential fitting problems.

2: Minimize the Variable Projection functional $\psi(\boldsymbol{\sigma})$ in (2.3.6) to get the best exponents $\boldsymbol{\sigma}$.

3: Solve the coefficients $\boldsymbol{\alpha} = \Phi(\boldsymbol{\sigma})^\dagger \boldsymbol{y}$.

Output: $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$.

2.3.3 Osborne's Modified Prony Method

The modified Prony method was introduced by Osborne and Smyth in [80, 81]. Given a differential equation

$$\sum_{k=1}^{p+1} \xi_k D^{k-1} y = 0, \tag{2.3.7}$$

where D is the differential operator, the modified Prony method gives estimation for the coefficients ξ_k. The solution to (2.3.7) could be a sum of complex exponentials, damped and undamped sinusoids, and real exponentials. We only focus on the following specific problem of fitting the sum of p real exponential functions

$$y(t) = \sum_{j=1}^{p} \alpha_j e^{-\sigma_j t} \tag{2.3.8}$$

to experimental data evaluated at n data points, where α_j and σ_j are assumed to be real, the σ_j distinct and nonnegative. In addition, without loss of generality, the domain for t is assumed to be the unit interval and $t_i = i/n$, $i = 1, \ldots, n$.

Difference and recurrence equations

The differential equation (2.3.7) can be represented in three forms: the ODE, the difference equation and the recurrence equation. The modified Prony method is based on these three forms of equations.

Let Π be the forward shift operator defined by $\Pi y(t) = y\left(t + \frac{1}{n}\right)$, and let Δ be the divided difference operator $\Delta = n(\Pi - I)$. Suppose the polynomial

$$p_\xi(z) = \sum_{k=1}^{p+1} \xi_k z^{k-1} \tag{2.3.9}$$

has distinct roots $-\sigma_j$ for $j = 1, \ldots, p$. Then the three types of equations are as follows.

1. The ODE:
$$\prod_{j=1}^{p}(D + \sigma_j I)y(t) = 0 \tag{2.3.10}$$

2. Difference equation. Since $\Delta e^{-\sigma_j t} = -e^{-\sigma_j t}\zeta_j$ with $\zeta_j = n(1 - e^{-\sigma_j/n})$, we have

$$\prod_{j=1}^{p}(\Delta + \zeta_j I)y(t) = 0, \tag{2.3.11}$$

which can be written as
$$\sum_{k=1}^{p+1} \gamma_k \Delta^{k-1} y(t) = 0 \tag{2.3.12}$$

for some suitable choice of γ_k. The $\{\gamma_k\}$ are called the difference form Prony parameters. The ζ_j and γ_k represent discrete approximations to σ_j and ξ_k in the sense that $\zeta_j \to \sigma_j$ and $\gamma_k \to \xi_k$ as $n \to \infty$.

3. Recurrence equation. Since $\Pi e^{-\sigma_j t} = e^{-\sigma_j t}\rho_j$ with $\rho_j = e^{-\sigma_j/n}$, we have

$$\prod_{j=1}^{p}(\Pi - \rho_j I)y(t) = 0, \tag{2.3.13}$$

which can be written as
$$\sum_{k=1}^{p+1} \delta_k \Pi^{k-1} y(t) = 0 \tag{2.3.14}$$

30

for some suitable choice of δ_k. The $\{\delta_k\}$ are called the recurrence form Prony parameters.

Let c_k satisfy the following equation in which the difference relation and the recurrence relation are equivalent,

$$\sum_{k=1}^{p+1} \gamma_k \Delta^{k-1} = \sum_{k=1}^{p+1} c_k \Pi^{k-1}. \tag{2.3.15}$$

Then $\boldsymbol{c}$ could be solved as

$$c_j = \sum_{k=j}^{p+1} (-1)^{k-j} \binom{k-1}{j-1} n^{k-1} \gamma_k. \tag{2.3.16}$$

This can also be represented by matrices, $\boldsymbol{c} = \mathbf{U}\boldsymbol{\gamma}$, where $\mathbf{U}$ is the nonsingular matrix

$$\mathbf{U} = \begin{pmatrix} 1 & -1 & 1 & \cdots & & \cdots & (-1)^p \\ & 1 & -2 & 3 & & & \vdots \\ & & 1 & -3 & 6 & & \vdots \\ & & & \ddots & & \ddots & \vdots \\ & & & & \ddots & & \binom{p}{2} \\ & & & & & 1 & -\binom{p}{1} \\ & & & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & n & & & \\ & & \ddots & & \\ & & & & n^p \end{pmatrix} \tag{2.3.17}$$

and $\boldsymbol{c} = (c_1, \ldots, c_{p+1})^T$. The original recurrence parameter $\boldsymbol{\delta}$ is defined so that $\delta_{p+1} = 1$. We may easily observe that $\boldsymbol{c}$ and $\boldsymbol{\delta}$ are rescaled versions of each other: $\boldsymbol{\delta} = \boldsymbol{c}/c_{p+1}$.

The algorithm

Denote $y_i = y(t_i)$, $i = 1, \ldots, n$. Let $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ and X_δ be the $n \times (n - p)$ matrix

$$\mathbf{X}_\delta = \begin{pmatrix} \delta_1 & & & \\ \delta_2 & \delta_1 & & \\ \vdots & \delta_2 & \ddots & \\ \delta_{p+1} & \vdots & \ddots & \delta_1 \\ & \delta_{p+1} & & \delta_2 \\ & & \ddots & \vdots \\ & & & \delta_{p+1} \end{pmatrix}. \tag{2.3.18}$$

Rewriting the recurrence equation (2.3.14), we have

$$\mathbf{X}_\delta^T \boldsymbol{y} = 0, \tag{2.3.19}$$

Alternatively, we may write the rescaled version of the matrix above to be

$$\mathbf{X} = \begin{pmatrix} c_1 & & & \\ c_2 & c_1 & & \\ \vdots & c_2 & \ddots & \\ c_{p+1} & \vdots & \ddots & c_1 \\ & c_{p+1} & & c_2 \\ & & \ddots & \vdots \\ & & & c_{p+1} \end{pmatrix}. \tag{2.3.20}$$

and

$$\mathbf{X}^T \boldsymbol{y} = 0, \tag{2.3.21}$$

which corresponds to the difference equation (2.3.12).

Next, we will use the Varpro method to fit the exponential function. Let $\mathbf{A}$ be the $n \times p$ matrix function of σ with $\mathbf{A}_{ij} = e^{-\sigma_j t_i}$, and

$$\boldsymbol{y} = \mathbf{A}(\boldsymbol{\sigma})\boldsymbol{\alpha}, \tag{2.3.22}$$

32

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^T$. Then $\mathbf{A}$ is orthogonal to all columns of $\mathbf{X}_\delta$ and $\mathbf{X}$. The variable projection functional defined in (2.3.6) can be written as

$$\psi(\boldsymbol{\sigma}) = \boldsymbol{y}^T(\mathbf{I} - \mathbf{P_A})\boldsymbol{y}, \tag{2.3.23}$$

where $\mathbf{P_A} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the orthogonal projection onto the column space of $\mathbf{A}$. Since $\mathbf{A}$ is orthogonal to $\mathbf{X}$, we can reparametrize to the Prony parameters as

$$\psi(\boldsymbol{\gamma}) = \boldsymbol{y}^T\mathbf{P_X}\boldsymbol{y} = \boldsymbol{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{2.3.24}$$

where $\mathbf{P_X}$ is the orthogonal projection onto the common column space of $\mathbf{X}$ and $\mathbf{X}_\delta$.

Now, the least squares problem is equivalent to minimization of ψ with respect to γ. In fact, one can show that the derivative of ψ can be written

$$\frac{\partial \psi}{\partial \boldsymbol{\gamma}} = 2\mathbf{B}(\boldsymbol{\gamma})\boldsymbol{\gamma}, \tag{2.3.25}$$

where $\mathbf{B}$ is the symmetric $(p+1) \times (p+1)$ matrix function of γ with elements

$$\mathbf{B}_{ij} = \boldsymbol{y}^T\mathbf{X}_i(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_j^T\boldsymbol{y} - \boldsymbol{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_i^T\mathbf{X}_j(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{2.3.26}$$

and where $\mathbf{X}$ is defined in (2.3.20) and $\mathbf{X}_j = \partial\mathbf{X}/\partial\gamma_j$.

The modified Prony method minimizes $\psi(\gamma)$ in (2.3.24) by repeatedly solving (2.3.25) until $\frac{\partial \psi}{\partial \boldsymbol{\gamma}} = 0$. In fact, it is formulated using Lagrange multipliers subject to the constraint $\boldsymbol{\gamma}^T\boldsymbol{\gamma} = 1$ so that a necessary condition for minimization to (2.3.24) is

$$[\mathbf{B}(\boldsymbol{\gamma}) - \lambda\mathbf{I}]\boldsymbol{\gamma} = 0. \tag{2.3.27}$$

The method solves (2.3.27) iteratively by updating $\boldsymbol{\gamma}$ and λ until the Lagrange multiplier λ is sufficiently small. At the end of iterations, we get $\boldsymbol{\gamma}$ as an approximation to the coefficients $\boldsymbol{\xi}$ in ODE (2.3.7). We then solve for the roots of the characteristic polynomial to get $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$, and $\boldsymbol{\alpha}$ finally follows from (2.3.22). In practice, however, we can alternatively implement the modified Prony method by directly minimizing (2.3.24) using algorithms like quasi-Newton or Nelder-Mead. Algorithm 5 summarizes the steps for the modified Prony method.

Algorithm 5 Modified Prony method

Input: $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ at regular time nodes $t_k = k \times \delta t$.

1: Build the matrix $\mathbf{X}$ as function of $\boldsymbol{\gamma}$.

2: Start with an initial $\boldsymbol{\gamma}_0$, and a tolerance ϵ.

3: **while** $\lambda^{(n)} > \epsilon \|\mathbf{B}\|$ **do**

4:      Solve $[\mathbf{B}(\boldsymbol{\gamma}^{(n)}) - \lambda^{(n+1)}\mathbf{I}]\boldsymbol{\gamma}^{(n+1)} = 0$, with $\boldsymbol{\gamma}^{(n+1)T}\boldsymbol{\gamma}^{(n+1)} = 1$

5:      $\lambda^{(n+1)}$ is the closest to zero of such solutions.

6: **end while**

7: Find roots of characteristic polynomial of (2.3.7) to get $\boldsymbol{\zeta}$, using coefficient $\boldsymbol{\gamma}_n$ obtained in the last step.

8: Exponents are given by $\sigma_j = -\dfrac{1}{n} \ln \left(1 - \dfrac{\zeta_j}{n} \right)$

9: Solve the coefficients $\boldsymbol{\alpha} = \mathbf{A}(\boldsymbol{\sigma})^\dagger \boldsymbol{y}$.

Output: $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$.

2.3.4   ODEs for three sites $N = 3$

Suppose that $w(t) = W'(t)$ is the p.d.f. of extinction times. We apply Laplace transform to (2.2.14) and get the following system:

$$\tilde{w}_0(s) = 1, \tag{2.3.28}$$

$$s\tilde{w}_1(s) = \lambda_1 \tilde{w}_2 - (\lambda_1 + \mu_1)\tilde{w}_1 + \mu_1 \tilde{w}_0, \tag{2.3.29}$$

$$s\tilde{w}_2(s) = \lambda_2 \tilde{w}_3 - (\lambda_2 + \mu_2)\tilde{w}_2 + \mu_2 \tilde{w}_1, \tag{2.3.30}$$

$$s\tilde{w}_3(s) = -\mu_3 \tilde{w}_3 + \mu_3 \tilde{w}_2. \tag{2.3.31}$$

where $\tilde{w}(s)$ is the Laplace transform of $w(t)$, This system of algebraic equations can be solved for $\tilde{w}_1$:

$$(\xi_1 + \xi_2 s + \xi_3 s^2 + s^3)\tilde{w}_1(s) = \xi_1 + \xi_0 s + \xi_{-1} s^2 \tag{2.3.32}$$

where the $\{\xi_k\}$ are defined in terms of birth and death rates

$$\xi_3 = \lambda_1 + \lambda_2 + \mu_1 + \mu_2 + \mu_3, \tag{2.3.33}$$

$$\xi_2 = \lambda_1\lambda_2 + \lambda_2\mu_1 + \mu_1\mu_2 + \lambda_1\mu_3 + \mu_1\mu_3 + \mu_2\mu_3, \tag{2.3.34}$$

$$\xi_1 = \mu_1\mu_2\mu_3, \tag{2.3.35}$$

$$\xi_0 = \mu_1\mu_2 + \mu_1\mu_3 + \lambda_2\mu_1, \tag{2.3.36}$$

$$\xi_{-1} = \mu_1. \tag{2.3.37}$$

We will now deal with the vector $\boldsymbol{\xi}$ instead of $\{\boldsymbol{\lambda}, \boldsymbol{\mu}\}$, because the transition rates can be inferred once $\boldsymbol{\xi}$ is known.

Directly computing the birth and death rates from these equations using the modified Prony method is subject to instability. For better results, we consider adding some moments of the extinction times as constraints. Let T_k be the k-th moment of exit times starting at site $n = 1$, and take the k-th derivative of $\tilde{w}_1(s) = \frac{d^k}{ds^k} \int_0^\infty e^{-st} w_1(t)\, dt = (-1)^k \int_0^\infty t^k e^{-st} w_1(t)\, dt$. Setting $s = 0$, we have

$$\tilde{w}_1^{(k)}(0) = (-1)^k \int_0^\infty t^k w_1(t)\, dt = (-1)^k\, T_k \tag{2.3.38}$$

2.3.5   General ODE for any N

In general, if N is arbitrary in the birth-death process, the Laplace-transformed p.d.f. can be expressed as:

$$F(s) = \left(\sum_{j=0}^{N} \xi_{j+1} s^j \right) \tilde{w}_1(s) - \sum_{k=-1}^{N-2} \xi_{-k} s^{k+1} = 0, \tag{2.3.39}$$

where $\xi_{N+1} = 1$ always holds. We may iteratively take the derivative of (2.3.39) with respect to s, set $s = 0$, and use the relation in (2.3.38). Notice that the definition for the parameters $\{\xi_j\}$ are different for chains with different lengths, and these derivative equations form a set of constraints on the moments of the data. Given the number of sites N (finite), we could represent all constraints in a matrix form:

$$G\boldsymbol{\xi} = \boldsymbol{b}, \tag{2.3.40}$$

where $\boldsymbol{\xi} = (\xi_{2-N}, \ldots, \xi_{-1}, \xi_0, \xi_1, \ldots, \xi_N)^T$, and $\boldsymbol{b} = (0, \ldots, 0, -N!)^T$, and

$$G = \left[\begin{array}{cccccc|cccccccc} 0 & \cdots & 0 & 0 & 0 & 0 & -1! & -T_1 & 1! & 0 & \cdots & & & 0 \\ 0 & \cdots & 0 & 0 & 0 & -2! & 0 & T_2 & -2!T_1 & 2! & 0 & \cdots & & 0 \\ 0 & \cdots & 0 & 0 & -3! & 0 & 0 & -T_3 & 3T_2 & -3!T_1 & 3! & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -4! & 0 & 0 & 0 & T_4 & -4T_3 & 12T_2 & -4!T_1 & 4! & 0 & \cdots & 0 \\ 0 & \cdots & -5! & 0 & 0 & 0 & 0 & -T_5 & 5T_4 & -20T_3 & 60T_2 & -5!T_1 & 5! & \cdots & 0 \\ \vdots & \reflectbox{$\ddots$} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ -N! & \cdots & 0 & 0 & 0 & 0 & 0 & g_{N1} & g_{N2} & g_{N3} & g_{N4} & g_{N5} & \cdots & g_{NN} & g_{N,N+1} \end{array}\right]. \tag{2.3.41}$$

The left block of G is an anti-diagonal matrix and the right block of G has elements

$$g_{ij} = (-1)^{i-j+1} T_{i-j+1} \prod_{k=0}^{j-2} (i-k), \quad \text{for } 1 \le j \le N+1. \tag{2.3.42}$$

2.3.6 Estimation of Rates by Minimization

In order to improve the result using the nonlinear least squares methods, we will use the constraints introduced in (2.3.40). The Modified Prony's Method minimizes the Variable projection functional with respect to the parameter $\boldsymbol{\gamma}$, which is an approximation to coefficients of the characteristic polynomial $\boldsymbol{\xi}$. The details of this method were described in Section 2.3.3. However, we will directly work with the parameter $\boldsymbol{\xi}$ by defining a new objective function

$$\tilde{\psi}(\xi_1, \xi_2, \xi_3) = \psi(\gamma_1, \gamma_2, \gamma_3) = \psi\left(\xi_1 - \frac{\xi_1\xi_3}{2n}, \xi_2 - \frac{\xi_2\xi_3 - 3\xi_1}{2n}, \xi_3 - \frac{\xi_3^2 - 2\xi_2}{2n}\right). \tag{2.3.43}$$

Consider the constraints matrix G in (2.3.41) for three sites, we have

$$\begin{bmatrix} 0 & -1 & -T_1 & 1 & 0 \\ -2 & 0 & T_2 & -2T_1 & 2 \\ 0 & 0 & -T_3 & 3T_2 & -6T_1 \end{bmatrix} \begin{bmatrix} \xi_{-1} \\ \xi_0 \\ \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -6 \end{bmatrix} \tag{2.3.44}$$

Note that we can compute any moment of the data in theory to improve the result, but the high-order moments have larger errors. Hence we only choose the first three moments here. This is an underdetermined linear system, so we may find a solution

(with least 2-norm) such that $G\boldsymbol{x}_0 = \boldsymbol{b}$. Suppose the singular value decomposition of G is $G = USV^*$, then the last two columns $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of V will be the basis for the nullspace of G. Then

$$\boldsymbol{\xi} = \boldsymbol{x}_0 + c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 \tag{2.3.45}$$

is a solution to (2.3.44). We can then search in the coefficients (c_1, c_2) such that $\tilde{\psi}$ is minimized. The true solution is just the last three elements in $\boldsymbol{\xi}$.

When we search for the coefficients $\boldsymbol{c}$, it is helpful to include the gradient. By the chain rule, we have

$$\frac{\partial \tilde{\psi}}{\partial \boldsymbol{c}} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{\xi}} \frac{\partial \psi}{\partial \boldsymbol{\gamma}}, \tag{2.3.46}$$

where

$$\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} 1 - \frac{\xi_3}{2n} & \frac{3}{2n} & 0 \\[2mm] 0 & 1 - \frac{\xi_3}{2n} & \frac{1}{n} \\[2mm] -\frac{\xi_1}{2n} & -\frac{\xi_2}{2n} & 1 - \frac{\xi_3}{n} \\[2mm] 0 & 0 & 0 \\[2mm] 0 & 0 & 0 \end{bmatrix}, \tag{2.3.47}$$

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{c}} = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \end{bmatrix}. \tag{2.3.48}$$

where n is the number of time nodes used in [0,1], and the last derivative term in the product $\frac{\partial \tilde{\psi}}{\partial \boldsymbol{\gamma}}$ is given by the modified Prony method in (2.3.25).

Once the estimations for $\boldsymbol{\xi}$ is computed, one can follow the flow chart 2.4 to obtain the transition rates $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, through a series of intermediate computations with respect to different variables shown in the chart.

2.3.7 Numerical Results

We explore the error behavior of the following six methods that compute the birth/death rates. Notice that the last 3 methods include moments while the first three do not.
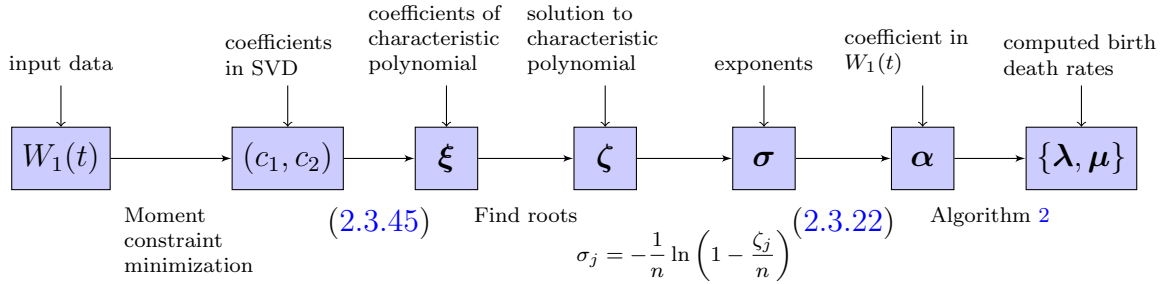
Figure 2.4: Flow chart of the entire inference procedure for $N = 3$. The input data is c.d.f. $W_1(t)$ given at equispaced time nodes. One first finds the SVD of the matrix G in (2.3.41), and coefficients (c_1, c_2). Then the coefficients of characteristic polynomial for the corresponding ODE can be recovered as $\boldsymbol{\xi}$. By solving the roots of this polynomial, one obtains $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$ as specified in the modified Prony method. The coefficients of hyperexponential $\boldsymbol{\alpha}$ will be then solved by ordinary least squares. Finally, the transition rates are calculated via Algorithm 2 using $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$.

(1) Direct minimization of $\tilde{\psi}$ via *fminunc* based on BFGS[1] quasi-newton method, starting at random initial point, without any additional information.

(2) Minimization of $\tilde{\psi}$ via *fminunc* starting at random initial point, with the gradient of objective function. This is Osborne's modified Prony method.

(3) Minimization of $\tilde{\psi}$ via *fminsearch* based on Nelder-Mead simplex algorithm (gradient free method), starting at random initial point, without any additional information.

(4) Minimization of $\tilde{\psi}$ via the new proposed moments method with *fminsearch*, starting at random initial point, with moments for the first passage times.

(5) Minimization of $\tilde{\psi}$ via the new proposed moments method with *fminunc* without gradient, starting at random initial point, with moments for the first passage times.

(6) Minimization of $\tilde{\psi}$ via the new proposed moments method with *fminunc* plus gradient, starting at random initial point, with moments for the first passage times.

Note that there is a subtle difference between method (2) and the modified Prony method: we just need to minimize $\tilde{\psi}$ whereas Osborne's modified Prony method sets up a nonlinear eigenvalue problem for $\tilde{\psi}$.

---

[1] The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is an iterative method for solving unconstrained nonlinear optimization problems [35].

2.3.7.1 Three-Site BDP ($N = 2$)

A total of 10^6 birth-death processes, starting from site 1 and ending in site 0, are simulated in MATLAB, with rates $\boldsymbol{\lambda} = (0.65, 0), \boldsymbol{\mu} = (0.6, 0.48)$. The c.d.f. and moments are estimated from these exit times. For the moment constraints method implementation, we choose 256 equispaced time nodes in the interval $(0, 1]$.

Consider $\boldsymbol{\nu} = (\lambda_1, \mu_1, \mu_2)$ as a new vector. We sample random birth and death rates that are within the interval $(0, 1]$, and denote the exact rates by $\boldsymbol{\nu}^*$. Also note that $\lambda_2 = 0$ is assumed and there is no need to infer this parameter. Figure 2.5 shows how the vector $\boldsymbol{\nu}$ changes after applying each of the six methods. The triangular markers represent the final relative distances (relative error) to the exact solution. It is clear from these plots that all six methods work properly and give accurate results when $N = 2$. We may use any of these methods to infer the birth-death rates in the three-site chain.

2.3.7.2 Four-Site BDP ($N = 3$)

A total of 10^6 birth-death processes, starting from site 1 and ending in site 0, are simulated in MATLAB, with rates $\boldsymbol{\lambda} = (0.65, 0.35, 0), \boldsymbol{\mu} = (0.6, 0.48, 0.3)$. The CDF and moments are estimated from these exit times, with an error of 0.2% in CDF and 2.4% in the first 5 moments. For the moment constraints method implementation, we choose 256 equispaced time nodes in the interval $(0, 1]$.

Consider $\boldsymbol{\nu} = (\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3)$ as a new vector. We sample random birth and death rates that are within the interval $(0, 1]$, and denote the exact rates by ν^*. Also note that $\lambda_3 = 0$ is assumed and there is no need to infer this parameter. Figure 2.6 shows how the vector $\boldsymbol{\nu}$ changes after applying each of the six methods. The blue crosses represent the results where the birth-death rates improved their accuracy (got closer to $\boldsymbol{\nu}^*$) while the magenta squares represent the results that reduced their accuracy (got further from $\boldsymbol{\nu}^*$). Methods that do not use moments generally yield bad results and incorporating the moments usually improves the inference.

Figure 2.5: Methods comparison for three sites BDP ($N = 2$). All methods have the same initial condition in each iteration. In each plot, x-axis is the relative distance from the initial guess to the exact transition rates $\boldsymbol{\nu}^*$, and y-axis is the relative distance from the resulting transition rates to $\boldsymbol{\nu}^*$. The red lines are $y = x$. The triangles are all below the red line, indicating that the methods have actually decreased the distance to the exact solution. All methods work well, insensitive of initial guesses.

We further investigate the best performing method (6) in Figure 2.7. It shows a histogram of transition rates calculated from method (6). Although over 90% of the final transition rates show an improvement over the initial guesses, the method is sensitive to initial conditions and does not infer all the sites with uniform accuracy, due to error accumulation in larger sites. The other methods perform even worse.

As the birth-death chain grows longer, these method become unreliable and unstable, if we only have pure extinction times from the BDP. This motivates the need for additional data, which we will discuss next.

2.4 Numerical Estimation of Transition Rates from Conditional Extinction Times

In this section, our problem set up is different in that we assume we also have access to the maximal site for each trajectory, as well as the extinction time. This additional data turns out to render the inference problem much better-posed.

2.4.1 Governing Equations for the Birth Death Process

Consider the same birth-death chain as in previous sections which is on a lattice with sites labeled $\{0, 1, 2, \ldots, N\}$, where N is known and finite: see Figure 2.8. A particle starts at site 1 and executes a random walk which we write as $\mathcal{X} = X(t)$: $X(t)$ is a random walk on the non-negative integers. At site i, the rightward (leftward) hopping rate is λ_i (μ_i). When the particle reaches site 0, we record the time of extinction. When this experiment is repeated many times, we may use the resulting data to compute the *extinction time distribution* $W(t)$.

The matrix that determines this process is defined as

$$A^{(N)} = \begin{pmatrix} -(\lambda_1+\mu_1) & \mu_2 & & & \\ \lambda_1 & -(\lambda_2+\mu_2) & \mu_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \lambda_{N-2} & -(\lambda_{N-1}+\mu_{N-1}) & \mu_N \\ & & & \lambda_{N-1} & -(\lambda_N+\mu_N) \end{pmatrix} \tag{2.4.1}$$

Figure 2.6: Methods comparison for four sites BDP ($N = 3$). All methods have the same initial condition in each iteration. The black lines are $y = x$. Blue crosses stand for results that are better than initial guess, and magenta squares correspond to results that get worse. The percentage on top of each plot is the proportion of results that actually get closer to exact rates.
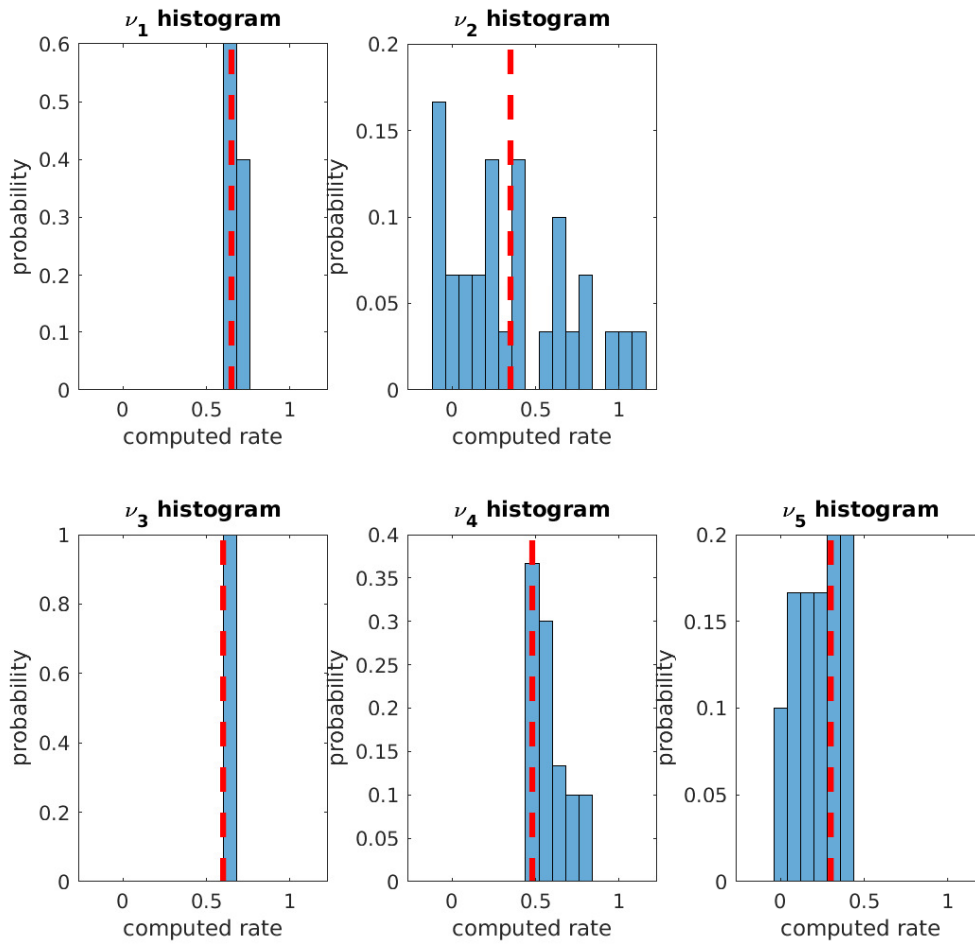
Figure 2.7: Histogram for transition rates $\boldsymbol{\nu} = (\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3)$ computed by method (6), which takes gradient and moments into consideration. Vertical dashed red lines are the exact transition rates. The rates are mostly accurate for $\nu_1 = \lambda_1$ and $\nu_3 = \mu_1$, but not so for the other three transition rates.
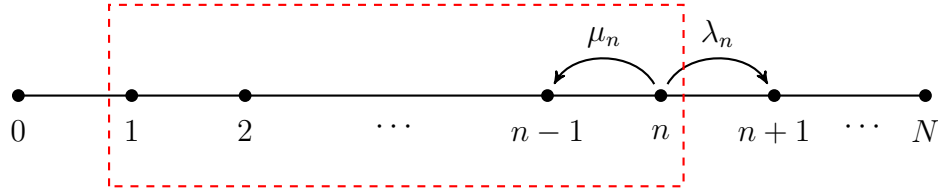
Figure 2.8: Birth death chain with $N + 1$ sites, with $\lambda_N = 0$. Our algorithm uses the extinction times of the process given that the positions of particles never exceed site n (always remain in the dashed box).

with $\lambda_N = 0$. Note that the superscript to this square matrix corresponds to its size. It turns out that this is the transpose of the infinitesimal generator matrix defined in (2.1.14) with first row and column removed, since this BDP satisfies a forward equation, and that definition is from a backward equation. If $P_k(t) = \mathbb{P}[X(t) = k]$ is the probability that the random walker is in site k at time t, then the vector of probabilities $\boldsymbol{P} = (P_1, \ldots, P_N)^T$ satisfies the forward master equation

$$\dot{\boldsymbol{P}} = A^{(N)}\boldsymbol{P}, \quad \boldsymbol{P}(0) = (1, 0, \ldots, 0)^T, \tag{2.4.2}$$

and it is a simple matter to find $W(t)$ from $P_1(t)$ (see Section 2.4.2).

When N is known, finding all the transition rates $\{\lambda_i, \mu_i\}_{1 \leq i \leq N}$ from $W(t)$ amounts to an exponential fitting problem, which is very ill-conditioned [104]. To overcome this difficulty, we assume that for every trajectory $X_j(t)$ we also record the *maximal site* of the particle, n_j. The maximal site for the random walker is simply the largest site number that it attains before exiting. By grouping trajectories according to their maximal site and computing the statistics of the extinction times of each group, we are able to accurately infer transition rates for birth death chains of length 11 from about 5×10^7 extinction times with relative error of a few percent.

To be more precise, if $X_j(t)$ is the jth trajectory, then τ_j is the jth extinction time defined as $\tau_j = \inf\{\tau : X_j(\tau) = 0\}$ and $n_j = \max_{0 \leq t \leq \tau_j} X_j(t)$ is the maximal site of the jth trajectory. If $S_n = \{\tau_j : n_j \leq n\}$ is the set of extinction times corresponding to trajectories whose maximal site does not exceed n, then for a finite sample of trajectories, we have $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_N$. The algorithm that we propose essentially takes as input $|S_n|$ and $\bar{S}_n$ (cardinality and mean of S_n) to infer λ_n and μ_n for each n.

Random walkers that exit from the dashed red box in Figure 2.8 can exit at site 0 or $n+1$. We informally call the times that correspond to exit at site 0 $(n+1)$ "left" ("right") extinction times. Then, the distribution of extinction times for the birth-death process, conditioned on trajectories not exceeding site n is identical to the left extinction time distribution out of the sublattice $\{1, \ldots, n\}$. By finding analytical expressions for the moments of this distribution and matching them to the observed moments, we may infer the transition rates on the lattice. This forms the basis of our method.

2.4.2 Extinction times and Probability Fluxes

We assume that sites 0 and $n+1$ are absorbing in the sense that if the particle reaches site 0 or $n+1$ ("exits"), it stays at these sites for all time. We use the superscript n to distinguish the subproblem from the entire chain. Define the probability that the random walker is at site k at time t as

$$P_k^{(n)}(t) = \mathbb{P}[X(t) = k]. \tag{2.4.3}$$

If we take the $n \times n$ leading principal submatrix of $A^{(N)}$, then it follows that the conditional probabilities $\boldsymbol{P}^{(n)} = (P_1^{(n)}, P_2^{(n)}, \ldots, P_n^{(n)})^T$ satisfy the forward master equations

$$\dot{\boldsymbol{P}}^{(n)} = A^{(n)} \boldsymbol{P}^{(n)}, \quad \boldsymbol{P}^{(n)}(0) = \boldsymbol{e}_1^{(n)} = \underbrace{(1, 0, \ldots, 0)^T}_{n \text{ elements}}, \tag{2.4.4}$$

for $1 \leq n \leq N$. Eq. (2.4.4) is the starting point for the reconstruction process.

Now we introduce the two random variables

• $E^{(n)} \in \{0, n+1\}$: a binary random variable which represents the exit site of the random walker on the sublattice $\{1, \ldots, n\}$:

$$\mathbb{P}[E^{(n)} = 0] = \Pi^{(n)}, \tag{2.4.5}$$
$$\mathbb{P}[E^{(n)} = n+1] = \Pi_*^{(n)},$$

and $\Pi^{(n)} + \Pi_*^{(n)} = 1$.

• $T^{(n)}$: the extinction time of the random walker, defined to be the time at which the walker arrives either at site 0 or $n+1$ for the first time. Conditioning on $E^{(n)}$, let the density of extinction times $T^{(n)}$ be $w_L^{(n)}(t)$ and $w_R^{(n)}(t)$:

$$
\begin{aligned}
P(t \leq T^{(n)} \leq t + dt | E^{(n)} = 0) &= w_L^{(n)}(t)dt, \\
P(t \leq T^{(n)} \leq t + dt | E^{(n)} = n+1) &= w_R^{(n)}(t)dt,
\end{aligned}
$$

and we let $W_L^{(n)}(t)$ and $W_R^{(n)}(t)$ be the corresponding CDFs.

Now we show how extinction times are related to probability fluxes. The probability of the particle being at site 0 at time $t + dt$ is given by

$$P_0^{(n)}(t + dt) = P_1^{(n)}(t)\mu_1 dt + P_0^{(n)}(t) \times 1. \tag{2.4.6}$$

Note that the probability of hopping left in time dt from site 1 is $\mu_1 dt$ and the probability of staying at site 0 in time dt is 1 since site 0 is absorbing. Eq. (2.4.6) implies that

$$\frac{dP_0^{(n)}(t)}{dt} = \mu_1 P_1^{(n)}(t). \tag{2.4.7}$$

Lemma 2.4. *The flux out of site 1, $\mu_1 P_1^{(n)}(t)$, and the left extinction time density $w_L^{(n)}(t)$ are related through*

$$\mu_1 P_1^{(n)}(t) = w_L^{(n)}(t)\Pi^{(n)}, \tag{2.4.8}$$

where μ_1 is the death rate from site 1 and $\Pi^{(n)}$ is defined in eq. (2.4.5).

Proof. If the random walker is at site 0 at time t, it must have arrived there either at t or before. Then

$$
\begin{aligned}
P_0^{(n)}(t) &= \mathbb{P}[T^{(n)} \leq t, E^{(n)} = 0], \\
\Rightarrow P_0^{(n)}(t) &= \mathbb{P}[T \leq t | E^{(n)} = 0] \, \mathbb{P}[E^{(n)} = 0], \\
\Rightarrow P_0^{(n)}(t) &= W_L^{(n)}(t)\Pi^{(n)}, \\
\Rightarrow \frac{d}{dt}P_0^{(n)}(t) &= w_L^{(n)}(t)\Pi^{(n)}, \\
\Rightarrow \mu_1 P_1^{(n)}(t) &= w_L^{(n)}(t)\Pi^{(n)},
\end{aligned}
$$

using eq. (2.4.7). $\qquad\square$

2.4.3 Algorithm for reconstructing transition rates

Our algorithm for transition rate reconstruction requires the following as input: for each n, the fraction of random walks that exit and whose maximal site does not exceed n; and the mean extinction time for these conditional random walks. For each $n \geq 1$, $\Pi^{(n)}$ and $\mathbb{E}[T^{(n)}|E^{(n)} = 0] \equiv M^{(n)}$ yield $\{\lambda_n, \mu_n\}$: see Fig. 2.9.

Figure 2.9: Flow chart of the algorithm presented in this paper. $\Pi^{(n)}$ is the probability of left exit and $M^{(n)}$ is the mean of the extinction times, all conditioned on that the particles remain in the domain of $\{1, \ldots, n\}$ before exiting. At each site, a pair of birth and death rate at that site is recovered.

2.4.3.1 Inference of μ_1 and λ_1

In the first step, we recover the birth and death rates at site 1. Note that $\boldsymbol{P}^{(1)}(t)$ only contains a single element, and the forward master equation can be written as

$$\dot{P}^{(1)} = A^{(1)} P^{(1)},$$

with

$$A^{(1)} = -(\lambda_1 + \mu_1) \qquad \text{and} \qquad P^{(1)}(0) = 1.$$

This simple ODE has solution

$$P^{(1)}(t) = e^{-(\lambda_1 + \mu_1)t}.$$

Suppose we only consider left extinction times, with $n = 1$, generated by all trajectories that directly arrive at site 0 from site 1. These extinction times are exponentially distributed with parameter $\lambda_1 + \mu_1$:

$$W_L^{(1)}(t) = \mathbb{P}\left[T^{(1)} \leq t | E^{(1)} = 0\right] = 1 - e^{-(\lambda_1 + \mu_1)t}.$$

It follows from the property of exponential distribution that

$$\lambda_1 + \mu_1 = \frac{1}{\mathbb{E}\left[T^{(1)}|E^{(1)} = 0\right]} = \frac{1}{M^{(1)}}. \tag{2.4.9}$$

In the next step, we use (2.4.8) to get

$$\mu_1 P_1^{(1)}(t) = w_L^{(1)}(t)\Pi^{(1)}$$

$$\Rightarrow \mu_1 = \frac{\Pi^{(1)}}{\int_0^\infty P_1^{(1)}(t')\,dt'}$$

$$\Rightarrow \mu_1 = \Pi^{(1)}(\lambda_1 + \mu_1) = \frac{\Pi^{(1)}}{M^{(1)}} \tag{2.4.10}$$

$$\Rightarrow \lambda_1 = \frac{1 - \Pi^{(1)}}{M^{(1)}}. \tag{2.4.11}$$

We now have obtained the forms of μ_1 and λ_1 in terms of $\Pi^{(1)}$ and $M^{(1)}$.

2.4.3.2   Inference of μ_2 and λ_2

The forward master equations are for $\boldsymbol{P}^{(2)}(t)$ are

$$\dot{\boldsymbol{P}}^{(2)} = A^{(2)}\boldsymbol{P}^{(2)}$$

where

$$A^{(2)} = \begin{pmatrix} -(\lambda_1 + \mu_1) & \mu_2 \\ \lambda_1 & -(\lambda_2 + \mu_2) \end{pmatrix}.$$

By (2.4.8), we have that

$$\mu_1 P_1^{(2)}(t) = w_L^{(2)}(t)\Pi^{(n)},$$

$$\Rightarrow \mu_1 \int_0^\infty P_1^{(2)}(t')dt' = \Pi^{(2)},$$

$$\Rightarrow w_L^{(2)}(t) = \frac{P_1^{(2)}(t)}{\int_0^\infty P_1^{(2)}(t')dt'}.$$

We now introduce the Laplace transform $\mathcal{L}\{P(t)\} = \tilde{P}(s)$. Then the transformed equation for $P_1^{(2)}(t)$ satisfies

$$\left[s^2 + \xi_2^{(2)}s + \xi_1^{(2)}\right]\tilde{P}_1^{(2)}(s) = s + \eta_2^{(2)}, \tag{2.4.12}$$

where $\xi_2^{(2)} = \lambda_1 + \mu_1 + \lambda_2 + \mu_2$, $\xi_1^{(2)} = \lambda_1\lambda_2 + \mu_1\mu_2 + \lambda_2\mu_1$ and $\eta_2^{(2)} = \lambda_2 + \mu_2$. Taking derivatives with respect to s, we have

$$\left[2s + \xi_2^{(2)}\right]\tilde{P}_1^{(2)}(s) + \left[s^2 + \xi_2^{(2)}s + \xi_1^{(2)}\right]\frac{d\tilde{P}_1^{(2)}(s)}{ds} = 1, \tag{2.4.13}$$

and when $s = 0$,

$$(\lambda_1 \lambda_2 + \mu_1 \mu_2 + \lambda_2 \mu_1) \tilde{P}_1^{(2)}(0) - (\lambda_2 + \mu_2) \;=\; 0, \quad (2.4.14)$$

$$(\lambda_1 + \mu_1 + \lambda_2 + \mu_2) \tilde{P}_1^{(2)}(0) + (\lambda_1 \lambda_2 + \mu_1 \mu_2 + \lambda_2 \mu_1) \left. \frac{d\tilde{P}_1^{(2)}(s)}{ds} \right|_{s=0} \;=\; 1. \quad (2.4.15)$$

Eqs. (2.4.14) and (2.4.15) can be rewritten as

$$\left[\frac{\lambda_1 + \mu_1}{\mu_1} \Pi^{(2)} - 1 \right] \lambda_2 + \left[\Pi^{(2)} - 1 \right] \mu_2 \;=\; 0, \quad (2.4.16)$$

$$\left[1 - M^{(2)}(\lambda_1 + \mu_1) \right] \lambda_2 + \left[1 - M^{(2)} \mu_1 \right] \mu_2 \;=\; \frac{\mu_1}{\Pi^{(2)}} - \lambda_1 - \mu_1, \quad (2.4.17)$$

a linear system for λ_2, μ_2 where

$$\Pi^{(2)} = \mu_1 \tilde{P}_1^{(2)}(0) \quad \text{and} \quad M^{(2)} = -\frac{\mu_1}{\Pi^{(2)}} \left. \frac{d\tilde{P}_1^{(2)}(s)}{ds} \right|_{s=0}. \quad (2.4.18)$$

Assuming λ_1 and μ_1 are known from the $n = 1$ case, solving eqs. (2.4.16) and (2.4.17) allows us to compute λ_2 and μ_2 from the conditional moments $\Pi^{(2)}$ and $M^{(2)}$.

2.4.3.3 Inference of μ_n and λ_n for $n \geq 3$

Now we consider the n-th site after computing the birth and death rates for the first $n-1$ sites. The Laplace transformed ODEs of $\tilde{\boldsymbol{P}}^{(n)}(s) = [\tilde{P}_1^{(n)}(s), \ldots, \tilde{P}_n^{(n)}(s)]^T$ can be represented in the following matrix form:

$$(sI_n - A^{(n)}) \tilde{\boldsymbol{P}}^{(n)}(s) = \boldsymbol{e}_1^{(n)} \quad (2.4.19)$$

where $\boldsymbol{e}_1^{(n)} = [1, 0, \ldots, 0]^T$ has n elements and I_n is the identity matrix of size $n \times n$. Whenever s is not an eigenvalue of $A^{(n)}$, we have that $\tilde{\boldsymbol{P}}^{(n)}(s) = (sI_n - A^{(n)})^{-1} \boldsymbol{e}_1^{(n)}$. Let the characteristic polynomial of $A^{(n)}$ be

$$q(s; \lambda_1, \mu_1) = s^n + \xi_n^{(n)} s^{n-1} + \cdots + \xi_2^{(n)} s + \xi_1^{(n)} = \det(sI_n - A^{(n)}), \quad (2.4.20)$$

where the $\{\xi_i^{(n)}\}$ are the coefficients of the characteristic polynomial of $A^{(n)}$, and they can be written as functions of birth and death rates:

$$\xi_i^{(n)} = \xi_i^{(n)}(\lambda_1, \ldots, \lambda_n; \mu_1, \ldots, \mu_n). \quad (2.4.21)$$

In addition, define another set of coefficients

$$\eta_i^{(n)} = \xi_i^{(n)}(\lambda_1 = 0, \lambda_2, \ldots, \lambda_n; \mu_1 = 0, \mu_2, \ldots, \mu_n). \qquad (2.4.22)$$

Finally, we note that the coefficient of of s^n in (2.4.20) is 1 and for notational convenience, define

$$\xi_{n+1}^{(n)} = 1; \quad \xi_{n+k}^{(n)} = 0 \text{ for } k \geq 2$$

$$\eta_{n+1}^{(n)} = 1; \quad \eta_{n+k}^{(n)} = 0 \text{ for } k \geq 2$$

Lemma 2.5. *The Laplace-transformed probability $\tilde{P}_1^{(n)}(s)$, the first element in $\tilde{\mathbf{P}}^{(n)}(s)$ from eq (2.4.19), is a rational function in s satisfying*

$$\left[s^n + \xi_n^{(n)} s^{n-1} + \cdots + \xi_2^{(n)} s + \xi_1^{(n)} \right] \tilde{P}_1^{(n)}(s) = s^{n-1} + \eta_n^{(n)} s^{n-2} + \cdots + \eta_3^{(n)} s + \eta_2^{(n)}. \quad (2.4.23)$$

for some constants $\{\xi_i^{(n)}\}_{i=1}^n$ and $\{\eta_i^{(n)}\}_{i=2}^n$.

Proof. Let $\hat{A}^{(k-1)}$ be the $(k-1) \times (k-1)$ submatrix of $A^{(k)}$ with the first row and first column removed, so that

$$\hat{A}^{(k-1)} = \begin{pmatrix} -(\lambda_2 + \mu_2) & \mu_3 & & & & \\ \lambda_2 & -(\lambda_3 + \mu_3) & \mu_4 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \lambda_{k-2} & -(\lambda_{k-1} + \mu_{k-1}) & \mu_k \\ & & & \lambda_{k-1} & -(\lambda_k + \mu_k) \end{pmatrix}. \quad (2.4.24)$$

By Cramer's rule [22] and the definition of the determinant in terms of its cofactor expansion, the first element of the solution vector $\tilde{\boldsymbol{P}}^{(n)}(s)$ can be calculated as

$$\tilde{P}_1^{(n)}(s) = \frac{\det \left(\begin{array}{c|c} 1 & -\mu_2 [\boldsymbol{e}_1^{(n-1)}]^T \\ \hline \mathbf{0} & sI_{n-1} - \hat{A}^{(n-1)} \end{array} \right)}{\det(sI_n - A^{(n)})} = \frac{\det(sI_{n-1} - \hat{A}^{(n-1)})}{\det(sI_n - A^{(n)})}.$$

Now introduce the polynomial

$$\det(sI_{n-1} - \hat{A}^{(n-1)}) = s^{n-1} + c_{n-1} s^{n-2} + c_{n-2} s^{n-3} + \ldots + c_2 s + c_1, \quad (2.4.25)$$

50

for some coefficients $c_{n-1}, c_{n-2}, \ldots, c_1$. Then it is clear that

$$q(s; 0, 0) = s^n + \eta_n^{(n)} s^{n-1} + \cdots + \eta_3^{(n)} s^2 + \eta_2^{(n)} s + \eta_1^{(n)}, \tag{2.4.26}$$

$$= \det\left(\begin{array}{c|c} s & -\mu_2 [e_1^{(n-1)}]^T \\ \hline 0 & sI_{n-1} - \hat{A}^{(n-1)} \end{array}\right), \quad \text{by definition of characteristic polynomial}$$

$$= s \det(sI_{n-1} - \hat{A}^{(n-1)}), \tag{2.4.27}$$

from the definitions in eqs. (2.4.20) and (2.4.22) and the fact that the $(1, 1)$ and $(2, 1)$ entries of $A^{(n)}$ are zero when $\lambda_1 = \mu_1 = 0$. Because $q(s = 0; \lambda_1 = 0, \mu_1 = 0) = 0$, it follows that $\eta_1^{(n)} = 0$. Eqs. (2.4.26) and (2.4.27) imply that

$$s^n + \eta_n^{(n)} s^{n-1} + \cdots + \eta_3^{(n)} s^2 + \eta_2^{(n)} s + \underbrace{\eta_1^{(n)}}_{=0} = s^n + c_{n-1} s^{n-1} + c_{n-2} s^{n-2} + \ldots + c_2 s^2 + c_1 s.$$

Comparing coefficients, we find that $\eta_n^{(n)} = c_{n-1}$, $\eta_{n-1}^{(n)} = c_{n-2}, \ldots, \eta_2^{(n)} = c_1$, so that eq. (2.4.25) becomes

$$\det(sI_{n-1} - \hat{A}^{(n-1)}) = s^{n-1} + \eta_n^{(n)} s^{n-2} + \eta_{n-1}^{(n)} s^{n-3} + \ldots + \eta_3^{(n)} s + \eta_2^{(n)},$$

$$\Rightarrow \tilde{P}_1^{(n)}(s) = \frac{s^{n-1} + \eta_n^{(n)} s^{n-2} + \cdots + \eta_3^{(n)} s + \eta_2^{(n)}}{s^n + \xi_n^{(n)} s^{n-1} + \cdots + \xi_2^{(n)} s + \xi_1^{(n)}}. \tag{2.4.28}$$

$\square$

Since the expression $\frac{\Pi^{(n)}}{\mu_1}$ appears frequently in later analysis, we define the notation

$$r^{(n)} = \frac{\Pi^{(n)}}{\mu_1} \tag{2.4.29}$$

and use it in the analysis below.

Lemma 2.6 (Moment and Exit Probability). *Let* $M^{(n)} \equiv \mathbb{E}\left[T^{(n)} | E^{(n)} = 0\right]$, *then*

$$\Pi^{(n)} = \mu_1 \tilde{P}_1^{(n)}(0), \tag{2.4.30}$$

$$M^{(n)} = -\frac{1}{r^{(n)}} \left.\frac{d\tilde{P}_1^{(n)}(s)}{ds}\right|_{s=0}. \tag{2.4.31}$$

Proof. The Laplace transform of $P_1^{(n)}(t)$ is $\tilde{P}_1^{(n)}(s) = \int_0^\infty e^{-st} P_1^{(n)}(t)\, dt$. Integrating both sides of (2.4.8), we have that

$$\Pi^{(n)} = \mu_1 \int_0^\infty P_1^{(n)}(t')\, dt'$$

$$\Rightarrow w_L^{(n)}(t) = \frac{P_1^{(n)}(t)}{\int_0^\infty P_1^{(n)}(t')\, dt'}.$$

Therefore, eq. (2.4.30) holds:

$$\tilde{P}_1^{(n)}(0) = \int_0^\infty P_1^{(n)}(t)\, dt = r^{(n)}.$$

If we differentiate the Laplace transform $\tilde{P}_1^{(n)}(s)$, then (2.4.31) is validated by

$$\frac{d\tilde{P}_1^{(n)}(s)}{ds} = -\int_0^\infty t e^{-st} P_1^{(n)}(t)\, dt$$

$$\Rightarrow \left.\frac{d\tilde{P}_1^{(n)}(s)}{ds}\right|_{s=0} = -\int_0^\infty t P_1^{(n)}(t)\, dt$$

$$= -r^{(n)} \int_0^\infty t w_L^{(n)}(t)\, dt = -r^{(n)} M^{(n)},$$

using eq. (2.4.8). $\qquad\qquad\square$

By setting $s = 0$ in (2.4.23) and its derivative equation, we find two constraints involving the exit probability and the first moment $M^{(n)}$ of the conditional extinction times using (2.4.30,2.4.31):

$$\xi_1^{(n)} r^{(n)} = \eta_2^{(n)}, \tag{2.4.32}$$

$$\left(\xi_2^{(n)} - \xi_1^{(n)} M^{(n)} \right) r^{(n)} = \eta_3^{(n)}. \tag{2.4.33}$$

Lemma 2.7 (Recurrence Relations). *The coefficients $\xi_1^{(n)}$, $\xi_2^{(n)}$ and $\eta_2^{(n)}$, $\eta_3^{(n)}$ are linear in λ_n and μ_n, satisfying*

$$\xi_1^{(n)} = (\lambda_n + \mu_n)\xi_1^{(n-1)} - \mu_n \lambda_{n-1}\xi_1^{(n-2)}, \tag{2.4.34}$$

$$\xi_2^{(n)} = \xi_1^{(n-1)} + (\lambda_n + \mu_n)\xi_2^{(n-1)} - \mu_n \lambda_{n-1}\xi_2^{(n-2)}. \tag{2.4.35}$$

and

$$\eta_2^{(n)} = (\lambda_n + \mu_n)\eta_2^{(n-1)} - \mu_n \lambda_{n-1}\eta_2^{(n-2)}, \tag{2.4.36}$$

$$\eta_3^{(n)} = \eta_2^{(n-1)} + (\lambda_n + \mu_n)\eta_3^{(n-1)} - \mu_n \lambda_{n-1}\eta_3^{(n-2)}. \tag{2.4.37}$$

Proof. From the definition of $A^{(n)}$ and by cofactor expansion, we have

$$\det(sI_n - A^{(n)}) = \det \begin{pmatrix} s+\lambda_1+\mu_1 & -\mu_2 & & & & \\ -\lambda_1 & s+\lambda_2+\mu_2 & -\mu_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\lambda_{n-2} & s+\lambda_{n-1}+\mu_{n-1} & -\mu_n \\ & & & -\lambda_{n-1} & s+\lambda_n+\mu_n \end{pmatrix},$$

$$= (s+\lambda_n+\mu_n)\det(sI_{n-1}-A^{(n-1)}) - \mu_n\lambda_{n-1}\det(sI_{n-2}-A^{(n-2)}).$$

Then,

$$\sum_{k=1}^{n+1} \xi_k^{(n)} s^{k-1} = (s+\lambda_n+\mu_n)(s^{n-1}+\xi_{n-1}^{(n-1)}s^{n-2}+\ldots+\xi_2^{(n-1)}s+\xi_1^{(n-1)})$$
$$\hspace{4cm} (2.4.38)$$
$$- \mu_n\lambda_{n-1}(s^{n-2}+\xi_{n-2}^{(n-2)}s^{n-3}+\ldots+\xi_2^{(n-2)}s+\xi_1^{(n-2)}).$$

By equating coefficients of (2.4.38) at $O(1)$ and $O(s)$, we can establish a recurrence relation between the coefficients of the characteristic polynomial for the n-site subproblem and the $n-1$ and $n-2$ site subproblems:

$$\xi_1^{(n)} = (\lambda_n+\mu_n)\xi_1^{(n-1)} - \mu_n\lambda_{n-1}\xi_1^{(n-2)},$$
$$\xi_2^{(n)} = \xi_1^{(n-1)} + (\lambda_n+\mu_n)\xi_2^{(n-1)} - \mu_n\lambda_{n-1}\xi_2^{(n-2)}.$$

It is clear from this recurrence that $\xi_1^{(n)}$ and $\xi_2^{(n)}$ are *linear* in the transition rates λ_n and μ_n since $\xi_1^{(n-1)}$ only depends on $\lambda_1,\ldots,\lambda_{n-1}, \mu_1,\ldots,\mu_{n-1}$ and $\xi_1^{(n-2)}$ only depends on $\lambda_1,\ldots,\lambda_{n-2}, \mu_1,\ldots,\mu_{n-2}$. A similar argument applied to $\det(sI_{n-1}-\hat{A}^{(n-1)})$ shows that $\{\eta_2^{(n)}\}$ and $\{\eta_3^{(n)}\}$ are linear in λ_n and μ_n also:

$$\eta_2^{(n)} = (\lambda_n+\mu_n)\eta_2^{(n-1)} - \mu_n\lambda_{n-1}\eta_2^{(n-2)},$$
$$\eta_3^{(n)} = \eta_2^{(n-1)} + (\lambda_n+\mu_n)\eta_3^{(n-1)} - \mu_n\lambda_{n-1}\eta_3^{(n-2)}.$$

$\square$

The way of computing (λ_n, μ_n) for the n-site subproblem is to rewrite (2.4.32) and substitute eqs. (2.4.34)-(2.4.37) into (2.4.32) and (2.4.33):

$$V_1^{(n)} \begin{pmatrix} \xi_1^{(n)} \\ \xi_2^{(n)} \\ \eta_2^{(n)} \\ \eta_3^{(n)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad V_2^{(n)} \begin{pmatrix} \sigma_n \\ \mu_n \end{pmatrix} = \begin{pmatrix} \xi_1^{(n)} \\ \xi_2^{(n)} \\ \eta_2^{(n)} \\ \eta_3^{(n)} \end{pmatrix} - \begin{pmatrix} 0 \\ \xi_1^{(n-1)} \\ 0 \\ \eta_2^{(n-1)} \end{pmatrix}, \qquad (2.4.39)$$

where

$$V_1^{(n)} = \begin{pmatrix} r^{(n)} & 0 & -1 & 0 \\ -M^{(n)} r^{(n)} & r^{(n)} & 0 & -1 \end{pmatrix}, \qquad V_2^{(n)} = \begin{pmatrix} \xi_1^{(n-1)} & -\lambda_{n-1} \xi_1^{(n-2)} \\ \xi_2^{(n-1)} & -\lambda_{n-1} \xi_2^{(n-2)} \\ \eta_2^{(n-1)} & -\lambda_{n-1} \eta_2^{(n-2)} \\ \eta_3^{(n-1)} & -\lambda_{n-1} \eta_3^{(n-2)} \end{pmatrix}, \qquad (2.4.40)$$

and $\sigma_n = \lambda_n + \mu_n$. Eliminating the vector $\left(\xi_1^{(n)}, \xi_2^{(n)}, \eta_2^{(n)}, \eta_3^{(n)} \right)^T$, we find that

$$V_1^{(n)} V_2^{(n)} \begin{pmatrix} \sigma_n \\ \mu_n \end{pmatrix} = -V_1^{(n)} \begin{pmatrix} 0 \\ \xi_1^{(n-1)} \\ 0 \\ \eta_2^{(n-1)} \end{pmatrix} = \begin{pmatrix} 0 \\ \eta_2^{(n-1)} - r^{(n)} \xi_1^{(n-1)} \end{pmatrix}. \qquad (2.4.41)$$

If the matrix $V_1^{(n)} V_2^{(n)}$ is invertible, σ_n and μ_n are uniquely determined, and so are λ_n and μ_n.

Theorem 2.2. *Given exact data $\{\Pi^{(n)}, M^{(n)}\}, n = 1, 2, \ldots, N$ generated by some underlying birth-death process, the rates $(\lambda_n, \mu_n), n = 1, \ldots, N$ are uniquely determined. In particular, the matrix*

$$F^{(2)} = \begin{pmatrix} (\lambda_1 + \mu_1) r^{(2)} - 1 & \Pi^{(2)} - 1 \\ 1 - M^{(2)}(\lambda_1 + \mu_1) & 1 - M^{(2)} \mu_1 \end{pmatrix} \qquad (2.4.42)$$

from eqs. (2.4.16) and (2.4.17) is invertible and the 2×2 matrix $V_1^{(n)} V_2^{(n)}$ (where $V_1^{(n)}$ and $V_2^{(n)}$ are defined in eqs. (2.4.40)) is invertible for $n \geq 3$.

Proof. Consider the case $n = 2$. Then, we see that

$$\det F^{(2)} = \lambda_1 \left(r^{(2)} - M^{(2)} \right), \tag{2.4.43}$$

$$= -\frac{\lambda_1^2 \mu_2}{(\lambda_2 + \mu_2)(\lambda_1 \lambda_2 + \mu_1 \lambda_2 + \mu_1 \mu_2)} < 0, \tag{2.4.44}$$

where we used the relations

$$\Pi^{(2)} = \frac{\mu_1(\lambda_2 + \mu_2)}{\lambda_1 \lambda_2 + \mu_1 \mu_2 + \lambda_2 \mu_1},$$

$$M^{(2)} = \frac{\lambda_2^2 + 2\lambda_2 \mu_2 + \lambda_1 \mu_2 + \mu_2^2}{(\lambda_2 + \mu_2)(\lambda_1 \lambda_2 + \lambda_2 \mu_1 + \mu_1 \mu_2)}.$$

Therefore (λ_2, μ_2) are uniquely determined. Now consider the case $n \geq 3$. For reference, define

$$\tilde{V}_2^{(n)} = \begin{pmatrix} \xi_1^{(n-1)} & \xi_1^{(n-2)} \\ \xi_2^{(n-1)} & \xi_2^{(n-2)} \\ \eta_2^{(n-1)} & \eta_2^{(n-2)} \\ \eta_3^{(n-1)} & \eta_3^{(n-2)} \end{pmatrix}. \tag{2.4.45}$$

(Compare $\tilde{V}_2^{(n)}$ to the definition of $V_2^{(n)}$ in eq. (2.4.40)). To show that $V_1^{(n)} V_2^{(n)}$ is invertible, it is sufficient to show that $\det \left(V_1^{(n)} \tilde{V}_2^{(n)} \right) \neq 0$. We split the proof into two parts. First, we find a simple expression for the determinant. Second, we show using induction that this expression is always non-zero.

Expression for Determinant. The determinant depends on $r^{(n)}$ and $M^{(n)}$, which are determined by the (perfect) data. Using the recurrence relations (2.4.34)-(2.4.37), we note that

$$r^{(n)} = \tilde{P}_1^{(n)}(0) = \frac{\eta_2^{(n)}}{\xi_1^{(n)}} = \frac{\sigma_n \eta_2^{(n-1)} - \mu_n \lambda_{n-1} \eta_2^{(n-2)}}{\sigma_n \xi_1^{(n-1)} - \mu_n \lambda_{n-1} \xi_1^{(n-2)}}, \tag{2.4.46}$$

and from Lemma 2.6,

$$
\begin{aligned}
-M^{(n)}r^{(n)} &= \left.\frac{d\tilde{P}_1^{(n)}}{ds}\right|_{s=0} = \frac{\eta_3^{(n)}\xi_1^{(n)} - \eta_2^{(n)}\xi_2^{(n)}}{\xi_1^{(n)^2}}, \\
&= \frac{\left(\eta_2^{(n-1)} + \sigma_n\eta_3^{(n-1)} - \mu_n\lambda_{n-1}\eta_3^{(n-2)}\right)\left(\sigma_n\xi_1^{(n-1)} - \mu_n\lambda_{n-1}\xi_1^{(n-2)}\right)}{\left(\sigma_n\xi_1^{(n-1)} - \mu_n\lambda_{n-1}\xi_1^{(n-2)}\right)^2} \\
&\quad - \frac{\left(\sigma_n\eta_2^{(n-1)} - \mu_n\lambda_{n-1}\eta_2^{(n-2)}\right)\left(\xi_1^{(n-1)} + \sigma_n\xi_2^{(n-1)} - \mu_n\lambda_{n-1}\xi_2^{(n-2)}\right)}{\left(\sigma_n\xi_1^{(n-1)} - \mu_n\lambda_{n-1}\xi_1^{(n-2)}\right)^2}. \quad (2.4.47)
\end{aligned}
$$

After substituting (2.4.46) and (2.4.47) into the definition of $V_1^{(n)}$ in (2.4.40) and performing some algebra,

$$
\begin{aligned}
\det(V_1^{(n)}\tilde{V}_2^{(n)}) &= \frac{\lambda_{n-1}\mu_n(\eta_2^{(n-1)}\xi_1^{(n-2)} - \eta_2^{(n-2)}\xi_1^{(n-1)})^2}{(\lambda_{n-1}\mu_n\xi_1^{(n-2)} - \sigma_n\xi_1^{(n-1)})^2}, \\
&= \frac{\lambda_{n-1}\mu_n(\eta_2^{(n-1)}\xi_1^{(n-2)} - \eta_2^{(n-2)}\xi_1^{(n-1)})^2}{\xi_1^{(n)^2}}. \quad (2.4.48)
\end{aligned}
$$

The denominator is always nonzero because $\xi_1^{(n)} = (-1)^n \det(A^{(n)}) \neq 0$: it is well known that the eigenvalues of the infinitesimal generator matrix (and its submatrices) of a birth-death process and are all negative [69], and the matrix $A^{(n)}$ is the transpose of a submatrix of such an infinitesimal generator. It remains to show that this expression is non-zero for $n \geq 3$.

Induction Proof. First we show that $\det(V_1^{(3)}\tilde{V}_2^{(3)})$ is non-zero. Because

$$
\begin{aligned}
\xi_1^{(2)} &= \lambda_1\lambda_2 + \mu_1\mu_2 + \lambda_2\mu_1, \\
\eta_2^{(2)} &= \lambda_2 + \mu_2, \\
\xi_1^{(1)} &= \lambda_1 + \mu_1, \\
\eta_2^{(1)} &= 1, \\
\Rightarrow \det(V_1^{(3)}\tilde{V}_2^{(3)}) &= \frac{\lambda_2\mu_3\lambda_1^2\mu_2^2}{\xi_1^{(3)^2}} > 0.
\end{aligned}
$$

Now assume that $\det(V_1^{(n)}\tilde{V}_2^{(n)})$ is non-zero. Then

$$
\eta_2^{(n-1)}\xi_1^{(n-2)} - \eta_2^{(n-2)}\xi_1^{(n-1)} \neq 0.
$$

56

It suffices to show that $\eta_2^{(n)}\xi_1^{(n-1)} - \eta_2^{(n-1)}\xi_1^{(n)} \neq 0$. Using (2.4.34) and (2.4.36),

$$\eta_2^{(n)}\xi_1^{(n-1)} - \eta_2^{(n-1)}\xi_1^{(n)}$$

$$= \left(\sigma_n\eta_2^{(n-1)} - \mu_n\lambda_{n-1}\eta_2^{(n-2)}\right)\xi_1^{(n-1)} - \eta_2^{(n-1)}\left(\sigma_n\xi_1^{(n-1)} - \mu_n\lambda_{n-1}\xi_1^{(n-2)}\right),$$

$$= \mu_n\lambda_{n-1}\left(\eta_2^{(n-1)}\xi_1^{(n-2)} - \eta_2^{(n-2)}\xi_1^{(n-1)}\right) \neq 0.$$

Therefore $\det(V_1^{(n)}\tilde{V}_2^{(n)})$ is non-zero for $n \geq 3$. Therefore $V_1^{(n)}V_2^{(n)}$ is invertible for $n \geq 3$. From (2.4.41), σ_n and μ_n are uniquely determined and so (λ_n, μ_n) are also uniquely determined. $\qquad\square$

In summary, at step $n \geq 3$, we must solve the linear system

$$V_1^{(n)}V_2^{(n)}\begin{pmatrix}\sigma_n \\ \mu_n\end{pmatrix} = \begin{pmatrix}0 \\ \eta_2^{(n-1)} - r^{(n)}\xi_1^{(n-1)}\end{pmatrix},$$

which we may write as

$$F^{(n)}\nu_n = G^{(n)}, \tag{2.4.49}$$

where

$$F^{(n)} = V_1^{(n)}V_2^{(n)}\begin{pmatrix}1 & 1 \\ 0 & 1\end{pmatrix}, \quad \nu_n = \begin{pmatrix}\lambda_n \\ \mu_n\end{pmatrix}, \quad G^{(n)} = \begin{pmatrix}0 \\ \eta_2^{(n-1)} - r^{(n)}\xi_1^{(n-1)}\end{pmatrix}. \tag{2.4.50}$$

The $F^{(n)}$ and $G^{(n)}$ depend on the previous transition rates $\nu_1, \ldots, \nu_{n-1}$ and from Theorem 2.2, $F^{(n)}$ is invertible. For reference, the entries of $F^{(n)}$ are:

$$F_{11}^{(n)} = r^{(n)}\xi_1^{(n-1)} - \eta_2^{(n-1)}, \tag{2.4.51}$$

$$F_{12}^{(n)} = r^{(n)}\left(\xi_1^{(n-1)} - \lambda_{n-1}\xi_1^{(n-2)}\right) - \eta_2^{(n-1)} + \lambda_{n-1}\eta_2^{(n-2)}, \tag{2.4.52}$$

$$F_{21}^{(n)} = r^{(n)}\left(\xi_2^{(n-1)} - \xi_1^{(n-1)}M^{(n)}\right) - \eta_3^{(n-1)}, \tag{2.4.53}$$

$$F_{22}^{(n)} = r^{(n)}\left[\xi_2^{(n-1)} - M^{(2)}\xi_1^{(n-1)} - \lambda_{n-1}\left(\xi_2^{(n-2)} - M^{(2)}\xi_1^{(n-2)}\right)\right]$$

$$+ \lambda_{n-1}\eta_3^{(n-2)} - \eta_3^{(n-2)} \tag{2.4.54}$$

The following notation will be used in the next theorem about error propagation:

$$\boldsymbol{x}^* = (\lambda_1^*, \ldots, \lambda_{n-1}^*, \mu_1^*, \ldots, \mu_{n-1}^*; \Pi^{(1)*}, \ldots, \Pi^{(n)*}, M^{(1)*}, \ldots, M^{(n)*}), \tag{2.4.55}$$

$$\boldsymbol{\delta x} = (\delta\lambda_1, \ldots, \delta\lambda_{n-1}, \delta\mu_1, \ldots, \delta\mu_{n-1}; \delta\Pi^{(1)}, \ldots, \delta\Pi^{(n)}, \delta M^{(1)}, \ldots, \delta M^{(n)})$$

where the birth-death rates with asterisks stand for exact rates and $\Pi^{(n)^*}, M^{(n)^*}$ are the exact data. In contrast, the elements in $\boldsymbol{\delta x}$ are perturbations to the corresponding elements. By Theorem 2.2, the exact transition rates at site n depend on the rates at sites $1, 2, \ldots, n-1$:

$$\lambda_n^* = f_1^{(n)}(\lambda_1^*, \ldots, \lambda_{n-1}^*, \mu_1^*, \ldots, \mu_{n-1}^*; \Pi^{(1)^*}, \ldots, \Pi^{(n)^*}, M^{(1)^*}, \ldots, M^{(n)^*}), \quad (2.4.56)$$

$$\mu_n^* = f_2^{(n)}(\lambda_1^*, \ldots, \lambda_{n-1}^*, \mu_1^*, \ldots, \mu_{n-1}^*; \Pi^{(1)^*}, \ldots, \Pi^{(n)^*}, M^{(1)^*}, \ldots, M^{(n)^*}), \quad (2.4.57)$$

where $f_i^{(n)} : \mathbb{R}^{4n-2} \to \mathbb{R}$ for $i = 1, 2$.

Theorem 2.3 (Error Propagation with site number). *Let $\nu_n = (\lambda_n, \mu_n)^T, D_n = (\Pi^{(n)}, M^{(n)})^T$ and let ν_n^* be the exact transition rate at site n. Suppose that all first derivatives of $f_1^{(n)}$ and $f_2^{(n)}$ in eqs. (2.4.56, 2.4.57) are bounded in a small neighborhood $B(\boldsymbol{x}^*, r)$ of $\boldsymbol{x}^*$ in equation (2.4.55), i.e. there exists $r, R > 0$ such that for any $\boldsymbol{x} \in B(\boldsymbol{x}^*, r)$,*

$$\|\nabla f_m^{(n)}(\boldsymbol{x})\|_\infty \le R/2, \quad (2.4.58)$$

for $1 \le n \le N$ and $m = 1, 2$. Then, if a small error δD_k, such that $\|\delta D_k\|_\infty < r$, is introduced into the data $\{D_k\}_{k=1}^n$ at each site such that $D_k = D_k^* + \delta D_k$ for $k = 1, \ldots, n$, the error of birth-death rates at site n satisfies

$$\|\delta \nu_n\|_\infty \le \sum_{j=1}^{n} R(1+R)^{n-j} \|\delta D_j\|_\infty. \quad (2.4.59)$$

Proof. The analysis is fairly standard and makes use of Taylor series expansions. The exact rates satisfy

$$\lambda_n^* = f_1^{(n)}(\boldsymbol{x}^*), \quad \mu_n^* = f_2^{(n)}(\boldsymbol{x}^*) \quad (2.4.60)$$

where $f_i^{(n)} : \mathbb{R}^{4n-2} \to \mathbb{R}$ for $i = 1, 2$.

Now consider a perturbation $\boldsymbol{\delta x}$ such that $\|\boldsymbol{\delta x}\|_\infty < r$. Then by the mean value theorem for multivariate functions, the errors at each site satisfy

$$f_1^{(n)}(\boldsymbol{x}^* + \boldsymbol{\delta x}) = f_1^{(n)}(\boldsymbol{x}^*) + \nabla f_1^{(n)}(\boldsymbol{z}_1^{(n)}) \cdot (\boldsymbol{\delta \lambda}_{n-1}, \boldsymbol{\delta \mu}_{n-1}) = \lambda_n^* + \delta \lambda_n$$

$$f_2^{(n)}(\boldsymbol{x}^* + \boldsymbol{\delta x}) = f_2^{(n)}(\boldsymbol{x}^*) + \nabla f_2^{(n)}(\boldsymbol{z}_2^{(n)}) \cdot (\boldsymbol{\delta \lambda}_{n-1}, \boldsymbol{\delta \mu}_{n-1}) = \mu_n^* + \delta \mu_n$$

for some $\boldsymbol{z}_1^{(n)} = \boldsymbol{x}^* + c_1^{(n)} \boldsymbol{\delta x}, \boldsymbol{z}_2^{(n)} = \boldsymbol{x}^* + c_2^{(n)} \boldsymbol{\delta x}$ with $c_i^{(n)} \in (0, 1), i = 1, 2$. By equations (2.4.60), we have

$$
\delta\lambda_n = \sum_{i=1}^{n-1} \left(\left. \frac{\partial f_1^{(n)}}{\partial \lambda_i} \right|_{\boldsymbol{z}_1^{(n)}} \delta\lambda_i + \left. \frac{\partial f_1^{(n)}}{\partial \mu_i} \right|_{\boldsymbol{z}_1^{(n)}} \delta\mu_i \right)
$$

$$
+ \sum_{i=1}^{n} \left(\left. \frac{\partial f_1^{(n)}}{\partial \Pi^{(i)}} \right|_{\boldsymbol{z}_1^{(n)}} \delta\Pi^{(i)} + \left. \frac{\partial f_1^{(n)}}{\partial M^{(i)}} \right|_{\boldsymbol{z}_1^{(n)}} \delta M^{(i)} \right),
$$

$$
\delta\mu_n = \sum_{i=1}^{n-1} \left(\left. \frac{\partial f_2^{(n)}}{\partial \lambda_i} \right|_{\boldsymbol{z}_2^{(n)}} \delta\lambda_i + \left. \frac{\partial f_2^{(n)}}{\partial \mu_i} \right|_{\boldsymbol{z}_2^{(n)}} \delta\mu_i \right)
$$

$$
+ \sum_{i=1}^{n} \left(\left. \frac{\partial f_2^{(n)}}{\partial \Pi^{(i)}} \right|_{\boldsymbol{z}_2^{(n)}} \delta\Pi^{(i)} + \left. \frac{\partial f_2^{(n)}}{\partial M^{(i)}} \right|_{\boldsymbol{z}_2^{(n)}} \delta M^{(i)} \right).
$$

Define the matrices

$$
R_k^{(n)} = \begin{pmatrix} \left. \frac{\partial f_1^{(n)}}{\partial \lambda_k} \right|_{\boldsymbol{z}_1^{(n)}} & \left. \frac{\partial f_1^{(n)}}{\partial \mu_k} \right|_{\boldsymbol{z}_1^{(n)}} \\ \left. \frac{\partial f_2^{(n)}}{\partial \lambda_k} \right|_{\boldsymbol{z}_2^{(n)}} & \left. \frac{\partial f_2^{(n)}}{\partial \mu_k} \right|_{\boldsymbol{z}_2^{(n)}} \end{pmatrix} \tag{2.4.61}
$$

and

$$
S_k^{(n)} = \begin{pmatrix} \left. \frac{\partial f_1^{(n)}}{\partial \Pi^{(k)}} \right|_{\boldsymbol{z}_1^{(n)}} & \left. \frac{\partial f_1^{(n)}}{\partial M^{(k)}} \right|_{\boldsymbol{z}_1^{(n)}} \\ \left. \frac{\partial f_2^{(n)}}{\partial \Pi^{(k)}} \right|_{\boldsymbol{z}_2^{(n)}} & \left. \frac{\partial f_2^{(n)}}{\partial M^{(k)}} \right|_{\boldsymbol{z}_2^{(n)}} \end{pmatrix} \tag{2.4.62}
$$

Therefore,

$$
\delta\nu_n = \sum_{k=1}^{n-1} R_k^{(n)} \delta\nu_k + \sum_{k=1}^{n} S_k^{(n)} \delta D_k. \tag{2.4.63}
$$

In general, we can repeatedly substitute to get $\delta\nu_n$ in terms of $\{\delta D_k\}_{k=1}^{n}$ only:

$$
\delta\nu_n = \sum_{j=1}^{n} \left(S_j^{(n)} + \sum_{j \le k_1 < n} S_j^{(k_1)} R_{k_1}^{(n)} + \sum_{j \le k_1 < k_2 < n} S_j^{(k_1)} R_{k_1}^{(k_2)} R_{k_2}^{(n)} + \dots \right.
$$
$$
\left. + \sum_{j \le k_1 < \dots < k_i < n} S_j^{(k_1)} R_{k_1}^{(k_2)} \cdots R_{k_i}^{(n)} + \dots + S_j^{(j)} R_j^{(j+1)} \cdots R_{n-1}^{(n)} \right) \delta D_j \tag{2.4.64}
$$

By equation (2.4.58) we have $\|R_k^{(n)}\|_\infty \le R$ and $\|S_k^{(n)}\|_\infty \le R$ for all $1 \le i, j \le N$. Then the binomial expansion yields

$$
||\delta\nu_n||_\infty \le R \sum_{j=1}^{n} (1+R)^{n-j} \|\delta D_j\|_\infty
$$

$\square$

Hence at each site n, the error in the birth-death rates (λ_n, μ_n) is the result of accumulating the errors from sites 1 to $n-1$.

Corollary 2.1. *Suppose $D = \max_j \|\delta D_j\|_\infty$, then equation (2.4.59) becomes*

$$\|\delta\nu_n\|_\infty \le D[(1+R)^n - 1]. \tag{2.4.65}$$

In this special case where all errors in data are bounded by D, we expect that the error in the birth-death rates grows exponentially.

2.4.3.4 Algorithm Details

The input data is an array of extinction times T with their corresponding maximum sites. In the preprocessing step, we group these extinction times by their maximum sites, so that the n-th group contains all trajectories with maximum sites not exceeding n. $\Pi^{(n)}$ is computed as the proportion of number of extinction times in this group out of the total number of simulations, and $M^{(n)}$ is given by the mean extinction times within this group. Graphically, only extinction times within the "red box" are processed, as in Figure 2.8, and the red box grows larger with each preprocessing step.

The implementation of the algorithm starts with the inference at site 1 and site 2. In order to keep track of the recurrence relation in Lemma 2.7, we only need to focus on the "feature vector" defined as $\boldsymbol{u}_n = \left(\xi_1^{(n)}, \xi_2^{(n)}, \eta_2^{(n)}, \eta_3^{(n)}\right)^T$ at each site n. For site 1, we have

$$\boldsymbol{u}_1 = \begin{pmatrix} \xi_1^{(1)} \\ \xi_2^{(1)} \\ \eta_2^{(1)} \\ \eta_3^{(1)} \end{pmatrix} = \begin{pmatrix} \lambda_1 + \mu_1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \tag{2.4.66}$$

For site 2, we have

$$
\boldsymbol{u}_2 = \begin{pmatrix} \xi_1^{(2)} \\ \xi_2^{(2)} \\ \eta_2^{(2)} \\ \eta_3^{(2)} \end{pmatrix} = \begin{pmatrix} \lambda_1\lambda_2 + \mu_1\mu_2 + \lambda_2\mu_1 \\ \lambda_1 + \mu_1 + \lambda_2 + \mu_2 \\ \lambda_2 + \mu_2 \\ 1 \end{pmatrix} \tag{2.4.67}
$$

Then we can define the matrix

$$
Z = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n \end{pmatrix} = \begin{pmatrix} \xi_1^{(1)} & \xi_1^{(2)} & \cdots & \xi_1^{(n)} \\ \xi_2^{(1)} & \xi_2^{(2)} & \cdots & \xi_2^{(n)} \\ \eta_2^{(1)} & \eta_2^{(2)} & \cdots & \eta_2^{(n)} \\ \eta_3^{(1)} & \eta_3^{(2)} & \cdots & \eta_3^{(n)} \end{pmatrix} \tag{2.4.68}
$$

where the j-th column is the feature vector at site j.

At each step $j \geq 3$, we need to solve linear system defined by (2.4.49) and (2.4.50) to obtain $\{\lambda_j, \mu_j\}$, and update the j-th column of Z by Lemma 2.7. Details are given in Algorithm 6.

There is an alternative way to infer the transition rates, instead of directly using the recurrence relations in Lemma 2.7. This method is technically simpler, yet more computationally intensive when the number of sites is large. We include the details of this method in Appendix A.

2.4.4 Numerical Results

Example 1: 5-site birth death chain. First we present a simple result of a birth-death chain with only 5 sites ($N = 4$) and some pre-determined rates. All the rates are about the same order of magnitude, see Fig. 2.10.

The extinction time data is generated by simulating the birth-death process with these given rates, and we evaluate the reconstruction in comparison to them: the error is computed using the infinity norm. With about 5×10^6 extinction times, we can infer the rates to very good accuracy – about 0.11% and 0.37% relative errors in $\{\lambda_k\}$ and $\{\mu_k\}$, respectively. First, the sum of birth and death rates $\lambda_1 + \mu_1$ follows from the

---

Algorithm 6 Inference of birth and death rates up to site N
in a birth death chain given conditional extinction times

1: Input: An array of extinction times T along with maximal sites from repeated
    simulation of a birth death process by Algorithm 3.
2: Initialize: Compute the conditional probabilities of left exit $\{\Pi^{(1)}, \ldots, \Pi^{(N)}\}$, and
    mean of conditional extinction times $\{M^{(1)}, \ldots, M^{(N)}\}$, at each site. Initialize Z
    as a $4 \times N$ zero matrix.
3: At site 1, $\begin{pmatrix} \lambda_1 \\ \mu_1 \end{pmatrix} = \begin{pmatrix} (1 - \Pi^{(1)})/M^{(1)} \\ \Pi^{(1)}/M^{(1)} \end{pmatrix}$, as in (2.4.10, 2.4.11). Feature vector $Z_{:,1} =$
    $\boldsymbol{u}_1$ is defined in (2.4.66).
4: **if** $N == 1$ **then return** $\{\mu_1, \lambda_1\}$
5: **end if**
6: At site 2, $\begin{pmatrix} \lambda_2 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} r^{(2)}(\lambda_1 + \mu_1) - 1 & \Pi^{(2)} - 1 \\ 1 - M^{(2)}(\lambda_1 + \mu_1) & 1 - M^{(2)}\mu_1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1/r^{(2)} - \lambda_1 - \mu_1 \end{pmatrix}$, see
    (2.4.16, 2.4.17). Feature vector $Z_{:,2} = \boldsymbol{u}_2$ is defined as (2.4.67).
7: **for** $j = 3 : N$ **do**
8:     Compute $V_1^{(j)}$ and $V_2^{(j)}$ as in (2.4.40).
9:     Form the matrices $F^{(j)}$ and $G^{(j)}$ as in (2.4.49) and (2.4.50).
10:     Solve $[\lambda_j, \mu_j]^T = (F^{(j)})^{-1}G^{(j)}$.
11:     Update $Z_{:,j} = V_2^{(j)} \begin{pmatrix} \lambda_j + \mu_j \\ \mu_j \end{pmatrix} + \begin{pmatrix} 0 \\ Z_{1,j-1} \\ 0 \\ Z_{3,j-1} \end{pmatrix}$.
12:     **if** $j == N$ **then**
13:         $\lambda_j = 0$
14:     **end if**
15: **end for**
16: Output: $\{\mu_1, \ldots, \mu_N\}$ and $\{\lambda_1, \ldots, \lambda_{N-1}\}$

---

mean extinction time conditioned on immediate exit through eq. (2.4.9). The *fraction* of times corresponding to immediate exit then yields μ_1 and λ_1 separately through (2.4.10) and (2.4.11). Next, λ_2 and μ_2 are computed by (2.4.16, 2.4.17). Then for $n = 3, 4$, we compute the n-th columns of matrix Z as in (2.4.68) and obtain $\{\lambda_n, \mu_n\}$ simultaneously. Notice that the last birth rate λ_N is always assumed to be zero and no inference is necessary at that site.
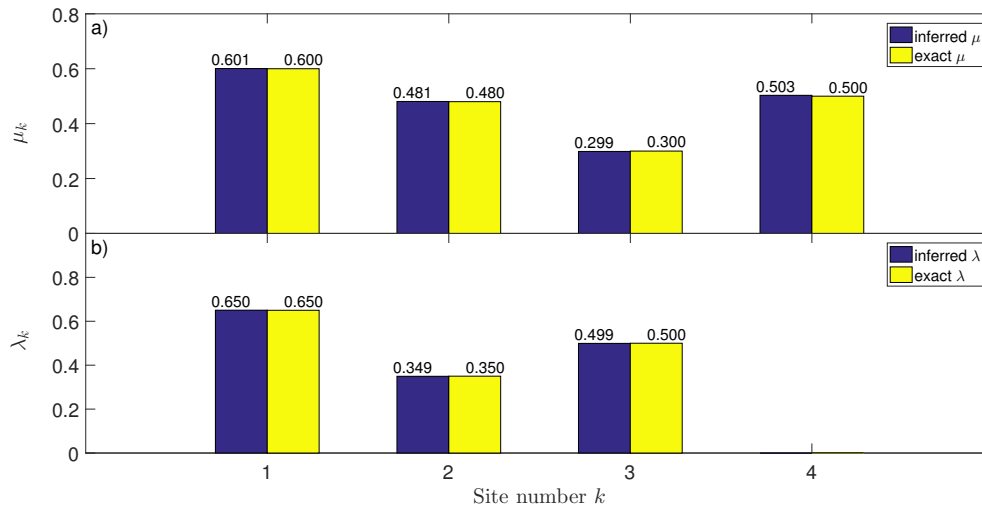
Figure 2.10: Bar plots of the inference results in a 5-site birth death chain. The top subplot (a) contains rates for μ_k and bottom subplot (b) for λ_k. The bars in dark blue represent numerically approximated rates, and yellow bars stand for exact rates. On top of each bar is the value associated with it.

Example 2: 11-site birth death chain. We now test our reconstruction algorithm on a longer chain. In this result, 5×10^7 extinction times are simulated from an 11-site birth-death chain. Following the same steps, we have the inference results with a relative error in $\{\lambda_k\}$ and $\{\mu_k\}$ to be 3.29% and 3.71%, respectively: see Fig. 2.11.

Example 3: 9-site birth death chain with a bottleneck. This birth-death chain has a "bottleneck" between sites 3 and 4 so that λ_3 and μ_4 are much smaller than the rates at the other sites – it is very difficult for the particle to transition from site 3 to 4 or
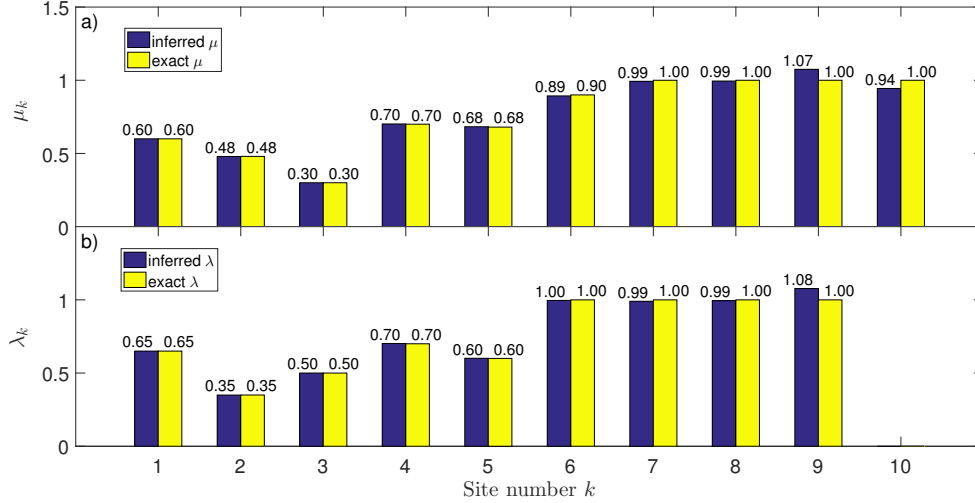
Figure 2.11: Bar plots of the inference results in a 11-site birth death chain. The top subplot (a) contains rates for μ_k and bottom subplot (b) for λ_k. The bars in dark blue represent numerically approximated rates, and yellow bars stand for exact rates.

from site 4 to 3: see Fig. 2.12. Upon application of our algorithm, we find that the maximum error in the $\{\lambda_k\}$ and $\{\mu_k\}$ are 1.17% and 1.24% respectively.
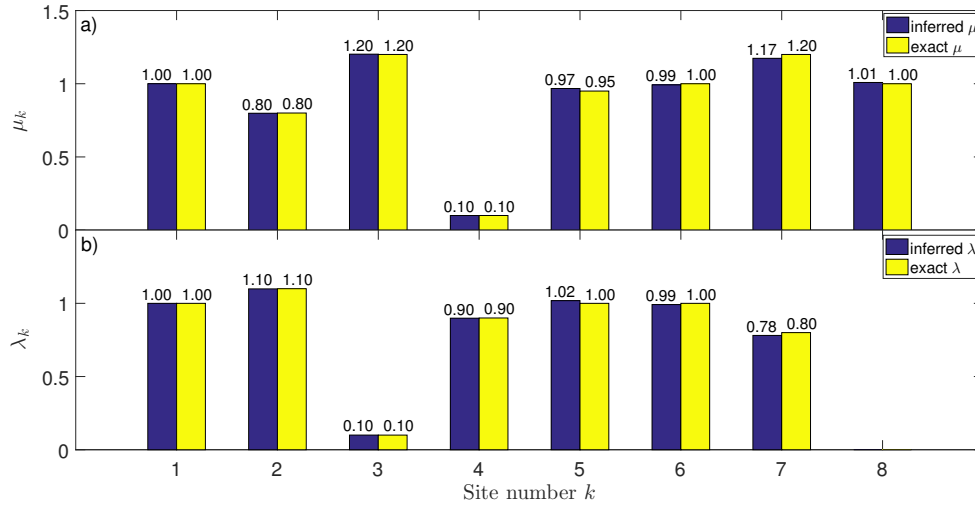
Figure 2.12: Bar plots of the inference results in a 9-site birth death chain with a bottleneck between sites 3 and 4. The top subplot (a) contains rates for μ_k and bottom subplot (b) for λ_k. The bars in dark blue represent numerically approximated rates, and yellow bars stand for exact rates.

Example 4: 9-site birth death chain with a sticky site. This example corresponds to a potential landscape with multiple minima, with one minimum much deeper than the others. The transition rates into this "sticky" site are of moderate magnitude, but the rates *out* of the site are small in comparison. For the same number of extinction times, the presence of the sticky site reduces the overall accuracy of the inference. With 5×10^7 extinction times, the errors are 7.40% and 8.29% relative errors in $\{\lambda_k\}$ and $\{\mu_k\}$, respectively: see Fig. 2.13. Specifically, the errors in the transition rates are larger after the sticky site than they are before. This is because the random walk samples sites 5-8 a lot less frequently than sites 1-4.
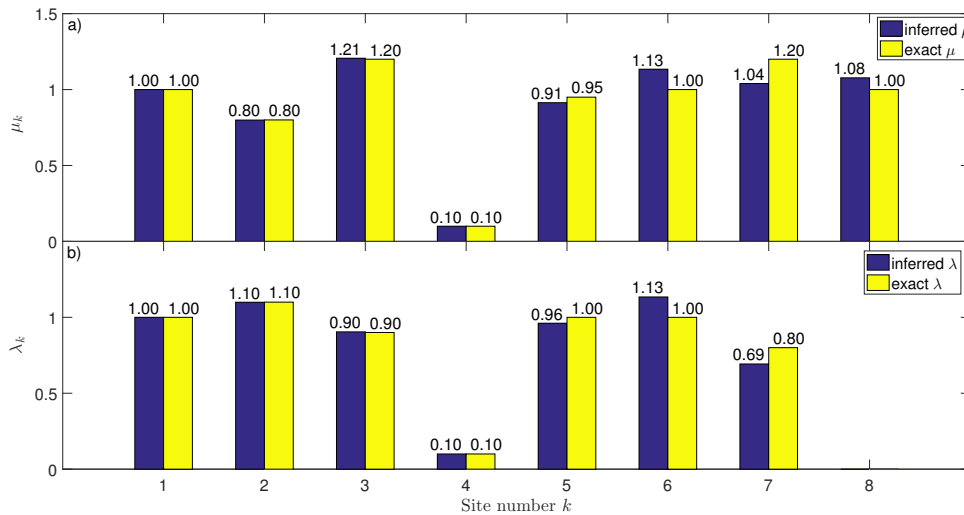
Figure 2.13: Bar plots of the inference results in a 9-site birth death chain where site 4 is "sticky" (i.e. both rates out of site 4 are relatively small). The top subplot (a) contains rates for μ and bottom subplot (b) for λ. The bars in dark blue represent numerically approximated rates, and yellow bars stand for exact rates.

Example 5: 9-site potential well with a single shallow minimum. In this example, the potential landscape has one local minimum that is very shallow compared to the others: the transition rates *out* of this minimum are large compared to the other rates. The random walker spends very little time at this site, rendering it almost invisible with respect to extinction times. One sees that the inference at the "invisible" site is very

poor compared to the other sites. The relative error at this site is 19.96% for $\{\lambda_6\}$ and 19.38% for $\{\mu_6\}$: see Fig. 2.14. The errors at this site dominate the errors at the other sites.
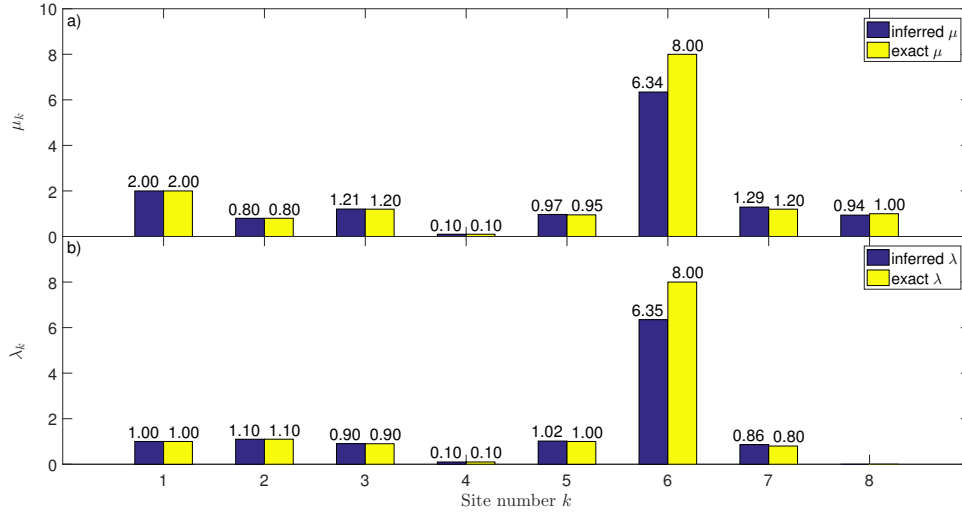
Figure 2.14: Inference results for a 9-site birth death chain, representing a potential landscape with multiple minima where one minimum is very shallow: the rates out of site 6 are much larger than the rates at the other sites. The top subplot (a) contains rates for μ and bottom subplot (b) for λ. The bars in dark blue represent numerically approximated rates, and yellow bars stand for exact rates.

Example 6: 11-site birth death chain error propagation. In this example, we show how error propagates with site number. The birth and death rates are all equal to 1, and 1×10^8 extinction times are generated from the Monte Carlo simulation. Bootstrap is done by taking random samples of size 5×10^6 from these extinction times and errors at each site are computed. This resampling procedure is repeated 50 times, and Figures 2.15 and 2.16 display the mean error and 95% confidence intervals from the above 50 samples, where it is clear that errors for both $\{\lambda_k\}$ and $\{\mu_k\}$ increase exponentially with site number, as a result of Theorem 2.3.

Example 7: Minimum number of extinction times required. Finally, we consider the minimum number of extinction times required such that relative error for both λ_k and
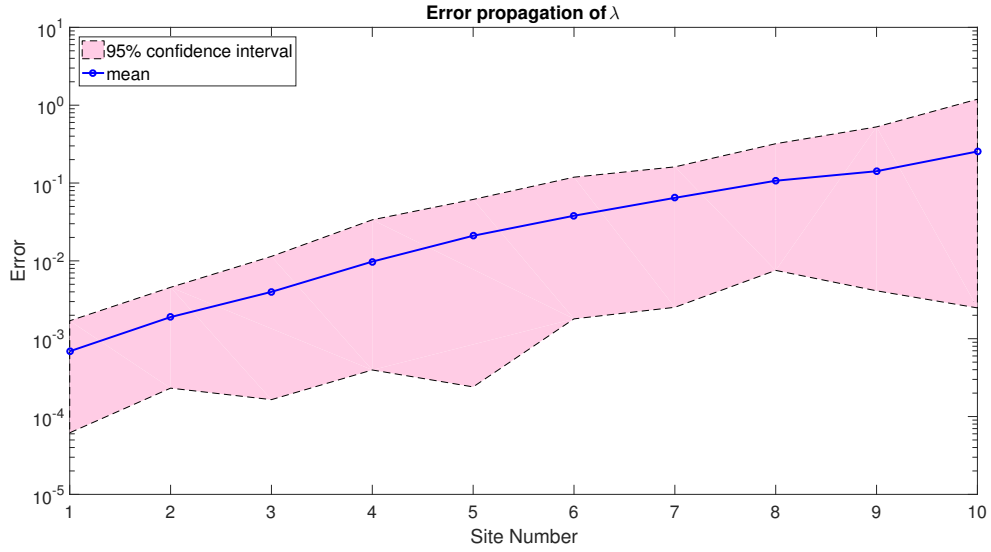
Figure 2.15: Error plot for λ at each site, taken as the average of 50 random samples of 1×10^8 extinction times. The linear fit of the mean error of λ is given by $\ln[\text{Error}(\lambda)] = 0.643 \times [\text{Site Number}] - 7.449$.
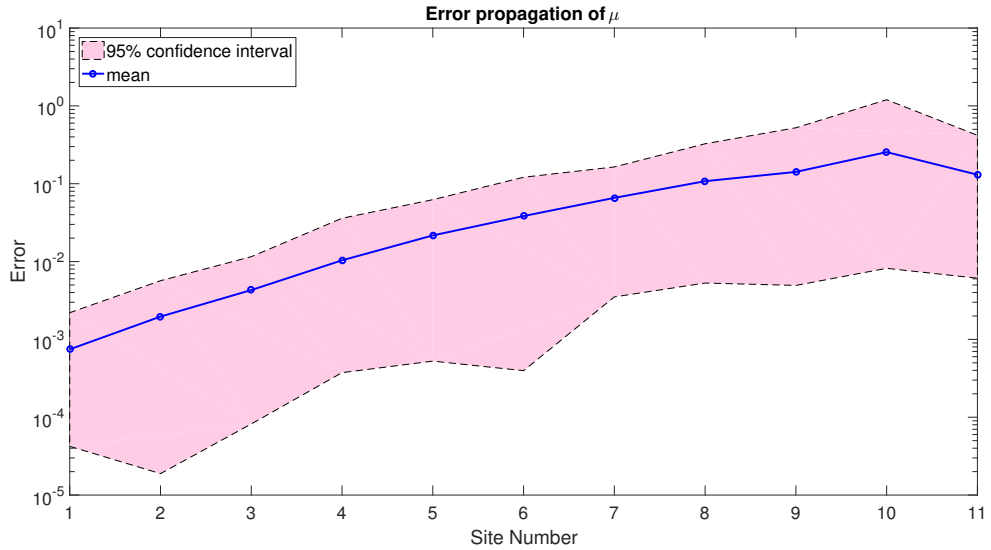
Figure 2.16: Error plot for μ at each site, taken as the average of 50 random samples of 1×10^8 extinction times. The linear fit of the mean error of μ is given by $\ln[\text{Error}(\mu)] = 0.5598 \times [\text{Site Number}] - 7.072$.

N	(Estimated) Minimum number of ETs
1	5.0×10^1
2	1.5×10^2
3	5.0×10^2
4	1.0×10^4
5	1.0×10^5
6	3.0×10^5
7	2.0×10^6
8	4.0×10^6
9	7.0×10^6

Table 2.1: Number of extinction times required for rates to have relative error below 15%, on chains of different lengths. The birth-death rates are on the same order of magnitude.

μ_k, $1 \le k \le N$, are below 15%. In order to estimate the mimimum number of extinction times required in a chain of length $N + 1$, we take 50 bootstrap samples of size m and record the average relative errors in the birth-death rates. We repeat this process, doubling m each time and then choose the smallest m that yields a relative error below 15%. We applied this process to a chain where all birth-death rates are about the same order of magnitude. Our results are given in Table 2.1, and Fig. 2.17. It is obvious from the plot that minimum number of extinction times required is increasing exponentially with the length of chain.
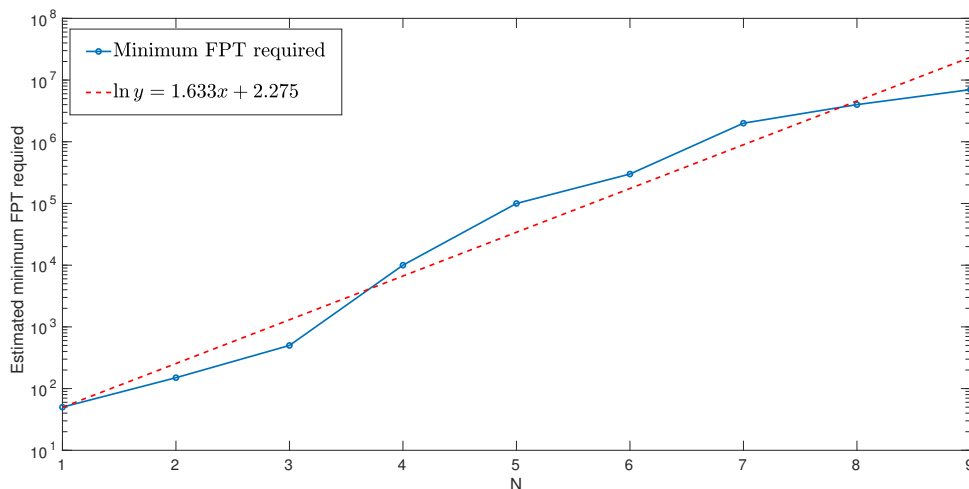
Figure 2.17: Number of extinction times required for rates to have relative error below 15%. The dashed red line is fit by data points on the blue line.

2.4.5 Summary

In summary, we have presented a method for extracting the kinetic rates of large proteins, with multiple folding domains, from extinction times (when all domains have folded) and "maximal sites" (the maximum number of unfolded domains before extinction). Both of these quantities can, in principle, be computed from AFM time traces.

The inference relies on the recurrence relation specified in Lemma 2.7 and starts with base cases when $n = 1, 2$, and inference of each subsequent site depends on its previous sites by solving a linear system. If the data $\Pi^{(n)}$ and $M^{(n)}$ are exactly given, meaning that they correspond to the statistics of an underlying birth-death process, then the birth-death rates are uniquely determined, and we proved that a small perturbation in site 1 will propagate exponentially throughout the chain, given that the first derivatives of $f_1^{(n)}$ and $f_2^{(n)}$ in Theorem 2.3 are bounded near the exact solution. With sufficient data (about 50 million for a chain of length 8 - 12), the method can compute these rates to very good accuracy and is capable of detecting bottlenecks or extreme values in the chain. In general, the number of extinction times one needs to obtain reasonable results grows exponentially with the total length of the chain.

There still remain many theoretical challenges to interpreting single-molecule AFM data. For example, inference from extinction times in the "transmission" problem where absorption/extinction is at site N rather than site 0 is severely ill-posed. Can some aspect of time-trace data be used to better-condition this problem? Also, bifurcations or loops in the birth-death chain representing multiple pathways to a final unfolded state have not yet been explored. Finally, it remains to be seen if our method can be adapted to force-ramp data, or how to proceed if transition times between metastable configurations are not exponentially distributed.

Chapter 3

INFERENCE IN NUCLEAR MAGNETIC RESONANCE PROBLEMS

In this chapter, inference problems arising in Nuclear Magnetic Resonance (NMR) and Magnetic Resonance Imaging (MRI) are discussed. First, a brief introduction to NMR and MRI with their physical backgrounds is given, and the relation between the inverse problem in NMR to the inverse Laplace transform is demonstrated. Due to the ill-posedness of the inverse Laplace transform, regularization methods are used to treat these problems. Then, NMR relaxometry is investigated mathematically in detail, where a new method to process some specific signals is proposed at the end.

3.1 Background

3.1.1 Introduction to Nuclear Magnetic Resonance

NMR is a physical phenomenon in which atomic nuclei absorb and re-emit electromagnetic (EM) radiation in a magnetic field [97, 32]. It is a non-destructive testing technique for observation of quantum mechanical magnetic properties of atoms, and is widely used in medical imaging [66, 23], and structural determination [12, 20]. The NMR phenomenon is based on the inherent spin of neutrons and protons that make up the atomic nucleus. This spin, determined by the spin quantum number, results from the intrinsic angular momentum of the nucleus. A non-zero spin is associated with a non-zero magnetic moment via the gyromagnetic ratio γ. This magnetic moment of the nucleus precesses in an external magnetic field, similar to the motion of a classical gyroscope in a gravitational field: see Figure 3.1.

In NMR experiments, nuclear spins are first aligned (polarized) in an external magnetic field. This alignment is subsequently perturbed by radio-frequency (RF) pulses with different frequencies, in order to trigger changes in nuclear spin and generate
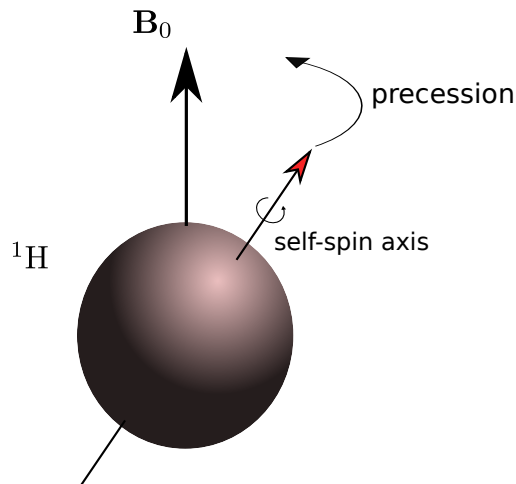
Figure 3.1: A hydrogen nucleus precesses in a magnetic field $\mathbf{B}_0$. The nucleus has an intrinsic spin from angular momentum, whose axis rotates about the $\mathbf{B}_0$ axis.

NMR signals. The RF pulse is produced by a synthesizer and applied in the direction orthogonal to the main magnetic field $\mathbf{B}_0$. An essential fact about atomic nuclei is that they exist in discrete energy states (spin levels), separated by finite energy differences. In the presence of an external magnetic field $\mathbf{B}_0$, this energy difference is given by

$$\Delta E = \gamma \hbar \mathbf{B}_0, \tag{3.1.1}$$

where $\hbar$ is the Planck constant divided by 2π. Transition between spin states occurs only when exactly the correct amount of energy is absorbed or emitted [60]. According to the Planck-Einstein relation, the electromagnetic radiation energy absorbed or emitted is proportional to its frequency ω, given by

$$\Delta E = \hbar \omega. \tag{3.1.2}$$

Combining (3.1.1) and (3.1.2) gives

$$\omega = \gamma \mathbf{B}_0. \tag{3.1.3}$$

This ω is the frequency of EM radiation required to activate a transition between two spin levels, and is referred to as the Larmor precession frequency [65, 15]. Equation (3.1.3) implies that the Larmor frequency only depends on the type of nucleus and the
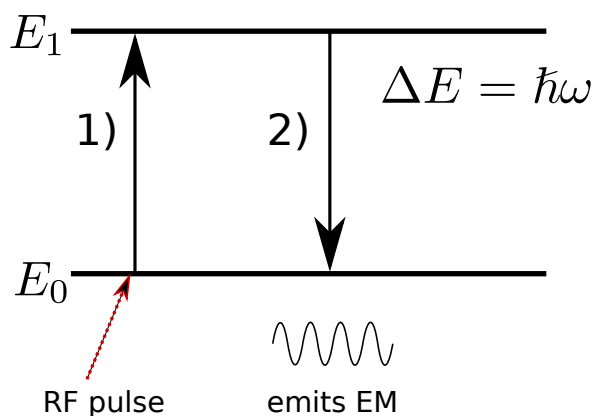
Figure 3.2: RF excitation of nuclei out of thermodynamic equilibrium. 1) A RF pulse of correct frequency ω is applied on a nucleus, causing the spin level to move from E_0 to E_1. 2) When the pulse is turned off, the nucleus decays back to initial state E_0. During this decay, the precessing magnetization vector of the nucleus induces voltage signals in the receiver coil.

magnetic field. When a RF pulse with Larmor precession frequency is applied, nuclear spin moves to a higher level. The nucleus returns to its original state after the pulse is turned off, and emits EM waves: see Figure 3.2.

However, not all nuclei give rise to NMR signals. If the numbers of both protons and neutrons in a nucleus are even, such as ^{12}C and ^{16}O, the overall nuclear spin is zero and no NMR absorption effect is exhibited; hence these nuclei are not used in practice. The most commonly studied nuclei in NMR include ^{1}H, ^{13}C, ^{17}O, etc, which possess a non-zero spin, as well as other nuclei with an odd number of neutrons and/or protons.

Relaxation is the decay of magnetized nuclei to their initial orientations and states. If a sample of nuclei is exposed to a pulse of RF energy, the magnetization vectors are pushed away from the $\mathbf{B_0}$ axis, after which they begin to precess. Voltage signals are induced in the receiver coil by these precessing magnetization vectors, according to Faraday's law of induction. Generally, NMR signals deteriorate with time after the RF pulse is turned off; this phenomenon is called "free induction decay" (FID). This deterioration in NMR signal arises from the nuclear magnetization and over-population of nuclei in high-energy spin states. As a result, an NMR signal is only observed when the RF pulse causes precession, or when (3.1.3) is satisfied. Relaxation experiments mainly measure two useful quantities T_1 and T_2, namely the
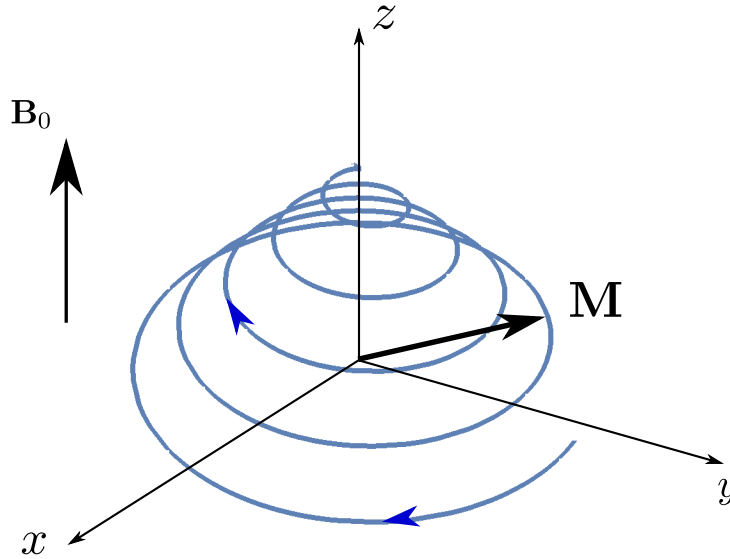
Figure 3.3: Activated by RF pulse, the spin orientation of a nucleus changes. The magnetization vector **M** precesses and finally returns to its initial orientation parallel to the magnetic field. The relaxation time of $\mathbf{M}_z$ in longitudinal direction is measured by T_1, and that of transverse component $\mathbf{M}_{xy}$ is measured by T_2.

"spin-lattice" and "spin-spin" relaxation times [41]. T_1 measures the time it takes for nuclei to release the absorbed energy to their ambient environment, and relax to their initial low-energy distribution. Dephasing exists simultaneously, due to loss of alignment of spin directions, and is measured by T_2. Formally, let **M** be the nuclear spin magnetization vector in $\mathbb{R}^3$. T_1 is the relaxation time of the component $\mathbf{M}_z$ that is parallel to the magnetic field, usually defined as the longitudinal direction (z-axis), whereas T_2 is the relaxation time of the transverse component $\mathbf{M}_{xy}$ that is perpendicular to the magnetic field (xy-plane): see Figure 3.3. T_1 is much larger than T_2 in general.

3.1.2 Introduction to Magnetic Resonance Imaging

MRI, one of the most important applications of NMR, is a medical imaging technique [66, 38], which is prominently used in diagnostic medicine and biomedical research [77, 63, 71]. Multidirectional and high-contrast images of organs in human

bodies, for example, are generated by MRI scanners equipped with strong magnetic fields, electric field gradients and radio wave emitters. These images help diagnosis based on compositional changes in the organ, and assist physicians with appropriate treatments and therapies. Compared to computed tomography (CT) and Positron-Emission Tomography (PET), MRI is often considered as a better choice in medical diagnosis, because it doesn't expose patients to X-rays and ionizing radiation which could potentially lead to cell abnormalities and cancer development. However, MRI operates at a high cost, and its usage remains controversial on patients with implanted devices such as pacemakers [42].

The largest component in a MRI scanner is the main magnet. It is usually made from a superconducting solenoid which generates a homogeneous and stable magnetic field in the scanner. By themselves, the main magnet and RF system only produce signals of a single frequency by equation (3.1.3). Therefore, MRI scanners are also equipped with another component, namely the gradient coils, that modulate the main magnetic field at different spatial points [96]. In practice, three orthogonal magnetic fields $\mathbf{G}_x, \mathbf{G}_y$ and $\mathbf{G}_z$ are generated in addition to the main magnetic field $\mathbf{B}_0$. The strength of these magnetic fields vary linearly along each direction, thus forming the gradient system. After the RF pulse is applied, the gradients act on the precessing nuclei. With the gradients, signals from different locations exhibit different frequencies and phases, thus making spatial localization possible.

Hydrogen atoms ^{1}H are most frequently used to generate detectable signals from biological organisms, due to their abundance in water, sharp MR signal, and fast acquisition time. From these properties, one can essentially map the resonance frequency of hydrogen atoms to their positions in the tissue. On the other hand, different tissue types contain different amounts of water and yield MR signals that decay at different rates.

Our collaborators at the National Institutes of Health use MRI to study osteoarthritis [1], a joint disease resulting from breakdown of the joint cartilage and underlying bone [68] and a major cause of disability worldwide. Their goal is to achieve

greater specificity in MR studies of cartilage macromolecules and develop therapies from early diagnosis. Cartilage is mainly composed of water, but it stores its water in different "pools", for example free water, collagen and proteoglycans. Distinct pools are expected to have distinct water mobility, and exhibit different magnetic resonance properties, and different relaxation times T_1 and T_2. Generally, ^{1}H nuclei in different tissues resonate at the same frequency, but relax at different rates. Hence we may estimate the composition of tissues by examining the relaxation times of Hydrogen atoms therein.

The image of tissues normally includes information of both location (coordinates of each pixel) and composition (intensity of each pixel). Both of them are derived from the voltage signal, or intensity, assigned to that pixel. The electric field gradient system in the MRI scanner is responsible for localizing the MR signal. The applied magnetic field establishes a one-to-one mapping from locations of ^{1}H nuclei to signal frequencies, and the inverse Fourier transform (IFT) is used in computing the signal frequency from the voltage in the electric field gradient system. Since IFT is well-posed [11], the location of points in the tissue that emit signals of a particular frequency can be accurately determined. On the other hand, the intensity of image at a given pixel is determined from a series of voltage signals, which is related to the relaxation times T_1 and T_2 at that pixel. In order to identify the type of corresponding tissue at that pixel, we need to infer the relaxation times by applying the inverse Laplace transform (ILT). Unfortunately, ILT is a well-known ill-posed Fredholm equation of the first kind [72, 31], meaning that the solution from ILT may not be unique, may not exist, or may not depend continuously on the data [54]. The solution could be highly sensitive to noise and allow multiple different solutions to a given dataset. For this reason, determination of relative amounts of proteoglycan, collagen and water in the cartilage is likely to be uncertain, rendering it difficult to predict the characteristics of different types of tissue. Hence the focus from here will be on ILT and determining the composition of tissues from the sizes of water pools they consist of along with their relaxation times.

3.1.3 NMR Relaxometry Formulation

The main interest in this chapter is two-dimensional relaxometry problems. For illustration purposes, one-dimensional relaxometry is considered first.

1D relaxometry

For a Fredholm integral equation of the first kind, we define the kernel function

$$K(u,v) = e^{-u/v}. \tag{3.1.4}$$

Suppose that the unknown distribution of spin-spin relaxation time is $F(T_2)$, which identifies the composition of the tissue being experimented. The relaxation measurement y is given by

$$y(t) = \int_0^\infty K(t, T_2)F(T_2)dT_2, \tag{3.1.5}$$

where $K(t, T)$ is the kernel given in (3.1.4). Discretizing in the domain of T_2 with increments $\Delta T_{2,j}$, (3.1.5) can be written in matrix form as

$$\boldsymbol{y} = \mathbf{K}\boldsymbol{f}, \tag{3.1.6}$$

where $\boldsymbol{y}_i = y(t_i)$, $\mathbf{K}_{ij} = K(t_i, T_{2,j})\Delta T_{2,j}$ and $\boldsymbol{f}_j = F(T_{2,j})$. The goal is to recover $F(T_2)$ given measurement $y(t)$.

2D relaxometry

In two-dimensions, the same kernel (3.1.4) is used. The joint distribution of spin-lattice and spin-spin relaxation times $F(T_1, T_2)$ of various ^{1}H atoms characterizes the composition of tissue under investigation. The measurement data Y has expression

$$Y(t_1, t_2) = \int_0^\infty \int_0^\infty (1 - 2K(t_1, T_1))F(T_1, T_2)K(t_2, T_2)dT_1dT_2, \tag{3.1.7}$$

Although the integrand contains $1 - 2K(t_1, T_1)$, a simpler version will be considered:

$$Y(t_1, t_2) = \int_0^\infty \int_0^\infty K(t_1, T_1)F(T_1, T_2)K(t_2, T_2)dT_1dT_2, \tag{3.1.8}$$

76

Equation (3.1.7) can be solved in the same manner as (3.1.8). However, there needs to be a separate one-dimensional analysis on the marginal $Y = \int_0^\infty F(T_2)K(t_2, T_2)\, dT_2$. In general, this requires an additional control experiment with a sufficiently large t_1 value, in which $K(t_1, T_1) \approx 0$ and data is sampled from the marginal distribution.

In matrix form, upon discretizing in t_1 and t_2, equation (3.1.8) becomes

$$Y(t_{1,i}, t_{2,j}) = \sum_n \sum_m K(t_{1,i}, T_{1,m})F(T_{1,m}, T_{2,n})K(t_{2,j}, T_{2,n}), \qquad (3.1.9)$$

or

$$\mathbf{Y} = \mathbf{K}_1 \mathbf{F} \mathbf{K}_2^T. \qquad (3.1.10)$$

Note that the discretizations in T_1 and T_2 are usually different experimentally so that $\mathbf{K}_1 \neq \mathbf{K}_2$ in general. The matrix components are given by

$$\mathbf{Y}_{ij} = Y(t_{1,i}, t_{2,j}), \qquad (3.1.11)$$

$$(\mathbf{K}_1)_{ij} = K(t_{1,i}, T_{1,j})\Delta T_{1,j}, \qquad (3.1.12)$$

$$(\mathbf{K}_2)_{ij} = K(t_{2,i}, T_{2,j})\Delta T_{2,j}, \qquad (3.1.13)$$

$$\mathbf{F}_{mn} = F(T_{1,m}, T_{2,n}). \qquad (3.1.14)$$

The relaxation times T_1 and T_2 distinguish different tissue types. The goal is to infer $F(T_1, T_2)$ from measurements $Y(t_1, t_2)$.

3.1.4 Ill-posedness of Inverse Laplace Transform

The inverse Laplace transform often arises in exponential analysis to deal with signals related to decaying functions of time $y(t)$. When the decay is described by a continuous distribution (spectral function) $x(\xi)$, the problem can be formulated as a Fredholm equation of the first kind:

$$y(t) = \int_0^\infty K_0(t, \xi)x(\xi)d\xi. \qquad (3.1.15)$$

where $K_0(t, \xi) = e^{-\xi t}$. In special cases when the spectral function is a sum of delta functions

$$x(\xi) = \sum_{i=1}^n \alpha_i \delta(\xi - \xi_i), \qquad (3.1.16)$$

the problem (3.1.15) reduces to multi-exponential analysis,

$$y(t) = \sum_{i=1}^{n} \alpha_i e^{-\xi_i t}. \tag{3.1.17}$$

which will be discussed in Chapter 4.

In theory, one can solve the ILT problems via the inversion formula:

$$x(\xi) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} y(t) e^{\xi t} dt, \tag{3.1.18}$$

where c lies to the right of all the poles of $y(t)$ in the complex plane. This inversion is feasible only if $y(t)$ is given analytically. In practice, however, signals are only measured in real time (not complex), and hence the Bromwich integral (3.1.18) in the complex plane cannot be computed from experimental data. For this reason, the ILT must be solved from the integral equation (3.1.15), which is known to be ill posed.

The ill-posedness of ILT can also be explained in terms of eigenfunctions of the integral equation (3.1.15). Suppose that $\phi_\omega^\pm(\xi)$ are two sets of orthogonal eigenfunctions corresponding to eigenvalues $\lambda_\omega^\pm$, where the plus (minus) sign stands for positive (negative) eigenvalues, i.e.

$$\int_0^\infty K_0(t,\xi)\phi_\omega^\pm(\xi)d\xi = \lambda_\omega^\pm \phi_\omega^\pm(t). \tag{3.1.19}$$

which is introduced in [72]. Both data signal y and spectral function x can be expanded in these eigenfunctions as

$$x(\xi) = \int_0^\infty x_\omega^+ \phi_\omega^+(\xi)d\omega + \int_0^\infty x_\omega^- \phi_\omega^-(\xi)d\omega. \tag{3.1.20}$$

Substituting (3.1.20) into (3.1.15), the data can be expressed as

$$y(t) = \int_0^\infty x_\omega^+ \lambda_\omega^+ \phi_\omega^+(\xi)d\omega + \int_0^\infty x_\omega^- \lambda_\omega^- \phi_\omega^-(\xi)d\omega. \tag{3.1.21}$$

where the coefficients are given by orthogonality:

$$x_\omega^\pm = \frac{1}{\lambda_\omega^\pm} \int_0^\infty y(t)\phi_\omega^\pm(t)dt, \tag{3.1.22}$$

and therefore one can represent the spectral function as

$$x(\xi) = \int_0^\infty d\omega\,\phi_\omega^+(\xi)\frac{1}{\lambda_\omega^+}\int_0^\infty dt\,y(t)\phi_\omega^+(t) + \int_0^\infty d\omega\,\phi_\omega^-(\xi)\frac{1}{\lambda_\omega^-}\int_0^\infty dt\,y(t)\phi_\omega^-(t). \quad (3.1.23)$$

It can been shown that the eigenvalues $\lambda_\omega^\pm$ decrease very rapidly to zero as $\omega \to \infty$, so that the accuracy of coefficients $x_\omega^\pm$ in (3.1.22) become extremely susceptible to noise in $y(t)$. As a consequence, one can never obtain the exact solution to (3.1.15) in practice, and the integral limits of ω in (3.1.23) must be truncated to $[0, \omega_{\max}]$ for a certain threshold $\omega_{\max}$ in order to avoid larger errors.

Fortunately, in many real experiments, one can take advantage of a priori information about the function $x(\xi)$ to extract more meaningful solutions from all possible candidates. This is known as regularization of the inverse problem. A general problem setting is that we have two functionals $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A}$ measures the agreement between data and solution, and $\mathcal{B}$ is concerned with the smoothness or stability of the solution with possible prior information taken into account. If only $\mathcal{A}$ is minimized, the solution agrees with the data to high accuracy, but it is prone to be unstable and physically unrealistic. For a rudimentary example, one can fit a high-order polynomial to noisy data, achieving 100% accuracy and minimizing the discrepancy from solution to data, while the data could just be generated from an underlying linear function. On the other hand, minimizing $\mathcal{B}$ alone yields a solution that is smooth, stable, or consistent with the prior information, but is unrelated to the measured data. For instance, if one believes the solution is nearly linear, then $\mathcal{B}$ may be chosen as a integral of the norm of second derivative.

Most inverse problem methods involve a trade-off between the optimization of $\mathcal{A}$ and $\mathcal{B}$:

$$\min \quad \mathcal{A} + \mu\mathcal{B} \quad (3.1.24)$$

for some regularization parameter $\mu \in (0, \infty)$, and the selection of "best" value of μ may be based on some criterion or be entirely subjective. This regularization parameter makes a compromise between *a priori* expectation in $\mathcal{B}$ with *a posteriori* knowledge in

$\mathcal{A}$ [86]. Inverse problems can be solved by different choices of $\mathcal{A}, \mathcal{B}$ and μ. We shall discuss some possible regularization methods in the following context.

3.1.5 Regularization Methods

We now discuss regularization methods in a discrete setting, and define $t = \{t_1, \dots, t_m\}$ and $\xi = \{\xi_1, \dots, \xi_n\}$. The Laplace kernel $K_0(t, \xi)$ is represented in the matrix form $\mathbf{K}$ such that $\mathbf{K}_{ij} = K_0(t_i, \xi_j)$. Without regularization, the goal is to solve

$$\min_{\boldsymbol{x}} \|\mathbf{K}\boldsymbol{x} - \boldsymbol{y}\|_2, \tag{3.1.25}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the discretized solution vector and data vector, respectively. Suppose the (real) singular value decomposition (SVD) of $\mathbf{K}$ is given by

$$\mathbf{K} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{3.1.26}$$

where $\mathbf{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_n]$ and $\mathbf{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n]$ are orthogonal matrices, and $\boldsymbol{\Sigma} = \mathrm{diag}[\sigma_1, \dots, \sigma_n]$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$. Then the least squares solution is given by the Moore-Penrose pseudo-inverse [83] formula

$$\boldsymbol{x}_{LS} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\boldsymbol{y} = \sum_{i=1}^{n} \frac{\langle \boldsymbol{u}_i, \boldsymbol{y} \rangle}{\sigma_i} \boldsymbol{v}_i. \tag{3.1.27}$$

Unfortunately, similar to (3.1.23), the least squares solution is problematic in practice because the errors are significantly amplified when singular values become too small. A necessary condition for obtaining good regularized solutions is to satisfy the discrete Picard condition [46], i.e. the Fourier coefficient $\langle \boldsymbol{u}_i, \boldsymbol{y} \rangle$ must on average decay to zero faster than the generalized singular values.

A regularization method imposes a filtering factor $\{r_i\}$ in order to damp the effect from division by small singular values in its SVD expansion,

$$\boldsymbol{x}_{reg} = \sum_{i=1}^{n} r_i \frac{\langle \boldsymbol{u}_i, \boldsymbol{y} \rangle}{\sigma_i} \boldsymbol{v}_i, \tag{3.1.28}$$

where we assume $0 \leq r_i \leq 1$ for damping purposes. From this perspective, a large class of regularization methods differ in their filtering factors. Nevertheless, there are also other regularization methods that cannot be described by filtering factors.

Tikhonov regularization

Tikhonov regularization [85, 101] is one of the most popular regularization methods for solving inverse problems. This method searches for an approximate solution that solves the following optimization problem as in integral equation (3.1.15):

$$\min \left\| \int_0^\infty K_0(t, \sigma)x(\sigma)d\sigma - y(t) \right\| + \mu \mathcal{B}[x(\sigma)], \tag{3.1.29}$$

where $\mathcal{B}$ is the smoothing functional containing *a priori* information (or subjective conjecture), and $\mu \geq 0$ is the regularization parameter. In the discrete case, the objective is usually in the following form:

$$\min_{\boldsymbol{x}} \|\mathbf{K}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu\|\boldsymbol{x}\|_2^2. \tag{3.1.30}$$

A useful method for choosing μ is called the "L-curve" introduced in [47]. With an appropriate choice, (3.1.30) could be solved by many optimization techniques [35] such as gradient descent, quasi-Newton, Nelder-Mead, differential evolution [98], etc. It can be shown that the solution to (3.1.30) is given by normal equation

$$\boldsymbol{x}_{reg} = (\mathbf{K}^T\mathbf{K} + \mu\mathbf{I})^{-1}\mathbf{K}^T\boldsymbol{y} = \sum_{i=1}^n r_i \frac{\langle \boldsymbol{u}_i, \boldsymbol{y}\rangle}{\sigma_i}\boldsymbol{v}_i, \tag{3.1.31}$$

with filtering factor

$$r_i = \frac{\sigma_i^2}{\sigma_i^2 + \mu}. \tag{3.1.32}$$

Given a fixed μ, it is clear that the filtering factors for larger singular values are closer to 1, while the ones corresponding to smaller singular values are close to 0, which diminishes the noise amplification effect.

Truncated SVD (TSVD)

The TSVD method [45] is simply obtained by truncating least squares solution (3.1.27) as singular values decrease to some threshold $\sigma_{\min}$. The solution is given by (3.1.28) with filtering factor

$$r_i = \begin{cases} 1, & \text{for } \sigma_i \geq \sigma_{\min} \\ 0, & \text{for } \sigma_i < \sigma_{\min} \end{cases} \tag{3.1.33}$$

and the 1-norm of the residual is minimized in this method. The challenge in this method is where to truncate the series.

Lasso

Lasso [100], short for "least absolute shrinkage and selection operator", is a popular modern regularization method that uses the 1-norm as the smoothing functional compared to the Tikhonov regularization. Its objective function is

$$\min_{\boldsymbol{x}} \|\mathbf{K}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu \|\boldsymbol{x}\|_1. \tag{3.1.34}$$

Because of the nature of the 1-norm constraint, the Lasso method produces a sparse solution vector $\boldsymbol{x}$ and gives easily interpretable models. It exhibits the interpretability property of subset selection techniques and stability like Tikhonov regularization. The Lasso method essentially finds the first point where the contours of residual hit the constraint region defined by the 1-norm, which are rhomboids in multi-dimensional space. If the intersection is at a corner, then one estimated parameter becomes zero, hence producing sparsity in the solution.

Furthermore, another method called elastic net [109] is an extension to both Lasso and Tikhonov regularization, which outperforms Lasso when the number of predictors (n) is much larger than the number of observations (m). However, the choices of two tuning parameters would be challenging.

3.2 Regularization in 2D NMR Relaxometry

In this section, regularization methods are applied to NMR relaxation problems. First, we discuss the simple one-dimensional case in which we want to solve $\mathbf{K}\boldsymbol{f} = \boldsymbol{y}$. In order to obtain the spin-spin time distribution $f(T_2)$, a direct method such as L_2 Tikhonov regularization can be used. The residual to be minimized is

$$r(\boldsymbol{f}) = (\boldsymbol{y} - \mathbf{K}\boldsymbol{f})^T(\boldsymbol{y} - \mathbf{K}\boldsymbol{f}) + \mu \boldsymbol{f}^T \boldsymbol{f} \tag{3.2.1}$$

and the Tikhonov regularization solution is given by solving $dr/d\boldsymbol{f} = 0$ in (3.2.1):

$$\boldsymbol{f} = (\mathbf{K}^T\mathbf{K} + \mu \mathbf{I})^{-1}\mathbf{K}^T\boldsymbol{y}, \tag{3.2.2}$$

where μ is the regularization parameter, and $\mathbf{I}$ is the identity matrix. In fact, [17] points out that extension to two-dimensional relaxometry framework followed by projection onto the T_2 axis would yield more stable results than direct inference in one dimension.

3.2.1 One-Parameter Regularization

In two dimensions, one can also use the Tikhonov regularization with only one tuning parameter μ to solve for $\mathbf{F}$ in (3.1.10). The problem setup is the following:

$$\min_{\mathbf{F}} \|\mathbf{K}_1 \mathbf{F} \mathbf{K}_2^T - \mathbf{Y}\|_F^2 + \mu \|\mathbf{F}\|_F^2, \tag{3.2.3}$$

with Frobenius norms. First, the vectorization of a matrix $\mathbf{X}_{m \times n}$ is defined as stacking the columns of that matrix:

$$\text{vec}(\mathbf{X}) = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \end{pmatrix}_{(mn) \times 1}, \tag{3.2.4}$$

where $\mathbf{X}_i$ is the i-th column of the matrix $\mathbf{X}$. When implementing the minimization problems numerically, it is always convenient to convert matrices to vectors instead. This involves the Kronecker product in the following relation:

$$\text{vec}(\mathbf{A}\mathbf{B}\mathbf{C}^T) = (\mathbf{C} \otimes \mathbf{A})\text{vec}(\mathbf{B}). \tag{3.2.5}$$

With vectorization, an equivalent problem to (3.2.3) is

$$\min_{\mathbf{F}} \|(\mathbf{K}_2 \otimes \mathbf{K}_1)\text{vec}(\mathbf{F}) - \text{vec}(\mathbf{Y})\|_2^2 + \mu \|\text{vec}(\mathbf{F})\|_2^2 \tag{3.2.6}$$

with 2-norms, or combining two terms together:

$$\min_{\mathbf{F}} \left\| \begin{pmatrix} \mathbf{K}_2 \otimes \mathbf{K}_1 \\ \mu \mathbf{I} \end{pmatrix} \text{vec}(\mathbf{F}) - \begin{pmatrix} \text{vec}(\mathbf{Y}) \\ \mathbf{0} \end{pmatrix} \right\|_2^2. \tag{3.2.7}$$

SVD structure of one-parameter regularization

Now we shall analyze this optimization problem in its SVD form[1]. Define the dimensions for the discretized 2D NMR problem as following:

$$\mathbf{K}_1 \in \mathbb{R}^{N_1 \times M_1}, \mathbf{K}_2 \in \mathbb{R}^{N_2 \times M_2}, \mathbf{F} \in \mathbb{R}^{M_1 \times M_2}, \mathbf{Y} \in \mathbb{R}^{N_1 \times N_2},$$

$$\boldsymbol{y} = \text{vec}(\mathbf{Y}), \ \boldsymbol{f} = \text{vec}(\mathbf{F}), \ \mathbf{I} = \text{identity of size } M_1 M_2 \times M_1 M_2.$$

The reduced SVD for $\mathbf{K}_i$ is defined as

$$\mathbf{K}_i = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T, \quad i = 1, 2, \tag{3.2.8}$$

where $\mathbf{U}_i \in \mathbb{R}^{N_i \times M_i}, \boldsymbol{\Sigma}_i \in \mathbb{R}^{M_i \times M_i}$ and $\mathbf{V}_i \in \mathbb{R}^{M_i \times M_i}$. By the properties of the Kronecker product [103], we have

$$\mathbf{K}_2 \otimes \mathbf{K}_1 = (\mathbf{U}_2 \boldsymbol{\Sigma}_2 \mathbf{V}_2^T) \otimes (\mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^T) = (\mathbf{U}_2 \otimes \mathbf{U}_1)(\boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1)(\mathbf{V}_2 \otimes \mathbf{V}_1)^T. \tag{3.2.9}$$

Suppose that $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are given as

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \sigma_{1,1} & & & \\ & \sigma_{1,2} & & \\ & & \ddots & \\ & & & \sigma_{1,M_1} \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} \sigma_{2,1} & & & \\ & \sigma_{2,2} & & \\ & & \ddots & \\ & & & \sigma_{2,M_2} \end{pmatrix}, \tag{3.2.10}$$

then the singular values of $\mathbf{K}_2 \otimes \mathbf{K}_1$ are all pairwise products of singular values of $\mathbf{K}_1$ and $\mathbf{K}_2$. Define

$$\widetilde{\mathbf{U}} = \mathbf{U}_2 \otimes \mathbf{U}_1, \ \widetilde{\mathbf{K}} = \mathbf{K}_2 \otimes \mathbf{K}_1, \ \widetilde{\mathbf{V}} = \mathbf{V}_2 \otimes \mathbf{V}_1, \tag{3.2.11}$$

with the singular values of $\widetilde{\mathbf{K}}$ sorted in descending order $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \ldots \geq \widetilde{\sigma}_{M_1 M_2} \geq 0$. The columns of $\widetilde{\mathbf{U}} = [\widetilde{\boldsymbol{u}}_1, \ldots, \widetilde{\boldsymbol{u}}_{M_1 M_2}]$ and $\widetilde{\mathbf{V}} = [\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_{M_1 M_2}]$ are also rearranged accordingly. From (3.1.31), the solution of (3.2.6) can be written as

$$\boldsymbol{f}_\mu = \sum_{i=1}^{M_1 M_2} \widetilde{r}_i \frac{\langle \widetilde{\boldsymbol{u}}_i, \boldsymbol{y} \rangle}{\widetilde{\sigma}_i} \widetilde{\boldsymbol{v}}_i, \tag{3.2.12}$$

---

[1] This calculation follows from the one presented in the following unpublished article by Hasan Celik, Ariel Hafftka, Wojtec Czaja, Richard G. Spencer. "Singular Value Decomposition Analysis of the Stabilization of the Inverse Laplace Transform of Multiexponential Decays through Extension into a Second Dimension".

with filtering factor

$$\widetilde{r}_i = \frac{\widetilde{\sigma}_i^2}{\widetilde{\sigma}_i^2 + \mu}. \tag{3.2.13}$$

3.2.2 Two-Parameter Regularization

One parameter regularization is simple and fast in reconstructing the underlying distribution, but it only does so by minimizing the 2-norm of residual. In cases that the distribution $F(T_1, T_2)$ is anisotropic with specific directions, one parameter alone may not be enough to recover this information. This motivates our investigation into two-parameter regularization below. Without pretending to be too rigorous, one can try to apply pseudo-inverses on both sides of (3.1.10) to obtain

$$\mathbf{F} = (\mathbf{K}_1^T\mathbf{K}_1 + \mu_1\mathbf{I}_1)^{-1}\mathbf{K}_1^T\mathbf{YK}_2(\mathbf{K}_2^T\mathbf{K}_2 + \mu_2\mathbf{I}_2)^{-1}. \tag{3.2.14}$$

This equation has, in general, two regularization parameters μ_1 and μ_2 which could be different since the kernels $\mathbf{K}_1$ and $\mathbf{K}_2$ don't have to be the same.

To show this, we start with equation $\mathbf{K}_1\mathbf{FK}_2^T = \mathbf{Y}$ as in (3.1.10) and introduce the new variable

$$\mathbf{X} = \mathbf{FK}_2^T. \tag{3.2.15}$$

Then Tikhonov regularization of equation $\mathbf{K}_1\mathbf{X} = \mathbf{Y}$ is given by

$$\min_{\mathbf{X}} \|\mathbf{K}_1\mathbf{X} - \mathbf{Y}\|_F^2 + \mu_2\|\mathbf{X}\|_F^2, \tag{3.2.16}$$

or

$$(\mathbf{K}_1^T\mathbf{K}_1 + \mu_1\mathbf{I}_1)\mathbf{X} = \mathbf{K}_1^T\mathbf{Y}. \tag{3.2.17}$$

The equivalence between (3.2.16) and (3.2.17) is demonstrated in Lemma 3.1. Minimizing (3.2.16) or solving (3.2.17) yields $\mathbf{X}$. This matrix is then used to solve the regularized version of (3.2.15) for $\mathbf{F}$:

$$\min_{\mathbf{F}} \|\mathbf{FK}_2^T - \mathbf{X}\|_F^2 + \mu_1\|\mathbf{F}\|_F^2, \tag{3.2.18}$$

or

$$\mathbf{F}(\mathbf{K}_2^T\mathbf{K}_2 + \mu_2\mathbf{I}_2) = \mathbf{XK}_2. \tag{3.2.19}$$

The procedure to find the regularized $\mathbf{F}$ involves first minimizing (3.2.16), then (3.2.18). It's clear that this process is equivalent to (3.2.14) because (3.2.17) implies

$$(\mathbf{K}_1^T\mathbf{K}_1 + \mu_1\mathbf{I}_1)\mathbf{X}\mathbf{K}_2 = \mathbf{K}_1^T\mathbf{Y}\mathbf{K}_2 \tag{3.2.20}$$

or

$$(\mathbf{K}_1^T\mathbf{K}_1 + \mu_1\mathbf{I})\mathbf{F}(\mathbf{K}_2^T\mathbf{K}_2 + \mu_2\mathbf{I}) = \mathbf{K}_1^T\mathbf{Y}\mathbf{K}_2, \tag{3.2.21}$$

using (3.2.19). We then recover (3.2.14) by taking inverses on both sides.

Lemma 3.1 (Normal equation)**.** *The least squares problem with Frobenius norm*

$$\arg\min_{\mathbf{X}} \|\mathbf{M}\mathbf{X} - \mathbf{B}\|_F^2 + \mu\|\mathbf{X}\|_F^2 \tag{3.2.22}$$

is equivalent to the normal equation

$$(\mathbf{M}^T\mathbf{M} + \mu\mathbf{I})\mathbf{X} = \mathbf{M}^T\mathbf{B}, \tag{3.2.23}$$

where $\mathbf{M}, \mathbf{X}, \mathbf{B}$ *are all matrices,* $\mathbf{I}$ *is the identity matrix, and* μ *is the regularization parameter.*

Proof. Vectorizing (3.2.22), we get

$$\arg\min_{\mathbf{X}} \|(\mathbf{I} \otimes \mathbf{M})\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{B})\|_2^2 + \mu\|\text{vec}(\mathbf{X})\|_2^2. \tag{3.2.24}$$

Let $\boldsymbol{x} = \text{vec}(\mathbf{X})$ and $\boldsymbol{b} = \text{vec}(\mathbf{B})$. Next, we take the derivative with respect to $\boldsymbol{x}$ and set it to zero:

$$0 = \boldsymbol{x}^T(\mathbf{I} \otimes \mathbf{M})^T(\mathbf{I} \otimes \mathbf{M})\boldsymbol{x} - 2((\mathbf{I} \otimes \mathbf{M})\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b} + \mu\boldsymbol{x}^T\boldsymbol{x} \tag{3.2.25}$$

so that

$$[(\mathbf{I} \otimes \mathbf{M})^T(\mathbf{I} \otimes \mathbf{M}) + \mu\mathbf{I}']\boldsymbol{x} = (\mathbf{I} \otimes \mathbf{M})^T\boldsymbol{b}. \tag{3.2.26}$$

Notice that the sizes of identity matrices $\mathbf{I}$ and $\mathbf{I}'$ are different. Since

$$(\mathbf{I} \otimes \mathbf{M})^T(\mathbf{I} \otimes \mathbf{M}) = \text{diag}(\mathbf{M}^T \cdots \mathbf{M}^T)\,\text{diag}(\mathbf{M} \cdots \mathbf{M}) = \mathbf{I} \otimes (\mathbf{M}^T\mathbf{M}), \tag{3.2.27}$$

we can "unvectorize" (3.2.26) to get

$$(\mathbf{M}^T\mathbf{M} + \mu\mathbf{I}')\mathbf{X} = \mathbf{M}^T\mathbf{B}. \tag{3.2.28}$$

$\square$

SVD structure of two-parameter regularization

Again, we analyze this optimization problem in its SVD form. Recall the dimensions for the discretized 2D NMR problem are

$$\mathbf{K}_1 \in \mathbb{R}^{N_1 \times M_1}, \mathbf{K}_2 \in \mathbb{R}^{N_2 \times M_2}, \mathbf{F} \in \mathbb{R}^{M_1 \times M_2}, \mathbf{Y} \in \mathbb{R}^{N_1 \times N_2}, \mathbf{X} \in \mathbb{R}^{M_1 \times N_2},$$

$$\boldsymbol{y} = \text{vec}(\mathbf{Y}), \ \boldsymbol{f} = \text{vec}(\mathbf{F}), \ \boldsymbol{x} = \text{vec}(\mathbf{X}),$$

$$\mathbf{I}_1 = \text{identity of size } N_2 \times N_2, \mathbf{I}_2 = \text{identity of size } M_1 \times M_1.$$

The reduced SVD for $\mathbf{K}_i$ is already defined in (3.2.8). The two-parameter regularization consists of two separate steps

$$(1) \quad \mathbf{K}_1 \mathbf{X} = \mathbf{Y} \Rightarrow (\mathbf{I}_1 \otimes \mathbf{K}_1)\boldsymbol{x} = \boldsymbol{y}, \tag{3.2.29}$$

$$(2) \quad \mathbf{F}\mathbf{K}_2^T = \mathbf{X} \Rightarrow (\mathbf{K}_2 \otimes \mathbf{I}_2)\boldsymbol{f} = \boldsymbol{x}. \tag{3.2.30}$$

Step (1)

The first step is to optimize the following

$$\underset{\boldsymbol{x}}{\arg\min} \, \|(\mathbf{I}_1 \otimes \mathbf{K}_1)\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu_1 \|\boldsymbol{x}\|_2^2, \tag{3.2.31}$$

with the regularization parameter μ_1, and the solution can be written

$$\boldsymbol{x}_{\mu_1}^\dagger = \sum_{k=1}^{M_1 N_2} \frac{\sigma_k^{(1)}}{\left(\sigma_k^{(1)}\right)^2 + \mu_1} \langle \boldsymbol{w}_k^{(1)}, \boldsymbol{y} \rangle \boldsymbol{z}_k^{(1)}, \tag{3.2.32}$$

where $\boldsymbol{w}_k^{(1)}$ is the k-th column of $\mathbf{I}_1 \otimes \mathbf{U}_1$, $\boldsymbol{z}_k^{(1)}$ is the k-th column of $\mathbf{I}_1 \otimes \mathbf{V}_1$, $\{\sigma_k^{(1)}\}$ are the singular values of $\mathbf{I}_1 \otimes \mathbf{K}_1$, and $\mathbf{U}_1$ is defined in (3.2.8). If we denote $\mathbf{U}_1 = \left[\boldsymbol{u}_1^{(1)}, \ldots, \boldsymbol{u}_{M_1}^{(1)}\right]$, then

$$\mathbf{I}_1 \otimes \mathbf{U}_1 = \begin{pmatrix} \boxed{\mathbf{U}_1} & & & & \\ & \boxed{\mathbf{U}_1} & & & \\ & & \boxed{\mathbf{U}_1} & & \\ & & & \ddots & \\ & & & & \boxed{\mathbf{U}_1} \end{pmatrix} \tag{3.2.33}$$

with

$$\boldsymbol{w}_k^{(1)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{u}_{k\%N_1}^{(1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{3.2.34}$$

where $\%$ is the modulo operator. The first and last indices of $\boldsymbol{u}_{k\%N_1}^{(1)}$ in $\boldsymbol{w}_k^{(1)}$ are $\lfloor k/M_1 \rfloor N_1 + 1$ and $\lceil k/M_1 \rceil N_1$, respectively. The vector $\boldsymbol{z}_k^{(1)}$ follows the same pattern as $\boldsymbol{w}_k^{(1)}$.

Step (2)

In the second step, we use the result of $\boldsymbol{x}_{\mu_1}^\dagger$ computed from the previous step to optimize

$$\arg\min_{\boldsymbol{f}} \|(\mathbf{K}_2 \otimes \mathbf{I}_2)\boldsymbol{f} - \boldsymbol{x}\|_2^2 + \mu_2 \|\boldsymbol{f}\|_2^2, \tag{3.2.35}$$

with the solution

$$\boldsymbol{f}_{\mu_2}^\dagger = \sum_{k=1}^{M_1 M_2} \frac{\sigma_k^{(2)}}{(\sigma_k^{(2)})^2 + \mu_2} \langle \boldsymbol{w}_k^{(2)}, \boldsymbol{x}_{\mu_1}^\dagger \rangle \boldsymbol{z}_k^{(2)}, \tag{3.2.36}$$

where $\boldsymbol{w}_k^{(2)}$ is the k-th column of $\mathbf{U}_2 \otimes \mathbf{I}_2$, $\boldsymbol{z}_k^{(2)}$ is the k-th column of $\mathbf{V}_2 \otimes \mathbf{I}_2$, and $\{\sigma_k^{(2)}\}$ are the singular values of $\mathbf{K}_2 \otimes \mathbf{I}_2$. If we denote $\mathbf{U}_2 = \left[\boldsymbol{u}_1^{(2)}, \ldots, \boldsymbol{u}_{M_2}^{(2)} \right]$, then

$$\mathbf{U}_2 \otimes \mathbf{I}_2 = \begin{pmatrix} u_{1,1}^{(2)}\mathbf{I}_2 & u_{2,1}^{(2)}\mathbf{I}_2 & \cdots & u_{M_2,1}^{(2)}\mathbf{I}_2 \\ u_{1,2}^{(2)}\mathbf{I}_2 & u_{2,2}^{(2)}\mathbf{I}_2 & \cdots & u_{M_2,2}^{(2)}\mathbf{I}_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_{1,N_2}^{(2)}\mathbf{I}_2 & u_{2,N_2}^{(2)}\mathbf{I}_2 & \cdots & u_{M_2,N_2}^{(2)}\mathbf{I}_2 \end{pmatrix} \tag{3.2.37}$$

with

$$\boldsymbol{w}_k^{(2)} = \begin{pmatrix} 0 \\ \vdots \\ u_{\lceil k/M_1 \rceil, 1}^{(2)} \\ \vdots \\ 0 \\ \hline \vdots \\ \vdots \\ \hline 0 \\ \vdots \\ u_{\lceil k/M_1 \rceil, N_2}^{(2)} \\ \vdots \\ 0 \end{pmatrix}. \tag{3.2.38}$$

The nonzero elements in the $\boldsymbol{w}_k$ are all elements of the vector $\boldsymbol{u}_{\lceil k/M_1 \rceil}^{(2)}$, and they are in the $(k\%M_1)$-th diagonal position in each small block. The right vector $\boldsymbol{z}_k^{(2)}$ has a similar pattern.

Result of two steps

Substituting the result of (3.2.32) into (3.2.36), we get the following formula for the two-parameter regularization:

$$\boldsymbol{f}_{\mu_1,\mu_2}^\dagger = \sum_{k=1}^{M_1 M_2} \sum_{j=1}^{M_1 N_2} \left(\frac{\sigma_k^{(2)}}{\left(\sigma_k^{(2)}\right)^2 + \mu_2} \right) \left(\frac{\sigma_j^{(1)}}{\left(\sigma_j^{(1)}\right)^2 + \mu_1} \right) \langle \boldsymbol{y}, \boldsymbol{w}_j^{(1)} \rangle \langle \boldsymbol{z}_j^{(1)}, \boldsymbol{w}_k^{(2)} \rangle \boldsymbol{z}_k^{(2)}.$$
$$\tag{3.2.39}$$

We further explore the structures of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ in (3.2.8) such that

$$\mathbf{I}_1 \otimes \boldsymbol{\Sigma}_1 = \begin{pmatrix} \boxed{\boldsymbol{\Sigma}_1} & & & \\ & \boxed{\boldsymbol{\Sigma}_1} & & \\ & & \ddots & \\ & & & \boxed{\boldsymbol{\Sigma}_1} \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 \otimes \mathbf{I}_2 = \begin{pmatrix} \boxed{\sigma_{2,1}\mathbf{I}_2} & & & \\ & \boxed{\sigma_{2,2}\mathbf{I}_2} & & \\ & & \ddots & \\ & & & \boxed{\sigma_{2,M_2}\mathbf{I}_2} \end{pmatrix}$$

$$\tag{3.2.40}$$

This leads to the following mappings:

$$\sigma_j^{(1)} = \sigma_{1,(j-1)\%M_1+1} \tag{3.2.41}$$

$$\sigma_k^{(2)} = \sigma_{2,\lceil k/M_1\rceil} \tag{3.2.42}$$

Alternatively, we may rewrite the two parameter regularization formula (3.2.39) as

$$\boldsymbol{f}_{\mu_1,\mu_2}^{\dagger} = \sum_{k=1}^{M_1 M_2} \sum_{j=1}^{M_1 N_2} \left(\frac{\sigma_{2,\lceil k/M_1\rceil}}{\sigma_{2,\lceil k/M_1\rceil}^2 + \mu_2} \right) \left(\frac{\sigma_{1,(j-1)\%M_1+1}}{\sigma_{1,(j-1)\%M_1+1}^2 + \mu_1} \right) \langle \boldsymbol{y}, \boldsymbol{w}_j^{(1)} \rangle \langle \boldsymbol{z}_j^{(1)}, \boldsymbol{w}_k^{(2)} \rangle \boldsymbol{z}_k^{(2)}.$$

$$\tag{3.2.43}$$

In practice, $\boldsymbol{y}$ will be noisy; all other quantities such as $\sigma_{1,i}, \sigma_{2,j}, \boldsymbol{w}_j^{(1)}, \boldsymbol{w}_k^{(2)}, \boldsymbol{z}_j^{(1)}$ and $\boldsymbol{z}_k^{(2)}$ come from SVD decompositions of the exact kernels.

3.2.3 Numerical Results

In this section, we demonstrate some numerical results for one and two-dimensional NMR relaxometry problems.

1D NMR

The 1D Picard coefficients with one parameter are the unregularized coefficents of the vector $\boldsymbol{v}_i$ in series expansion (3.1.27), given by

$$\text{1D Picard coefficients} = \frac{\langle \boldsymbol{u}_i, \boldsymbol{y} \rangle}{\sigma_i}. \tag{3.2.44}$$

By examining the Picard coefficient, we can see how fast the solution blows up with error in the data.

We first graph the Picard coefficients of one-dimensional NMR problem for different levels of signal-to-noise ratio (SNR) in Figure 3.4. Without regularization,
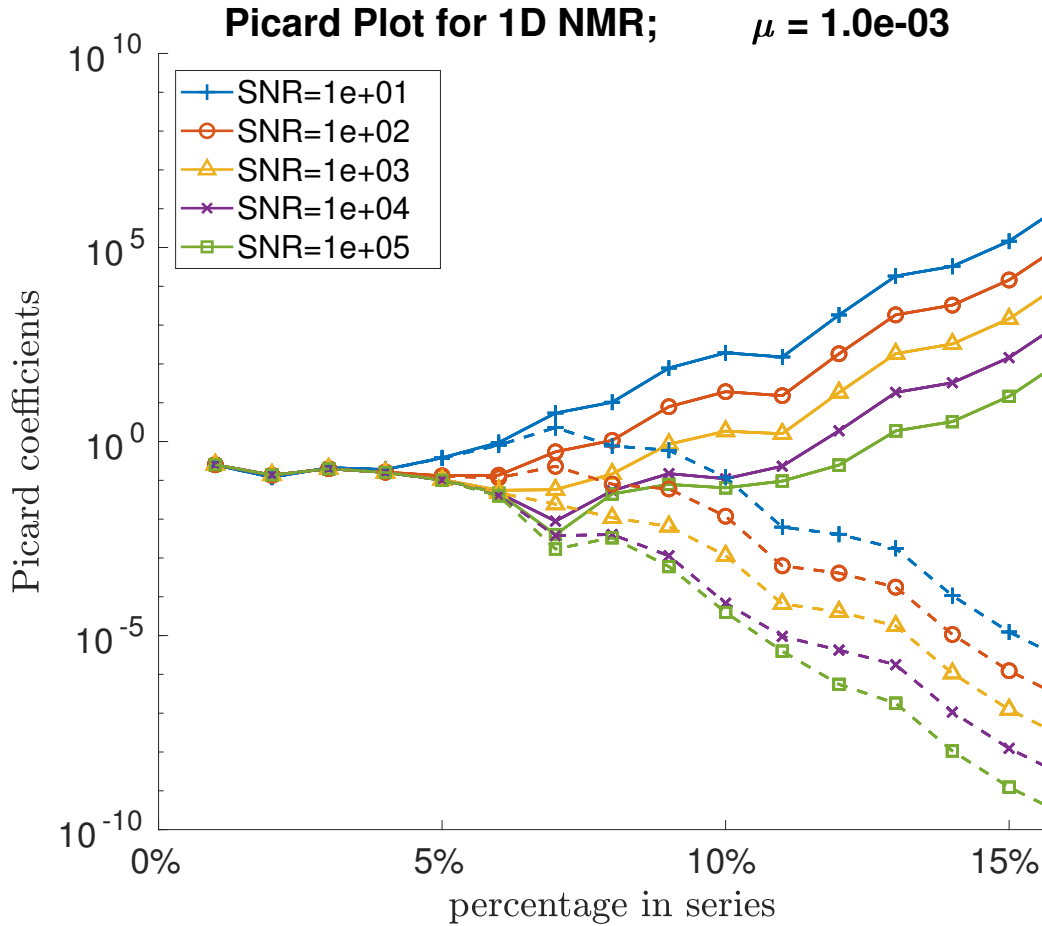
Figure 3.4: Picard plots for one-dimensional NMR. The Picard coeffients are plotted against the percentage of series taken in expansion (3.1.27) for different levels of SNR. Lines with same color has same SNR. Solid line: without regularization. Dashed line: with regularization parameter $\mu = 0.001$.

it is clear that the Picard coefficient blows up quickly when the noise is present, and the effective percentage of series that can be used to approximate solution is only a few percent. On the contrary, when a single parameter is used as a filter, the Picard coefficients stay small and thus the noise is controlled.

2D NMR

In 2D problems, we compare the one-parameter and two-parameter regularization results. The 2D 1-parameter Picard coefficients are the unregularized coefficients

in (3.2.12):

$$\text{2D 1-parameter Picard coefficients} = \frac{\langle \widetilde{\boldsymbol{u}}_i, \boldsymbol{y} \rangle}{\widetilde{\sigma}_i}. \tag{3.2.45}$$

The 2D 2-parameter Picard coefficients are the unregularized coefficients in (3.2.39):

$$\text{2D 2-parameter Picard coefficients} = \frac{\langle \boldsymbol{y}, \boldsymbol{w}_j^{(1)} \rangle \langle \boldsymbol{z}_j^{(1)}, \boldsymbol{w}_k^{(2)} \rangle}{\sigma_k^{(2)} \sigma_j^{(1)}}. \tag{3.2.46}$$

The SVD analysis for the two-dimensional NMR problem is shown in Figures 3.5 and 3.6. We graph the Picard coefficients with or without regularization subject to $\text{SNR} = 1 \times 10^3$. One-parameter regularization (3.2.12) is compared with two-parameter regularization (3.2.39). For the one-parameter regularization, the singular values $\{\tilde{\sigma}_i\}$ are pairwise products of singular values of $\mathbf{K}_1$ and $\mathbf{K}_2$, whose size is $M_1 M_2$. In Figure 3.5, the series (3.2.12) has been sorted in decreasing order of the singular values $\tilde{\sigma}_i$. On the other hand, the singular values $\{\sigma_k^{(2)} \sigma_j^{(1)}\}$ are also calculated as the pairwise products of singular values of $\mathbf{K}_1$ and $\mathbf{K}_2$. However, duplicates exists in these singular values, due to the Kronecker products with identity matrices. In Figure 3.6, the series (3.2.39) is also sorted according to decreasing order of singular values.

In both graphs, the regularized Picard coefficients are bounded for all terms taken in the series, while the unregularized coefficients grow exponentially with more terms in the series. In addition, while the two-parameter regularization SVD formula provides more flexibility over the parameter choices, it results in an explosion in the number of terms in the expansion and a much more oscillatory result compared to the one-parameter method. The duplicates in two-parameter series expansion accounts for the oscillation in the result: the terms with the same singular value are grouped together in one chunk, but the sorting within this chunk is not specified. The behaviors of both methods are similar given the same percentage of series used, and it clearly shows that two-parameter regularization is not as good as one-parameter regularization. In view of simplicity and efficiency, we choose one-parameter Tikhonov regularization for inference in the two-dimensional problem.

Futhermore, we show some numerical results for the reconstruction of the two-dimensional distribution $F(T_1, T_2)$ in Figures 3.7, 3.8, 3.9. They are similar to the
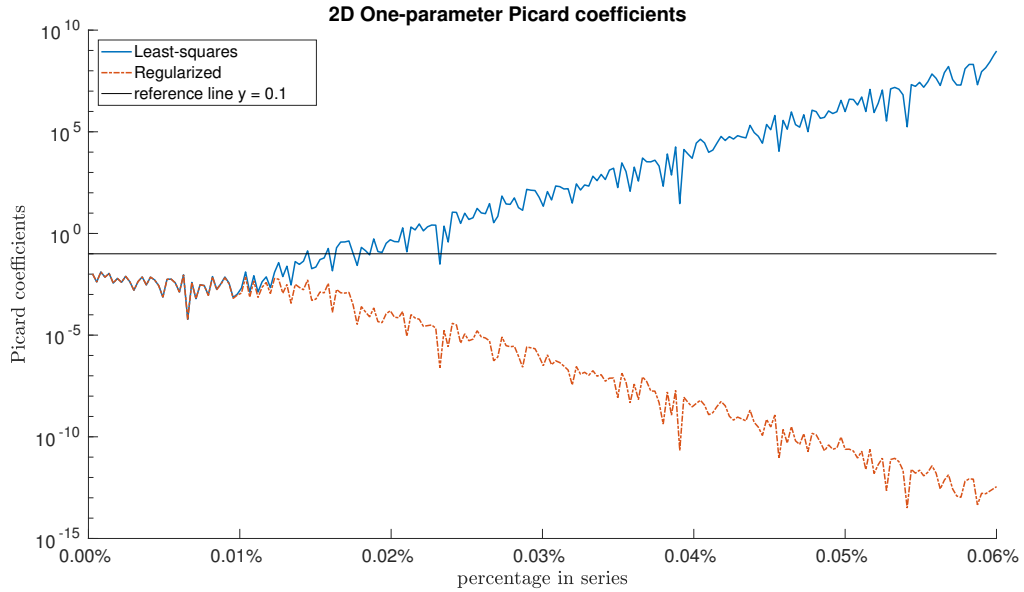
Figure 3.5: Picard plot for two-dimensional NMR with one-parameter regularization. The Picard coeffients in (3.2.45) are plotted against the percentage of series taken in expansion (3.2.12) for SNR $= 1 \times 10^3$. Solid blue line: without regularization. Dashed red line: with *one* regularization parameter $\mu = 1 \times 10^{-2}$.
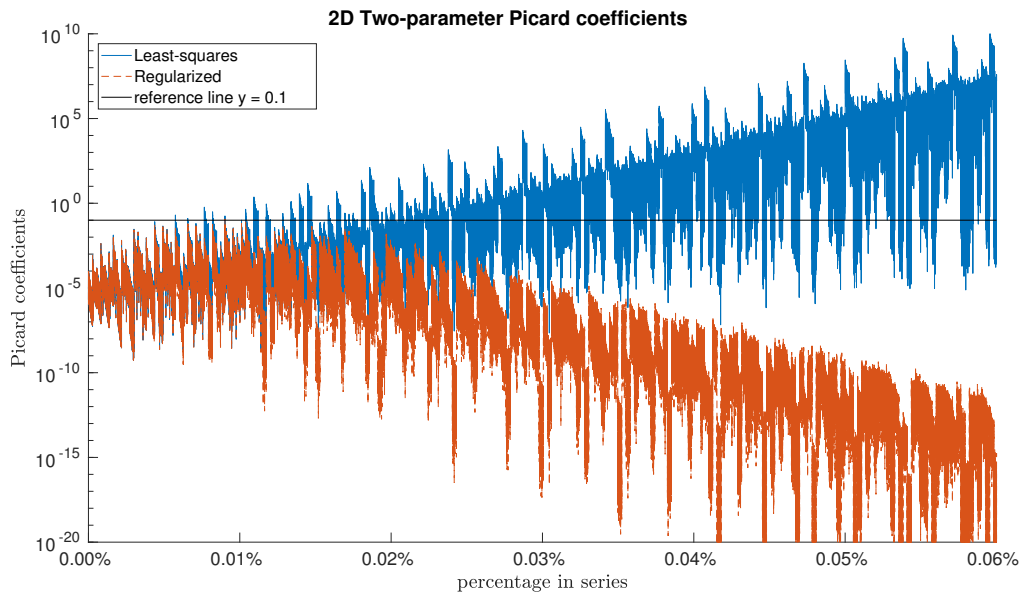
Figure 3.6: Picard plot for two-dimensional NMR with two-parameter regularization. The Picard coeffients in (3.2.46) are plotted against the percentage of series taken in expansion (3.2.39) for SNR $= 1 \times 10^3$. Solid blue line: without regularization. Dashed red line: with *two* regularization parameters $\mu_1 = 1 \times 10^{-2}$ and $\mu_2 = 8 \times 10^{-3}$.

plots given in [17]. The graph on the left is the input peak representing the underlying distribution $F(T_1, T_2)$. The simulated data $Y(t_1, t_2)$, an exponential surface, is then generated by sampling from this distribution according to (3.1.8). With the simulated data, we attempted to reconstruct the original distribution $F(T_1, T_2)$ with both one-parameter and two-parameter Tikhonov regularization methods.

Our main interest is in peak locations, because this information indicates the type of tissue from magnetic imaging. On the contrary, two parameters regularization method as described in Section (3.2.2) cannot resolve the two peaks. The regularization parameter for the one-parameter method can be chosen by the L-curve method [47] mentioned earlier. For the two-parameter method, we have tried many different regularization parameters similar to the value used in the one-parameter regularization, it either results in merged peaks or scattered points. Furthermore, one-parameter regularization scheme is much faster than the two-parameter scheme.

From physical perspective, the peaks must have positive magnitudes. As a result, this reconstruction procedure is restricted by the nonnegative nonlinear least squares (NNLS) method, which is implemented via the KKT condition [64]. This is a very important constraint, without which the method produces much poorer results. Aside from this, one should also note that the SVD analysis in the previous section does not account for the positivity constraint.

3.3   Directional Total Variation Regularization

We now introduce another type of regularization by using the directional total variation (DTV) as the smoothing functional in (3.1.24). Total variation (TV) is a measure of the variation in an image [18] that is often used for denoising in images [105, 5]. In anisotropic images with a dominant direction, the DTV can be used to obtain higher quality denoised images [4, 59]. The DTV for a function $u = u(x, y)$ with bounded variation in the domain Ω is defined as

$$\text{DTV}_{a,\theta}(u) = \iint_{\Omega} \sqrt{(D_\theta u)^2 + (aD_{\theta^\perp}u)^2}\, dxdy \qquad (3.3.1)$$
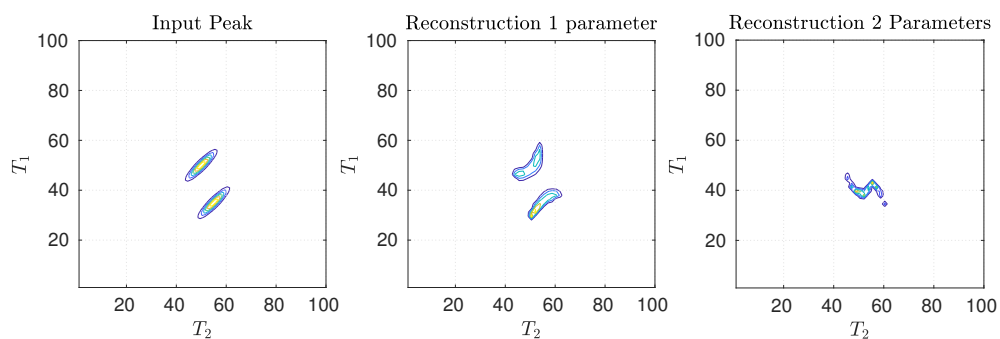
Figure 3.7: Reconstruction of two-dimensional NMR distribution. The horizontal axis is the spin-spin time, and the vertical axis is the spin-lattice time. (a) Left: The MR signal is generated by this underlying distribution $F(T_1, T_2)$ with two major peaks of the same shape. (b) Middle: One-parameter Tikhonov regularization is used to reconstruct the distribution, with regularization parameter $\mu = 2 \times 10^{-4}$. This method accurately recovers the original distribution, especially the positions of peaks, which are used to determine the type of tissue under imaging. (c) Right: Two-parameter Tikhonov regularization is used, with similar regularization parameters $\mu_1 = 1 \times 10^{-4}$ and $\mu_2 = 5 \times 10^{-4}$. Two peaks cannot be resolved from this method.
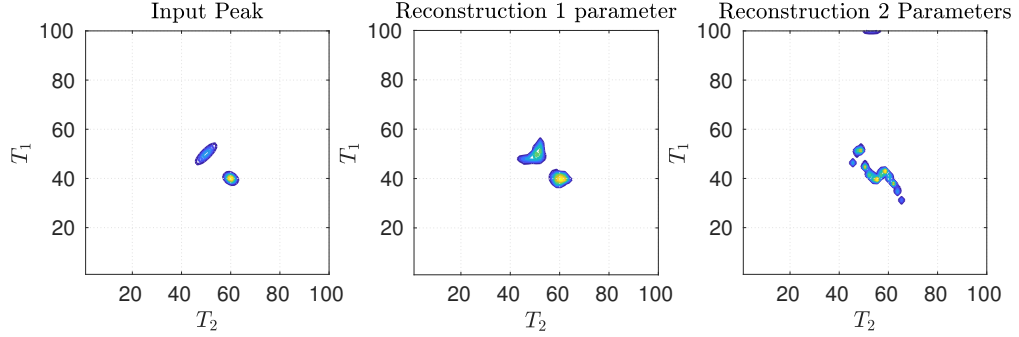
Figure 3.8: Reconstruction of two-dimensional NMR distribution. (a) Left: The MR signal is generated by this underlying distribution $F(T_1, T_2)$ with two major peaks, with different covariances and orientations. (b) Middle: One-parameter Tikhonov regularization with regularization parameter $\mu = 1.5 \times 10^{-4}$. (c) Right: Two-parameter Tikhonov regularization with regularization parameters $\mu_1 = 1 \times 10^{-4}$ and $\mu_2 = 2 \times 10^{-4}$. Two peaks cannot be resolved from this method.

where $D_\theta u$ denotes the directional derivative of u in the direction with unit vector $(\cos\theta, \sin\theta)$, and $\theta^\perp = \theta + \pi/2$. As specified in [59], we focus on the ellipse $E^{a,\theta}(0)$ $(a < 1)$ whose major axis is in the direction $(\cos\theta, \sin\theta)$ with length 2, and minor axis in the direction $(-\sin\theta, \cos\theta)$ with length $2a$. In addition, the Laplace transform operator is defined as

$$(\widetilde{K}F)(p,q) = \iint_\Omega e^{-px} F(x,y) e^{-qy} dx dy. \tag{3.3.2}$$

Note that this is the traditional definition of two-dimensional Laplace transform, different from the NMR conventional form in (3.1.8). In order to relate these two different Laplace transforms, we introduce $\alpha_i = 1/T_i$ for $i = 1, 2$, then the NMR data in (3.1.8) can be written as

$$Y(t_1, t_2) = \int_0^\infty \int_0^\infty e^{-\alpha_1 t_1} f(\alpha_1, \alpha_2) e^{-\alpha_2 t_2} d\alpha_1 d\alpha_2, \tag{3.3.3}$$

where

$$f(\alpha_1, \alpha_2) = \frac{F(\alpha_1^{-1}, \alpha_2^{-1})}{\alpha_1^2 \alpha_2^2}. \tag{3.3.4}$$
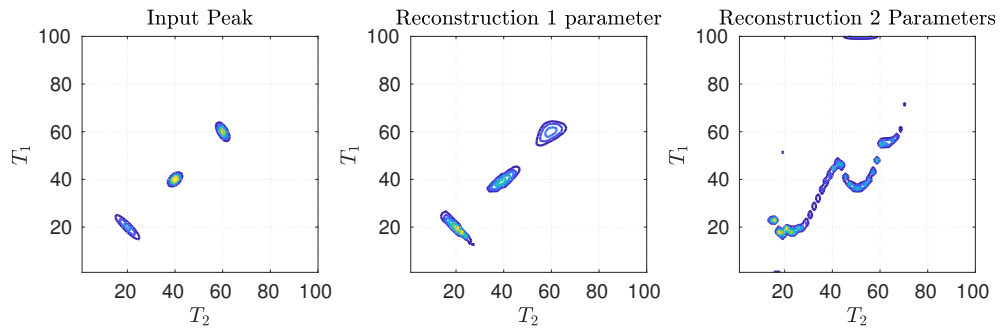
Figure 3.9: Reconstruction of two-dimensional NMR distribution. (a) Left: The MR signal is generated by this underlying distribution $F(T_1, T_2)$ with three major peaks, with different covariances and orientations. (b) Middle: One-parameter Tikhonov regularization with regularization parameter $\mu = 1.5 \times 10^{-4}$. (c) Right: Two-parameter Tikhonov regularization with regularization parameters $\mu_1 = 1 \times 10^{-4}$ and $\mu_2 = 2 \times 10^{-4}$. Two peaks cannot be resolved from this method.

With the above definition, we define the new regularized objective function as

$$I(f) = \frac{1}{2} \iint_{\Omega} (\widetilde{K}f - Y)^2 dt_1 dt_2 + \lambda \, \mathrm{DTV}_{a,\theta}(f), \tag{3.3.5}$$

where $\Omega = [0,\infty) \times [0,\infty)$, $\widetilde{K}$ is the two-dimensional Laplace transform operator, Y is the input data, λ is the regularization parameter. In addition, a, θ can be chosen according to prior information. For instance, we may be able to assess the anisotropy and degree of anisotropy of the distribution f. This is essentially different from the optimization problem in the Bayram paper [4] in two ways. First, the identity kernel is used by Bayram whereas the Laplace kernel is used here. On the other hand, the ultimate goal for Bayram is image denoising in which both input and output are images. However, in our proposed DTV scheme here, we try to recover the underlying distribution for relaxation times/rates given measured data on the exponential surface $Y(t_1, t_2)$.

Furthermore, define the integrand in (3.3.1) as $H_{a,\theta}(u) = \sqrt{(D_\theta u)^2 + (aD_{\theta^\perp}u)^2}$, then by definition of the directional derivative,

$$H_{a,\theta}(u) = \sqrt{(\cos^2\theta + a^2\sin^2\theta)u_x^2 + 2(1-a^2)\cos\theta\sin\theta\, u_x u_y + (a^2\cos^2\theta + \sin^2\theta)u_y^2}. \tag{3.3.6}$$

The integrals in (3.3.5) can be combined as

$$I(f) = \iint_{\Omega} L(f, f_{\alpha_1}, f_{\alpha_2})\, d\alpha_1 d\alpha_2 \tag{3.3.7}$$

where

$$L(f, f_{\alpha_1}, f_{\alpha_2}) = \frac{1}{2}(\widetilde{K}f - Y)^2 + \lambda H_{a,\theta}(f). \tag{3.3.8}$$

Then the minimum of $I(f)$ is the solution to the Euler-Lagrange equation

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial f_{\alpha_1}}\right) - \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial f_{\alpha_2}}\right) = 0, \tag{3.3.9}$$

or

$$(\widetilde{K}^*\widetilde{K}f - \widetilde{K}^*Y) - \lambda \left\{ \frac{\partial}{\partial \alpha_1} \frac{(\cos^2\theta + a^2\sin^2\theta)f_{\alpha_1} + (1-a^2)\cos\theta\sin\theta\, f_{\alpha_2}}{H_{a,\theta}(f)} \right.$$
$$\left. + \frac{\partial}{\partial \alpha_2} \frac{(a^2\cos^2\theta + \sin^2\theta)f_{\alpha_2} + (1-a^2)\cos\theta\sin\theta\, f_{\alpha_1}}{H_{a,\theta}(f)} \right\} = 0. \tag{3.3.10}$$

Since the Laplace transform operator is self-adjoint, the first two terms in (3.3.10) can be expressed as

$$(K^*Y)(\alpha_1, \alpha_2) = \iint_\Omega e^{-\alpha_1 t_1} Y(t_1, t_2) e^{-\alpha_2 t_2} dt_1 dt_2, \qquad (3.3.11)$$

$$(K^*Kf)(\alpha_1, \alpha_2) = \iint_\Omega \frac{f(\hat{\alpha}_1, \hat{\alpha}_2)}{(\alpha_1 + \hat{\alpha}_1)(\alpha_2 + \hat{\alpha}_2)} d\hat{\alpha}_1 d\hat{\alpha}_2. \qquad (3.3.12)$$

The solution to the Euler-Lagrange equation (3.3.10) is found from gradient descent, implemented through a diffusion equation [93],

$$\frac{\partial f}{\partial t} = \frac{\partial L}{\partial f} - \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial f_{\alpha_1}}\right) - \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial f_{\alpha_2}}\right), \text{ for } t > 0, \ \alpha_1, \alpha_2 \in \Omega,$$

$$f(\alpha_1, \alpha_2, 0) \text{ is given at } t = 0, \qquad (3.3.13)$$

$$\frac{\partial f}{\partial \boldsymbol{n}} = 0 \text{ on } \partial\Omega, \text{ where } \boldsymbol{n} \text{ is the outward normal vector.}$$

As t increases, the solution f will stabilize and become the minimum of the objective $I(f)$ in (3.3.5).

We implemented this algorithm with various regularization parameters and numerical schemes. Unfortunately, the numerical results are not satisfactory and do not recover the original distribution $F(T_1, T_2)$. One possible problem is the nonlinear diffusion terms make the CFL condition more complicated. By adding another regularization parameter β in the denominators of (3.3.10), we can reduce the effect of the nonlinear diffusion and avoid dividing by zero on the boundaries. When β is small, nonlinear effects dominates and solution shows random block structures as those related to total variation [93], but the blocks are not at the correct locations. On the other hand, when β is large enough, the problem stabilizes as it becomes more like a normal diffusion problem, in which solution flattens and no particular peak is displayed. For this reason, we don't show any result in this section, but it's still worth the future work on this method.

3.4 Summary

In summary, we have discussed the reconstruction problems in NMR and MRI. Given measurements of the experiments, we aim to recover the underlying distribution

for relaxation times. These problems are ill-conditioned because of the inverse Laplace transform, and we compared the one-parameter and the two-parameter Tikhonov regularization methods in order to resolve the ill-posedness of reconstruction. We showed that the one-parameter regularization method is easy to use and interpret, and it resulted in accurate reconstruction of original distributions. On the contrary, the two-parameter regularization method yields more unstable result due to larger matrices sizes, and the difficult choices for parameters, so that it is even more susceptible to noise in the data.

We also proposed the method with DTV as regularizer to the NMR problems, in which formulations are given but numerical results have not been given. Both the two-parameter Tikhonov regularization and the DTV method use two tuning paramters, which provides more flexibility to the problem. It has the potential to reconstruct distributions which are highly anisotropic, and we hope to further investigate this in the future.

Chapter 4

ESTIMATION OF PARAMETERS IN EXPONENTIAL FITTING PROBLEMS

A common problem that arose in the Birth-Death Process (BDP) and Nuclear Magnetic Resonance (NMR) problems is exponential fitting. In the BDP, we were trying to find the best parameters in the c.d.f. of extinction times (2.2.19), which is a sum of exponential functions. The NMR problems, on the other hand, aim at recovering the underlying distribution of relaxation times in (3.1.15). It reduces to the same exponential fitting problem as in the BDP if the spectral function $x(\xi)$ in (3.1.15) is simply a sum of delta functions. In fact, exponential fitting is used in describing many physical phenomena that can be described through differential equations, such as radioactive decay, spin relaxometry, and chemical reaction kinetics. With real datasets, these problems amount to recovering the decay constants and coefficients of exponential functions for the underlying process. This problem is challenging since inferring the exponents leads to a nonlinear least squares problem, and is potentially ill-conditioned. In view of this, we study exponential fitting in more detail in this chapter. Some classic methods of exponential fitting are introduced first. We then propose the moment constraint method for two-term exponential fitting in Section 4.3.2, which utilizes the moments of data points on top of the modified Prony method. The moment constraint method overcomes some of the difficulties that other methods may have, and it yields more stable results. The chapter ends with some numerical examples and special application to four-term exponential fitting problems.

4.1 Some Classic Methods for Exponential Fitting

Many researchers [54, 50, 84, 49] have investigated a variety of exponential fitting methods, and we discuss some of them in this chapter. Some of these classic methods are the basis of our new approach. Formally, a general exponential fitting problem with p terms estimates parameters $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$ from n noisy measurements $\boldsymbol{y}$ at regular times $t_k = k \times \delta t$ for,

$$y_k = y(t_k) = \epsilon_k + \sum_{j=1}^{p} \alpha_j e^{\omega_j(\delta t)k}, \text{ for } k = 1, \ldots, n, \tag{4.1.1}$$

where ϵ_i is the noise in each measurement. In general, the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ are complex, but they are real numbers in the BDP and NMR problems discussed in previous chapters. In the following sections, we demonstrate some popular methods for exponential fitting problems. In addition, the Variable projection method and the modified Prony method are already discussed in Section 2.3.

4.1.1 Prony's Method

Prony's method, proposed in 1795 [87], is the foundation of many other linear methods that identify the exponential coefficients using algebraic methods such as eigenvalue analysis and polynomial root finding. These methods include Prony least squares method [48], matrix pencil method [53] and Kung's method [62]. Prony's method assumes an autoregressive model of order p in which y_k depends linearly on its previous p data points

$$y_k = -\sum_{j=1}^{p} a_{p-j} y_{k-j}, \quad (k > p) \tag{4.1.2}$$

and in matrix form it can be written as

$$\begin{pmatrix} y_1 & y_2 & \cdots & y_p \\ y_2 & y_3 & \cdots & y_{p+1} \\ \vdots & \vdots & & \vdots \\ y_p & y_{p+1} & \cdots & y_{2p-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{p-1} \end{pmatrix} = - \begin{pmatrix} y_{p+1} \\ y_{p+2} \\ \vdots \\ y_{2p} \end{pmatrix}, \tag{4.1.3}$$

or

$$\mathbf{H}\boldsymbol{a} = -\boldsymbol{b} \tag{4.1.4}$$

where $\mathbf{H}$ is referred to as a Hankel matrix. Equation (4.1.2) can be expressed in another way as

$$\begin{pmatrix} y_{k-p+1} \\ y_{k-p+2} \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{p-2} & -a_{p-1} \end{pmatrix} \begin{pmatrix} y_{k-p} \\ y_{k-p+1} \\ \vdots \\ y_{k-1} \end{pmatrix}, \qquad (4.1.5)$$

or

$$\boldsymbol{b}_k = \mathbf{A}\boldsymbol{b}_{k-1}, \quad (k > p), \qquad (4.1.6)$$

where $\boldsymbol{b}_k = (y_{k-p+1}, \ldots, y_k)^T$. The matrix $\mathbf{A}$ above is a companion matrix [30]. If it has distinct eigenvalues $\{\lambda_j\}$, it is diagonalizable $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, and its characteristic polynomial is given by

$$c(t) = t^p + a_{p-1}t^{p-1} + \cdots + a_1 t + a_0. \qquad (4.1.7)$$

Then if $\boldsymbol{e}_p = (0, \ldots, 0, 1)^T$ is of size $p \times 1$, we can express y_k as

$$y_k = \boldsymbol{e}_p^T \boldsymbol{b}_k = \boldsymbol{e}_p^T \mathbf{V}\mathbf{\Lambda}^{k-p}\mathbf{V}^{-1}\boldsymbol{b}_p = \sum_{j=1}^{p} \underbrace{[\lambda_j^{-p}(\boldsymbol{e}_p^T\mathbf{V})_j(\mathbf{V}^{-1}\boldsymbol{b}_p)_j]}_{\alpha_j} \lambda_j^k, \qquad (4.1.8)$$

and according to the assumption in (4.1.1), we let

$$\lambda_j = e^{\omega_j \delta t} \quad \Rightarrow \quad \omega_j = \frac{\ln \lambda_j}{\delta t}. \qquad (4.1.9)$$

Since the coefficients $\boldsymbol{a}$ are given by (4.1.3), the exponents $\boldsymbol{w}$ can be obtained by finding the roots of characteristic polynomial (4.1.7) and using relation (4.1.9). Unfortunately, due to the high sensitivity of polynomial root-finding, Prony's method yields biased estimates if noise is present in the data, but the method has been rectified later by other authors [52, 79]. Algorithm 7 summarizes the steps for Prony's method.

Algorithm 7 Prony's method

Input: $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ at regular time nodes $t_k = k \times \delta t$.

1: Build the Hankel matrix $\mathbf{H}$ and vector $\boldsymbol{b}$ in (4.1.3), and solve $\boldsymbol{a} = \mathbf{H}^{-1}\boldsymbol{b}$.

2: Form the characteristic polynomial $c(t) = t^p + a_{p-1}t^{p-1} + \cdots + a_1 t + a_0$. Calculate its roots λ_j.

3: Calculate the exponents by $\omega_j = \dfrac{\ln \lambda_j}{\delta t}$.

Output: $\boldsymbol{\omega}$.

4.1.2 Matrix Pencil Method

The matrix pencil method [53] is a subspace-based method that considers the model

$$y_k = y(t_k) = \sum_{j=1}^{p} \alpha_j z_j^k, \tag{4.1.10}$$

where a change of variable $z_j^k = e^{\omega_j k (\delta t)}$ is used. The Hankel matrix of size $(L+1) \times (M+1)$ is given by

$$\mathbf{H} = \begin{pmatrix} y_0 & y_1 & \cdots & y_M \\ y_1 & y_2 & \cdots & y_{M+1} \\ \vdots & \vdots & & \vdots \\ y_L & y_{L+1} & \cdots & y_{N-1} \end{pmatrix}, \tag{4.1.11}$$

in which $N - M = L + 1$ is satisfied. A Vandermonde decomposition can be derived directly from (4.1.10) as

$$\mathbf{H} = \mathbf{A}\mathbf{D}\mathbf{B}^T, \tag{4.1.12}$$

where $\mathbf{A}$, $\mathbf{D}$ and $\mathbf{B}$ are Vandermonde matrices as the following

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \vdots & \vdots & & \vdots \\ z_1^L & z_2^L & \cdots & z_n^L \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \vdots & \vdots & & \vdots \\ z_1^M & z_2^M & \cdots & z_n^M \end{pmatrix}^T. \tag{4.1.13}$$

The matrix $\mathbf{A}$ is shift-invariant. If $\mathbf{Z} = \mathrm{diag}(\boldsymbol{z})$, then it is easy to show that

$$\mathbf{A}_{2:L+1,:} = \mathbf{A}_{1:L,:}\mathbf{Z}, \tag{4.1.14}$$

where $\mathbf{A}_{2:L+1,:}$ and $\mathbf{A}_{1:L,:}$ are the submatrices of $\mathbf{A}$ formed by truncating its first row and last row, respectively. On the other hand, $\mathbf{H}$ has the SVD

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{4.1.15}$$

We can see that $\mathbf{U}$ and $\mathbf{A}$ represent the same subspace with different bases, and therefore can be connected via a nonsingular matrix $\mathbf{G}$:

$$\mathbf{U} = \mathbf{A}\mathbf{G}. \tag{4.1.16}$$

Note that $\mathbf{U}$ inherits the shift-invariance from $\mathbf{A}$:

$$\mathbf{U}_{2:L+1,:} = \mathbf{A}_{2:L+1,:}\mathbf{G} = \mathbf{A}_{1:L,:}\mathbf{Z}\mathbf{G} = \mathbf{U}_{1:L,:}\mathbf{G}^{-1}\mathbf{Z}\mathbf{G}. \tag{4.1.17}$$

Given the SVD of the Hankel matrix $\mathbf{H}$, we use the pseudo-inverse to compute

$$\mathbf{G}^{-1}\mathbf{Z}\mathbf{G} = (\mathbf{U}_{1:L,:})^{\dagger}\mathbf{U}_{2:L+1,:}. \tag{4.1.18}$$

Therefore we obtain the elements in $\mathbf{Z}$ by calculating the eigenvalues of its similar matrix $\mathbf{G}^{-1}\mathbf{Z}\mathbf{G}$, and thus recovering the exponential coefficients. Similar subspace-based methods include Kung's method (also known as HSVD) [62, 3] and HTLS [44]. Algorithm 8 summarizes the steps for the matrix pencil method.

Algorithm 8 Matrix Pencil method

Input: $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ at regular time nodes $t_k = k \times \delta t$.

1: Singular value decompostion $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where the size of $\mathbf{U}$ is $(L+1) \times (M+1)$

2: Obtain $\mathbf{U}_{1:L,:}$ by removing the last row of $\mathbf{U}$, and $\mathbf{U}_{2:L+1,:}$ by removing the first row of $\mathbf{U}$

3: Compute $\{z_j\}$ as the eigenvalues of $(\mathbf{U}_{1:L,:})^{\dagger}\mathbf{U}_{2:L+1,:}$

4: Exponents are given by $\omega_j = \dfrac{\ln z_j}{\delta t}$

Output: $\boldsymbol{\omega}$.

4.1.3 Recursive Fitting

We now consider the recursive fitting method [34]. This is used in traffic measurements and communication networks that analyze the *long-tail distributions*, i.e., tails that decay more slowly than exponentials. However, it could potentially be used for exponential fitting as well.

Examples of long-tail distributions include Perato distribution and Weibull distribution. Given the c.d.f. $F(t)$, its complement (ccdf) is denoted by $F^c(t) = 1 - F(t)$. The goal of this method is to fit a mixture of exponentials H_k to the c.d.f of the long-tail distribution, accurate within a finite interval $[t_1, t_2]$ for suitably small t_1 and suitably large t_2. Suppose the ccdf of the hyperexponential distribution H_k is

$$H_k^c(t) = \sum_{i=1}^k \alpha_i e^{-\sigma_i t}, \qquad (4.1.19)$$

with $\alpha_i \geq 0$ for all i, $\sum_{i=1}^k \alpha_i = 1$, and $0 < \sigma_1 < \ldots < \sigma_k$. The main idea is to fit these exponentials to $F(t)$ recursively on different time scales. Starting with the slowest decaying pair (α_1, λ_1), if σ_2 is much larger than σ_1, then the rest of the sum $\sum_{i=2}^k e^{-\sigma_i t}$ would be negligible compared to $e^{-\sigma_1 t}$ on a sufficiently large time scale t. In this case, we could find the pair (α_1, λ_1) by single exponential fitting from the data $F^c(t)$, without considering the remaining exponential terms. We then subtract the term $\alpha_1 e^{-\sigma_1 t}$ from both $F^c(t)$ and $H_k^c(t)$, and choose the second largest time scale to recover the second pair (α_2, λ_2). This procedure is recursively applied until all exponents and coefficients are obtained.

This recursive fitting procedure will work only if σ_{i+1} is significantly larger than σ_i for all i. This only applies to ccdf $F^c(t)$ which are log-convex, and equivalently, the p.d.f. $f(t)$ must have decreasing failure rate (DFR) [34]. Furthermore, this method can also be applied to fit hyperexponential distribution to data directly. However, it has been shown that the performance is generally better if a simple long-tail distribution (with only a few parameters) is fit to the data first, and then a hyperexponential distribution is fit to that long-tail distribution from recursive fitting procedure. Algorithm 9 summarizes the steps for the recursive fitting method.

Algorithm 9 Recursive procedure for fitting a hyperexponential c.d.f. to a given long-tail c.d.f.

Input: Long-tail distribution $F(t)$, and its complement $F^c(t)$. Let $F_1^c(t) = F^c(t)$.

1: Choose the number k of exponential components of the hyperexponential distribution we want to fit.

2: Choose time scales $0 < c_k < c_{k-1} < \ldots < c_1$, with ratios c_i/c_{i+1} sufficiently large for all i.

3: Find a suitable b such that $1 < b < c_i/c_{i+1}$ for all i.

4: **for** $i = 1 : k - 1$ **do**

5: $\quad$ Choose a pair (α_i, σ_i) to match the ccdf $F_i^c(t)$ at $t = xc_i$ for $x = 1$ and b.

6: $\quad \sigma_i = \dfrac{1}{(b-1)c_i} \ln \dfrac{F_i^c(c_i)}{F_i^c(bc_i)}$

7: $\quad \alpha_i = F_i^c(c_i)e^{\sigma_i c_i}$

8: $\quad$ Subtract the fitted term from the current ccdf, $F_{i+1}^c(xc_i) = F_i^c(xc_i) - \alpha_i e^{-\sigma_i xc_i}$ for $x = 1$ and b.

9: $\quad i \leftarrow i + 1$

10: **end for**

11: $\alpha_k = 1 - \sum_{i=1}^{k-1} \alpha_i$

12: $\sigma_k = \dfrac{1}{c_k} \ln(\alpha_k/F_k^c(c_k))$

Output: The hyperexponential c.d.f. $H_k(t) = 1 - \sum_{i=1}^{k} \alpha_i e^{-\sigma_i t}$ that approximates the long-tail distribution $F(t)$

4.2 The Expectation-Maximization Algorithm

In this section, we consider the expectation-maximization (EM) algorithm, which is an iterative scheme that arises in many probablistic models. The EM algorithm will be exploited later in our new method as well. Many probablistic models are used in modeling physical and biological data. However, the available data are often incomplete. The EM algorithm is able to estimate the parameters in probablistic models with incomplete data. It is especially useful in Gaussian mixture models and clustering.

4.2.1 Coin Flip Example

We start with an illustrative example from [26] that seeks to estimate the probabilities of head for two coins A and B with unknown biases $\boldsymbol{\theta} = (\theta_A, \theta_B)$. This means that coin A will land on heads with probability θ_A, and similar for coin B.

The following procedure is repeated 12 times: (1) randomly choose one of the two coins (equally likely); (2) toss the chosen coin ten times independently, and record the number of heads. If the identity of the coin is known (the coins are marked), then we simply aggregate the number of heads in all experiments with coin A (B), and compute a maximum likelihood estimation (MLE) of θ_A (θ_B). Now consider a more challenging problem when the identity of coins are masked during the whole experiment, and the only quantities known are the number of heads for each independent coin tosses.

In order to estimate the model parameter, the EM algorithm computes the probabilities for each possible completion of the missing data, i.e., the identities of each chosen coin, using the current parameter $\boldsymbol{\theta}^{(n)}$ after n iterations. These probabilities are used to create a weighted training set, and maximum likelihood estimation is applied on the weighted training data that yields the next parameter $\boldsymbol{\theta}^{(n+1)}$.

In summary, the EM algorithm alternates between the following two steps iteratively until convergence. (1) E-step: Guess a probability distribution over the completions of missing data, given current parameters; (2) M-step: Reestimate the model parameters by maximizing the expected log-likelihood over these completions.

We implemented the coin toss problem with the EM algorithm, started with initial biases $\boldsymbol{\theta} = (0.6, 0.5)$. In each iteration, the E-step calculates the probabilities of coin identities in 12 trials using Bayes' rule, and assigns weighted heads and tails to both coins according to the experiment results. In particular, suppose that the current bias is (θ_A, θ_B), and h heads out of 10 tosses have been observed. Then the probability

that coin A is chosen is given by Bayes' rule as

$$p_A = \mathbb{P}(A|H = h) = \frac{\mathbb{P}(H = h|A)/2}{\mathbb{P}(H = h|A)/2 + \mathbb{P}(H = h|B)/2} \tag{4.2.1}$$

$$= \frac{\binom{10}{h}\theta_A^h(1 - \theta_A)^{10-h}}{\binom{10}{h}\theta_A^h(1 - \theta_A)^{10-h} + \binom{10}{h}\theta_B^h(1 - \theta_B)^{10-h}}$$

$$= \frac{\theta_A^h(1 - \theta_A)^{10-h}}{\theta_A^h(1 - \theta_A)^{10-h} + \theta_B^h(1 - \theta_B)^{10-h}}. \tag{4.2.2}$$

We then accumulate all weighted heads and tails for both coins and use maximum likelihood to derive the biases of them. Table 4.1 contains both experiments results of 12 repetitions, and model computations of EM algorithm after one iteration.

4.2.2   Formal Definition

We now present a more formal definition of the EM algorithm given by Dempster [24]. It is an approach to iterative computation of maximum likelihood estimates when the observations are regarded as incomplete data. Consider two sample spaces $\mathcal{X}$ and $\mathcal{Y}$, and a many-to-one mapping $\mathcal{X} \to \mathcal{Y}$ between them. The observed "incomplete" data is $\boldsymbol{y} \in \mathcal{Y}$, and the associated "complete" data $\boldsymbol{x} \in \mathcal{X}$ is not observed directly. More specifically, $\boldsymbol{x}$ only lies in $\mathcal{X}(\boldsymbol{y})$, the subset of $\mathcal{X}$ determined by the equation $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$.

There are two types of densities to consider. First we have the complete-date sampling densities $f(\boldsymbol{x}|\boldsymbol{\phi})$, which depends on parameter $\boldsymbol{\phi}$. Their corresponding incomplete-date sampling densities are denoted $g(\boldsymbol{y}|\boldsymbol{\phi})$, and are related by

$$g(\boldsymbol{y}|\boldsymbol{\phi}) = \int_{\mathcal{X}(\boldsymbol{y})} f(\boldsymbol{x}|\boldsymbol{\phi}) \, d\boldsymbol{x}. \tag{4.2.3}$$

Notice that many $f(\boldsymbol{x}|\boldsymbol{\phi})$ could generate the same $g(\boldsymbol{y}|\boldsymbol{\phi})$. The EM algorithm finds a value $\boldsymbol{\phi}$ that maximizes $g(\boldsymbol{y}|\boldsymbol{\phi})$ given the observation $\boldsymbol{y}$. However, it is actually related to the complete-data specification $f(\boldsymbol{x}|\boldsymbol{\phi})$. We consider the simplest and most useful case where the complete-data density $f(\boldsymbol{x}|\boldsymbol{\phi})$ is in the regular exponential family:

$$f(\boldsymbol{x}|\boldsymbol{\phi}) = b(\boldsymbol{x}) \exp(\boldsymbol{\phi}\boldsymbol{t}(\boldsymbol{x})^T)/a(\boldsymbol{\phi}), \tag{4.2.4}$$

Experimental Results			Model Computations				
				Coin A		Coin B	
Trial	H	T	p_A	H	T	H	T
1	9	1	0.80	7.20	0.80	1.80	0.20
2	4	6	0.35	1.40	2.10	2.60	3.90
3	7	3	0.65	4.55	1.95	2.45	1.05
4	5	5	0.45	2.25	2.25	2.75	2.75
5	8	2	0.73	5.84	1.46	2.16	0.54
6	3	7	0.27	0.81	1.89	2.19	5.11
7	1	9	0.14	0.14	1.26	0.86	7.74
8	4	6	0.35	1.40	2.10	2.60	3.90
9	4	6	0.35	1.40	2.10	2.60	3.90
10	5	5	0.45	2.25	2.25	2.75	2.75
11	9	1	0.80	7.20	0.80	1.80	0.20
12	6	4	0.55	3.30	2.20	2.70	1.80
Subtotal	-	-	-	37.74	21.16	27.26	33.84
$\boldsymbol{\theta}^{(1)}$	-	-	-	0.64		0.45	

Table 4.1: **Left of bold vertical line:** In each trial, one of the two coins A and B is randomly chosen (and remains anonymous) and tossed ten times in a row. This procedure is repeated 12 times and number of heads (H) and tails (T) are recorded. **Right of bold vertical line:** Result of coin toss experiment after one iteration in EM algorithm, i.e., after one E-step and one M-step. Coin biases are initially set to $\boldsymbol{\theta}^{(0)} = (0.6, 0.5)$. For each trial, we calculate the probability that the chosen coin is A with Bayes' rule given current parameter $\boldsymbol{\theta}^{(0)}$, and denoted this probability p_A. This probability is then multiplied into the total heads and tails in that trial, so that we get weighted heads and tails for both coins A and B. At the end of this iteration, a cumulative count of heads and tails are calculated for both coins, and MLE is used to get the next parameter $\boldsymbol{\theta}^{(1)} = (0.64, 0.45)$. If the EM iterations proceed until convergence, the final parameter we get is $\boldsymbol{\theta}^{(n)} = (0.76, 0.39)$, which is close to the true biases $(0.4, 0.8)$ up to permutations. It is important to note that the EM does not distinguish the biases between A and B.

where $\boldsymbol{\phi}$ is a $1 \times r$ vector parameter in a convex set Ω, $\boldsymbol{t}(\boldsymbol{x})$ denotes a $1 \times r$ vector of complete-data sufficient statistics [1] Suppose that $\boldsymbol{\phi}^{(p)}$ is the current value of $\boldsymbol{\phi}$ after p iterations, the EM algorithm is presented in Algorithm 10 for exponential family distributions.

Algorithm 10 Expectation-Maximization algorithm

Input: An initial guess of parameter $\boldsymbol{\phi}^{(0)}$, and a tolerance ϵ.
 1: **while** $\|\boldsymbol{\phi}^{(n+1)} - \boldsymbol{\phi}^{(n)}\| > \epsilon$ **do**
 2:     E-step: Estimate the complete-data sufficient statistics $\boldsymbol{t}(\boldsymbol{x})$:

$$\boldsymbol{t}^{(n)} = \mathbb{E}[\boldsymbol{t}(\boldsymbol{x})|\boldsymbol{y}, \boldsymbol{\phi}^{(n)}]. \tag{4.2.5}$$

 3:     M-step: Use maximum likelihood to determine $\boldsymbol{\phi}^{(n+1)}$ as the solution to

$$\mathbb{E}[\boldsymbol{t}(\boldsymbol{x})|\boldsymbol{\phi}] = \boldsymbol{t}^{(n)}. \tag{4.2.6}$$

 4: **end while**
Output: The parameter $\boldsymbol{\phi}^{(n+1)}$ after convergence.

Recall that the goal of EM algorithm is to maximize the incomplete-data density $g(\boldsymbol{y}|\boldsymbol{\phi})$ over all possible parameters $\boldsymbol{\phi} \in \Omega$. We now explain why reiteration of EM steps eventually leads to the best parameter $\boldsymbol{\phi}^*$ of $\boldsymbol{\phi}$.

Under the exponential-family assumption, the problem is equivalent to maximizing its log-likelihood function

$$L(\boldsymbol{\phi}) = \log g(\boldsymbol{y}|\boldsymbol{\phi}). \tag{4.2.7}$$

Let k represent the conditional density of $\boldsymbol{x}$ given data $\boldsymbol{y}$ and parameter $\boldsymbol{\phi}$, and notice that f is independent of the data $\boldsymbol{y}$. Then

$$k(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\phi}) = \frac{f(\boldsymbol{x}|\boldsymbol{\phi})}{g(\boldsymbol{y}|\boldsymbol{\phi})}, \tag{4.2.8}$$

---

[1] A statistic $t = T(X)$ is *sufficient* for an underlying parameter θ if the conditional probability distribution of data X, given the statistics $t = T(X)$, does not depend on θ [16], i.e. $\mathbb{P}(x|t, \theta) = \mathbb{P}(x|t)$. For example, the sample mean is sufficient for the mean μ of a normal distribution with known variance, since the data and μ are independent once sample mean has been calculated. In contrast, the sample median is not sufficient for the mean, because the sample data X can provide extra information about the population mean.

so that (4.2.7) can be written as

$$L(\boldsymbol{\phi}) = \log f(\boldsymbol{x}|\boldsymbol{\phi}) - \log k(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\phi}). \tag{4.2.9}$$

Since the term $a(\boldsymbol{\phi})$ is a normalization term, it is automatically determined once other terms are known. For exponential families, we have

$$k(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\phi}) = b(\boldsymbol{x}) \exp(\boldsymbol{\phi}\boldsymbol{t}(\boldsymbol{x})^T)/a(\boldsymbol{\phi}|\boldsymbol{y}), \tag{4.2.10}$$

where

$$a(\boldsymbol{\phi}|\boldsymbol{y}) = \int_{\mathcal{X}(\boldsymbol{y})} b(\boldsymbol{x}) \exp(\boldsymbol{\phi}\boldsymbol{t}(\boldsymbol{x})^T) \, d\boldsymbol{x}. \tag{4.2.11}$$

The analogous expression for $a(\boldsymbol{\phi})$ in (4.2.4) is

$$a(\boldsymbol{\phi}) = \int_{\mathcal{X}} b(\boldsymbol{x}) \exp(\boldsymbol{\phi}\boldsymbol{t}(\boldsymbol{x})^T) \, d\boldsymbol{x}, \tag{4.2.12}$$

with the only difference being in the sample spaces. Then, the likelihood function (4.2.7) simplifies to

$$L(\boldsymbol{\phi}) = -\log a(\boldsymbol{\phi}) + \log a(\boldsymbol{\phi}|\boldsymbol{y}). \tag{4.2.13}$$

Differentiating (4.2.12) and denoting $\boldsymbol{t}(\boldsymbol{x})$ by $\boldsymbol{t}$, we get

$$\frac{\partial}{\partial \boldsymbol{\phi}} \log a(\boldsymbol{\phi}) = \frac{1}{a(\boldsymbol{\phi})} \int_{\mathcal{X}} \boldsymbol{t}(\boldsymbol{x})^T b(\boldsymbol{x}) \exp(\boldsymbol{\phi}\boldsymbol{t}(\boldsymbol{x})^T) \, d\boldsymbol{x} = \int_{\mathcal{X}} \boldsymbol{t}^T f(\boldsymbol{x}|\boldsymbol{\phi}) \, d\boldsymbol{x} = \mathbb{E}[\boldsymbol{t}|\boldsymbol{\phi}]. \tag{4.2.14}$$

Similarly, differentiating (4.2.11) results in

$$\frac{\partial}{\partial \boldsymbol{\phi}} \log a(\boldsymbol{\phi}|\boldsymbol{y}) = \mathbb{E}[\boldsymbol{t}|\boldsymbol{y}, \boldsymbol{\phi}]. \tag{4.2.15}$$

Therefore, the derivative to the log-likelihood (4.2.7) can be written as the difference of an unconditional and a conditional expectation of the sufficient statistics

$$\frac{\partial}{\partial \boldsymbol{\phi}} L(\boldsymbol{\phi}) = -\mathbb{E}[\boldsymbol{t}|\boldsymbol{\phi}] + \mathbb{E}[\boldsymbol{t}|\boldsymbol{y}, \boldsymbol{\phi}]. \tag{4.2.16}$$

If the EM algorithm converges to $\boldsymbol{\phi}^*$, then $\boldsymbol{\phi}^{(n)} = \boldsymbol{\phi}^{(n+1)} = \boldsymbol{\phi}^*$ in the limit as $n \to \infty$. Combining the E-step (4.2.5) and M-step (4.2.6) yields $\mathbb{E}[\boldsymbol{t}|\boldsymbol{\phi}^*] = \mathbb{E}[\boldsymbol{t}|\boldsymbol{y}, \boldsymbol{\phi}^*]$, or $\frac{\partial}{\partial \boldsymbol{\phi}} L(\boldsymbol{\phi}) = 0$ at $\boldsymbol{\phi} = \boldsymbol{\phi}^*$. Thus the repeated application of EM algorithm leads to maximization of the log-likelihood $L(\boldsymbol{\phi})$. Dempster also extends this proof to a less-restricted case when $f(\boldsymbol{x}|\boldsymbol{\phi})$ is not in the exponential family, and later removes all reference to exponential families. We will skip these cases here.

4.3 Modified Osborne's Method with Moment Constraints

We now present a new method for estimating the parameters $\{\alpha_k, \sigma_k\}$, $1 \leq k \leq N$ in a hyperexponential *probability* distribution, whose survival probability function (i.e., complement of the cumulative distribution function) can be represented by

$$S(t) = \sum_{k=1}^{N} \alpha_k e^{-\sigma_k t}, \qquad (4.3.1)$$

from M samples of the distribution. It is important to notice that these are samples for survival times, instead of direct samples of S. For this reason, this exponential fitting problem may also be seen as a kernel density estimation problem. The coefficients α_k and the exponents σ_k satisfy $\sum_{k=1}^{N} \alpha_k = 1$, $\alpha_k > 0$, and $\sigma_k > 0$. The problem above is one typical example of mixture density estimation problems, which have been studied since the 1980s [39]. The predominant methods [88] to tackle these problems include the method of moments [82], the method of maximum likelihood [106], and the expectation-maximization (EM) algorithm [24, 26].

Our technique implements a modified version of Osborne's least-squares method [81] but uses the sample moments from the data to improve stability, along with ideas from the EM algorithm and Tikhonov regularization. One main difficulty with estimating σ_k arises when two of the σ_k are of similar size, and the sample moments help to alleviate this problem to a certain extent.

4.3.1 Osborne's Method for $N = 2$

We now review Osborne's method for exponential fitting for $N = 2$. The data $\{y_1, \ldots, y_M\}$ is sampled from the survival probability function $S(t) \equiv 1 - W(t)$ at random times, where $W(t)$ is the corresponding cumulative distribution function (c.d.f.). Then we construct the numerical survival function $\boldsymbol{S}$ from the data on the interval $(0, 1]$ with n regular time nodes. Let $w(t) = W'(t)$ be the probability density function (p.d.f.) and assume that it is the solution to the differential equation

$$\mathcal{L}w(t) = 0, \qquad (4.3.2)$$

with

$$\mathcal{L} = D^2 + \xi_2 D + \xi_1, \tag{4.3.3}$$

for some constants ξ_1 and ξ_2, and $D \equiv \dfrac{d}{dt}$ is the differential operator. Let σ_1 and σ_2 be the roots of the characteristic equation. Then $\xi_2 = -(\sigma_1 + \sigma_2)$ and $\xi_1 = \sigma_1 \sigma_2$.

The modified Prony algorithm [81], instead of using the operator $\mathcal{L}$, works with discretized data and discretized coefficients (γ_1, γ_2) corresponding to the following difference equation:

$$(\Delta^2 + \gamma_2 \Delta + \gamma_1)\, w(t) = (\Delta - \zeta_1)(\Delta - \zeta_2)\, w(t) = 0. \tag{4.3.4}$$

It is well-known that the original Prony method yields biased estimates for the exponential fitting problem and is very sensitive to noise. Unfortunately, the modified Prony method, unlike its description in [81], is also very sensitive to the initial iteration point when there is noise in the data. The modified Prony method is described in Section (2.3.3), and the main goal is to minimize the variable projection functional:

$$\psi(\boldsymbol{\gamma}) = \boldsymbol{S}^T P_{\mathbf{X}} \boldsymbol{S} = \boldsymbol{S}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{S}, \tag{4.3.5}$$

where $\boldsymbol{S}$ is the vector of discretized survival function values on $(0, 1]$, and $\mathbf{X} = \mathbf{X}(\boldsymbol{\gamma})$ is a matrix defined as in eq (2.3.20). Specifically for $p = 2$, $\mathbf{X}$ is an $n \times (n-2)$ matrix

$$\mathbf{X}(\boldsymbol{\gamma}) = \begin{bmatrix} c_1 & & & \\ c_2 & c_1 & & \\ c_3 & c_2 & \ddots & \\ & c_3 & \ddots & c_1 \\ & & \ddots & c_2 \\ & & & c_3 \end{bmatrix}, \tag{4.3.6}$$

where each column of the above matrix is

$$\boldsymbol{c} = \begin{bmatrix} 1 & -1 & 1 \\ & 1 & -2 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & n & \\ & & n^2 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}, \tag{4.3.7}$$

with $\gamma_3 = 1$.

114

4.3.2   Moment Constraint Method

This is our main contribution to the exponential fitting problem. The Laplace transform $\tilde{w}(u)$ of the p.d.f. satisfies

$$(u^2 + \xi_2 u + \xi_1)\,\tilde{w}(u) = (u + \xi_2)\,w(0) + w'(0). \tag{4.3.8}$$

Let T_k be the k-th moment of the data $W(t)$, then

$$T_k = \int_0^\infty t^k w(t)\,dt. \tag{4.3.9}$$

Consider the Laplace transform of $w(t)$, if we differentiate it k times,

$$\tilde{w}^{(k)}(u) = \frac{d^k}{du^k}\int_0^\infty e^{-ut} w(t)\,dt = (-1)^k \int_0^\infty t^k e^{-ut} w(t)\,dt, \tag{4.3.10}$$

and therefore

$$\tilde{w}^{(k)}(0) = (-1)^k\, T_k. \tag{4.3.11}$$

We thus get constraints by differentiating eq. (4.3.8),

$$-\xi_0 - T_1\xi_1 + \xi_2 = 0, \tag{4.3.12}$$

$$T_2\xi_1 - 2T_1\xi_2 = -2, \tag{4.3.13}$$

where $\xi_0 = \alpha_1\sigma_1 + \alpha_2\sigma_2$.

In order to improve the result using nonlinear least squares methods, we propose to add a certain number of constraints to the optimization. The roots of the characteristic equation of (4.3.4) are denoted $\{\zeta_k\}$, and they are related to $\{\sigma_k\}$ by

$$\zeta_k = n(1 - e^{-\sigma_k/n}) = \sigma_k - \frac{\sigma_k^2}{2n} + \frac{\sigma_k^3}{6n^2} + O(1/n^3), \quad k = 1, 2 \tag{4.3.14}$$

Then,

$$\begin{aligned}
\gamma_1 &= \zeta_1\zeta_2 \\
&= n^2(1 - e^{-\sigma_1/n})(1 - e^{-\sigma_2/n}), \\
&= \sigma_1\sigma_2\left(1 - \frac{\sigma_1 + \sigma_2}{2n} + \frac{\sigma_1^2 + \sigma_2^2}{6n^2}\right) + O(1/n^3), \\
&= \xi_1 - \frac{\xi_1\xi_2}{2n} + \frac{\xi_1\xi_2^2 - 2\xi_1^2}{6n^2} + O(1/n^3).
\end{aligned} \tag{4.3.15}$$

Similarly,

$$
\begin{aligned}
\gamma_2 &= \zeta_1 + \zeta_2 \\
&= n(1 - e^{-\sigma_1/n}) + n(1 - e^{-\sigma_2/n}), \\
&= \sigma_1 + \sigma_2 - \frac{\sigma_1^2 + \sigma_2^2}{2n} + \frac{\sigma_1^3 + \sigma_2^3}{6n^2} + O(1/n^3), \\
&= \xi_2 - \frac{\xi_2^2 - 2\xi_1}{2n} + \frac{\xi_2^3 - 3\,\xi_1\xi_2}{6n^2} + O(1/n^3).
\end{aligned}
\tag{4.3.16}
$$

Instead of minimizing with respect to (γ_1, γ_2) in the objective function ψ in (4.3.5), we directly minimize with respect to the parameters (ξ_1, ξ_2) in (4.3.2) by defining a new objective function

$$
\tilde{\psi}(\xi_1, \xi_2) = \psi(\gamma_1, \gamma_2) \approx \psi\left(\xi_1 - \frac{\xi_1\xi_2}{2n}, \ \xi_2 - \frac{\xi_2^2 - 2\xi_1}{2n}\right)
\tag{4.3.17}
$$

This comes from the second order relations in (4.3.15, 4.3.16), and is accurate to order $\mathcal{O}(1/n^2)$, where n is the size of $\boldsymbol{S}$ in (4.3.5). Consider the constraints matrix for the two exponential problem by rewriting (4.3.12)(4.3.13),

$$
\begin{bmatrix} -1 & -T_1 & 1 \\ 0 & T_2 & -2T_1 \end{bmatrix}
\begin{bmatrix} \xi_0 \\ \xi_1 \\ \xi_2 \end{bmatrix}
=
\begin{bmatrix} 0 \\ -2 \end{bmatrix}
\tag{4.3.18}
$$

or $C\boldsymbol{\xi} = \boldsymbol{b}$. This is an underdetermined linear system, so we may find a solution (with smallest 2-norm) such that $C\boldsymbol{x}_0 = \boldsymbol{b}$. Suppose the singular value decomposition of C is $C = USV^*$, then the last column $\boldsymbol{v}_1$ of V will be the basis for the nullspace of C. Then $\boldsymbol{\xi} = \boldsymbol{x}_0 + c\boldsymbol{v}_1$ is a solution to (4.3.18). We can then find the coefficient c such that $\tilde{\psi}$ is minimized. We ignore ξ_0 and the true solution is just the last two elements in $\boldsymbol{\xi}$.

When we search for the coefficient c, having the gradient improves the convergence. By equation (4.3.17), we have

$$
\frac{\partial \tilde{\psi}}{\partial c} = \frac{\partial \boldsymbol{\xi}}{\partial c} \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{\xi}} \frac{\partial \psi}{\partial \boldsymbol{\gamma}},
\tag{4.3.19}
$$

where

$$\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} 0 & 0 \\ 1 - \dfrac{\xi_2}{2n} & -\dfrac{\xi_1}{n} \\ -\dfrac{\xi_1}{2n} & 1 - \dfrac{\xi_2}{n} \end{bmatrix}, \qquad (4.3.20)$$

$$\frac{\partial \boldsymbol{\xi}}{\partial c} = \boldsymbol{v}_1^T. \qquad (4.3.21)$$

where n is the number of time nodes used in $(0,1]$, and the last derivative term in the product $\frac{\partial \tilde{\psi}}{\partial \boldsymbol{\gamma}}$ was derived in Osborne's paper and is given by (2.3.25).

In summary, our method for estimating σ_1 and σ_2 essentially finds (ξ_1, ξ_2) that minimizes Osborne's objective function (4.3.17) subject to the linear constraints (4.3.18). T_1 and T_2 are the first and second moments of the p.d.f. which are estimated from the sample data. Once (ξ_1, ξ_2) are found, σ_1, σ_2 are calculated as the roots of characteristic polynomial with coefficients (ξ_1, ξ_2) and α_1, α_2 are obtained from least squares:

$$\boldsymbol{\alpha} = A(\boldsymbol{\sigma})^{\dagger} \boldsymbol{S}, \qquad (4.3.22)$$

where $A_{ij} = e^{-\sigma_j t_i}$.

Furthermore, in order to improve performance when the sizes of exponents are very different, we can sample the data on different time scales. All the calculations above remain unchanged and the variable projection functional (4.3.17) is also unchanged. Define $t' = t/\lambda \in (0, 1/\lambda]$, where $\lambda \ll 1$ and the survival function (4.3.1) becomes

$$S(t') = \alpha_1 e^{-\sigma_1' t'} + \alpha_2 e^{-\sigma_2' t'}, \qquad (4.3.23)$$

where

$$\sigma_k' = \lambda \sigma_k. \qquad (4.3.24)$$

Then

$$\xi_1' = \sigma_1' \sigma_2' = \lambda^2 \xi_1 \qquad (4.3.25)$$

$$\xi_2' = \sigma_1' + \sigma_2' = \lambda \xi_2 \qquad (4.3.26)$$

Specifically, we have the following relations

$$\gamma_1' = \xi_1' - \frac{\xi_1'\xi_2'}{2n\lambda} + \cdots = \lambda^2\left(\xi_1 - \frac{\xi_1\xi_2}{2n} + \cdots\right) = \lambda^2\gamma_1 \qquad (4.3.27)$$

$$\gamma_2' = \xi_2' - \frac{\xi_2'^2 - 2\xi_1'}{2n\lambda} + \cdots = \lambda\left(\xi_2 - \frac{\xi_2^2 - 2\xi_1}{2n} + \cdots\right) = \lambda\gamma_2 \qquad (4.3.28)$$

$$(4.3.29)$$

Therefore, the variable projection functional is

$$\tilde{\psi}(\xi_1', \xi_2') = \psi(\gamma_1', \gamma_2') = \psi(\lambda^2\gamma_1, \lambda\gamma_2). \qquad (4.3.30)$$

Now the matrix from (2.3.20) is

$$\boldsymbol{c}' = \begin{bmatrix} 1 & -1 & 1 \\ & 1 & -2 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & n\lambda & \\ & & (n\lambda)^2 \end{bmatrix} \begin{bmatrix} \gamma_1' \\ \gamma_2' \\ \gamma_3' \end{bmatrix} = \lambda^2\boldsymbol{c}, \qquad (4.3.31)$$

and since $X' = \lambda^2 X$, we have

$$\mathbf{X}'(\mathbf{X}'^T\mathbf{X})^{-1}\mathbf{X}'^T = \lambda^2\mathbf{X}(\lambda^4\mathbf{X}^T\mathbf{X})^{-1}\lambda^2\mathbf{X}^T = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T. \qquad (4.3.32)$$

Hence the variable projection functional is unchanged under a different sampling time scale

$$\tilde{\psi}(\xi_1', \xi_2') = \tilde{\psi}(\xi_1, \xi_2). \qquad (4.3.33)$$

Note that, however, the constraint matrix is different because the new moments satisfy $T_k' = T_k/\lambda^k$. The new constraints are given by

$$\begin{bmatrix} -1 & -T_1' & 1 \\ 0 & T_2' & -2T_1' \end{bmatrix} \begin{bmatrix} \xi_0' \\ \xi_1' \\ \xi_2' \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \qquad (4.3.34)$$

The moment constraint method is summarized in Algorithm 11.

Algorithm 11 Moment constraint method for $N = 2$

Input: Random samples $\boldsymbol{y} = \{y_1, \ldots, y_M\}$ from the target survival function $S(t) = \sum_{k=1}^{2} \alpha_k e^{-\sigma_k t}$ with unknown parameters.

1: Choose a suitable value of λ either using Tikhonov regularization to determine the approximate position of exponents, or using prior knowledge.

2: Generate the numerical survival function $\boldsymbol{S}$ on the interval $(0, 1/\lambda]$ from data $\boldsymbol{y}$.

3: Form the constraint matrix C as in (4.3.34) using moments of data $\boldsymbol{y}$, such that $C\boldsymbol{\xi} = \boldsymbol{b}$.

4: Find a solution $\boldsymbol{x}_0$ of least 2-norm such that $C\boldsymbol{x}_0 = \boldsymbol{b}$. Then $\boldsymbol{\xi} = \boldsymbol{x}_0 + c\boldsymbol{v}_1$ is a solution to (4.3.34) for any c.

5: Search for the constant c^* that minimizes variable projection functional $\tilde{\psi}$ as in (4.3.17) using minimization techniques.

6: $\boldsymbol{\xi} = \boldsymbol{x}_0 + c^*\boldsymbol{v}_1$.

7: Find roots of characteristic polynomial $D^2 + \xi_2 D + \xi_1 = 0$, denoted by σ_1 and σ_2.

8: Find linear coefficients by least squares $\boldsymbol{\alpha} = A(\boldsymbol{\sigma})^\dagger \boldsymbol{S}$, with $A_{ij} = e^{-\sigma_j t_i}$.

Output: Exponents $\boldsymbol{\sigma}$ and linear coefficients $\boldsymbol{\alpha}$ in the target survival function $S(t)$.

4.3.3 Numerical Results

In all the results for this section, we compare our method to four other methods. For illustrative purposes, we list our method as method (1), and index the other methods as the following: (2) Maximum Likelihood Estimation with moments, (3) Matlab's `fit` function with the option `exp2`, (4) Maximum Likelihood Estimation without moments, and (5) Osborne's Method (which does not use sample moments).

(1) This is the Moment Constraint Method proposed in this Chapter.

(2) *MLE with moment constraints.* We can write the log-likelihood function for (4.3.1) as

$$\ln \mathcal{L}(c_1, c_2, \sigma_1, \sigma_2; t_1, \ldots, t_n) = \sum_{i=1}^{n} \ln(c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}) \qquad (4.3.35)$$

subject to the constraint

$$\frac{c_1}{\sigma_1} + \frac{c_2}{\sigma_2} = 1, \tag{4.3.36}$$

because $c_i = \alpha_i \sigma_i$ for $i = 1, 2$. We include the first two moment constraints to the constraint problem, so that the log-likelihood function becomes

$$G(c_1, c_2, \sigma_1, \sigma_2, \mu_0, \mu_1, \mu_2; t_1, \ldots, t_n) = \sum_{i=1}^{M} \ln(c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i})$$

$$+ \mu_0 \left(\frac{c_1}{\sigma_1} + \frac{c_2}{\sigma_2} - 1 \right) + \mu_1 \left(\frac{c_1}{\sigma_1^2} + \frac{c_2}{\sigma_2^2} - T_1 \right) + \mu_2 \left(\frac{2c_1}{\sigma_1^3} + \frac{2c_2}{\sigma_3^3} - T_2 \right),$$
$$\tag{4.3.37}$$

where T_k is the k-th moment of the data $\{t_i\}_{i=1}^{M}$.

(3) *Matlab fit with 'exp2' option.* Directly fit a two-exponents model to data using Matlab's built-in fit function. In particular, we take the samples from $S(t) = 1 - W(t)$ on the interval $(0, \lambda]$, and Matlab will yield all unknown parameters.

(4) *MLE with no moment constraints.* Similar as method (2), we optimize the log-likelihood function:

$$G(c_1, c_2, \sigma_1, \sigma_2, \mu_0; t_1, \ldots, t_n) = \sum_{i=1}^{M} \ln(c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}) + \mu_0 \left(\frac{c_1}{\sigma_1} + \frac{c_2}{\sigma_2} - 1 \right). \tag{4.3.38}$$

Then the maximum is obtained when the following equations are satisfied:

$$\sum_{i=1}^{M} \frac{e^{-\sigma_1 t_i}}{c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}} - \frac{\mu_0}{\sigma_1} = 0 \tag{4.3.39}$$

$$\sum_{i=1}^{M} \frac{e^{-\sigma_2 t_i}}{c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}} - \frac{\mu_0}{\sigma_2} = 0 \tag{4.3.40}$$

$$c_1 \sum_{i=1}^{M} \frac{t_i e^{-\sigma_1 t_i}}{c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}} - \frac{\mu_0 c_1}{\sigma_1^2} = 0 \tag{4.3.41}$$

$$c_2 \sum_{i=1}^{M} \frac{t_i e^{-\sigma_2 t_i}}{c_1 e^{-\sigma_1 t_i} + c_2 e^{-\sigma_2 t_i}} - \frac{\mu_0 c_2}{\sigma_2^2} = 0 \tag{4.3.42}$$

$$\frac{c_1}{\sigma_1} + \frac{c_2}{\sigma_2} - 1 = 0 \tag{4.3.43}$$

(5) *Osborne's method (without moments).* Direct minimization of the variable projection functional

$$\psi(\boldsymbol{\gamma}) = \boldsymbol{S}^T P_{\mathbf{X}} \boldsymbol{S} = \boldsymbol{S}^T \mathbf{X}^{-1} (\mathbf{X}^T \mathbf{X}) \mathbf{X}^T \boldsymbol{S} \tag{4.3.44}$$

with respect to the coefficients $\{\gamma_1, \gamma_2\}$, where P_X is the orthogonal projection onto the common column space of $\mathbf{X}$ and $\mathbf{X}_\delta$.

Two-term exponential fitting

In this example, data is generated by the survival function $S(t) = 0.3e^{-\sigma_1 t} + 0.7e^{-\sigma_2 t}$, where σ_1 is fixed to be 1, and σ_2 taking different values in log-space between $10^{0.1}$ to $10^{1.5}$. In specific, 500 sample times are generated by choosing two exponential distributions according to their occurrence probability (0.3 and 0.7 respectively), and a c.d.f. is computed from these samples. Here, the moment constraint method is implemented on a fixed time scale $(0, 1]$.

Our moment constraint method is compared against four other methods on each of the σ_2 used, with a random initial guess of $\boldsymbol{\xi}$ for each value of σ_2, but the same guess was used across all five methods. The comparison results are displayed in Figures 4.1, 4.2, 4.3, 4.4. Since the sample data are noisy, the relative error (defined as absolute error rescaled by the norm of exponents) is also noisy. For better interpretation, the 30-term moving average is taken on the errors to smooth out these curves.

From these graphs, we can see that method 1 in magenta curves yields better result than the other four methods in almost all situations. The performance of method 1 is even better when σ_1 is very close to σ_2. On the other hand, all methods have similar errors when the ratio σ_2/σ_1 is relatively large. MLE methods often result in outliers despite inclusion of moments, whereas the moment constraint method is more stable in this sense. In addition, including moments constraint generally improves the results.

4.3.4 Scale Detection and Four-term Exponential Fitting

Our method detects a widely disparate σ_k by first solving a Tikhonov-regulaized least squares problem [86]. We recast (4.3.1) as a first-kind integral equation

$$\int_{-\infty}^{0} e^{st}\alpha(s)ds = 1 - W(t) \equiv S(t). \tag{4.3.45}$$

In other words, we are trying to find the inverse Laplace Transform of the survival function $S(t)$. This is well-known to be an ill-posed problem [31]. The integral equation
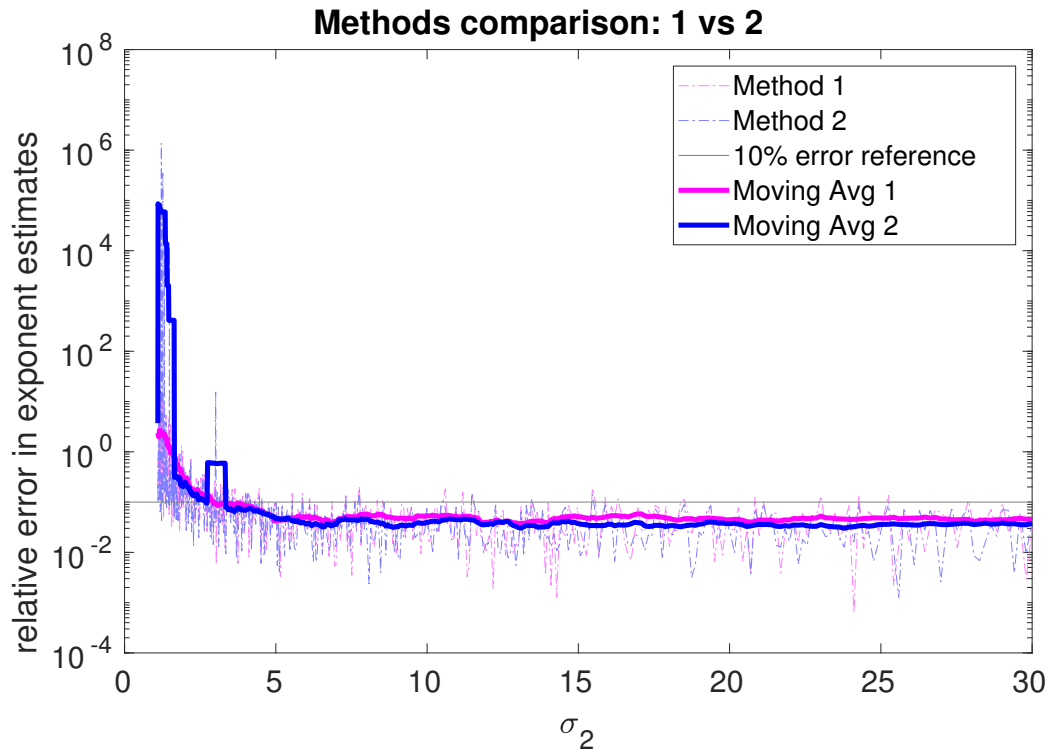
Figure 4.1: (1) Moment constraint method (magenta) compared with (2) MLE with moments (blue). Dashed lines are actual relative errors in exponent estimate, and solid lines are the 30-term moving average of the corresponding error curves. The horizontal black line is the 10% error for reference.
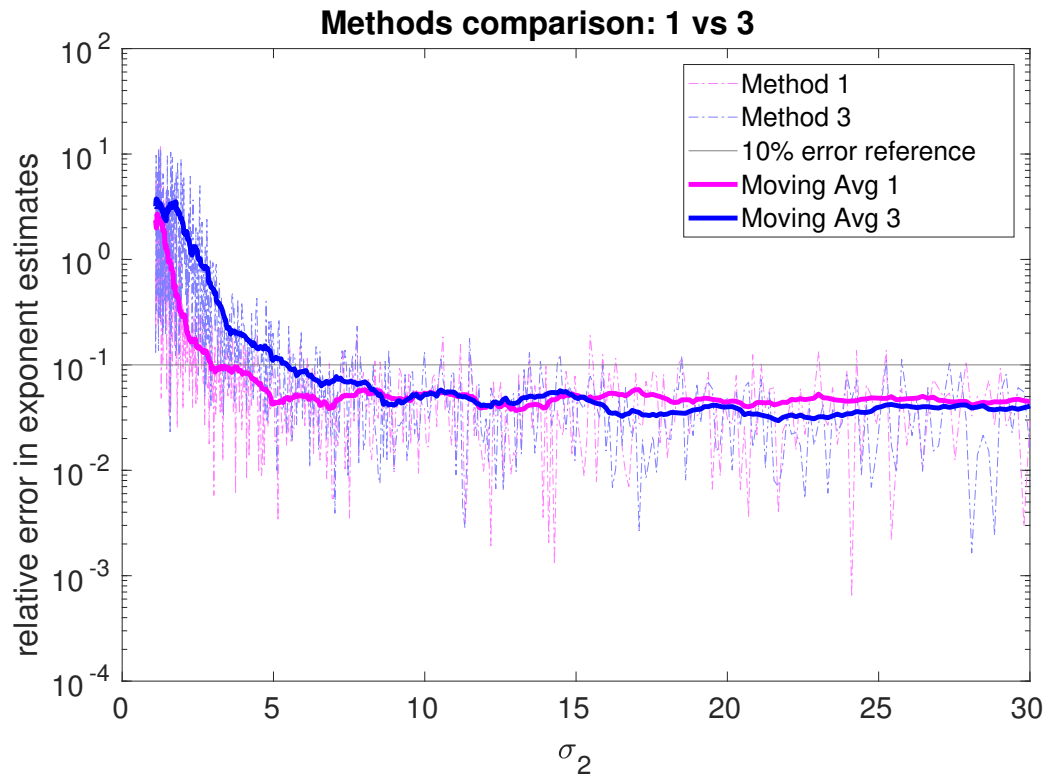
Figure 4.2: (1) Moment constraint method (magenta) compared with (3) Matlab 'exp2' fit (blue). Dashed lines are actual relative errors in exponent estimate, and solid lines are the 30-term moving average of the corresponding error curves. The horizontal black line is the 10% error for reference.
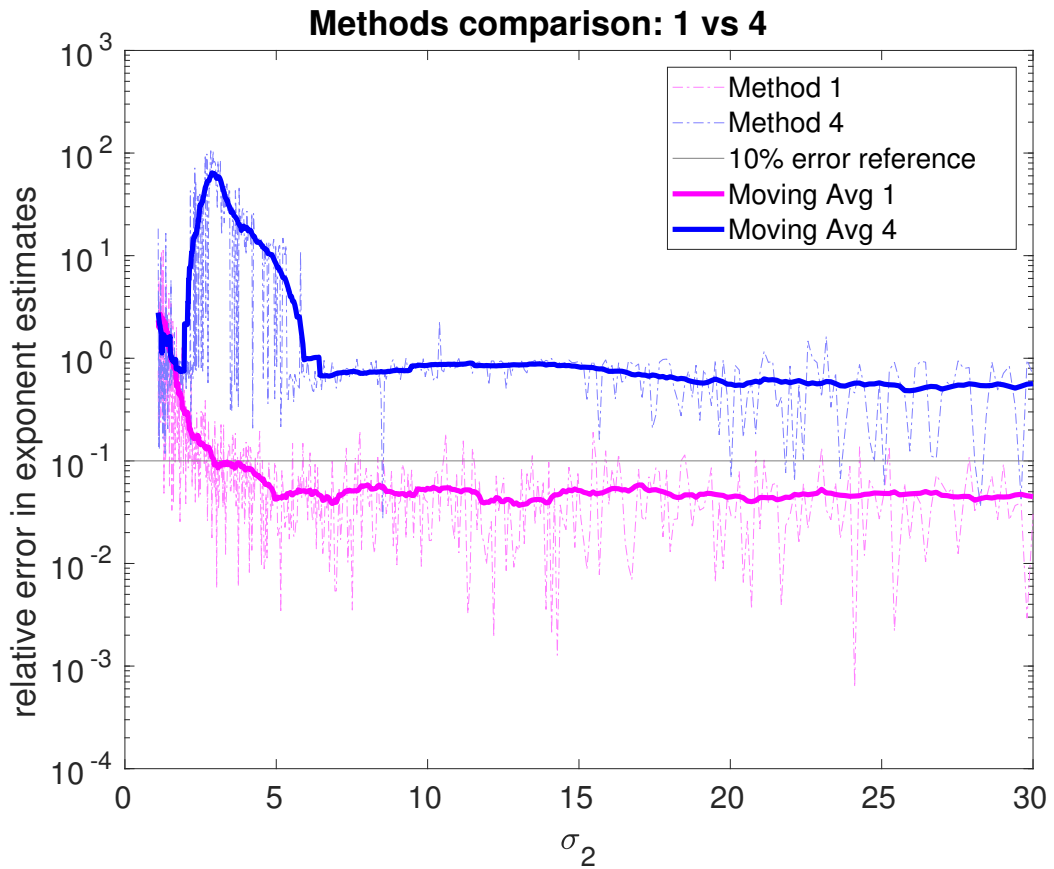
Figure 4.3: (1) Moment constraint method (magenta) compared with (4) MLE without moments (blue). Dashed lines are actual relative errors in exponent estimate, and solid lines are the 30-term moving average of the corresponding error curves. The horizontal black line is the 10% error for reference.
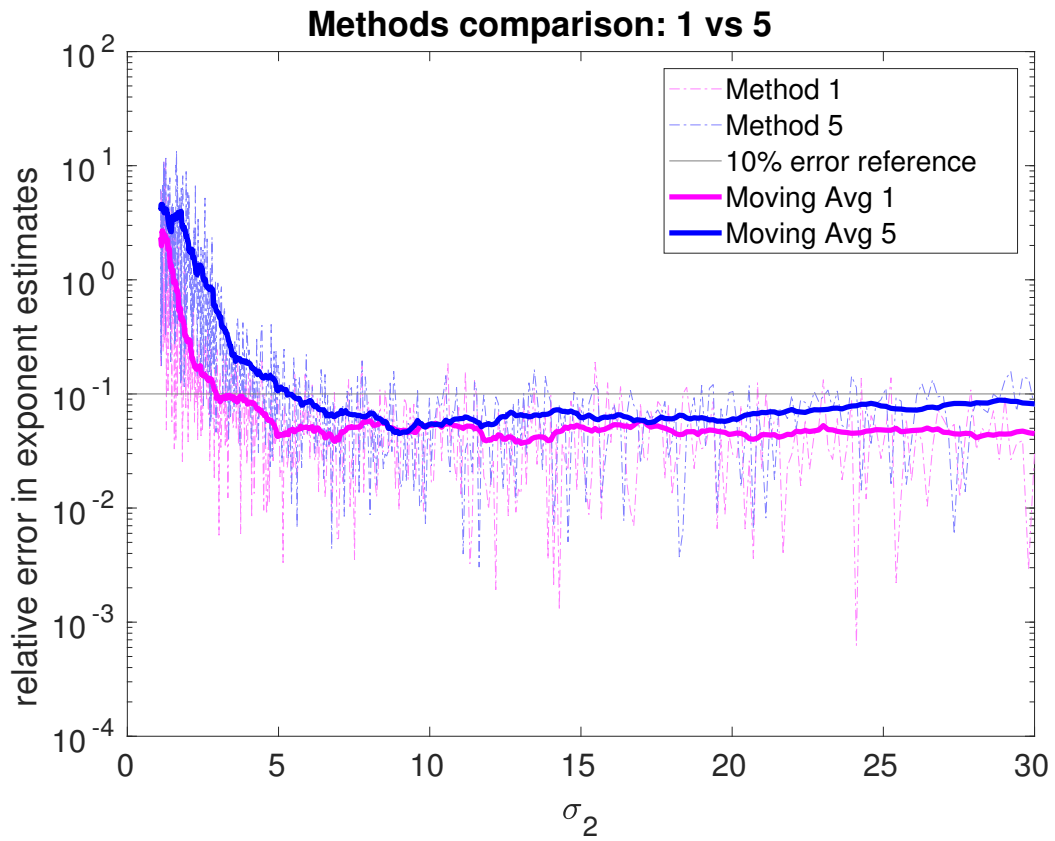
Figure 4.4: (1) Moment constraint method (magenta) compared with (5) Osborne's method (blue). Dashed lines are actual relative errors in exponent estimate, and solid lines are the 30-term moving average of the corresponding error curves. The horizontal black line is the 10% error for reference.

is discretized on $[-L, 0]$ using $\{s_k\}$, $k = 1, \ldots, \nu$ with $\Delta s = L/(\nu - 1)$ to give

$$A(\boldsymbol{\sigma})\boldsymbol{\alpha} = \boldsymbol{S} \tag{4.3.46}$$

where $A_{ij} = -e^{-t_i s_j}\Delta s$, $\alpha_j = \alpha(s_j)$, $S_i = S(t_i)$. Tikhonov regularization proceeds in the usual way by solving the regularized least squares problem

$$(\boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I})\boldsymbol{\alpha} = A^T\boldsymbol{S}, \tag{4.3.47}$$

for some small regularization parameter μ.

We try to apply this scale detection in addition to our proposed method, and it proves useful in some special four-term exponential fitting problems. Consider the function

$$S(t) = \sum_{i=1}^{4} \alpha_i e^{-\sigma_i t}, \tag{4.3.48}$$

where $\sigma_1 < \sigma_2 < \sigma_3 < \sigma_4$, with $\sigma_1/\sigma_2 = O(1), \sigma_3/\sigma_4 = O(1)$ and $\sigma_2 \ll \sigma_3$, meaning that σ_1, σ_2 are close, and σ_3, σ_4 are close.

We first use Tikhonov regularization to find the approximate values of the exponents. A wide range of possible values for the regularization parameters balance goodness-of-fit with the stability of the result. Since there are two groups of exponents, we pick an appropriate regularization parameter that yields two estimated exponents $\hat{\sigma}_1, \hat{\sigma}_2$ and use them as initial guesses for two separate two-exponential fitting problems:

$$S(t) = \sum_{i=1}^{4} \alpha_i e^{-\sigma_i t} \approx \sum_{i=1}^{2} \hat{\alpha}_i e^{-\hat{\sigma}_i t}. \tag{4.3.49}$$

Meanwhile, by least-squares, we also get estimates for the coefficients $\hat{\alpha}_1, \hat{\alpha}_2$.

Next, similar to the idea of the EM algorithm, we split the original dataset into two groups. The first group contains the data that is more likely to be generated by the slow-decaying exponents, and the second group contains the rest of data belonging to fast-decaying exponents. Bayes' rule is applied to compute the probability of each data point being generated from the slow-decaying exponents. Specifically, if there are

M random samples, the probability of the data point y_i belonging to the first group (slow-decaying exponents) is

$$\mathbb{P}\{y_i \in \text{Group } 1\} = \frac{\hat{\alpha}_1 \hat{\sigma}_1 e^{-\hat{\sigma}_1 y_i}}{\hat{\alpha}_1 \hat{\sigma}_1 e^{-\hat{\sigma}_1 y_i} + \hat{\alpha}_2 \hat{\sigma}_2 e^{-\hat{\sigma}_2 y_i}}, \quad \text{for } i = 1, \ldots, M. \qquad (4.3.50)$$

Then, we compare it to a uniform random number to determine final group assignment. If this probability is smaller than this random threshold, we assign the data point to group 1, otherwise to group 2. Finally, we fit two exponentials for each set of data (assuming different time scales λ_1 and λ_2 associated with each set) using our proposed moment-constraint method.

Four-term exponential fitting numerical results

This is another special application of two-exponential fitting, where we have two pairs of close exponents with different orders of magnitude. We may treat this problem as two double exponential fitting problem on two different time scales. Data is generated from (4.3.49). The inference results are shown in Table 4.2. The estimated values are followed by exact values of corresponding exponents. The last two columns are the rough estimates of $\hat{\sigma}_1$ and $\hat{\sigma}_2$ for the two groups we have mentioned in the previous section. Two-term exponential fitting is applied in these two groups at the end.

Sometimes, the algorithm would lead to complex conjugate solutions and we may simply rectify it by repeating the algorithm with a different intial guess or another random number seed, since real exponents are expected. However, when the exponents are very close together, most of the well-known methods will fail. However, when the exponents σ_k $(k = 1, 2, 3, 4)$ are not too close together and σ_2 and σ_3 are well-separated, our method can usually recover the four exponents with reasonable accuracy. We conclude that our method can roughly recover four-term exponential c.d.f.s with this special format.

Example	σ_1	σ_2	σ_3	σ_4	$\hat{\sigma}_1$	$\hat{\sigma}_2$
1	1.0 (1)	3.3 (3)	46.9 (40)	53.3 (60)	2	50
2	1.1 (1)	6.3 (5)	80.8 (80)	104.8 (100)	3	92
3	5.4 (5)	11.8 (10)	68.1 (60)	94.0 (110)	8	80
4	11.1 (10)	23.0 (20)	144.0 (100)	171.1 (200)	16	157

Table 4.2: Results of four-exponential fitting. The exponents satisfy $\sigma_1 < \sigma_2 < \sigma_3 < \sigma_4$, with $\sigma_1/\sigma_2 = O(1), \sigma_3/\sigma_4 = O(1)$ and $\sigma_2 \ll \sigma_3$. In each example, the table shows the estimated results for all exponents, with their true values in the parentheses. The last two columns in the table are the estimation for two groups resulted from Tikhonov regularization.

4.4 Summary

In summary, we have studied methods for exponential fitting, which are central to both the BDP and NMR problems we studied in previous chapters. Most popular methods fail when exponents are sufficiently close together. We have proposed the moment constraint method, based on the modified Prony method, to fit two-term exponential survival probability functions from random sample times. Our method overcomes some of the difficulties of recovering the exponents when they are close, yields more stable and reliable results compared with maximum likelihood methods and Matlab built-in fitting function. However, it doesn't resolve the exponents when they are sufficiently close, due to the fundamental limit to the resolution of exponents [8]. The moment constraint method is potentially useful for kernel density estimation given the sample times, and we have successfully applied this method in four-term exponential fitting problems under special circumstances.

Chapter 5

CONCLUSION

In this dissertation, we have investigated the problem of estimating and inferring parameters in exponential models that describe phenomena in biophysics and imaging given measured data. Exponential analysis is often an inverse problem which is highly ill-conditioned, in a sense that the solution may not exist, may not be unique or may not be stable. We have carefully studied three different problems related to the central topic of exponential analysis: birth-death process (BDP), nuclear magnetic resonance (NMR), and direct exponential fitting.

The BDP is mainly concerned with inferring transition rates of proteins from their extinction time distributions. To analyze this problem, we model the reaction coordinate using a BDP. Specifically, we consider a protein with N domains where every domain that unfolds (folds) increases (decreases) the reaction coordinate by an integer so that it maps to the set of integers $\{0, 1, \ldots, N\}$. As input data, our method uses (i) the extinction time of trajectories, starting from when the protein leaves the 0 state for the first time and finishing when the protein re-enters state 0 for the first time and (ii) the maximum number of unfolded domains in the said trajectory. Since the maximum value n reached by each trajectory is known, we use the proportion of trajectories that do not exceed n and corresponding mean extinction times (ET) to recover the birth-death rates sequentially from 1 to N. In each step n, we focus on coefficients of the characteristic polynomial of the matrix that governs the BDP, relate it to its previous two states, and set up a recurrence relation. In general, the initial error will propagate with the site number exponentially. However, with sufficient amount of input data, we can recover the rates with relatively small error. For instance, given 50 million ETs of an 11-site BDP, we can recover the rates with a relative error about 3%.

Future works in this section include extending the method to transmission problems where the BDP becomes extinct at the largest state, as well as to more general Markov chains.

The NMR problem arises from estimating the relaxation time of a particular tissue in biological organisms from given data measurements. From this, the composition of that tissue can easily be inferred. Traditionally, this is done by applying the inverse Laplace transform, which is ill-posed. We proposed and compared Tikhonov regularization methods with one parameter and with two parameters. Picard coefficient analysis in singular value decomposition shows that the two-parameter regularization is not as good as the one-parameter regularization, due to its complexity and noise amplifying effect. We showed numerical examples of both methods and it was clear that the one-parameter regularization yielded better result than the corresponding two-parameter regularization method. In addition, we have proposed a method to use directional total variation (DTV) as a regularizer to the NMR problem, which essentially requires two tuning parameters. Mathematical formulations have been set up, but we need more numerical results that support this model. Implementing the DTV regularization numerically can be a main focus in the future.

Direct fitting of an exponential function to data is a problem with a long history, in which many researchers have investigated and created effective methods. A common pitfall for these classic method is that they cannot treat problems in which exponents are close to each other because of the fundamental resolution limit to such problems. For a specific type of two-term exponential function, which are survival functions of some probability distributions, we have proposed the moment constraint method to resolve the exponents from random time samples. This method is capable of recovering the exponents with a better accuracy and stability, even if the exponents are close together, and it beats maximum likelihood estimation, Matlab's built-in 'fit' function, and the modified Prony method. We have also implemented the moment constraint method in four-term exponential fitting problems whose exponents can be separated

into two groups by magnitude. By splitting the problem into two subproblems of two-term exponential fitting and using Tikhonov regularization with the EM algorithm, we produced results with reasonable accuracy. In the future, we are interested in extending this method to three-term and n-term exponential fitting where exponents are relatively close to each other.

BIBLIOGRAPHY

[1] Beth G Ashinsky, Christopher E Coletta, Mustapha Bouhrara, Vanessa A Lukas, Julianne M Boyle, David A Reiter, Corey P Neu, Ilya G Goldberg, and Richard G Spencer. Machine learning classification of oarsi-scored human articular cartilage using magnetic resonance imaging. *Osteoarthritis and cartilage*, 23(10):1704–1712, 2015.

[2] David Baker. A surprising simplicity to protein folding. *Nature*, 405(6782):39–42, 2000.

[3] H Barkhuijsen, R De Beer, and D Van Ormondt. Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals. *Journal of Magnetic Resonance (1969)*, 73(3):553–557, 1987.

[4] Ilker Bayram and Mustafa E Kamasak. Directional total variation. *IEEE Signal Processing Letters*, 19(12):781–784, 2012.

[5] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

[6] George I Bell. Models for the specific adhesion of cells to cells. *Science*, 200(4342):618–627, 1978.

[7] Herman JC Berendsen. A glimpse of the holy grail? *Science*, 282(5389):642–643, 1998.

[8] M Bertero, P Boccacci, and Edward Roy Pike. On the recovery and resolution of exponential relaxation rates from experimental data: a singular-value analysis of the laplace transform inversion in the presence of noise. *Proc. R. Soc. Lond. A*, 383(1784):15–29, 1982.

[9] Robert B Best, Emanuele Paci, Gerhard Hummer, and Olga K Dudko. Pulling direction as a reaction coordinate for the mechanical unfolding of single molecules. *The Journal of Physical Chemistry B*, 112(19):5968–5976, 2008.

[10] Robert F Botta, Carl M Harris, and William G Marchal. Characterizations of generalized hyperexponential distribution functions. *Stochastic Models*, 3(1):115–148, 1987.

[11] Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[12] Axel T Brunger, Paul D Adams, G Marius Clore, Warren L DeLano, Piet Gros, Ralf W Grosse-Kunstleve, Jian-Sheng Jiang, John Kuszewski, Michael Nilges, Navraj S Pannu, et al. Crystallography & nmr system: A new software suite for macromolecular structure determination. *Acta Crystallographica-Section D-Biological Crystallography*, 54(5):905–921, 1998.

[13] Joseph D Bryngelson, José Nelson Onuchic, Nicholas D Socci, and Peter G Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

[14] Kenneth P Bube and Robert Burridge. The one-dimensional inverse problem of reflection seismology. *SIAM review*, 25(4):497–559, 1983.

[15] M Büttiker. Larmor precession and the traversal time for tunneling. *Physical Review B*, 27(10):6178, 1983.

[16] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[17] Hasan Celik, Mustapha Bouhrara, David A Reiter, Kenneth W Fishbein, and Richard G Spencer. Stabilization of the inverse laplace transform of multiexponential decay through introduction of a second dimension. *Journal of Magnetic Resonance*, 236:134–139, 2013.

[18] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[19] Joshua C Chang, Pak-Wing Fok, and Tom Chou. Bayesian uncertainty quantification for bond energies and mobilities using path integral analysis. *Biophysical journal*, 109(5):966–974, 2015.

[20] Pictiaw Chen and Z Sun. A review of non-destructive methods for quality evaluation and sorting of agricultural products. *Journal of Agricultural Engineering Research*, 49:85–98, 1991.

[21] Shi-Jie Chen and Ken A Dill. Rna folding energy landscapes. *Proceedings of the National Academy of Sciences*, 97(2):646–651, 2000.

[22] Cramer and Gabriel. *Introduction a l'analyse des lignes courbes algebriques par Gabriel Cramer...* chez les freres Cramer & Cl. Philibert, 1750.

[23] Raymond Damadian. Tumor detection by nuclear magnetic resonance. *Science*, 171(3976):1151–1153, 1971.

[24] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[25] Luc Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265. ACM, 1986.

[26] Chuong B Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897, 2008.

[27] Christopher M Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–890, 2003.

[28] Charles R Doering, Khachik V Sargsyan, and Leonard M Sander. Extinction times for birth-death processes: Exact results, continuum asymptotics, and the failure of the fokker–planck approximation. *Multiscale Modeling & Simulation*, 3(2):283–299, 2005.

[29] Olga K Dudko, Gerhard Hummer, and Attila Szabo. Intrinsic rates and activation free energies from single-molecule pulling experiments. *Physical review letters*, 96(10):108101, 2006.

[30] Alan Edelman and H Murakami. Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776, 1995.

[31] Charles L Epstein and John Schotland. The bad truth about laplace's transform. *SIAM review*, 50(3):504–520, 2008.

[32] Richard R Ernst, Geoffrey Bodenhausen, Alexander Wokaun, et al. *Principles of nuclear magnetic resonance in one and two dimensions*, volume 14. Clarendon Press Oxford, 1987.

[33] John W Evans, William B Gragg, and Randall J LeVeque. On least squares exponential sum approximation with positive coefficients. *Mathematics of Computation*, 34(149):203–211, 1980.

[34] Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance evaluation*, 31(3-4):245–279, 1998.

[35] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[36] Danielle Florens-Zmirou. Approximate discrete-time schemes for statistics of diffusion processes. *Statistics: A Journal of Theoretical and Applied Statistics*, 20(4):547–557, 1989.

[37] Pak-Wing Fok and Tom Chou. Reconstruction of potential energy profiles from multiple rupture time distributions. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 466(2124):3479–3499, 2010.

[38] WD Foltz and DA Jaffray. Principles of magnetic resonance imaging. *Radiation research*, 177(4):331–348, 2012.

[39] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

[40] LB Freund. Characterizing the resistance generated by a molecular bond as it is forcibly separated. *Proceedings of the National Academy of Sciences*, 106(22):8818–8823, 2009.

[41] Horst Friebolin and Jack K Becconsall. *Basic one-and two-dimensional NMR spectroscopy*. VCH Weinheim, 1993.

[42] J Rod Gimbel and Emanuel Kanal. Can patients with implantable pacemakers safely undergo magnetic resonance imaging?, 2004.

[43] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[44] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[45] Per Christian Hansen. The truncatedsvd as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, 1987.

[46] Per Christian Hansen. The discrete picard condition for discrete ill-posed problems. *BIT Numerical Mathematics*, 30(4):658–672, 1990.

[47] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, 34(4):561–580, 1992.

[48] Francis Begnaud Hildebrand. *Introduction to numerical analysis*. Courier Corporation, 1987.

[49] Jeffrey Hokanson. *Numerically stable and statistically efficient algorithms for large scale exponential fitting*. PhD thesis, Rice University, 2013.

[50] Kenneth Holmström and Jöran Petersson. A review of the parameter estimation problem of fitting positive exponential sums to empirical data. *Applied Mathematics and Computation*, 126(1):31–61, 2002.

[51] Shui-Hung Hou. Classroom note: A simple proof of the leverrier–faddeev characteristic polynomial algorithm. *SIAM review*, 40(3):706–709, 1998.

[52] Alston Scott Householder. On prony's method of fitting exponential decay curves and multiple-hit survival curves. Technical report, 1950.

[53] Yingbo Hua and Tapan K Sarkar. Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(5):814–824, 1990.

[54] Andrei A Istratov and Oleg F Vyvenko. Exponential analysis in physical phenomena. *Review of Scientific Instruments*, 70(2):1233–1257, 1999.

[55] John A Jacquez and Carl P Simon. The stochastic si model with recruitment and deaths i. comparison with the closed sis model. *Mathematical biosciences*, 117(1-2):77–125, 1993.

[56] Niels Keiding. Maximum likelihood estimation in the birth-and-death process. *The Annals of Statistics*, pages 363–372, 1975.

[57] David G Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, pages 338–354, 1953.

[58] Leonard Kleinrock. *Queueing systems, volume 2: Computer applications*, volume 66. wiley New York, 1976.

[59] Rasmus Dalgas Kongskov, Yiqiu Dong, and Kim Knudsen. Directional total generalized variation regularization. *arXiv preprint arXiv:1701.02675*, 2017.

[60] Jason A Koutcher and C Tyler Burt. Principles of nuclear magnetic resonance. *Journal of nuclear medicine*, 25(1):101–111, 1984.

[61] Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

[62] Sun-Yuan Kung, K Si Arun, and DV Bhaskar Rao. State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem. *JOSA*, 73(12):1799–1811, 1983.

[63] Kenneth K Kwong, John W Belliveau, David A Chesler, Inna E Goldberg, Robert M Weisskoff, Brigitte P Poncelet, David N Kennedy, Bernice E Hoppel, Mark S Cohen, and Robert Turner. Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proceedings of the National Academy of Sciences*, 89(12):5675–5679, 1992.

[64] Charles L Lawson and Richard J Hanson. *Solving least squares problems*, volume 15. Siam, 1995.

[65] Malcolm H Levitt. *Spin dynamics: basics of nuclear magnetic resonance.* John Wiley & Sons, 2001.

[66] Zhi-Pei Liang and Paul C Lauterbur. *Principles of magnetic resonance imaging: a signal processing perspective.* SPIE Optical Engineering Press, 2000.

[67] Everett A Lipman, Benjamin Schuler, Olgica Bakajin, and William A Eaton. Single-molecule measurement of protein folding kinetics. *Science*, 301(5637):1233–1235, 2003.

[68] Anna Litwic, Mark H Edwards, Elaine M Dennison, and Cyrus Cooper. Epidemiology and burden of osteoarthritis. *British medical bulletin*, 105(1):185–199, 2013.

[69] Fumiaki Machihara. On the property of eigenvalues of some infinitesimal generator. *Operations Research Letters*, 2(3):123–126, 1983.

[70] Ajay P Manuel, John Lambert, and Michael T Woodside. Reconstructing folding energy landscapes from splitting probability analysis of single-molecule trajectories. *Proceedings of the National Academy of Sciences*, 112(23):7183–7188, 2015.

[71] PM Matthews and P Jezzard. Functional magnetic resonance imaging. *Journal of Neurology, Neurosurgery & Psychiatry*, 75(1):6–12, 2004.

[72] JG McWhirter and E R Pike. On the numerical inversion of the laplace transform and similar fredholm integral equations of the first kind. *Journal of Physics A: Mathematical and General*, 11(9):1729, 1978.

[73] Ingemar Nåsell. On the quasi-stationary distribution of the stochastic logistic epidemic. *Mathematical biosciences*, 156(1-2):21–40, 1999.

[74] Jan Nygaard Nielsen, Henrik Madsen, and Peter C Young. Parameter estimation in stochastic differential equations: an overview. *Annual Reviews in Control*, 24:83–94, 2000.

[75] Andres F Oberhauser, Paul K Hansma, Mariano Carrion-Vazquez, and Julio M Fernandez. Stepwise unfolding of titin under force-clamp atomic force microscopy. *Proceedings of the National Academy of Sciences*, 98(2):468–472, 2001.

[76] F Oesterhelt, M Rief, and HE Gaub. Single molecule force spectroscopy by afm indicates helical structure of poly (ethylene-glycol) in water. *New Journal of Physics*, 1(1):6, 1999.

[77] Seiji Ogawa, Tso-Ming Lee, Alan R Kay, and David W Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872, 1990.

[78] Mikael Oliveberg and Peter G Wolynes. The experimental survey of protein-folding energy landscapes. *Quarterly reviews of biophysics*, 38(3):245–288, 2005.

[79] MR Osborne. Some special nonlinear least squares problems. *SIAM Journal on Numerical Analysis*, 12(4):571–592, 1975.

[80] MR Osborne and Gordon K Smyth. A modified prony algorithm for fitting functions defined by difference equations. *SIAM journal on scientific and statistical computing*, 12(2):362–382, 1991.

[81] MR Osborne and Gordon K Smyth. A modified prony algorithm for exponential function fitting. *SIAM Journal on Scientific Computing*, 16(1):119–138, 1995.

[82] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[83] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955.

[84] Victor Pereyra and Godela Scherer. *Exponential data fitting and its applications*. Bentham Science Publishers, 2010.

[85] David L Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM (JACM)*, 9(1):84–97, 1962.

[86] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1996.

[87] R Prony. Essai experimental–,-. *J. de lEcole Polytechnique*, 1795.

[88] Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.

[89] Matthias Rief, Mathias Gautel, Alexander Schemmel, and Hermann E Gaub. The mechanical stability of immunoglobulin and fibronectin iii domains in the muscle protein titin measured by atomic force microscopy. *Biophysical journal*, 75(6):3008–3014, 1998.

[90] Geoffrey C Rollins and Ken A Dill. General mechanism of two-state protein folding kinetics. *Journal of the American Chemical Society*, 136(32):11420–11427, 2014.

[91] Robert Ros, Falk Schwesinger, Dario Anselmetti, Martina Kubon, Rolf Schäfer, Andreas Plückthun, and Louis Tiefenauer. Antigen binding forces of individually addressed single-chain fv antibody molecules. *Proceedings of the National Academy of Sciences*, 95(13):7402–7405, 1998.

[92] Sheldon M Ross. *Introduction to probability models.* Academic press, 2014.

[93] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[94] R Dustin Schaeffer, Alan Fersht, and Valerie Daggett. Combining experiment and simulation in protein folding: closing the gap for small model systems. *Current opinion in structural biology*, 18(1):4–9, 2008.

[95] Benjamin Schuler and Hagen Hofmann. Single-molecule spectroscopy of protein folding dynamics–expanding scope and timescales. *Current opinion in structural biology*, 23(1):36–47, 2013.

[96] Gerrit Schultz. *Magnetic resonance imaging with nonlinear gradient fields: signal encoding and image reconstruction.* Springer Science & Business Media, 2013.

[97] Charles P Slichter. *Principles of magnetic resonance*, volume 1. Springer Science & Business Media, 2013.

[98] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[99] Yohichi Suzuki and Olga K Dudko. Single-molecule rupture dynamics on multidimensional landscapes. *Physical review letters*, 104(4):048101, 2010.

[100] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[101] Andreĭ Nikolaevich Tikhonov, AV Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013.

[102] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[103] Charles F Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100, 2000.

[104] James M Varah. On fitting exponentials by nonlinear least squares. *SIAM journal on scientific and statistical computing*, 6(1):30–44, 1985.

[105] Curtis R Vogel and Mary E Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[106] John H Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5(3):329–350, 1970.

[107] Ronald W Wolff. Problems of statistical inference for birth and death queuing models. *Operations Research*, 13(3):343–357, 1965.

[108] Michael T Woodside and Steven M Block. Reconstructing folding energy landscapes by single-molecule force spectroscopy. *Annual review of biophysics*, 43:19–39, 2014.

[109] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendix

INFERENCE WITH LEVERRIER-FADDDEEV ALGORITHM

In light of lemma 2.7, we can assume the following forms:

$$\xi_i^{(n)} = C_{i1}^{(n)}\lambda_n + C_{i2}^{(n)}\mu_n + C_{i3}^{(n)}, \quad i = 1, 2, \tag{A.0.1}$$

$$\eta_j^{(n)} = \hat{C}_{j1}^{(n)}\lambda_n + \hat{C}_{j2}^{(n)}\mu_n + \hat{C}_{j3}^{(n)}, \quad j = 2, 3, \tag{A.0.2}$$

where $C^{(n)}$ and $\hat{C}^{(n)}$ are coefficients that depend on $\lambda_1, \ldots, \lambda_{n-1}$, $\mu_1, \ldots, \mu_{n-1}$; we describe how to compute $C^{(n)}$ and $\hat{C}^{(n)}$ in Section A. If we plug (A.0.1) and (A.0.2) into (2.4.32) and (2.4.33), and denote $r^{(n)} = \Pi^{(n)}/\mu_1$, then we get a linear system in two variables (λ_n, μ_n):

$$F^{(n)} \begin{bmatrix} \lambda_n \\ \mu_n \end{bmatrix} = G^{(n)}$$

where the 2×2 matrix $F^{(n)}$ and 2-vector $G^{(n)}$ are defined as

$$F^{(n)} = \begin{bmatrix} r^{(n)}C_{11}^{(n)} - \hat{C}_{21}^{(n)} & r^{(n)}C_{12}^{(n)} - \hat{C}_{22}^{(n)} \\ r^{(n)}(C_{21}^{(n)} - C_{11}^{(n)}M^{(n)}) - \hat{C}_{31}^{(n)} & r^{(n)}(C_{22}^{(n)} - C_{12}^{(n)}M^{(n)}) - \hat{C}_{32}^{(n)} \end{bmatrix}, \tag{A.0.3}$$

$$G^{(n)} = \begin{bmatrix} \hat{C}_{23}^{(n)} - r^{(n)}C_{13}^{(n)} \\ \hat{C}_{33}^{(n)} - r^{(n)}(C_{23}^{(n)} - C_{13}^{(n)}M^{(n)}) \end{bmatrix}. \tag{A.0.4}$$

This matrix is consistent with equations (2.4.51)–(2.4.54). The rates at current site are therefore given by $[\lambda_n, \mu_n]^T = (F^{(n)})^{-1}G^{(n)}$.

Implementation

The method for computing the coefficient matrices $C^{(n)}$ and $\hat{C}^{(n)}$ is the Leverrier-Faddeev (L-F) algorithm [51]. The L-F algorithm computes all coefficients of the

characteristic polynomial for a given $n \times n$ square matrix with time complexity $\mathcal{O}(n^4)$. In this problem, it suffices to get only the last three coefficients $\xi_3^{(n)}$, $\xi_2^{(n)}$ and $\xi_1^{(n)}$. Given the expression in (A.0.1), we have

$$
\begin{cases}
C_{i1}^{(n)} &= \xi_i^{(n)}(\lambda_1, \ldots, \lambda_{n-1}, 1; \mu_1, \ldots, \mu_{n-1}, 0) - \xi_i^{(n)}(\lambda_1, \ldots, \lambda_{n-1}, 0; \mu_1, \ldots, \mu_{n-1}, 0) \\
C_{i2}^{(n)} &= \xi_i^{(n)}(\lambda_1, \ldots, \lambda_{n-1}, 0; \mu_1, \ldots, \mu_{n-1}, 1) - \xi_i^{(n)}(\lambda_1, \ldots, \lambda_{n-1}, 0; \mu_1, \ldots, \mu_{n-1}, 0) \\
C_{i3}^{(n)} &= \xi_i^{(n)}(\lambda_1, \ldots, \lambda_{n-1}, 0; \mu_1, \ldots, \mu_{n-1}, 0)
\end{cases}
$$

for $i = 1, 2$. With (A.0.2), we also have

$$
\begin{cases}
\hat{C}_{i1}^{(n)} &= \eta_i^{(n)}(\lambda_2, \ldots, \lambda_{n-1}, 1; \mu_2, \ldots, \mu_{n-1}, 0) - \eta_i^{(n)}(\lambda_2, \ldots, \lambda_{n-1}, 0; \mu_2, \ldots, \mu_{n-1}, 0) \\
\hat{C}_{i2}^{(n)} &= \eta_i^{(n)}(\lambda_2, \ldots, \lambda_{n-1}, 0; \mu_2, \ldots, \mu_{n-1}, 1) - \eta_i^{(n)}(\lambda_2, \ldots, \lambda_{n-1}, 0; \mu_2, \ldots, \mu_{n-1}, 0) \\
\hat{C}_{i3}^{(n)} &= \eta_i^{(n)}(\lambda_2, \ldots, \lambda_{n-1}, 0; \mu_2, \ldots, \mu_{n-1}, 0)
\end{cases}
$$

for $j = 2, 3$. A concise and straightforward algorithm containing all steps is presented in Algorithm 12.

Algorithm 12 Inference of birth and death rates up to site N with L-F algorithm

---

1: Input: An array of extinction times T along with maximal site of repeated simulation of a birth death process.
2: Initialize: Compute the conditional probabilities of left exit $\{\Pi^{(1)}, \ldots, \Pi^{(N)}\}$, and mean of conditional extinction times $\{M^{(1)}, \ldots, M^{(N)}\}$.
3: At site 1, $\mu_1 = \Pi^{(1)}/M^{(1)}$ and $\lambda_1 = (1 - \Pi^{(1)})/M^{(1)}$.
4: **for** $j = 2 : N$ **do**
5:     Compute the coefficients of the characteristic polynomial of $A^{(j)}$ to obtain $C_{ij}^{(n)}$ and $\hat{C}_{ij}^{(n)}$ in section A. Note the true values of the rates λ_n and μ_n are not needed in this calculation.
6:     Form the constraint matrices $F^{(j)}$ and $G^{(j)}$ as in (A.0.3) and (A.0.4);
7:     Solve $[\lambda_j, \mu_j]^T = (F^{(j)})^{-1} G^{(j)}$.
8:     **if** $j == N$ **then**
9:         $\lambda_j = 0$
10:     **end if**
11: **end for**
12: Output $\{\mu_1, \ldots, \mu_N\}$ and $\{\lambda_1, \ldots, \lambda_{N-1}\}$