Dimensionality Reduction of the Chemical Master Equation

by

Midhun Kathanaruparambil Sukumaran

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	:	Roger B. Sidje Professor, Applied & Computational Math, Department of Mathematics, University of Alabama
Supervisor	:	Dr. Brian P. Ingalls Associate Professor, Dept. Applied Mathematics, University of Waterloo
Internal Member	:	Mohammad Kohandel Associate Professor, Dept. Applied Mathematics, University of Waterloo
Internal Member	:	David Siegel Professor, Dept. Applied Mathematics, University of Waterloo
Internal-External Member	:	Justin W.L. Wan Professor, David R. Cheriton School of Computer Science, University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The dynamics of biochemical systems show significant variability when the reactant populations are small. Standard approaches via deterministic modeling exclude such variability. A well established stochastic model, the Chemical master equation (CME), describes the dynamics of biochemical systems by representing the time evolution of the probability distribution of species' discrete states in a well-mixed reaction volume. However, the dimension of the CME (i.e. the number of transition states in the system) rapidly grows as the molecular population and number of reactions in the network increases. Also, the dynamics of biochemical systems typically vary over a wide range of time scales: a phenomenon referred to as stiffness. Large dimensions and stiffness pose challenges to numerical analysis of system behavior. By eliminating the fast modes, which correspond to fast time scales that are often not experimentally observed, a model reduction can be achieved. In our work, we apply such a model reduction to the CME. The slow and fast modes of the system correspond to small and large eigenvalues of the transition matrix of the CME. By a transformation, we exclude the fast modes to arrive at a truncated model. We propose a method based on eigenbasis transformations that provide efficient approximations that are accurate beyond a short initial time interval. We also present efficient algorithms for generation of the CME from a network and for computation of eigenbases. Finally, we describe how this reduction approach can be implemented to provide efficient time-step identification in a well-established scheme for an approximation of the CME (the so-called finite state projection).

Acknowledgements

To my life-coaches, my mother Usha Sukumaran and wife Lakshmi Ranjit: because I owe it all to you. Many Thanks!

I am grateful to my father K. P. Sukumaran, Lakshmi's parents Ranjit Vasavan and Suma Ranjit, brother Jidhin K. Sukumaran, and my daughter Veda Midhun who have provided me through moral and emotional support in my life. I am also grateful to my other family members, especially Mohanan Velliachan, and friends who have supported me along the way.

I would like to express my sincere gratitude to my advisor Prof. Brian P. Ingalls for the continuous support to my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. Life is wonderful with great people around. Thanks a lot, Brian, for building a fool to someone valuable.

I would also like to thank Dr. Marc R. Roussel for bringing me to this wonderland Canada and offering a great research opportunity to build my career.

I am extremely thankful to all math faculty members and my fellow graduate students at the University of Waterloo for providing all kinds of possible help.

Lastly, I sincerely thank all those who have directly or indirectly helped for the work reported herein.

Dedication

Ushamme, This is for you....

Table of Contents

List of Figures

List of Tables

Abbreviations

CLE Chemical Langevin Equation 3, 4

CME Chemical Master Equation x–xii, 3–10, 13–24, 32, 33, 41, 43, 44, 48–51, 55, 80, 82–84, 91–97, 102–106

FSP Finite State Projection 7–9, 15, 16, 19, 21, 94–98, 102, 104–106

LNA Linear Noise Approximation 4

MM Michaelis-Menten x–xii, 6, 19, 26, 32, 41, 74, 75, 78, 82, 88, 91

ODE Ordinary Differential Equation 5, 6, 8, 73, 76, 81

RRE Reaction Rate Equation 3, 4

RREF Row Reduced Echelon Form 28, 65, 66, 77

SSA Stochastic Simulation Algorithm 3, 4, 15, 93, 105

Chapter 1

Introduction

Biology is a field of science that studies the most complex system in the universe: life. The field uses a wide range of tools to understand various parts of the living organisms. Although the field has a long history which dates back thousands of years, we are still a long way from understanding or predicting complex mechanisms in their entirety. Over the past several decades, through developments in the Molecular Biology field, we have achieved an understanding of the molecular interactions that underlie cellular behaviors in living organisms. Historically, molecular biology analysis was done through a reductionist approach which analyzes biochemical systems by focusing on individual parts of the system. The reductionist approach is only effective for simple systems for which a comprehensive behavior of a system can be formulated from knowledge of its building blocks. However, this does not typically hold for complex systems because a change in one component may affect multiple parts unpredictably. As an analogy, we cannot hope to understand the complex functioning of the airplane by listing all its parts [1].

In earlier days of molecular biology, comprehensive study of intracellular systems was hampered by the limited scope of available measurement technologies. In contrast, due to improvements in high-throughput measurements in the last couple of decades, we are able to collect a huge amount of data which paved the way to study systems in their entirety (holism). The field of comprehensive study of the behavior of biomolecular systems is called (Molecular) Systems Biology [1, 2]. System biology studies call for the use of computational and mathematical modeling and analysis methods to interpret complex system behaviors. System biology has promising applications in a variety of fields including agriculture, manufacturing, biofuels, and most importantly health and disease.

System biology regularly employs dynamical models calibrated against high throughput experimental data [3]. Modeling techniques are commonly characterized as either bottom-up (built up from fundamental units) or top-down (built from the observations of the complete system) direction. However, in system biology, most modeling projects follow a middle-out approach, which starts with characterization of systems (such as cells or pathways), followed by top-down or bottom-up extension.

The modeling approaches focus on identification of two system features: static structure and temporal dynamics. Static structure consist of component interactions and the characterization of such connections, such as signaling or mass transfer. Beyond structural information, the most important analysis is the identification of the dynamical nature of these systems; how they perform naturally and how they respond to environmental factors. As a consequence, we can use predictions of system behavior to design interventions to achieve specific goals.

Mathematical modeling and simulation of biomolecular networks are achieved through a variety of methods, as shown in Figure 1.1. Molecular dynamics models capture biochem-

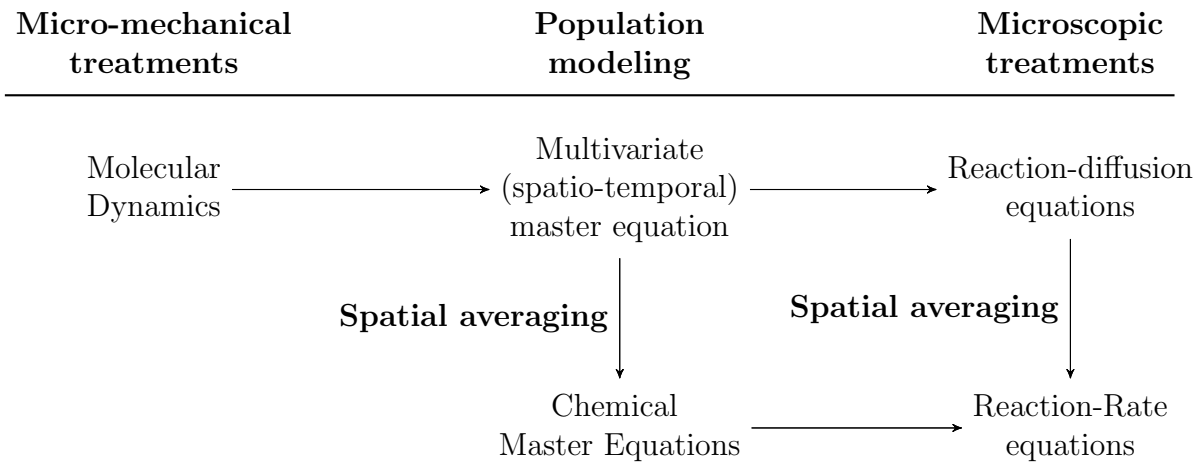

Figure 1.1: Three levels of modeling approaches to biochemical systems (modified from [4]).

ical system at the atomic level. By averaging atomic level internal degrees of freedom, spatio-temporal master equations can be derived to describe biochemical systems. Applying continuous approximations to the molecular population, reaction-diffusion equations can be used to approximate the system. Spatial averaging give rise to the Chemical Master Equation (CME) and Reaction Rate Equation (RRE) from the respective treatments as shown in the figure. Some other modeling approaches are described in the paper [5].

In the paper [6], the approximation from CME to RRE is further expanded as described in Figure 1.2. CME and SSA are derived as exact stochastic modeling approaches. Under the assumption of constant reaction propensity during a long time interval $\tau > \Delta t$, tau-leaping approximates the dynamical behavior of Chemical master equation [7, 8, 9]. CLE

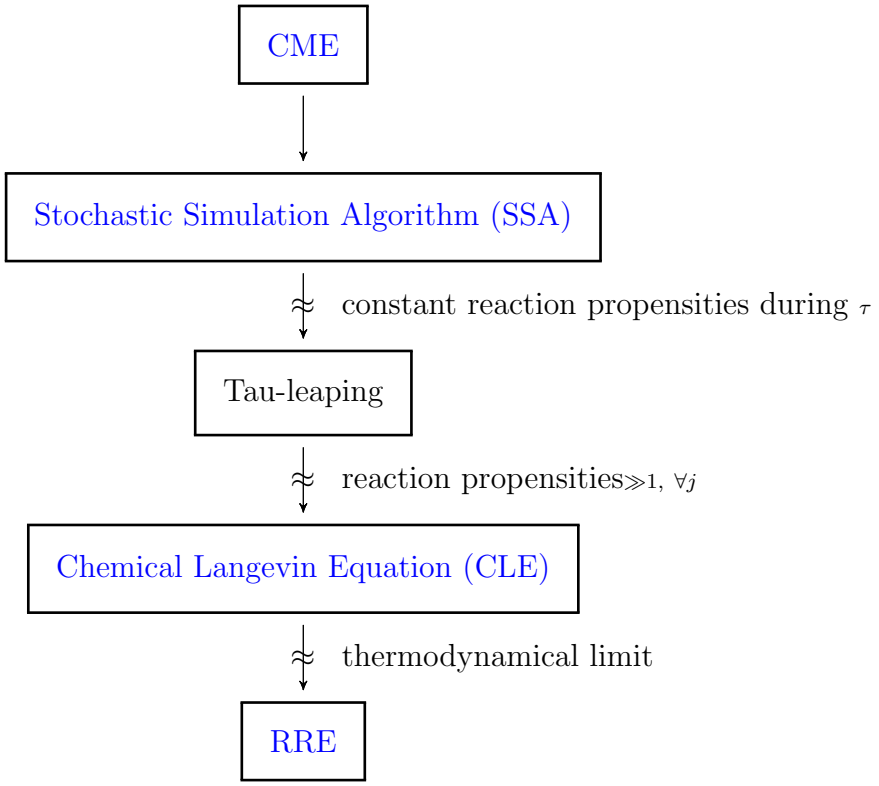

Figure 1.2: Exact stochastic modeling of chemical kinetics and approximations. Arrows represents exact representations of the source technique and arrows with $\approx$ symbol represents the approximations to the source technique under the condition indicated [6].

models [10, 11] are reduced sets of stochastic differential equations (compared to the CME) that approximate a biochemical system under the assumption of large reaction propensities. In the thermodynamic limit, biochemical systems can be further reduced to the reaction rate equations (derived from laws of mass-action). In a continuous population dynamics, stochastic approaches such as Moment Closure methods [12, 13], and the Linear Noise Approximation (LNA) [14] can also be used to approximate the dynamical behaviour of biochemical systems. A review of these methods can be found in the recent literature [15].

Out of these techniques, most mathematical models of biochemical systems published to date are represented by reaction rate equations (derived from laws of mass-action). Many such models can be accessed from Biomodel database at http://www.ebi.ac.uk/biomodels-main. These models are highly tractable in terms of construction, simulation, and analysis. However, a limitation of these differential equation-based models is that they are deterministic, and so cannot capture the variability that often dominates the behavior of biochemical networks, especially when molecular populations are small [16, 17, 18, 19]. To adequately address such cases, stochastic models are called for. The model most commonly applied to capture variability in well-mixed biochemical systems is the CME [20, 14], a linear system of Ordinary Differential Equation (ODE)s, which describes the dynamics of the probability distribution over all states in which the system may find itself [21, 22, 23].

Linear systems of ODEs are typically thought of as highly tractable. However, simulation and analysis of the CME is hampered by two major challenges. The first is dimension, i.e., the total number of states that the system can attain. For closed reaction networks in which all molecular populations are bounded, the CME is of finite dimension, but the dimension size grows explosively with the reactant population sizes. For example, a closed two step reaction chain A $\rightleftharpoons$ B $\rightleftharpoons$ C has 66 states when there are a total of 10 molecules present in the system. When there are 1000 molecules, the state dimension is 501501. When describing intracellular reaction networks, the state dimension is often too high for any meaningful computation to be carried out. To make matters worse, for open reaction networks in which molecular populations are not bounded (which is the most common case of interest in a biochemical context) the state space is infinite.

The second challenge when dealing with the CME is stiffness, i.e., wide range (often over

orders of magnitude) of the time-scales of dynamics within the system. This often makes direct simulation and analysis intractable. The timescales of a linear system of ODEs are quantified by the eigenvalues of the coefficient (transition) matrix. As an example, the eigenspectrum of the CME of a (Michaelis-Menten (MM)) biochemical system is shown in Figure 1.3; distinct times-scales are readily identified as separate groups of eigenvalues. Lower magnitude eigenvalues indicate slower timescales; large magnitude eigenvalues rep-

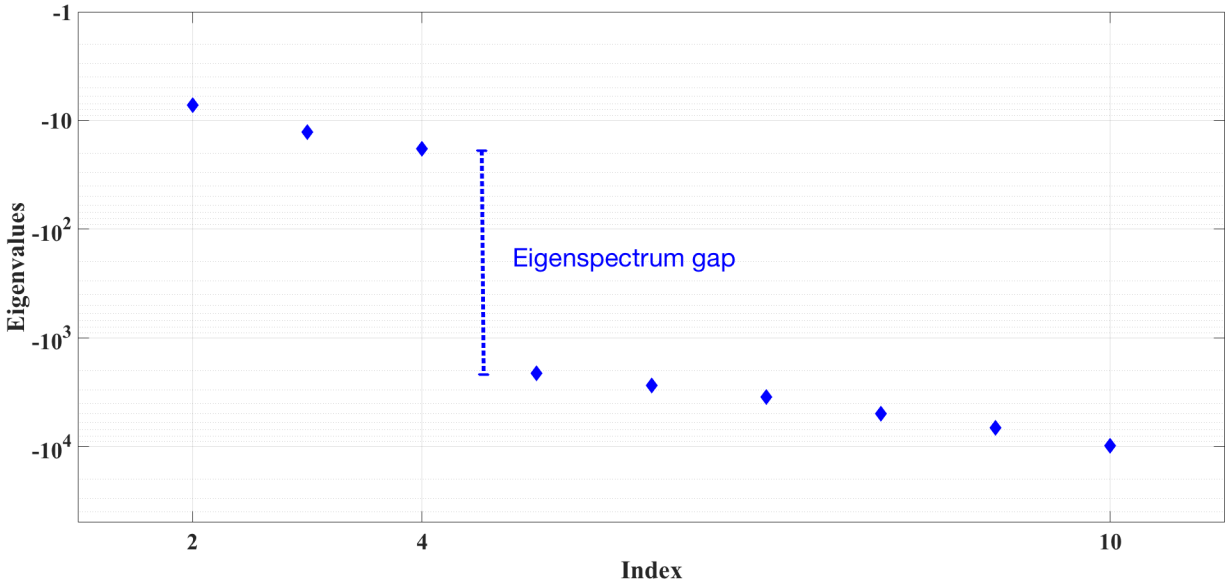

Figure 1.3: Eigenspectrum of a CME of the MM system $S_1 + S_2 \underset{5\times10^2}{\overset{10^3}{\rightleftharpoons}} S_3 \underset{2}{\overset{5}{\rightleftharpoons}} S_2 + S_4$ with initial state $\begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$. The real part of the eigenvalues of the corresponding matrix $\mathbf{D}$ are plotted in order of increasing magnitude. The zero eigenvalue does not appear on this log scale. The system operates on two distinct timescales as given by the eigenspectrum gap.

resent fast timescales. The gap between the two groups indicates the degree of time-scale separation. The larger this degree of separation is, the more challenging it will be to ana-

lyze or simulate the system directly. In contrast, the larger this gap, the more accurate will be a model reduction based on time-scale separation. Fast timescales are only reflected in initial behaviors over short times. After these initial transients, the CME can be approximated in terms of the slow time-scale behavior. Timescale separation approaches neglect fast transients to provide efficient analysis of system behavior over slower timescales. This thesis is devoted to such timescale separation approximations for the CME.

One of the widely accepted methods to approximate the chemical master equation is the Finite State Projection (FSP) algorithm developed by Munsky and Khammash [24]. The original algorithm presented in the paper is not a computationally efficient approach. Several improvements have appeared in the last decade [25]. A major improvement was achieved by implementing the method as a time-stepping algorithm [26, 27]. Alternatively, efficiency improvements were achieved by preconditioning the FSP algorithm [28, 25]. We follow a combined approach of multi-step FSP and preconditioning by dimensionality reduction using timescale separation.

Contribution of this thesis towards dynamical analysis of CME are follows:

1. Efficient generation of the components of a finite dimensional CME: State Space and Transition Matrix.

2. An algorithm for generating a stable partial eigenbasis of a matrix.

3. A structured formulation of initial condition for the reduced model proposed by Roussel and Zhu [4].

4. A generalized algorithm for the dimensionality reduction of a finite dimensional CME

using timescale separation approach without the requirement of widely separated timescales.

5. A generalized algorithm for the generation of reduced dimensional CME and a possible extension of generating a reduced network for systems with widely separated timescales.

6. An efficient algorithm for identifying the time-step for the multi-step FSP algorithm proposed by Burrage *et al.* [26].

The thesis is organized as follows:

Chapter 2 presents a derivation of the CME and reviews some of the standard methods to find the exact theoretical solution of a CME. Some commonly used approaches for approximating the solution for the CME are also discussed.

Chapter 3 presents a computational algorithm to generate the state space and transition matrix of a CME from the network description.

Chapter 4 presents algorithms for the dimensionality reduction of a CME. It begins with an exact reduced initial condition for the reduced CME proposed by Roussel and Zhu [4]. Next, a set of transformations is presented that facilitates reduction of a CME to a lower dimensional system of ODEs which has solutions that approximate the original. In particular, appropriate choice of transformation gives rise to probability conservation in the reduced system's solution. In addition, as a special case, for systems with significantly large separation in timescales, a reduction algorithm that does not rely on eigenbasis is also presented.

Chapter 5 presents a preconditioned FSP algorithm. For a large or even infinite dimensional CME, multi-step FSP is a standard approach to approximate the CME's solution. However, the projected finite dimensional CME often has large dimension for a reasonable time-step. In addition, using the eigenbasis transformation reduction algorithm in Chapter 4, an optimal time step generation is presented. To resolve these difficulties, a preconditioned FSP algorithm exploiting the reduction techniques from Chapter 4 is presented.

Chapter 6 concludes with discussions and future directions.

Chapter 2

Chemical Master Equation

In the 1960s, Oppenheim *et al.* studied the relationship between stochastic and deterministic models for simple systems [29]. This work was later extended to complex systems by Kurtz [30]. In 1967, McQuarrie investigated stochastic models of chemically reacting systems [31] which laid principles for the development of the mathematical model known as the Chemical Master Equation (CME). In 1992, Gillespie gave a rigorous derivation for the CME [21]. A derivation of the CME is presented here.

Consider a biochemical reaction network in thermal equilibrium consisting of k chemical species $S_1, S_2, \ldots, S_k$ involved in h reactions $R_1, R_2, \ldots, R_h$ in a well-stirred reaction vessel of fixed volume. Each reaction has the form

$$R_i : \sum_{j=1}^{k} a_{j,i} S_j \xrightarrow{r_i} \sum_{j=1}^{k} b_{j,i} S_j \tag{2.1}$$

where $a_{j,i}$ and $b_{j,i} \in \mathbb{Z}_{\geq 0}$ are the stoichiometric coefficients that indicate the number of

molecules of chemical species S_j consumed or produced in reaction R_i. Each reaction R_i is characterized by (i) a stochastic reaction rate constant $r_i \in \mathbb{R}_+$, and (ii) a stoichiometry vector

$$\mathbf{S}_i = \begin{bmatrix} b_{1,i} - a_{1,i} & b_{2,i} - a_{2,i} & \dots & b_{k,i} - a_{k,i} \end{bmatrix}^T. \tag{2.2}$$

The system behavior can be described by an update rule. To begin, we define the following:

1. A state of the system is described by a k-dimensional vector. By indexing all possible states of the system, a state is given by

$$\mathbf{N}_j = \begin{bmatrix} N_{j,1} & N_{j,2} & \dots & N_{j,k} \end{bmatrix}^T. \tag{2.3}$$

   where $N_{j,i}$ represents the number of molecules of species S_i in the j^{th} state. The state space $\mathbf{N}$ is defined as the matrix with j^{th} column equal to $\mathbf{N}_j$.

2. Let $\Delta t > 0$ be sufficiently small such that at most one reaction occurs in the time interval $[t, t + \Delta t]$.

3. Let $P(\mathbf{N}_j, t)$ be the probability that the system is in state $\mathbf{N}_j$ at time t. This probability is conditional on an initial state distribution $\{P(\mathbf{N}_j, 0)\}$ $(\forall j)$; following common convention, we suppress the conditional dependence.

4. The propensity of a reaction is the probability of its occurrence in unit time. Let $D_i(\mathbf{N}_j)$ be the propensity of reaction R_i when the system is in state $\mathbf{N}_j$.

Considering the state as a random variable, biochemical systems can be modeled as a stochastic process. For a biochemical system, future time evolution of the state variable only depends upon the present state of the system. Such models are called memory lacking or Markovian [10, 32] process.

Assume the probability distribution, $\{P(\mathbf{N}_j, t)\}$ $(\forall j)$, is known at some time t. Then to formulate the probability distribution at time $t + \Delta t$, consider all events that could lead to the state being $\mathbf{N}_j$ at time $t + \Delta t$. Consider two cases:

1. **No reaction occurs during the time interval** $[t, t + \Delta t]$: In this case, the system is in state $\mathbf{N}_j$ at time $t + \Delta t$ only if the system is in state $\mathbf{N}_j$ at time t. The probability of reaction R_i firing in the interval $[t, t + \Delta t]$ given the system is in state $\mathbf{N}_j$ at time t is $D_i(\mathbf{N}_j)\Delta t$. Then the probability of *any* reactions firing in the time interval $[t, t + \Delta t]$ given the system is in state $\mathbf{N}_j$ at time t is $\sum_{i=1}^{h} D_i(\mathbf{N}_j)\Delta t$. Then using probability conservation,

$$1 - \sum_{i=1}^{h} D_i(\mathbf{N}_j)\Delta t$$

   is the probability of no reaction firing during the time interval $[t, t + \Delta t]$.

2. **One reaction occurs during the time interval** $[t, t + \Delta t]$: Suppose reaction R_i fires once in the interval $[t, t + \Delta t]$ and the system is in state $\mathbf{N}_j - \mathbf{S}_i$ at time t. Then the corresponding state transition would be $\mathbf{N}_j - \mathbf{S}_i \to \mathbf{N}_j$. The probability of

reaction R_i firing given that the system is in state $\mathbf{N}_j - \mathbf{S}_i$ at time t is

$$D_i(\mathbf{N}_j - \mathbf{S}_i)\Delta t.$$

Together, these two scenarios allow us to derive the probability of attaining state $\mathbf{N}_j$ at time $t + \Delta t$ as

$$P(\mathbf{N}_j, t + \Delta t) = P(\mathbf{N}_j, t)\left(1 - \sum_{i=1}^{h} D_i(\mathbf{N}_j)\Delta t\right) + \sum_{i=1}^{h} P(\mathbf{N}_j - \mathbf{S}_i, t)\, D_i(\mathbf{N}_j - \mathbf{S}_i)\Delta t.$$

where the first term describes the probability of the system being in state $\mathbf{N}_j$ at time t and no reaction occurring in the interval $[t, t + \Delta t]$ and the second term describes the probability of the system being in states $\mathbf{N}_j - \mathbf{S}_i$ $(i = 1, 2, \ldots, h)$ and a corresponding reaction R_i occurring. Note that

$$P(\mathbf{N}_j, t + \Delta t) = P(\mathbf{N}_j, t) - \sum_{i=1}^{h} P(\mathbf{N}_j, t)\, D_i(\mathbf{N}_j)\Delta t + \sum_{i=1}^{h} P(\mathbf{N}_j - \mathbf{S}_i, t) D_i(\mathbf{N}_j - \mathbf{S}_i)\Delta t.$$

Subtracting $P(\mathbf{N}_j, t)$ on both sides and dividing throughout by Δt gives

$$\frac{P(\mathbf{N}_j, t + \Delta t) - P(\mathbf{N}_j, t)}{\Delta t} = \sum_{i=1}^{h} P(\mathbf{N}_j - \mathbf{S}_i, t)\, D_i(\mathbf{N}_j - \mathbf{S}_i) - P(\mathbf{N}_j, t)\, D_i(\mathbf{N}_j).$$

Taking the limit $\Delta t \to 0$ gives the CME:

$$\frac{d}{dt}P(\mathbf{N}_j, t) = \sum_{i=1}^{h} P(\mathbf{N}_j - \mathbf{S}_i, t)\, D_i(\mathbf{N}_j - \mathbf{S}_i) - P(\mathbf{N}_j, t)\, D_i(\mathbf{N}_j). \tag{2.4}$$

Considering all possible states in the system, the CME can be expressed in matrix form as

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t). \tag{2.5}$$

where $\mathbf{P}(t)$ is a vector with j^{th} component $P(\mathbf{N}_j, t)$, potentially infinite dimensional, and $\mathbf{D}$ is a corresponding transition operator. In the case where n, the number of possible states in the system, is finite, $\mathbf{P}(t)$ is a vector of length n and $\mathbf{D}$ is the *transition matrix* of size $n \times n$.

2.1 Exact Solution of the CME

A standard way of expressing the solution to a finite dimensional CME is given by

$$\mathbf{P}(t) = \exp(\mathbf{D}\, t)\mathbf{P}(0) \qquad\qquad \forall t \geq 0 \tag{2.6}$$

where the matrix exponential is

$$\exp(\mathbf{D}\, t) = \sum_{k=1}^{\infty} \frac{(\mathbf{D}\, t)^k}{k!}. \tag{2.7}$$

Alternatively using eigenvectors,

$$\exp(\mathbf{D}\, t) = \mathbf{R} \exp(\mathbf{J}\, t)\mathbf{R}^{-1} \tag{2.8}$$

where $\mathbf{J}$ is the Jordan block matrix with eigenvalues on the diagonal and $\mathbf{R}$ is an eigenmatrix with n linearly independent (generalized) eigenvectors. Then the solution of a finite dimensional CME (2.5) can be expressed as

$$\mathbf{P}(t) = \mathbf{R}\, \exp(\mathbf{J}\, t)\, \mathbf{R}^{-1}\mathbf{P}(0) \qquad \forall t \geq 0 \tag{2.9}$$

2.2 Approximation to the Solution of the CME

For most systems of interest, the CME has an enormously large dimension and exhibits a wide range of timescales. This create problems for computation of the exact solution even though techniques for numerical computation of the matrix exponential [33, 34, 35] and algorithms, such as ARPACK, for computing eigen-pairs of large sparse matrix [36] are available. Furthermore, for an infinite dimensional CME, things are even worse. One common approach for the characterization of probability distribution (solution of CME) is using sample paths which could be generated by algorithms such as Gillespie's Stochastic Simulation Algorithm (SSA) [37]. However, due to wide range of timescales in most biochemical systems, computation of sample paths is very slow and thus SSA is an inefficient approach. Algorithms such as tau-leaping [7, 8, 9], Slow-scale SSA [38], and the hybrid Slow Scale Tau-leaping Method [39] can be implemented to improve the efficiency of sample path generation, but these approaches are still insufficient to provide useful approximations of the CME for many systems of interest.

One widely accepted approach for approximating the CME is the Finite State Projec-

tion (FSP) algorithm [24]. The FSP algorithm truncates the CME[1] to a lower dimensional CME that is sufficient for capturing transitions between a subset of states that are probable in a given time interval $(0, t_f)$. However, the original FSP algorithm is not an efficient approach to solve the CME for a long time interval due to the enormous number of states that needs to be considered in the truncated CME. To resolve this problem, alternative approaches were introduced such as variable time-stepping algorithm [26], and preconditioning of the truncated transition matrix using timescale separation [28] and aggregation [40]. We introduced a combination of these three methods of aggregation, timescale separation, and multi-time stepping FSP to approximate the transient solution of the CME.

Using a wide gap in timescales, a reduction approach has been implemented for approximating solutions for the CME by Peleš *et al.* [28]. The algorithm is explained as follows: Consider the CME

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t)\,, \qquad \mathbf{P}(0) = \mathbf{P}_0\,, \qquad \mathbf{P} \in \mathbb{R}^n \tag{2.10}$$

where the eigenvalues of $\mathbf{D}$ has the property:

$$0 = \boxed{\Re(\lambda_1) \geq \Re(\lambda_2) \geq \cdots \geq \Re(\lambda_d)} \gg \boxed{\Re(\lambda_{d+1}) \geq \cdots \geq \Re(\lambda_n)} \quad \text{for some } d < n$$

[1] A detailed description of the algorithm is given in the Chapter 5.

Next, decompose[2] the transition matrix into two matrices as

$$\mathbf{D} = \mathbf{F} + \Delta \tag{2.11}$$

where $\mathbf{F}$ and Δ are chosen such that $\mathbf{F}$ has eigenvalues corresponding to fast timescales and Δ has eigenvalues corresponding to slow timescales. In addition, $\mathbf{F}$ is a block diagonal matrix with non-zero blocks on the diagonal which can be partitioned as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{F}_d \end{bmatrix} \tag{2.12}$$

These type of blocks for the transition matrix of a CME means the state transitions are absent between sets of states corresponding to each block. Each block is a representation of an independent system each of which may exhibit a steady state of their own represented by the null space of each block $\mathbf{F}_i$. In contrast, the original system, represented by $\mathbf{D}$, has transitions between the states of the blocks $\mathbf{F}_i$s. The transitions between these blocks are represented by Δ.

The algorithm requires a specific left ($\mathbf{L}$) and right ($\mathbf{R}$) null matrix of $\mathbf{F}$ which is

[2] A detailed description of the decomposition is given in Section 4.2.2

generated using left and right null vectors of each blocks independently:

$$\mathbf{F}_i\mathbf{R}_i = \mathbf{0} \qquad \text{and} \qquad \mathbf{F}_i^T\mathbf{L}_i = \mathbf{0} \qquad \forall i \in \{1, 2, \ldots, d\} \tag{2.13}$$

Next, combining $\mathbf{L}_i$s and $\mathbf{R}_i$s to form $\mathbf{L}$ and $\mathbf{R}$ as

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{L}_d \end{bmatrix} \qquad \text{and} \qquad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{R}_d \end{bmatrix} \tag{2.14}$$

where each $\mathbf{R}_i$ is a non-negative vector because they are the non-negative steady state solutions of the CMEs with transition matrix $\mathbf{F}_i$. In addition, $\mathbf{R}_i$s are scaled such that column sum is equal to 1. For each left null vector $\mathbf{L}_i$, each component is equal to 1.

Then

$$\tilde{\mathbf{P}}(t) = \mathbf{R} \exp\left(\mathbf{L}^T \Delta\mathbf{R}\, t\right) \mathbf{L}^T \mathbf{P}(0) \tag{2.15}$$

approximates the solution $\mathbf{P}$ as

$$\left\| \mathbf{P}(t) - \tilde{\mathbf{P}}(t) \right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad \forall t \geq \frac{1}{\Re(\lambda_{d+1})} \ln(\epsilon) \tag{2.16}$$

Note that (2.15) is in fact the solution of a reduced system:

$$\dot{\mathbf{X}}(t) = \left(\mathbf{L}^T \Delta\mathbf{R}\right) \mathbf{X}(t)\,, \qquad \mathbf{X}(0) = \mathbf{X}_0 = \mathbf{L}^T \mathbf{P}(0) \tag{2.17}$$

where $\mathbf{L}^T\Delta\mathbf{R}$ has properties of the transition matrix of a CME.

In timescale separation approaches, as implemented in [28], one common assumption is that timescales are well separated as shown in Figure 1.3 (reproduced here in Figure 2.1). However, such a gap may vanish as the number of molecules increases. For example, increasing the number of molecules of species S_1 and S_2 in the Michaelis-Menten (MM) system from 3 to 60 has CME with eigenspectrum as shown in Figure 2.1(B). Lack of a gap in the eigenspectrum makes it impossible to generate a useful decomposition into $\mathbf{F}$ and Δ. Around the same time that the FSP was published, Roussel and Zhu published an algorithm for dimensionality reduction of the CME using a timescale separation approach [4] without the assumption of a well separated eigenspectrum. However, the algorithm was lacking an exact initial condition for the reduced CME and hence was not directly useful. In this thesis, we solved this problem by generating an exact initial condition for the reduced system using a specific set of aggregations (eigenbasis projections -not necessarily using eigenvectors). We also developed a dimensionality reduction algorithm following Roussel and Zhu using the idea of eigenvector projection of [28]. In addition we also present an alternate algorithm for generating a reduced CME using a projection generated from the blocks of the transition matrix. These algorithms are presented in Chapter 4.

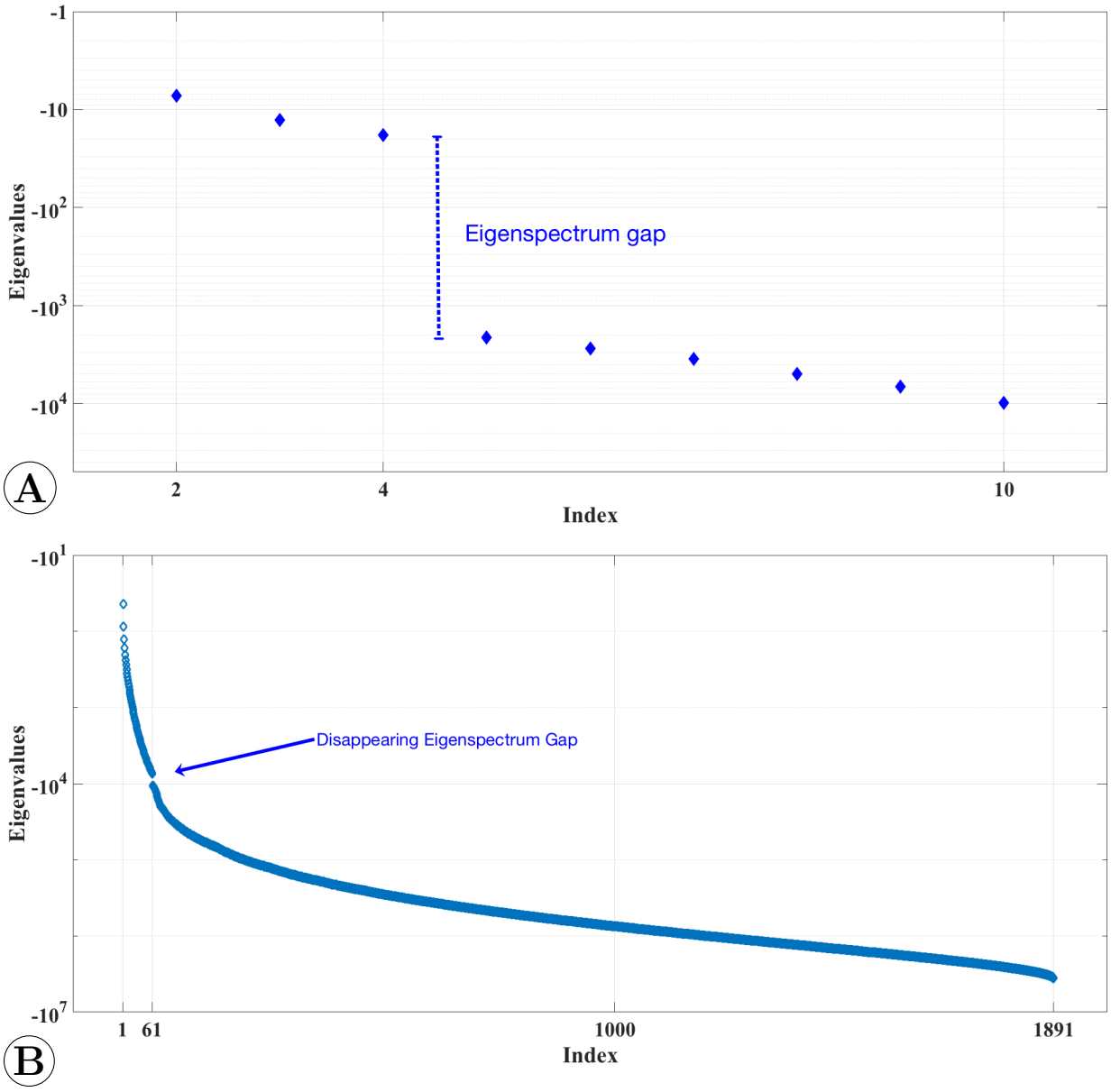

Figure 2.1: Eigenspectrum of a CME of the Michaelis-Menten system $S_1 + S_2 \underset{5\times10^2}{\overset{10^3}{\rightleftharpoons}} S_3 \overset{5}{\underset{2}{\rightleftharpoons}} S_2 + S_4$ with two different initial states $\begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix}$ is equal to (A) $\begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$ and (B) $\begin{bmatrix} 60 & 60 & 0 & 0 \end{bmatrix}$. The real part of the eigenvalues of the corresponding matrix $\mathbf{D}$ are plotted in order of increasing magnitude. The zero eigenvalue does not appear on this log scale.

Chapter 3

Algorithm for Generation of the State Space and Transition Matrix of of a CME

For simple networks with small molecule numbers, the state space and transition matrix for a Chemical Master Equation (CME) can be produced using pen and paper. As the dimension increases this process is inefficient. A general algorithm for this task is presented by Kan *et al.* [41]. However, for use in the Finite State Projection (FSP) algorithm, which is an iterative algorithm, this approach may be insufficient because a new CME is generated in each iteration. This chapter presents a novel efficient algorithm to solve this problem.

The straightforward approach presented by Kan *et al.* [41] is given in Algorithm 1. The algorithm accepts stoichiometry vectors $\{\mathbf{S}_i\}$ $(i = 1, 2, \ldots, h)$ and stochastic reaction

Algorithm 1: An algorithm for the CME [41]

Input:

$\{\mathbf{S}_i\} \leftarrow$ Stoichiometry vectors $\in \mathbb{Z}^{k\times 1}$ and $i = 1, 2, \ldots, h$;

$\{c_i\} \leftarrow$ Stochastic reaction rates $\in \mathbb{R}_{+}^{1\times h}$ and $i = 1, 2, \ldots, h$;

$\mathbf{N}_1 \leftarrow$ Initial state vector $\in \mathbb{Z}_{\geq 0}^{k\times 1}$;

begin

- $\mathbf{N} \leftarrow \mathbf{N}_1$;
- accessible state vector size, $n \leftarrow 1$;
- current state index, $j \leftarrow 1$;
- **while** $j \leq n$ **do**
  - Current reaction, $r \leftarrow 1$;
  - **while** $r \leq h$ **do**
    - Check if current reaction, R_r reacts from current state $\mathbf{N}_j$;
    - **if** *True* **then**
      - Target state, $\mathbf{N}_{\mathrm{k}} \leftarrow \mathbf{N}_{\mathrm{j}} + \mathbf{S}_r$;
      - check if target state $\mathbf{N}_{\mathrm{k}}$ is already in $\mathbf{N}$;
      - **if** *True* **then**
        - get the index i of the state in $\mathbf{N}$ that is equal to $\mathbf{N}_{\mathrm{k}}$;
      - **else**
        - add $\mathbf{N}_{\mathrm{k}}$ to $\mathbf{N}$;
        - $n \leftarrow n + 1$;
        - $i \leftarrow n$;
      - $h_r(\mathbf{N}_j) \leftarrow$ # of possible combinations reaction r can happen from $\mathbf{N}_j$;
      - $\mathbf{D}_{i,\mathrm{j}} \leftarrow c_r\, h_r(\mathbf{N}_j)$;
    - $r \leftarrow r + 1$;
  - $j \leftarrow j + 1$;
- update diagonal entries $\mathbf{D}_{jj} = -\sum_{i=1, i\neq j}^{n} \mathbf{D}_{ij}$;

Output:

$\mathbf{D} \leftarrow$ Transition matrix $\in \mathbb{R}^{n\times n}$;

$\mathbf{N} \leftarrow$ State space $\in \mathbb{Z}_{\geq 0}^{k\times n}$;

rates $\{\mathbf{c}_i\}$ $(i = 1, 2, \ldots, h)$ as the representation of the chemical reaction network. In addition, a single state vector $\mathbf{N}_1$ is needed to characterize the conservations (and also act as an initial seed for iterative construction of the state space). The algorithm iteratively identifies the complete state space. Each iteration starts with an incomplete state space (initially $\mathbf{N}_1$), called the source states, stored as a matrix. In each iteration, the algorithm generates new states (so called *target states*) that are reached through firing of each of the h reactions from the source states: $\{\mathbf{N}_{target} = \mathbf{N}_{source} + \mathbf{S}_i\}$ $(i = 1, 2, \ldots, h)$. Note that, some of the reactions are infeasible from some source states. For each feasible target state, a propensity is assigned to the corresponding reaction. The algorithm terminates when an iteration produces no new states.

The computational cost of this algorithm is mostly spent on finding the index for the assignment of the reaction propensity elements in the transition matrix. This is costly because the expanding state space $\mathbf{N}$ and the reaction propensity values are generated concurrently. Alternately, if we first generated the state space $\mathbf{N}$ alone, we could find all state transitions efficiently using a vectorized approach: $\{\mathbf{N}_{target}\} = \mathbf{N} + \{\mathbf{S}_i\}_{\times n}$ where $\{\mathbf{S}_i\}_{\times n}$ is a set of n column vectors with each column equal to $\mathbf{S}_i$. A similar approach can be applied to generate all reaction propensity values in the transition matrix. Furthermore, additional efficiency can be achieved by avoiding the iterative checking of the existence of target states in the state space.

We developed an approach for upfront state space generation based on moiety conservations. We then use a vectorized computation to generate the reaction propensity elements of the transition matrix. Optimized algorithms for the state space and the transition matrix of a CME are presented in the following sections. A comparison of the computational

time with Algorithm 1 is presented later in this chapter.

3.1 State Space

To generate the state space of a CME, we make use of the semi-positive moiety conservation laws as presented in the paper [42][1]. The moiety conservation laws of a system can be written in matrix form as

$$\mathbf{C}\mathbf{N}_j = \mathbf{b} \qquad \forall j \in \{1, 2, \ldots, n\} \tag{3.1}$$

where $\mathbf{C}$ represents the conservation law's coefficient matrix, $\mathbf{N}_j$ is a state, and $\mathbf{b}$ is constant vector of moiety totals. Note that, for $\mathbf{C}$, $\mathbf{N}_j$, and $\mathbf{b}$, all components are non-negative integers. The set of *all non-negative integer solutions of the linear system* 3.1 is the state space $\mathbf{N}$ of the CME. A novel method for finding the state space is presented as Algorithm 2.

The algorithm separates the species into dependent and independent species following conservation laws 3.1. Then the algorithm identifies feasible state components corresponding to the independent species. The corresponding dependent species components are then generated.

We introduce the algorithm through an example in the next section.

[1] A MATLAB implementation (`sbioconsmoiety` function) for generating semi-positive moiety conservation is available in the SIMBIOLOGY toolbox.

Algorithm 2: Algorithm for finding all non negative integer solutions of the linear system $\mathbf{C}\mathbf{N}_j = \mathbf{b}$ where $\mathbf{C}$, $\mathbf{N}_j$, and $\mathbf{b} \geq 0$ component-wise.

Input:

$\mathbf{C} \leftarrow$ (Moiety conservation law) coefficient matrix $\in \mathbb{Z}_{\geq 0}^{r \times k}$ `// where` $r = \texttt{rank}(\mathbf{C})$

$\mathbf{b} \leftarrow \mathbf{C}\mathbf{N}_1 \in \mathbb{Z}_{+}^{r \times 1}$ `// where` $\mathbf{N}_1$ `is a state vector in the state space`

$\mathbf{U} =$ `Least integer upper bound of variables in the linear system`

for $i = 1, 2, \ldots, k$ **do**
  $l = 0$;
  for $j = 1, 2, \ldots, r$ **do**
    if $\mathbf{C}_{j,i} \neq 0$ **then**
      $x_l = \mathbf{b}_j / \mathbf{C}_{j,i}$; $l = l + 1$;
  $U_i = \texttt{floor}(\min\{\mathbf{x}\})$ `// where` $\mathbf{x} = \{x_1, x_2, \ldots, x_l\}$

$\mathbf{N} =$ `Non-negative integer solutions of the linear system`

$\mathbf{I} \leftarrow \begin{bmatrix} I_1 & I_2 & \ldots & I_k \end{bmatrix}$ such that components in $\mathbf{U}(\mathbf{I})$ are sorted in decreasing order.;
$\mathbf{U} \leftarrow \mathbf{U}(\mathbf{I})$; $\mathbf{C} \leftarrow \mathbf{C}(:, \mathbf{I})$ `// Re-indexing`
$\mathbf{C}_{RREF} \leftarrow \texttt{RREF}(\mathbf{C})$ `// row reduced echelon form of` $\mathbf{C}$
$\mathbf{K} \leftarrow \begin{bmatrix} K_1 & K_2 & \ldots & K_r \end{bmatrix}$ `// Index of pivotal elements in` $\mathbf{C}_{RREF}$
$\mathbf{J} \leftarrow \begin{bmatrix} J_1 & J_2 & \ldots & J_{k-r} \end{bmatrix}$ `// Index of non-pivotal elements in` $\mathbf{C}_{RREF}$
$\mathbf{C}_{pivot} \leftarrow \mathbf{C}(:, \mathbf{K})$; `// Pivotal columns in` $\mathbf{C}$
$\mathbf{C}_{non-pivot} \leftarrow \mathbf{C}(:, \mathbf{J})$; `// Non-pivotal columns in` $\mathbf{C}$
$\mathbf{N}_{non-pivot} \leftarrow \begin{bmatrix} 0 & 1 & \ldots & U(J_1) \end{bmatrix}$;
for *j = 2,3,...,k-r* **do**
  $\mathbf{N}_{non-pivot} \leftarrow \begin{bmatrix} \mathbf{N}_{non-pivot} & \mathbf{N}_{non-pivot} & \ldots & \mathbf{N}_{non-pivot} \\ \vec{\mathbf{0}} & \vec{\mathbf{1}} & \ldots & \overrightarrow{\mathbf{U}(J_j)} \end{bmatrix}$;
  Remove columns that satisfy $\mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot}(:, l) > \mathbf{b}$;
$n \leftarrow$ number of columns in $\mathbf{N}_{non-pivot}$;
$\mathbf{B} \leftarrow \begin{bmatrix} \mathbf{b} & \mathbf{b} & \ldots & \mathbf{b} \end{bmatrix}$; `// where` $\mathbf{B} \in \mathbb{Z}^{r \times n}$
$\mathbf{N}_{pivot} \leftarrow \mathbf{C}_{pivot}^{-1}(\mathbf{B} - \mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot})$;
$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{pivot} \\ \mathbf{N}_{non-pivot} \end{bmatrix}$;
Re-index $\mathbf{N}$, according to the original state indexing;

Output:

$\mathbf{N} \leftarrow \{\mathbf{N}_j\}_{j=1}^{n}$; `// State space of the system`

3.1.1 Example

We use the Michaelis-Menten reaction network to illustrate. The Michaelis-Menten (MM) mechanism is defined as follows:

$$\mathrm{S}_1 + \mathrm{S}_2 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} \mathrm{S}_3 \underset{c_4}{\overset{c_3}{\rightleftharpoons}} \mathrm{S}_2 + \mathrm{S}_4 \tag{3.2}$$

where S_is represents species involved in the reaction network and c_i represents the stochastic reaction rate constant of the reaction R_i. There are 4 species and 4 reaction channels in the network.

We chose a molecular population of each species in the MM mechanism at some time as $N_1 = N_2 = N_3 = N_4 = 1$ where N_i represents the number of molecules of the species S_i. This gives a state, say $\mathbf{N}_1$, as

$$\mathbf{N}_1 = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.3}$$

The corresponding moiety conservation laws of the MM mechanism are

$$N_1 + N_3 + N_4 = 3 \tag{3.4}$$

$$N_2 + N_3 = 2. \tag{3.5}$$

Equivalently, in matrix form,

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{3.6}$$

where $\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ are the inputs of the Algorithm 2 in line 1 and 2.

We start with finding the least integer upper bound $\mathbf{U}$ of each species' molecular population (lines 3-8). We chose species N_1 and N_3 as examples to show the procedure. For species N_1, we set up two (number of rows) linear equations using the first column (corresponding to species S_1) in $\mathbf{C}$ and $\mathbf{b}$ as

$$1 \cdot U_1 = 3 \qquad \text{and} \qquad 0 \cdot U_1 = 2 \tag{3.7}$$

Canceling the inconsistent equation $0 \cdot U_1 = 2$ gives $U_1 = 3$. Similarly for species S_3, linear equations are

$$1 \cdot U_3 = 3 \qquad \text{and} \qquad 1 \cdot U_3 = 2 \tag{3.8}$$

which has two different solutions ($U_3 = 3$ and $U_3 = 2$). Taking the minimum of these two solutions provides the upper bound of species N_3. In this example, the `floor` function is not required due to the presence of integer solutions. Following the same procedures on

other species the least integer upper bound of each species' molecular population is given by

$$\mathbf{U} = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 2 & 3 \end{bmatrix} \qquad \text{(line 8)} \qquad (3.9)$$

Next find an index, $\mathbf{I}$, such that elements in $\mathbf{U}$ are sorted in descending order.

$$\mathbf{I} = \begin{bmatrix} 1 & 4 & 2 & 3 \end{bmatrix} \implies \mathbf{U}(\mathbf{I}) = \begin{bmatrix} 3 & 3 & 2 & 2 \end{bmatrix} \qquad \text{(line 9)} \qquad (3.10)$$

Re-index $\mathbf{U}$ and $\mathbf{C}$ column-wise using the index $\mathbf{I}$.

$$\mathbf{U} = \begin{bmatrix} U_1 & U_4 & U_2 & U_3 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 2 & 2 \end{bmatrix} \qquad (3.11)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \text{(line 10)} \qquad (3.12)$$

Let $\mathbf{K}$ and $\mathbf{J}$ be the pivotal and non-pivotal indices of species respectively found using the Reduced Row-Echelon Form (RREF) of the matrix $\mathbf{C}$. Since $\mathbf{C}$ is already in Row Reduced Echelon Form (RREF),

$$\mathbf{K} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \end{bmatrix} \qquad \text{(line 12)} \qquad (3.13)$$

$$\mathbf{J} = \begin{bmatrix} J_1 & J_2 \end{bmatrix} = \begin{bmatrix} 2 & 4 \end{bmatrix} \qquad \text{(line 13)} \qquad (3.14)$$

Partition $\mathbf{C}$ using the columns corresponding to the pivotal and non-pivotal indices:

$$\mathbf{C}_{pivot} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{(line 14)} \qquad (3.15)$$

$$\mathbf{C}_{non-pivot} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \text{(line 15)} \qquad (3.16)$$

Define $\mathbf{N}_{non-pivot} = \begin{bmatrix} 0 & 1 & \dots & \mathbf{U}(J_1) \end{bmatrix}$

$$\mathbf{N}_{non-pivot} = \begin{bmatrix} 0 & 1 & 2 & 3 \end{bmatrix} \qquad \text{(line 16)} \qquad (3.17)$$

Extend $\mathbf{N}_{non-pivot}$ using $\mathbf{U}(J_j) = 2$ as follows

$$\mathbf{N}_{non-pivot} = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix} \qquad \text{(line 18)} \qquad (3.18)$$

Next we determine which columns satisfy $\mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot} \leq \mathbf{b}$

$$\mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix} \qquad (3.19)$$

$$= \begin{bmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix} \qquad (3.20)$$

Eliminating columns that do not satisfy $\mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot} \leq \mathbf{b}$ gives

$$\mathbf{N}_{non-pivot} = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \end{bmatrix} \quad \text{(line 19)} \quad (3.21)$$

Recall that the state space is the non-negative integer solution of the linear system $\mathbf{CN} = \begin{bmatrix} \mathbf{b} & \mathbf{b} & \dots & \mathbf{b} \end{bmatrix}$ where the state space $\mathbf{N}$ can be partitioned as

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{pivot} \\ \mathbf{N}_{non-pivot} \end{bmatrix} \quad (3.22)$$

where $\mathbf{N}_{non-pivot}$, in the state space is already identified. To find $\mathbf{N}_{pivot}$,

$$\mathbf{CN} = \mathbf{B} \quad (3.23)$$

$$\implies \quad \mathbf{C}_{pivot}\mathbf{N}_{pivot} + \mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot} = \mathbf{B} \quad (3.24)$$

$$\implies \quad \mathbf{C}_{pivot}\mathbf{N}_{pivot} = \mathbf{B} - \mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot} \quad (3.25)$$

$$\implies \quad \mathbf{N}_{pivot} = \mathbf{C}_{pivot}^{-1}(\mathbf{B} - \mathbf{C}_{non-pivot}\mathbf{N}_{non-pivot}) \quad (3.26)$$

In this case,

$$\begin{aligned}
\mathbf{N}_{pivot} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \end{bmatrix} \right) \\
&= \begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \end{bmatrix} \\
&= \begin{bmatrix} 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad \text{(line 22)} \qquad (3.27)
\end{aligned}$$

Then combining 3.21 and 3.27, we have

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{pivot} \\ \mathbf{N}_{non-pivot} \end{bmatrix} = \begin{bmatrix} N_1 \\ N_2 \\ N_4 \\ N_3 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \end{bmatrix} \qquad \text{(line 23)} \qquad (3.28)$$

Note that, due to the re-indexing using $\mathbf{I}$ and partitioning of $\mathbf{C}$ as pivotal and non-pivotal parts, the row-index of the state space is changed to $\begin{bmatrix} 1 & 2 & 4 & 3 \end{bmatrix}$ where the first two components represents the pivotal species and second two components represents the non-

pivotal species. So, re-indexing the state space back to original indexing gives

$$\mathbf{N} = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \end{bmatrix} \qquad \text{(line 24)} \qquad (3.29)$$

satisfying eq. (3.6). In the next section, we present an algorithm to determine the transition matrix of a CME using the state space, $\mathbf{N}$,

3.2 Transition Matrix

An element $\mathbf{D}_{j,k}$ of the transition matrix $\mathbf{D}$ represents the total propensity for all reactions that transition from the state $\mathbf{N}_k$ to state $\mathbf{N}_j$. Generation of the transition matrix is presented in Algorithm 3. As a first step, we catalog the state transitions corresponding to all reactions. Next, reaction propensities for all such transitions are calculated. Finally, each reaction propensity is assigned to the transition matrix using the indices of the corresponding state transition. We will continue with the MM network to illustrate the algorithm.

3.2.1 Example

For the MM mechanism (3.2), the inputs to the Algorithm 3 are

Algorithm 3: Algorithm for finding the transition matrix of a CME

Input:

$\mathbf{N} \leftarrow \{\mathbf{N}_j\}_{j=1}^{n}$: State space of the system $\in \mathbb{Z}_{\geq 0}^{k \times n}$;

$\{\mathbf{S}_i\}_{i=1}^{h} \leftarrow$ set of Stoichiometry vectors $\in \mathbb{Z}^{k \times h}$;

$\{\mathbf{R}_i\}_{i=1}^{h} \leftarrow$ set of Reactant-Stoichiometry vectors $\in \mathbb{Z}_{\geq 0}^{k \times h}$;

$\mathbf{c} \leftarrow$ Reaction rate vector;

$\mathbf{I} =$ `Index of transition states in the state space`

for i = *1,2,...,* h **do**

  $\mathbf{S} = \begin{bmatrix} \mathbf{S}_i & \mathbf{S}_i & \dots & \mathbf{S}_i \end{bmatrix}$; `// where` $\mathbf{S} \in \mathbb{Z}^{k \times n}$

  $\hat{\mathbf{N}} = \mathbf{N} + \mathbf{S}$; `// transition state matrix due to reaction` R_i

  $\mathbf{I}(i,:) = \begin{bmatrix} I_1 & I_2 & \dots & I_n \end{bmatrix}$ such that j^{th} transition state $\hat{\mathbf{N}}(:,j) = \mathbf{N}(:,I_j)$;

    `// where` $I_j = 0$ `if` $\hat{\mathbf{N}}(:,j) \notin \mathbf{N}$

$\hat{\mathbf{D}} =$ `Reaction propensities of each transitions`

for i = *1,2,...,* h **do**

  $n_{r,i} \leftarrow$ number of reactant species in the reaction R_i;

  $\mathbf{J} = \begin{bmatrix} J_1 & J_2 & \dots & J_{n_{r,i}} \end{bmatrix}$; `// Index of reactant species in the reaction` R_i

  for j = *1,2,...,* n **do**

    $\mathbf{C}_{i,j} = \prod_{r=1}^{n_r} \texttt{nchoosek}(\mathbf{N}_{J_r,j}, \mathbf{R}_{J_r,i})$; `// where` $\mathbf{C} \in \mathbb{Z}_{\geq 0}^{h \times n}$

  $\hat{\mathbf{D}} = \text{diag}(\mathbf{c})\mathbf{C}$; `// where` $\hat{\mathbf{D}} \in \mathbb{Z}_{\geq 0}^{h \times n}$

$\mathbf{D} =$ `Transition matrix of the CME`

$\mathbf{D} = \mathbf{0}_{n \times n}$;

for i = *1,2,...,* h **do**

  for v = *1,2,...,* n **do**

    $D = \hat{\mathbf{D}}_{i,v}$;

    $u = \mathbf{I}_{i,v}$;

    if $D \neq 0$ *and* $u \neq 0$ **then**

      $\mathbf{D}_{u,v} = D$;

for j = *1,2,...,* h **do**

  $\mathbf{D}_{j,j} = -\sum_{i=1}^{h} \prod_{i=1}^{n_{r,i}} \mathbf{N}$

Output:

$\mathbf{D} \leftarrow$ Transition matrix of the CME $\in \mathbb{R}^{n \times n}$;

1. State space

$$\mathbf{N} = \begin{bmatrix} 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \end{bmatrix} \qquad \text{(line 1)} \qquad (3.30)$$

2. Stoichiometry vectors

$$\begin{matrix} R_1 & R_2 & R_3 & R_4 \\ \downarrow & \downarrow & \downarrow & \downarrow \end{matrix}$$

$$\{\mathbf{S}_i\} = \left\{ \begin{bmatrix} -1 \\ -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 1 \\ -1 \end{bmatrix} \right\} \begin{matrix} \leftarrow \text{S} \\ \leftarrow \text{E} \\ \leftarrow \text{C} \\ \leftarrow \text{P} \end{matrix} \qquad \text{(line 2)} \qquad (3.31)$$

3. Reactant-Stoichiometry vectors: These are versions of the stoichiometry vectors in

which only reactant species appear and the signs are flipped.

$$\begin{matrix} & R_1 & R_2 & R_3 & R_4 & \\ & \downarrow & \downarrow & \downarrow & \downarrow & \\ \{\mathbf{R}_i\} = \left\{ \right. & \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \left. \right\} & \begin{matrix} \leftarrow \text{S} \\ \leftarrow \text{E} \\ \leftarrow \text{C} \\ \leftarrow \text{P} \end{matrix} \end{matrix} \qquad \text{(line 3)} \tag{3.33}$$

4. Stochastic reactions rates

$$\mathbf{c} = \begin{bmatrix} 1000 & 500 & 5 & 2 \end{bmatrix} \qquad \text{(Line 4)} \tag{3.34}$$

We begin by finding the state transitions corresponding to each reaction in turn. For the first reaction R_1, we construct an n dimensional array by repeating the stoichiometry vector corresponding to reaction R_1 as

$$\mathbf{S} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \text{(line 6)} \tag{3.35}$$

Then all potential target states reached through reaction R_1 are given by

$$\hat{\mathbf{N}} = \mathbf{N} + \mathbf{S} = \begin{bmatrix} 2 & 1 & 0 & -1 & 1 & 0 & -1 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 \end{bmatrix} \qquad \text{(line 7)} \qquad (3.36)$$

Next, for each feasible target state, we identify its index in the state space $\mathbf{N}$, recorded as follows. Let $\mathbf{I}$ be a matrix with element $\mathbf{I}_{j,k}$ representing the index of the j^{th} column of $\hat{\mathbf{N}}$ in $\mathbf{N}$ reached through reaction R_k. If a column of $\hat{\mathbf{N}}$ is not in $\mathbf{N}$, then we set the corresponding value to 0. Then for the first iteration, the first row of $\mathbf{I}$ is given by

$$\mathbf{I}_1 = \begin{bmatrix} 5 & 6 & 7 & 0 & 8 & 9 & 0 & 0 & 0 \end{bmatrix} \qquad \text{(line 8)} \qquad (3.37)$$

Repeating for each reaction, the index matrix $\mathbf{I}$ is generated as

$$\mathbf{I} = \begin{bmatrix} 5 & 6 & 7 & 0 & 8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 \\ 0 & 0 & 0 & 0 & 2 & 3 & 4 & 6 & 7 \\ 0 & 5 & 6 & 7 & 0 & 8 & 9 & 0 & 0 \end{bmatrix} \qquad \text{(line 8)} \qquad (3.38)$$

Calculation of reaction propensities

We next use collision theory [21] to identify the propensity of a reaction. For reactions in which the stoichiometric coefficients for the reactant species are all 1 (a typical case),

the reaction propensity of $R_i : \sum_{j=1}^{k} S_j \xrightarrow{c_i} \sum_{j=1}^{l} b_{j,i} P_j$, following mass action law, can be calculated as

$$c_i \prod_{j=1}^{k} N_j \tag{3.39}$$

where N_j is the number of molecules of species S_j available. In the case where multiple copies of a reactant appear, we need to carefully address the number of possible collision combinations.

The number of possible collision combinations that can lead to reaction R_i from state state $\mathbf{N}_j$ is given by

$$\mathbf{C}_{i,j} = \prod_{r=1}^{n_{r,i}} \binom{\mathbf{N}_{r,j}}{\mathbf{R}_{r,i}} = \prod_{r=1}^{n_{r_i}} \frac{\mathbf{N}_{r,j}!}{(\mathbf{R}_{r,i}!)(\mathbf{N}_{r,j} - \mathbf{R}_{r,i})!} \tag{3.40}$$

where $\mathbf{R}_{r,i}$ is the r^{th} non-zero component in the Reactant-Stoichiometry vector $\mathbf{R}_i$, $\mathbf{N}_{r,j}$ is the component in the state vector $\mathbf{N}_j$ corresponding to the r^{th} reactant species, and $n_{r,i}$ is the number of reactant species. As an example, $\mathbf{C}_{1,3}$ is calculated as follows:

The Reactant-Stoichiometry vector, $\mathbf{R}_1$ is

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T \tag{3.41}$$

and state $\mathbf{N}_3$ is

$$\mathbf{N}_3 = \begin{bmatrix} 1 & 2 & 0 & 2 \end{bmatrix}^T \tag{3.42}$$

The indices of the reactant species (non-zero elements in $\mathbf{R}_1$) are

$$\mathbf{J} = \begin{bmatrix} 1 & 2 \end{bmatrix} \tag{3.43}$$

Then

$$\mathbf{N}_{\mathbf{J},3} = \begin{bmatrix} 1 & 2 \end{bmatrix} \qquad \text{and} \qquad \mathbf{R}_{\mathbf{J},1} = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{3.44}$$

Then

$$\mathbf{C}_{1,3} = \prod_{r=1}^{2} \binom{\mathbf{N}_{\mathbf{J},3}(r)}{\mathbf{R}_{\mathbf{J},1}(r)} = \binom{1}{1} \times \binom{2}{1} = 2$$

Considering each reaction and each state, collision combinations matrix $\mathbf{C}$ is

$$\mathbf{C} = \begin{bmatrix} 6 & 4 & 2 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 2 & 4 & 6 & 0 & 1 & 2 & 0 & 0 \end{bmatrix} \qquad \text{(line 13)} \tag{3.45}$$

We then determine the reaction propensity matrix $\hat{\mathbf{D}}$ using the stochastic reaction rates $\mathbf{c}$:

$$\hat{\mathbf{D}} = \text{diag}(\mathbf{c})\mathbf{C} \tag{3.46}$$

$$= \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 6 & 4 & 2 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 2 & 4 & 6 & 0 & 1 & 2 & 0 & 0 \end{bmatrix} \tag{3.47}$$

$$= \begin{bmatrix} 6000 & 4000 & 2000 & 0 & 2000 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 500 & 500 & 500 & 1000 & 1000 \\ 0 & 0 & 0 & 0 & 5 & 5 & 5 & 10 & 10 \\ 0 & 4 & 8 & 12 & 0 & 2 & 4 & 0 & 0 \end{bmatrix} \quad \text{(line 14)} \tag{3.48}$$

Next we use the indices collected in matrix $\mathbf{I}$ (3.38) to populate the off-diagonal entries

of $\mathbf{D}$.

$$\mathbf{D} = \begin{bmatrix} * & 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 5 & 500 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 5 & 500 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 5 & 0 & 0 \\ 6000 & 4 & 0 & 0 & * & 0 & 0 & 1000 & 0 \\ 0 & 4000 & 8 & 0 & 0 & * & 0 & 10 & 1000 \\ 0 & 0 & 2000 & 12 & 0 & 0 & * & 0 & 10 \\ 0 & 0 & 0 & 0 & 2000 & 2 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 4 & 0 & * \end{bmatrix} \quad \text{(line 21)} \tag{3.49}$$

Finally, as mentioned before, the diagonal element in the j^{th} column is negative the column sum of j^{th} column of $\mathbf{D}$ excluding the diagonal element,

$$\mathbf{D}_{j,j} = -\sum_{i=1, i \neq j}^{n} \mathbf{D}_{i,j} \tag{3.50}$$

In conclusion, for the MM Mechanism, the transition matrix $\mathbf{D}$ is

$$\mathbf{D} = \begin{bmatrix} -6000 & 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & -4004 & 0 & 0 & 5 & 500 & 0 & 0 & 0 \\ 0 & 0 & -2008 & 0 & 0 & 5 & 500 & 0 & 0 \\ 0 & 0 & 0 & -12 & 0 & 0 & 5 & 0 & 0 \\ 6000 & 4 & 0 & 0 & -2505 & 0 & 0 & 1000 & 0 \\ 0 & 4000 & 8 & 0 & 0 & -1507 & 0 & 10 & 1000 \\ 0 & 0 & 2000 & 12 & 0 & 0 & -509 & 0 & 10 \\ 0 & 0 & 0 & 0 & 2000 & 2 & 0 & -1010 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 4 & 0 & -1010 \end{bmatrix} \quad \text{(line 23)} \tag{3.51}$$

3.3 Computational Efficiency

Dimension (n)	Algorithm 2 & 3	Algorithm 1
66	0.05	0.02
5151	0.08	2.5
20301	0.2	28
45451	0.4	358
125751	1.1	3526

Table 3.1: Time, in seconds, taken to generate the CME using both methods in MATLAB for different dimensions of the CME of the MM mechanism. Computer configuration: Windows, Intel i5-6300U CPU, 2.4 GHz and 8GB RAM.

The computational time required to generate the state space and transition matrix in

MATLAB, of Algorithm 2 and 3 is compared to Algorithm 1 in Table 3.1.

Improvements were achieved in the Algorithm 2 and 3 by consolidating most of the steps into functions and vectorizing iterative steps. MATLAB scripts are available in the Github repository https://github.com/midhunks/Chemical-Master-Equation.

Chapter 4

Dimensionality Reduction and Approximation of the Chemical Master Equation

A finite (n) dimensional Chemical Master Equation (CME), with initial condition $\mathbf{P}(0)$ can be written in matrix form as

$$\dot{\mathbf{P}}(t) = \mathbf{D}\,\mathbf{P}(t)\,, \qquad \mathbf{P}(0) = \mathbf{P}_0\,, \qquad \mathbf{P}(t) \in \mathbb{R}^n. \tag{4.1}$$

Because the CME is a linear system, its solution can be expressed as

$$\mathbf{P}(t) = \mathbf{R}\,\exp(\mathbf{J}t)\,\mathbf{a} \qquad \forall t \geq 0 \tag{4.2}$$

where $\mathbf{J}$ is the Jordan matrix with eigenvalue λ_i on the i^{th} diagonal, $\mathbf{R}$ is an eigenmatrix with n linearly independent (generalized) eigenvector $\mathbf{R}_i$ at the column index i, and $\mathbf{a} = \mathbf{R}^{-1}\,\mathbf{P}_0$.

If $\mathbf{R}$ is composed of n *linearly independent eigenvectors*, i.e.

$$\mathbf{D}\,\mathbf{R} = \mathbf{R}\,\Lambda, \tag{4.3}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, then the solution (4.2) has the form

$$\mathbf{P}(t) = \mathbf{R}\,\exp(\Lambda t)\,\mathbf{a} \qquad\qquad \forall t \geq 0 \tag{4.4}$$

which can be expressed in summation form as

$$\mathbf{P}(t) = \sum_{i=1}^{n} a_i\,\exp(\lambda_i t)\,\mathbf{R}_i \qquad\qquad \forall t \geq 0 \tag{4.5}$$

For a CME, the real part of all eigenvalues ($\Re(\lambda)$) are non-positive [43]. Furthermore, assume the eigenvalues satisfy:

$$0 \geq \boxed{\Re(\lambda_1) \geq \ldots \geq \Re(\lambda_d)} > \boxed{\Re(\lambda_{d+1}) \geq \ldots \geq \Re(\lambda_n)}, \quad \text{for some } d \in \{1, 2, \ldots, n{-}1\} \tag{4.6}$$

Writing the solution (4.5) as

$$\mathbf{P}(t) = \sum_{i=1}^{d} a_i\,\exp(\lambda_i t)\,\mathbf{R}_i + \sum_{i=d+1}^{n} a_i\,\exp(\lambda_i t)\,\mathbf{R}_i \qquad\qquad \forall t \geq 0, \tag{4.7}$$

the condition (4.6) implies that, as t increases , $\exp(\lambda_j t)$ decays more rapidly for $j \in \{d+1, \ldots, n\}$ compared to $j \in \{1, \ldots, d\}$, and consequently

$$\mathbf{P}(t) \approx \tilde{\mathbf{P}}(t) = \sum_{i=1}^{d} a_i \, \exp(\lambda_i t) \, \mathbf{R}_i \qquad \forall \text{ sufficiently large } t \tag{4.8}$$

This thesis is devoted to exploiting this observation to arrive at an accurate approximation (past an initial transient) to the full system behavior from simulations of a reduced d-dimensional system. A theoretical formulation of this idea is presented in Theorem 1 using the notations in InfoBox 4.1 and assumptions in InfoBox 4.2.

Theorem 1. *Choose $d \in \{1, 2, \ldots, n\}$ such that the assumption in InfoBox 4.2 is true. Consider the solution $\mathbf{P}(t)$ of an n-dimensional linear initial value problem*

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t)\,, \qquad \mathbf{P}(0) = \mathbf{P}_0\,, \qquad \mathbf{P}(t) \in \mathbb{R}^n \tag{4.11}$$

in partitioned form as

$$\mathbf{P}(t) = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \exp\left(\begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} t \right) \begin{bmatrix} \mathbf{A}_d \\ \mathbf{A}_m \end{bmatrix} \tag{4.12}$$

where $\begin{bmatrix} \mathbf{A}_d \\ \mathbf{A}_m \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix}^{-1} \mathbf{P}(0).$

InfoBox 4.1: **Notations**

Let $\mathbf{D} \in \mathbb{R}^{n\times n}$ be a matrix with right and left eigenrelations

$$\mathbf{DR} = \mathbf{R}\Lambda\,, \qquad\qquad \mathbf{D}^T\mathbf{L} = \mathbf{L}\Lambda \tag{4.9a}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Also, let $\mathbf{T} \in \mathbb{R}^{n\times n}$ be an invertible matrix. Define

$$\hat{\mathbf{D}} = \mathbf{TDT}^{-1}\,, \qquad \hat{\mathbf{R}} = \mathbf{TR}\,, \qquad \hat{\mathbf{L}} = \mathbf{T}^{-T}\mathbf{L}. \tag{4.9b}$$

Partition $\mathbf{D}$, $\mathbf{L}$, $\mathbf{R}$, Λ, $\mathbf{T}$, $\hat{\mathbf{D}}$, $\hat{\mathbf{L}}$, and $\hat{\mathbf{R}}$ as

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{dd} & \mathbf{D}_{dm} \\ \mathbf{D}_{md} & \mathbf{D}_{mm} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_{dd} & \mathbf{L}_{dm} \\ \mathbf{L}_{md} & \mathbf{L}_{mm} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix}$$
$$\hat{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix}, \quad \hat{\mathbf{L}} = \begin{bmatrix} \hat{\mathbf{L}}_{dd} & \hat{\mathbf{L}}_{dm} \\ \hat{\mathbf{L}}_{md} & \hat{\mathbf{L}}_{mm} \end{bmatrix}, \quad \hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{T}_{dd} & \mathbf{T}_{dm} \\ \mathbf{T}_{md} & \mathbf{T}_{mm} \end{bmatrix} \tag{4.9c}$$

where the sub-indices of each block in the partitioned matrices indicate the dimension of the sub-matrix and $d + m = n$.

InfoBox 4.2: **Core Assumption**

Assume $\mathbf{D} \in \mathbb{R}^{n\times n}$ has linearly independent eigenvectors and eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ satisfying

$$\boxed{0 \geq \Re(\lambda_1) \geq \cdots \geq \Re(\lambda_d)} \; > \; \boxed{\Re(\lambda_{d+1}) \geq \cdots \geq \Re(\lambda_n)}, \quad d \in \{1, 2, \ldots, n-1\} \tag{4.10}$$

Follow notations in InfoBox 4.1 and assume the matrix $\hat{\mathbf{D}}$ and $\mathbf{T}$ are indexed such that the block matrices $\mathbf{T}_{dd}$ and $\hat{\mathbf{R}}_{dd}$ are invertible.

Define

$$\tilde{\mathbf{P}}(t) = \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \exp\left(\Lambda_{dd}\, t\right) \mathbf{A}_d. \tag{4.13}$$

Then for any given $\epsilon > 0$, *with* $t_\epsilon = \frac{1}{\Re(\lambda_{d+1})} \ln(\epsilon)$,

$$\left\| \mathbf{P}(t) - \tilde{\mathbf{P}}(t) \right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad \forall t \geq t_\epsilon. \tag{4.14}$$

The proof is given in the appendix (page 107).

In the work of Roussel and Zhu [4], a reduced system is achieved using timescale separation following (4.8). Although the procedure of Roussel and Zhu results in a reduced system which can generate a valid approximation as in (4.17) below, it was not directly useful because the reduced initial condition that corresponds to a particular solution of the original system could not be systematically identified. (Roussel and Zhu did provide a method to estimate the reduced initial condition by using knowledge of the biochemical system. They employed a linear programming approach where the objective function relied on the system's behavior during the initial transient which is, in general, hard to identify.) We propose a systematic way of identifying the reduced initial condition. The complete reduction algorithm including a novel error bound is presented in the following theorem.

Theorem 2. *Choose* $d \in \{1, 2, \ldots, n-1\}$ *such that the assumptions in InfoBox 4.2 hold.*

Consider an n-dimensional initial value problem

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t)\,, \qquad\qquad \mathbf{P}(0) = \mathbf{P}_0 \tag{4.15}$$

Then consider the d-dimensional initial value problem[1]

$$\dot{\mathbf{X}}(t) = \left(\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}\right)\mathbf{X}(t)\,, \quad \mathbf{X}(0) = \hat{\mathbf{R}}_{dd}\left(\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \hat{\mathbf{L}}_{md}\end{bmatrix}^T\begin{bmatrix}\hat{\mathbf{R}}_{dd}\\ \hat{\mathbf{R}}_{md}\end{bmatrix}\right)^{-1}\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \hat{\mathbf{L}}_{md}\end{bmatrix}^T\mathbf{T}\mathbf{P}_0 \tag{4.16}$$

Then for any ϵ $(0 < \epsilon < 1)$*, with* $t_\epsilon = \frac{1}{\Re(\lambda_{d+1})}\ln(\epsilon)$*,*

$$\left\|\mathbf{P}(t) - \mathbf{T}^{-1}\begin{bmatrix}\mathbf{I}_{dd}\\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}\end{bmatrix}\mathbf{X}(t)\right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad\qquad \forall t \geq t_\epsilon \tag{4.17}$$

The proof is given in the Appendix (page 113).

A pseudo-code implementation of the approximation in this theorem is presented in Algorithm 4.

The approximate n-dimensional solution of the CME from Theorem 2 satisfies probability conservation as stated below.

[1]Note, $\mathbf{T}$, $\mathbf{L}$, and $\mathbf{R}$ are invertible (full rank n). Then the sub-matrices of transformed eigenvector matrices $\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \hat{\mathbf{L}}_{md}\end{bmatrix}$ and $\begin{bmatrix}\hat{\mathbf{R}}_{dd}\\ \hat{\mathbf{R}}_{md}\end{bmatrix}$ has full rank d. Then the $d \times d$ matrix $\left(\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \hat{\mathbf{L}}_{md}\end{bmatrix}^T\begin{bmatrix}\hat{\mathbf{R}}_{dd}\\ \hat{\mathbf{R}}_{md}\end{bmatrix}\right)$ is invertible

Algorithm 4: Reduction Algorithm

Input:

$\mathbf{D} \leftarrow$ Transition matrix $\in \mathbb{R}^{n\times n}$;

$\mathbf{P}_0 \leftarrow$ Initial condition $\in \mathbb{R}^n$;

$d \leftarrow$ A number $\in \{1, 2, \ldots, n-1\}$;

$\epsilon \leftarrow$ error tolerance greater than zero;

begin

$\mathbf{T}_{nd} \leftarrow$ Left eigenbasis of $\mathbf{D}$ corresponding to d slow eigenvalues;

$\mathbf{I} \leftarrow$ An index such that top d columns of $\mathbf{T}_{nd}$ is an invertible matrix.;

$\mathbf{I}_{rev} \leftarrow$ an index to re-index the solution back to the original indexing.;

Re-Index $\mathbf{D}$, $\mathbf{P}_0$, $\mathbf{T}_{nd}$ using $\mathbf{I}$.;

$\mathbf{T} \leftarrow$ An invertible transformation matrix $\in \mathbb{R}^{m\times n}$ where the first d rows are equal to $\mathbf{T}_{nd}^T$;

$\mathbf{M}_{nd} \leftarrow$ Right eigenbasis of $\mathbf{D}$ corresponding to d slow eigenvalues;

$\mathbf{M}_{nd} \leftarrow$ Transpose of $\texttt{RREF}((\mathbf{TM})^T)$: Updated corresponding to the transformed matrix $\hat{\mathbf{D}} = \mathbf{TDT}^{-1}$;

$\mathbf{Q}_{dd} \leftarrow \hat{\mathbf{D}}_{dn}\mathbf{M}_{nd}$: Reduced Transition matrix;

$\mathbf{X}_0 \leftarrow$ Reduced initial condition. Varies with transformation used.;

$\mathbf{X}(t) \leftarrow$ Reduced solution of the system $\dot{\mathbf{X}}(t) = \mathbf{Q}_{dd}\mathbf{X}(t)\,, \mathbf{X}(0) = \mathbf{X}_0$;

$\tilde{\mathbf{P}}(t) \leftarrow \mathbf{T}^{-1}\mathbf{M}_{nd}\mathbf{X}(t)$;

Output:

$\tilde{\mathbf{P}}(t) \leftarrow$ Approximate solution of the CME $\dot{\mathbf{P}}(t) = \mathbf{DP}(t)\,, \mathbf{P}(0) = \mathbf{P}_0$ for times greater than $t_\epsilon = \frac{1}{\Re(\lambda_{d+1})}\log(\epsilon)$ with error bound $\mathcal{O}(\epsilon)$.

Corollary 1. *Suppose the hypothesis of Theorem 2 holds and* $\mathbf{1}_n^T\mathbf{P}(t) = \mathbf{1}_n^T\mathbf{P}_0\,, \forall t$. *Then*

$$\mathbf{1}_n^T\tilde{\mathbf{P}}(t) = \mathbf{1}_n^T\mathbf{P}_0 \quad \forall t \tag{4.18}$$

where $\tilde{\mathbf{P}}(t) = \mathbf{T}^{-1}\begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix}\mathbf{X}(t)$.

The proof is given in Appendix (page 117)

We next consider three cases for the transformation $\mathbf{T}$.

4.1 No Transformation: $\mathbf{T} = \mathbf{I}$

In the work of Roussel and Zhu [4], the reduction in Theorem 2 is developed for $\mathbf{T} = \mathbf{I}$, i.e. $\hat{\mathbf{D}} = \mathbf{D}$, $\hat{\mathbf{R}} = \mathbf{R}$, and $\hat{\mathbf{L}} = \mathbf{L}$. In Theorem 2 we extended the formulation of the reduced system with a systematic generation of the initial condition as (4.16) and we provided an upper bound for the error in the approximation as (4.17). An implementation of this algorithm will be presented in Section 4.3.

4.2 Transformation to Achieve Probability Conservation in the Reduced System

The CME describes the time evolution of a probability distribution. In particular, the solutions satisfy conservation of probability over time. By Corollary 1, the approximate solution $\tilde{\mathbf{P}}(t) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t)$ (eq (4.17)) also satisfy probability conservation. However, the reduced d dimensional system (4.16) constructed using $\mathbf{T} = \mathbf{I}$ has solutions that do not satisfy any conservation. Therefore, although it is useful for generating approximate solutions to a CME, this reduced system can't be interpreted directly as describing the evolution of a probability distribution. However, under appropriate choice of transformation $\mathbf{T}$, solutions of the reduced system satisfy a conservation, as described next.

Corollary 2. *Suppose the hypothesis of Theorem 2 is true . Let* $\mathbf{T}$ *be an invertible matrix such that each column sum of the first* d *rows is equal to* 1*, i.e.*

$$\mathbf{1}_d^T \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{dm} \end{bmatrix} \mathbf{T} = \mathbf{1}_n^T \tag{4.19}$$

Consider the reduced differential equation (4.16)

$$\dot{\mathbf{X}}(t) = \left(\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm} \hat{\mathbf{R}}_{md} \hat{\mathbf{R}}_{dd}^{-1} \right) \mathbf{X}(t). \tag{4.20}$$

Then all solutions of this system satisfy

$$\mathbf{1}_d^T \mathbf{X}(t) = \mathbf{1}_n^T \mathbf{P}_0 \qquad \forall t \tag{4.21}$$

The proof is given in the Appendix (page 119)

In the context of a CME, the column sum of each solutions is 1 because of the probability conservation ($\mathbf{1}_n^T \mathbf{P}(t) = 1\,, \forall t$). Then, under any transformation $\mathbf{T}$ that satisfies the condition (4.19), the solution of the system (4.20) satisfies probability conservation. i.e.

$$\mathbf{1}_d^T \mathbf{X}(t) = \mathbf{1}_d^T \mathbf{X}(0) = 1 \tag{4.22}$$

To identify a transformation for the CME so that probability conservation is maintained, $\mathbf{T}$ should satisfy the condition $\mathbf{1}_d^T \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{dm} \end{bmatrix} \mathbf{T} = \mathbf{1}_d^T \mathbf{T}_{dn} = \mathbf{1}_n^T$ which only constrains the top d rows. To begin a simple construction, we can fix the top-left $d \times d$ block

of $\mathbf{T}$ as an identity matrix, which also satisfies the invertibility condition of $\mathbf{T}_{dd}$ in the assumptions InfoBox 4.2. Next, each column of the top-right block of $\mathbf{T}$ should have a sum of 1. Two such transformation that we will explore in this thesis are

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^{T} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \qquad \text{and} \qquad \mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{D}_{dm}\mathbf{D}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.23}$$

where the bottom block matrices are chosen to ensure the invertibility of $\mathbf{T}$. (The first transformation is generated from the left eigenvectors of $\mathbf{D}$ and the second comes from the transition matrix $\mathbf{D}$ itself). Lemmas 6 and 7 given in Appendix (page 123 and 125) confirm that the column sum of the top right block of each of these transformation matrices is equal to $\mathbf{1}_m^T$. We will make use of a modified form of the second transformation to exploit the eigenspectrum (as in Figure 2.1(A)) for improving the efficiency of the computation of the reduced system and approximation. This modification is an alternate implementation of the reduction presented by Peleš [28].

4.2.1 Transformation Using Left Eigenvectors

In addition to the probability conservation, the choice of

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^{T} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.24}$$

is particularly made to achieve the following properties :

$$\hat{\mathbf{D}}_{dm} = \mathbf{0}_{dm} \tag{4.25a}$$

$$\hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm} \tag{4.25b}$$

$$\hat{\mathbf{L}}_{md} = \mathbf{0}_{md} \tag{4.25c}$$

which are proved in Corollary 3 (page 121). Using these properties, the reduced d-dimensional system (4.16) simplifies to

$$\dot{\mathbf{X}}(t) = \hat{\mathbf{D}}_{dd}\mathbf{X}(t) \tag{4.26}$$

Note that, due to the structure of $\mathbf{T}$, the inverse of $\mathbf{T}$ can be easily computed by taking the negative of the top right block. i.e.

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^{T} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.27}$$

The initial condition associated with the reduced system (4.16) is simplified as follows:

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd}\left(\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \mathbf{0}_{md} \end{bmatrix}^{T}\begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix}\right)^{-1}\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \mathbf{0}_{md} \end{bmatrix}^{T}\mathbf{T}\mathbf{P}_0 \tag{4.28}$$

$$= \hat{\mathbf{R}}_{dd}\left(\hat{\mathbf{L}}_{dd}^{T}\hat{\mathbf{R}}_{dd}\right)^{-1}\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \mathbf{0}_{md} \end{bmatrix}^{T}\mathbf{T}\mathbf{P}_0 \tag{4.29}$$

Now, note that $\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \mathbf{0}_{md} \end{bmatrix}$ is full rank d. Then, because of zero block, $\hat{\mathbf{L}}_{dd}$ is invertible. Then

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd}\hat{\mathbf{R}}_{dd}^{-1}\hat{\mathbf{L}}_{dd}^{-T}\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \mathbf{0}_{md} \end{bmatrix}^T \mathbf{T}\mathbf{P}_0 \tag{4.30}$$

$$= \hat{\mathbf{L}}_{dd}^{-T}\begin{bmatrix} \hat{\mathbf{L}}_{dd}^T & \mathbf{0}_{md}^T \end{bmatrix} \mathbf{T}\mathbf{P}_0 \tag{4.31}$$

$$= \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{md}^T \end{bmatrix} \mathbf{T}\mathbf{P}_0 \tag{4.32}$$

$$= \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^T \end{bmatrix} \mathbf{P}_0 \tag{4.33}$$

Combining (4.26) and (4.32), we have the reduced system of the form

$$\dot{\mathbf{X}}(t) = \hat{\mathbf{D}}_{dd}\mathbf{X}(t)\,, \qquad \mathbf{X}(0) = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^T \end{bmatrix} \mathbf{P}_0 \tag{4.34}$$

which has solution $\mathbf{X}(t)$ that be can be used to approximate the solution $\mathbf{P}$ as given in Theorem 2:

$$\left\| \mathbf{P}(t) - \mathbf{T}^{-1}\begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{dd}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) \right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad \forall t \geq \frac{1}{\Re(\lambda_{d+1})}\ln(\epsilon) \tag{4.35}$$

4.2.2 Transformation Using Transition Matrix Components

In this section, we develop an alternate derivation of the timescale separation algorithm first presented in [28]. For systems with large separation in timescales, the eigenvalues

satisfy

$$\boxed{0 \geq \Re(\lambda_1) \geq \Re(\lambda_2) \geq \ldots \geq \Re(\lambda_d)} \gg \boxed{\Re(\lambda_{d+1}) \geq \Re(\lambda_{d+2}) \geq \ldots \geq \Re(\lambda_n)} \tag{4.36}$$

as shown in Figure 2.1(A).

Then setting $\frac{\Re(\lambda_d)}{\Re(\lambda_{d+1})} = \delta$, we have

$$\frac{\Re(\lambda_i)}{\Re(\lambda_{d+1})} \leq \delta \approx 0\,, \qquad \forall i \in 1, 2, \ldots d \tag{4.37}$$

Using this idea, we split the transition matrix $\mathbf{D}$ as

$$\mathbf{D} = \mathbf{F} + \Delta \tag{4.38}$$

where $\mathbf{F}$ and Δ are chosen [2] such that $\mathbf{F}$ has eigenvalues corresponding to fast timescales and Δ has eigenvalues corresponding to slow timescales. Also, $\mathbf{F}$ and Δ are chosen such that the eigenspectra of $\mathbf{F}$ and Δ each have d zero eigenvalues. In addition, $\mathbf{F}$ and Δ are proper CME's transition matrices. Such a splitting can be achieved through a simple iterative procedure as follows: Assume, the eigenspectrum gap is of order 10^m, i.e. , $\Re(\lambda_{d+1}) = \Re(\lambda_d) - c\,10^m$ for some constant $c < 10$. Then each iteration follows a truncation as

$$\mathbf{F} = 10^{k+i} \left\lfloor \frac{\mathbf{D}}{10^{k+i}} \right\rceil, \qquad i \in \{1, 2, \ldots, m\} \tag{4.39}$$

where $10^k \geq |\Re(\lambda_d)|$ and $\lfloor \cdot \rceil$ is the function for rounding a number to nearest integer.

[2] In the paper [28], $\mathbf{F}$ and Δ are represented using H and ϵV respectively.

The stopping criteria for this approach is when $\mathbf{F}$ and $\Delta = \mathbf{D} - \mathbf{F}$ achieves exactly d zero eigenvalues.

In partitioned form $\mathbf{F}$ can be written as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{dd} & \mathbf{F}_{dm} \\ \mathbf{F}_{md} & \mathbf{F}_{mm} \end{bmatrix} \tag{4.40}$$

Following paper [28], the matrix $\mathbf{F}$ has a block structure (2.12) and rank $n-d$. In addition, all d blocks contain 1 eigenvalue equal to zero and so they each have one row (and column) dependent on other rows (and column) in the block. Re-indexing the matrix $\mathbf{F}$ such that those d-dependent rows in each block are at the top d rows of $\mathbf{F}$ leads the bottom $n-d$ rows of $\mathbf{F}$ has rank $n-d$. In particular, due to row and column re-indexing, $\mathbf{F}_{mm}$ will be invertible. A similar structural argument on correspondingly row re-indexed eigenmatrices $\mathbf{R}$ and $\mathbf{L}$ will give top $d \times d$ blocks, $\mathbf{R}_{dd}$ and $\mathbf{L}_{dd}$ invertible and diagonal.

Under the assumption, $\delta \to 0$, we have

$$\mathbf{D} = \mathbf{F}. \tag{4.41}$$

Then the transformation using blocks of $\mathbf{D}$ can be written in terms of $\mathbf{F}$ as

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{D}_{dm}\mathbf{D}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.42}$$

Note that, due to the structure of $\mathbf{T}$, the inverse of $\mathbf{T}$ can be easily computed by taking

the negative of the top right block. i.e.

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{F}_{dm}\mathbf{F}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.43}$$

Let $\hat{\mathbf{F}} = \mathbf{T}\mathbf{F}\mathbf{T}^{-1}$. Then right and left eigenvectors of $\hat{\mathbf{F}}$ corresponding to the d-zero eigenvalues satisfy the conditions:

$$\begin{bmatrix} \hat{\mathbf{F}}_{dd} & \hat{\mathbf{F}}_{dm} \\ \hat{\mathbf{F}}_{md} & \hat{\mathbf{F}}_{mm} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{dd} \\ \mathbf{0}_{md} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \hat{\mathbf{F}}_{dd} & \hat{\mathbf{F}}_{dm} \\ \hat{\mathbf{F}}_{md} & \hat{\mathbf{F}}_{mm} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{dd} \\ \mathbf{0}_{md} \end{bmatrix} \tag{4.44}$$

Using these conditions on left and right null matrices, we can simplify the initial condition for the reduced system as follows:

We have the reduced initial condition (4.16) as

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd} \left(\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \mathbf{T}\mathbf{P}_0 \tag{4.45}$$

Using Lemma 8 (given in page 126), we have $\hat{\mathbf{R}}_{dd}$ and $\hat{\mathbf{L}}_{dd}$ are invertible and $\hat{\mathbf{L}}_{md} = \mathbf{0}_{md}$.

Then

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd}\left(\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \mathbf{0}_{md}\end{bmatrix}^T\begin{bmatrix}\hat{\mathbf{R}}_{dd}\\ \hat{\mathbf{R}}_{md}\end{bmatrix}\right)^{-1}\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \mathbf{0}_{md}\end{bmatrix}^T\mathbf{T}\mathbf{P}_0 \tag{4.46}$$

$$= \hat{\mathbf{R}}_{dd}\left(\hat{\mathbf{L}}_{dd}^T\hat{\mathbf{R}}_{dd}\right)^{-1}\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \mathbf{0}_{md}\end{bmatrix}^T\mathbf{T}\mathbf{P}_0 \tag{4.47}$$

$$= \hat{\mathbf{R}}_{dd}\hat{\mathbf{R}}_{dd}^{-1}\hat{\mathbf{L}}_{dd}^{-T}\begin{bmatrix}\hat{\mathbf{L}}_{dd}\\ \mathbf{0}_{md}\end{bmatrix}^T\mathbf{T}\mathbf{P}_0 \tag{4.48}$$

$$= \begin{bmatrix}\mathbf{I}_{dd}\\ \mathbf{0}_{md}\end{bmatrix}^T\mathbf{T}\mathbf{P}_0 \tag{4.49}$$

$$= \begin{bmatrix}\mathbf{I}_{dd} & \mathbf{0}_{md}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{dm}^{-1}\\ \mathbf{0}_{md} & \mathbf{I}_{mm}\end{bmatrix}\mathbf{P}_0 \tag{4.50}$$

$$= \begin{bmatrix}\mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{dm}^{-1}\end{bmatrix}\mathbf{P}_0 \tag{4.51}$$

Then the reduced system (4.16) in Theorem 2 can be written as

$$\dot{\mathbf{X}}(t) = \left(\hat{\mathbf{D}}_{dd} - \hat{\mathbf{D}}_{dm}\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}\right)\mathbf{X}(t)\,, \qquad \mathbf{X}(0) = \begin{bmatrix}\mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1}\end{bmatrix}\mathbf{P}_0, \tag{4.52}$$

which approximates the solution $\mathbf{P}$ as

$$\left\|\mathbf{P}(t) - \mathbf{T}^{-1}\begin{bmatrix}\mathbf{I}_{dd}\\ \hat{\mathbf{R}}_{dd}\hat{\mathbf{R}}_{dd}^{-1}\end{bmatrix}\mathbf{X}(t)\right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad \forall t \geq \frac{1}{\Re(\lambda_{d+1})}\ln(\epsilon) \tag{4.53}$$

Note that in this case the error is dictated not by the user, but by the size of the gap in the eigenspectrum due to the additional assumption $\delta = 0$.

4.3 Implementation of the Reduction Procedure

In theory, the result in Theorem 2 can be applied to any linear system with transition matrix which has linearly independent eigenvectors. In practice, when addressing systems for which n (and possibly also d) are large, identifying eigenvalues and eigenvectors to reduce the system is an ill conditioned process due to computational difficulties in generating accurate eigenvectors. There are a number of aspects of the reduction scheme that demand special care to avoid the accumulation of numerical errors. In this section we will consider the following:

1. Selection of the reduced dimension d.

2. Computation of initial condition and transition matrix of the reduced system.

4.3.1 Selection of the Reduced Dimension d

For a given error tolerance ϵ $(0 < \epsilon < 1)$, Theorem 2 provides an acceptable approximation on the order of ϵ for all

$$t \geq t_\epsilon = \frac{1}{\Re(\lambda_{d+1})} \ln(\epsilon) > 0 \tag{4.54}$$

In this condition, t_ϵ depends upon the error tolerance ϵ and choice of d. Then for a given error tolerance ϵ, we can aim to achieve a particular t_ϵ by appropriate choice of d. In

particular, if the relation

$$\Re(\lambda_{d+1}) \leq \frac{1}{t_\epsilon} \ln(\epsilon) \qquad \text{(note that } \Re(\lambda_{d+1}) < 0\text{)} \tag{4.55}$$

is satisfied, then the corresponding index of the eigenvalue $\Re(\lambda_{d+1})$ can be used to find the reduced dimension d for which the error is bounded by ϵ for all time $t \geq t_\epsilon$.

Even though we can identify a corresponding eigenvalue λ_{d+1} using (4.55), we cannot deduce the information about the reduced dimension d without computing all eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{d+1}$. Algorithms such as `eigs`, MATLAB's implementation of the ARPACK routines [44, 45], efficiently computes a partial set of eigenvalues with a *reasonable accuracy*. However, in our experience, the `eigs` function is found to be erroneous while computing a large set of eigenvalues when the matrix is high dimensional. To resolve this, we identify eigenvalues iteratively. In each step, two (3 or more in case of complex and repeated) eigenvalues near to the previously identified eigenvalues are identified.

4.3.2 Computation of the Initial Condition and the Transition Matrix of the Reduced System Using a Semi-Orthogonal Eigenbasis

Computation of the initial condition and the transition matrix for the reduced system requires identification of eigenvectors. However, due to round-off errors, eigenvectors of large dimensional matrices loose linear independence and are not good for inverse operations. To resolve this, we propose an alternative approach, based on a numerically stable eigenbasis,

as follows. Recall, the reduced initial condition from Theorem 2 is given by

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd} \left(\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \mathbf{T}\mathbf{P}_0. \tag{4.56}$$

Because $\hat{\mathbf{R}}_{dd}$ is invertible,

$$\begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{R}_{dd} \tag{4.57}$$

where the matrix $\begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix}$ is a (scaled) Semi-Orthogonal matrix [46]. A semi orthogonal matrix is defined as follows: Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a semi-orthogonal matrix with $m \neq n$. Then exactly one the following is true for $\mathbf{A}$.

$$\mathbf{A}^T\mathbf{A} = \mathbf{I} \qquad \text{or} \qquad \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{4.58}$$

which is a diagonal matrix in our case due to the scaling of each column to achieve an identity block in the matrix. The semi-orthogonality is achieved in terms of rows of an identity matrix. Because this is a basis for the eigenspace, we call this matrix a Semi-Orthogonal Eigenbasis.

Next, define $\mathbf{M}_{nd} = \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix}$ which is a Semi-Orthogonal (right) Eigenbasis. Then

$$\begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} = \hat{\mathbf{M}}_{nd}\hat{\mathbf{R}}_{dd} \tag{4.59}$$

Because $\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}$ is a full rank matrix, we can identify d linearly independent rows in the matrix. Let $\mathbf{U}_{dd}$ be a $d \times d$ block matrix formed with those d linearly independent rows. Then $\mathbf{U}_{dd}$ is an invertible matrix. Then define $\mathbf{T}_{nd} = \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix} \mathbf{U}_{dd}^{-1}$ which is a Semi-Orthogonal (left) Eigenbasis. Then

$$\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix} = \mathbf{T}_{nd}\mathbf{U}_{dd} \tag{4.60}$$

Then

$$\mathbf{X}(0) = \hat{\mathbf{R}}_{dd}\left(\left(\mathbf{T}_{nd}\mathbf{U}_{dd}\right)^T \mathbf{M}_{nd}\hat{\mathbf{R}}_{dd}\right)^{-1}\left(\mathbf{T}_{nd}\mathbf{U}_{dd}\right)^T \mathbf{P}_0 \tag{4.61}$$

$$= \hat{\mathbf{R}}_{dd}\left(\mathbf{U}_{dd}^T\mathbf{T}_{nd}^T\mathbf{M}_{nd}\hat{\mathbf{R}}_{dd}\right)^{-1}\mathbf{U}_{dd}^T\mathbf{T}_{nd}^T\mathbf{P}_0 \tag{4.62}$$

$$= \hat{\mathbf{R}}_{dd}\hat{\mathbf{R}}_{dd}^{-1}\left(\mathbf{T}_{nd}^T\mathbf{M}_{nd}\right)^{-1}\mathbf{U}_{dd}^{-T}\mathbf{U}_{dd}^T\mathbf{T}_{nd}^T\mathbf{P}_0 \tag{4.63}$$

$$\implies \qquad \mathbf{X}(0) = \left(\mathbf{T}_{nd}^T\mathbf{M}_{nd}\right)^{-1}\mathbf{T}_{nd}^T\mathbf{P}_0 \tag{4.64}$$

Recall, the transition matrix of the reduced system (4.16) in Theorem 2 is given by

$$\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \tag{4.65}$$

By factoring, we can write

$$\begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{R}_{md}\mathbf{R}_{dd}^{-1} \end{bmatrix} \tag{4.66}$$

Then using $\mathbf{M}_{nd} = \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix}$, we have

$$\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} = \begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \end{bmatrix} \mathbf{M}_{nd} \tag{4.67}$$

We have achieved an expression for the initial condition and transition matrix in terms of left and right Semi-Orthogonal Eigenbases $\mathbf{T}_{nd}$ and $\mathbf{M}_{nd}$ as opposed to eigenvectors $\hat{\mathbf{R}}$ and $\hat{\mathbf{L}}$. We next present an approach for accurate generation of a Semi-Orthogonal Eigenbasis.

Semi-Orthogonal Eigenbasis Decomposition

Consider a partial eigenvalue relation of a matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ with d linearly independent eigenvectors as

$$\mathbf{K}\mathbf{V}_{nd} = \mathbf{V}_{nd}\Lambda_{dd} \tag{4.68}$$

In partitioned form,

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dm} \\ \mathbf{K}_{md} & \mathbf{K}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{dd} \\ \mathbf{V}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{dd} \\ \mathbf{V}_{md} \end{bmatrix} \Lambda_{dd} \tag{4.69}$$

Suppose $\mathbf{V}_{dd}$ invertible. Then right multiplying with $\mathbf{V}_{dd}^{-1}$ gives

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dm} \\ \mathbf{K}_{md} & \mathbf{K}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{dd} \\ \mathbf{V}_{md} \end{bmatrix} \mathbf{V}_{dd}^{-1} = \begin{bmatrix} \mathbf{V}_{dd} \\ \mathbf{V}_{md} \end{bmatrix} \Lambda_{dd}\mathbf{V}_{dd}^{-1} \tag{4.70}$$

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dm} \\ \mathbf{K}_{md} & \mathbf{K}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{V}_{md}\mathbf{V}_{dd}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{dd}\Lambda_{dd}\mathbf{V}_{dd}^{-1} \\ \mathbf{V}_{md}\Lambda_{dd}\mathbf{V}_{dd}^{-1} \end{bmatrix} \tag{4.71}$$

Define $\mathbf{X}_{md} = \mathbf{V}_{md}\mathbf{V}_{dd}^{-1}$. i.e., $\mathbf{V}_{md} = \mathbf{X}_{md}\mathbf{V}_{dd}$. Then

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dm} \\ \mathbf{K}_{md} & \mathbf{K}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{X}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{dd}\Lambda_{dd}\mathbf{V}_{dd}^{-1} \\ \mathbf{X}_{md}\mathbf{V}_{dd}\Lambda_{dd}\mathbf{V}_{dd}^{-1} \end{bmatrix} \tag{4.72}$$

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dm} \\ \mathbf{K}_{md} & \mathbf{K}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{X}_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{X}_{md} \end{bmatrix} \mathbf{V}_{dd}\Lambda_{dd}\mathbf{V}_{dd}^{-1} \tag{4.73}$$

Here, $\begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{X}_{md} \end{bmatrix} \mathbf{V}_{dd}$ is an eigenmatrix and thus $\begin{bmatrix} \mathbf{I}_{dd} \\ \mathbf{X}_{md} \end{bmatrix}$ is a right Semi-Orthogonal Eigenbasis of $\mathbf{K}$. The same procedure applied to $\mathbf{K}^T$ generates a left Semi-Orthogonal Eigenbasis.

Algorithm for the Semi-Orthogonal Eigenbasis Decomposition

We use the standard block-power iteration [47] to get the eigenvectors corresponding to the largest d eigenvalues. In block-power iteration, the eigenvectors corresponding to the large eigenvalues are generated as

$$\mathbf{K}^n\mathbf{V}_0 \xrightarrow{n\to\infty} \mathbf{V}\Lambda^n \tag{4.74}$$

where $\mathbf{V}$ is the partial set of eigenvectors. In each iteration $\mathbf{V}$ is normalized to avoid overflow.

In our method, we seek an eigenbasis which is a scaled semi-orthogonal matrix with an identity block. Then, instead of normalizing in every step as in the block-power iteration, we use the classic Row Reduced Echelon Form (RREF) method in which we scaling pivot

elements to one after row reducing the whole matrix. The RREF procedure makes sure that the new block matrix generated in every iteration has an identity block. In addition, it also keeps each column vector linearly independent. Implementation of this eigenbasis generation approach is given in Algorithm 5.

Algorithm 5: Implementation of the Semi-Orthogonal Eigenbasis Decomposition

Input:
- $\mathbf{K} \leftarrow$ A matrix $\in \mathbb{R}^{n \times n}$;
- $d \leftarrow$ A number $\in \{1, 2, \ldots, n-1\}$;
- $\delta \leftarrow$ tolerance greater than machine epsilon;
- $\mathbf{I} \leftarrow$ (optional) Index of d pivotal elements;

begin
- $\mathbf{V}_{old} \leftarrow$ A zero matrix of size $n \times d$;
- $\mathbf{V}_{new} \leftarrow$ A non-zero initial seed matrix of size $n \times d$;
- **while** $\|\mathbf{V}_{new} - \mathbf{V}_{old}\|_\infty > \delta$ **do Power iteration**
  - $\mathbf{V}_{old} \leftarrow \mathbf{V}_{new}$;
  - **for** $j = 1, 2, \ldots, d$ **do**
    - **if** $\mathbf{I}$ *is not given* **then**
      - $\mathbf{I}(j) \leftarrow$ Index of the absolute maximum component in the j^{th} row of $\mathbf{V}_{new}^T$;
    - $\mathbf{V}_{new}^T \leftarrow$ pivoting (without scaling) of $\mathbf{V}_{new}^T$ using the $\mathbf{I}(j)^{th}$ element;
  - $\mathbf{V}_{new} \leftarrow \mathbf{K}\mathbf{V}_{new}$;
- $\mathbf{V}_{new} \leftarrow$ Scale each column such that pivot elements are equal to 1.

Output:
- $\mathbf{V}_{new} \leftarrow$ Desired basis matrix $\in \mathbb{R}^{n \times d}$ with tolerance δ;
- $\mathbf{I} \leftarrow$ Index of d pivotal elements;

The seed vectors for the algorithm can be chosen as a set of random linearly independent vectors. Another possibility for the seeds is the use of eigenvectors computed using `eigs` which will leads to fast convergence. However, these may be linearly dependent vectors due to the round-off errors. In our experience, we can resolve this by introducing random

noise in the seed eigenvectors.

Implementation of the Semi-Orthogonal Eigenbasis in the CME's Reduction

Algorithm 5 generates an eigenbasis corresponding to the largest d eigenvalues. So, the basis generated is not directly useful for the reduction defined in terms of eigenvectors corresponding to the smallest eigenvalues. Recall that in Section 4.3.1, for the identification of the reduced dimension, we shifted the matrix $\mathbf{D}$ such that shifted eigenvalues satisfy the condition:

$$\boxed{\Re(\lambda_1) \geq \Re(\lambda_2) \cdots \geq \Re(\lambda_d)} > \boxed{\Re(\lambda_{d+1}) \geq \Re(\lambda_{d+2}) \cdots \geq \Re(\lambda_n)} \geq 0 \tag{4.75}$$

Then Algorithm 5 generates the eigenbasis of $\mathbf{D} - \Re(\lambda_n)\mathbf{I}_{nn}$ corresponding to small eigenvalues of $\mathbf{D}$.

Next, we explain the reduction approach using an example for all transformations discussed so far.

4.4 Example: Michaelis-Menten model

To illustrate the algorithm and results for all transformations, we use a Michaelis-Menten model:

$$\mathrm{S}_1 + \mathrm{S}_2 \underset{500}{\overset{1000}{\rightleftharpoons}} \mathrm{S}_3 \underset{1}{\overset{5}{\rightleftharpoons}} \mathrm{S}_2 + \mathrm{S}_4 \tag{4.76}$$

We choose a small molecular population to make it easy to interpret. Here we chose a feasible state vector as $\mathbf{N}_1 = \begin{bmatrix} N_1, & N_2, & N_3, & N_4 \end{bmatrix} = \begin{bmatrix} 3, & 3, & 0, & 0 \end{bmatrix}$. Then the state space of the system is generated using Algorithm 2 in Section 3.1 as

$$\mathbf{N} = \begin{bmatrix} 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 & 3 & 2 & 1 & 3 & 2 & 3 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 \end{bmatrix} \begin{matrix} \leftarrow N_1 \\ \leftarrow N_2 \\ \leftarrow N_3 \\ \leftarrow N_4 \end{matrix} \tag{4.77}$$

Initial condition $\mathbf{P}(\mathbf{N}, t = 0)$ is taken[3] as

$$\mathbf{P}(\mathbf{N}, t = 0) = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}^T \tag{4.78}$$

[3]The particular choice is for a better visualization of the solution's dynamics.

Next, the transition matrix is generated by Algorithm 3 in section 3.2 as

$$\mathbf{D} = \begin{bmatrix} -9000 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9000 & -4505 & 1000 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4000 & -2010 & 1500 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & -1515 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & -6003 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 6000 & -2507 & 1000 & 6 & 0 & 0 \\ 0 & 0 & 0 & 15 & 0 & 2000 & -1011 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & -3006 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 3000 & -509 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & -9 \end{bmatrix} \tag{4.79}$$

The eigenspectrum of the transition matrix $\mathbf{D}$ is shown in Figure 2.1 has a wide gap.

We chose the error tolerance ϵ equal to 10^{-6} which is small compared to the max probability 1 and $t_\epsilon = 10^{-2}$.

4.4.1 Selection of the Reduced Dimension d

The first step in the reduction is the choice of dimension d of the reduced model such that user's choice of error tolerance $\epsilon = 10^{-6}$ and $t_\epsilon = 10^{-2}$ are achieved. Following the

argument in Section 4.3.1 we seek an eigenvalue which satisfies condition (4.55), i.e.

$$\Re(\lambda_{d+1}) \leq \frac{1}{t_\epsilon}\log(\epsilon) = \frac{1}{10^{-2}}\log(10^{-6}) = -1381.6 \tag{4.80}$$

Then using MATLAB's `eigs` function, we identified two eigenvalues near zero. Next, repeating the process nearly newly identified eigenvalue until condition (4.80) is satisfied gives a set of eigenvalues as follows:

$$\lambda_1 = 0 \tag{4.81}$$

$$\lambda_2 = -7.2 \tag{4.82}$$

$$\lambda_3 = -12.7 \tag{4.83}$$

$$\lambda_4 = -18.1 \tag{4.84}$$

$$\lambda_5 = -2129 \tag{4.85}$$

The gap between eigenvalues λ_4 and λ_5 reflects the separation of timescale which corresponds to our choice of t_ϵ.

4.4.2 Identifying Initial Condition and Transition Matrix

No Transformation: (T = I)

Using Algorithm 5, a right eigenbasis matrix $\mathbf{M}_{nd}$ of $\mathbf{D}$ was found as

$$\mathbf{M}_{nd} = \begin{bmatrix} 4.89 \times 10^{-9} & -4.02 \times 10^{-6} & -7.54 \times 10^{-12} & 0.0139289 \\ 4.25 \times 10^{-8} & -4.15 \times 10^{-5} & -5.11 \times 10^{-11} & 0.250414 \\ 0 & 0 & 0 & 1 \\ -7.36 \times 10^{-8} & 1.78 \times 10^{-4} & 1.13 \times 10^{-10} & 0.664892 \\ -5.42 \times 10^{-5} & 0.041775 & 8.71 \times 10^{-8} & 2.71 \times 10^{-4} \\ -3.79 \times 10^{-4} & 0.500758 & 2.49 \times 10^{-7} & 1.90 \times 10^{-3} \\ 0 & 1 & 0 & 0 \\ 0.166809 & 2.39 \times 10^{-4} & -4.29 \times 10^{-4} & 1.77 \times 10^{-6} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.86}$$

We constructed $\mathbf{I_M}$ as a vector with components that represent row indices for re-indexing $\mathbf{M}_{nd}$ such that top $d \times d$ block is identity matrix. For this $\mathbf{M}_{nd}$,

$$\mathbf{I_M} = \begin{bmatrix} 9 & 7 & 10 & 3 & 1 & 2 & 4 & 5 & 6 & 8 \end{bmatrix} \tag{4.87}$$

Similarly, Algorithm 5 generates the left eigenbasis matrix $\mathbf{T}_{nd}$ of $\mathbf{D}$ as

$$\mathbf{T}_{nd} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.998776 & 3.24 \times 10^{-10} & 1.22 \times 10^{-3} & -5.66 \times 10^{-7} \\ 0.997123 & -1.32 \times 10^{-11} & 2.88 \times 10^{-3} & 5.74 \times 10^{-7} \\ 0.994467 & 3.05 \times 10^{-9} & 5.53 \times 10^{-3} & 6.45 \times 10^{-6} \\ 0 & 0 & 1 & 0 \\ -2.70 \times 10^{-4} & -2.72 \times 10^{-7} & 0.998918 & 1.35 \times 10^{-3} \\ -6.47 \times 10^{-4} & 1.74 \times 10^{-6} & 0.997403 & 3.24 \times 10^{-3} \\ 0 & 0 & 0 & 1.0 \\ -5.26 \times 10^{-8} & 1.43 \times 10^{-3} & -5.72 \times 10^{-4} & 0.999142 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.88}$$

Next, we re-index the rows of $\mathbf{M}_{nd}$, $\mathbf{T}_{nd}$, and $\mathbf{P}_0$ using the index $\mathbf{I}_{\mathbf{M}}$[4]. A corresponding re-indexing for $\mathbf{D}$ changes the index of both rows and columns.

Next, using the re-indexed $\mathbf{T}_{nd}$, $\mathbf{M}_{nd}$, and $\mathbf{P}_0$, the initial condition for the reduced system is computed as

$$\mathbf{X}(0) = \left(\mathbf{T}_{nd}^T \mathbf{M}_{nd}\right)^{-1} \mathbf{T}_{nd}^T \mathbf{P}_0 = \begin{bmatrix} 0.2076 \\ 0.1712 \\ 0.1941 \\ 0.0999 \end{bmatrix} \tag{4.89}$$

[4]A permutation matrix $\mathbf{M}$ can be created using $\mathbf{I}_M$ by assigning $\mathbf{M}_{i,j} = 1$ if $j = \mathbf{I}_{M_i}$, otherwise 0. Then the matrix $\mathbf{MA}$ is re-indexed row-wise, the matrix $\mathbf{AM}^T$ is re-indexed column-wise, and the matrix $\mathbf{MAM}^T$ is re-indexed both row-wise and column-wise

Finally, using the re-indexed $\mathbf{D}$ and $\mathbf{M}_{nd}$, the reduced transition matrix is computed as

$$\begin{bmatrix} \mathbf{D}_{dd} & \mathbf{D}_{dm} \end{bmatrix} \mathbf{M}_{nd} = \begin{bmatrix} -11 & -7.0 \times 10^{-4} & 1.1 & 4.6 \times 10^{-7} \\ 5.3 \times 10^{-3} & -8.6 & 11 & 7.7 \\ 14.0 & 3.2 & -9.5 & 5.0 \times 10^{-4} \\ 0 & 5.0 & 0 & -9.0 \end{bmatrix} \tag{4.90}$$

Then the solution of the reduced IVP

$$\dot{\mathbf{X}} = \left(\begin{bmatrix} \mathbf{D}_{dd} & \mathbf{D}_{dm} \end{bmatrix} \mathbf{M}_{nd} \right) \mathbf{X}\,, \qquad \mathbf{X}(0) = \left(\mathbf{T}_{nd}^T \mathbf{M}_{nd} \right)^{-1} \mathbf{T}_{nd}^T \mathbf{P}_0 \tag{4.91}$$

is computed using a non-stiff Ordinary Differential Equation (ODE) solver (`ode45`) or other methods. Then the approximate solution $\tilde{\mathbf{P}}$ is computed as

$$\tilde{\mathbf{P}}(t) = \mathbf{M}_{nd} \mathbf{X}(t) \tag{4.92}$$

This approximate solution is displayed in Figure 4.1(B) along with the exact solution in panel (A). The error in the approximation is shown in Figure 4.2. As expected, the approximate solution agrees with the exact solution after the initial transients.

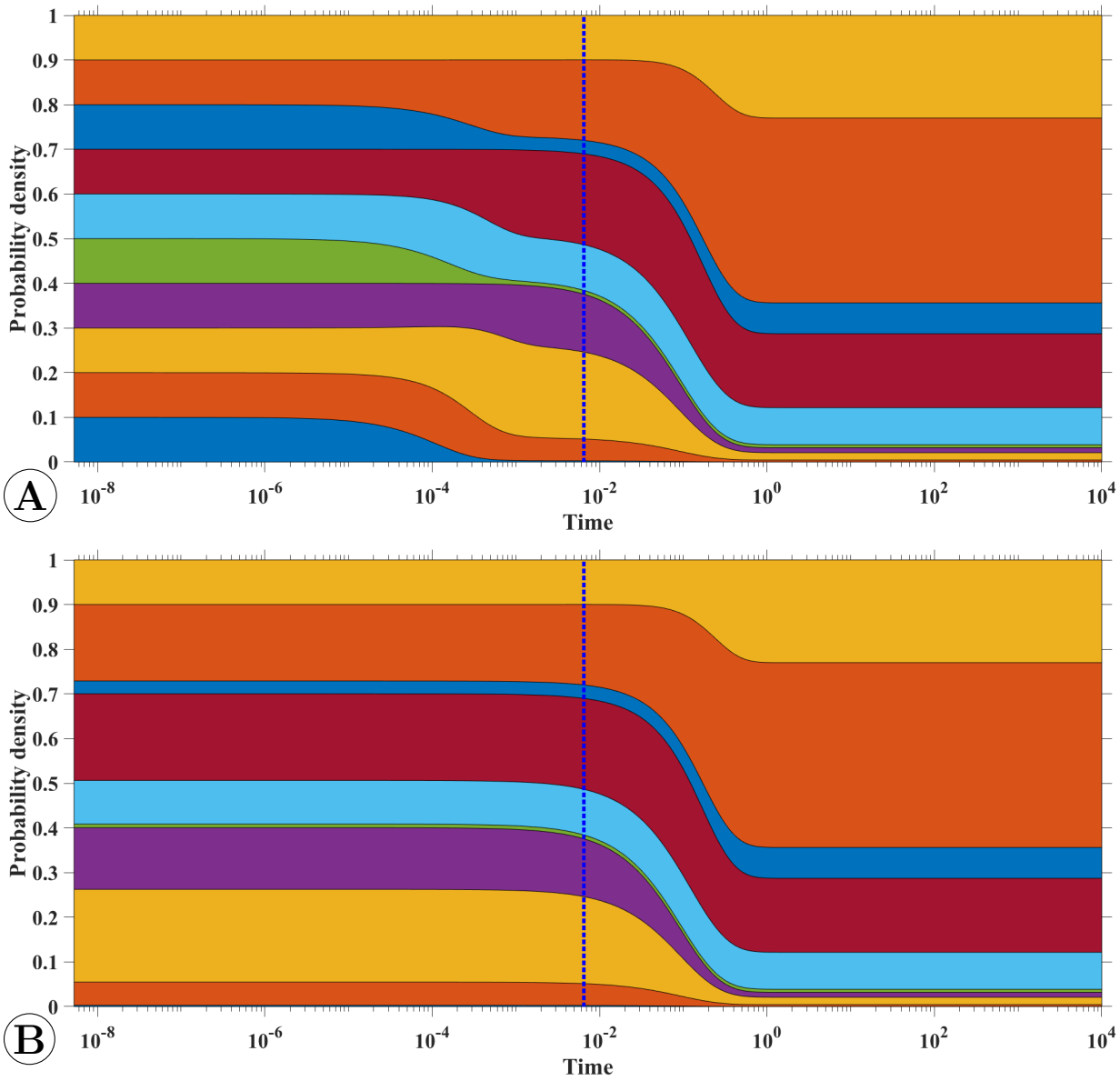

Figure 4.1: Exact (A) and approximate (B) solution of the Michaelis-Menten (MM) mechanism (4.76) with $\mathbf{N}_1 = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$. Blue dotted line represent the time lower-bound $t_\epsilon = 6.5 \times 10^{-3}$ (updated according to exact value of λ_{d+1}).

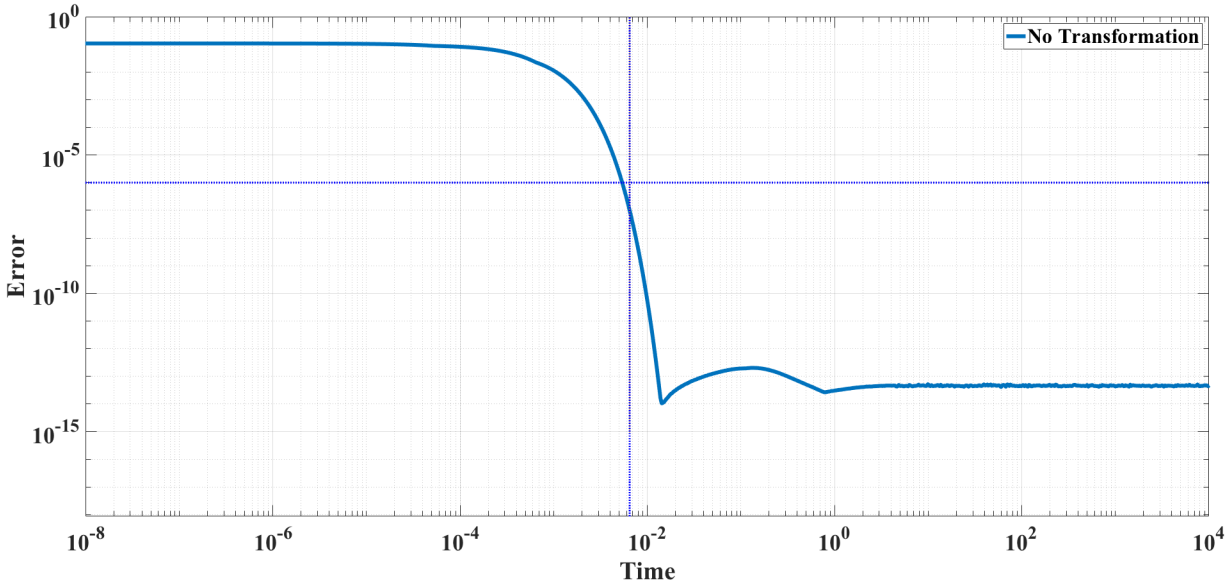

Figure 4.2: Error in the approximation due to no transformation approach for the MM mechanism (4.76) with $\mathbf{N}_1 = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$. Blue dotted lines represent the time lower-bound $t_\epsilon = 6.5 \times 10^{-3}$ (updated according to exact value of λ_{d+1}) and error tolerance $\epsilon = 10^{-6}$.

Transformation Using Left Eigenbasis

We constructed $\mathbf{I_T}$ as a vector with components that represent row indices for re-indexing $\mathbf{T}_{nd}$ such that top $d \times d$ block is identity matrix. For this $\mathbf{T}_{nd}$,

$$\mathbf{I_T} = \begin{bmatrix} 1 & 10 & 5 & 8 & 2 & 3 & 4 & 6 & 7 & 9 \end{bmatrix} \tag{4.93}$$

Next, we re-indexed the rows of the matrices $\mathbf{M}_{nd}$, $\mathbf{T}_{nd}$, and $\mathbf{P}_0$. A corresponding re-indexing for $\mathbf{D}$ changes the index of both rows and columns.

Next, using the re-indexed $\mathbf{T}_{nd}$ and $\mathbf{P}_0$, the initial condition for the reduced system is

computed as

$$\mathbf{X}(0) = \mathbf{T}_{nd}^{T}\mathbf{P}_0 = \begin{bmatrix} 0.39894 \\ 0.10014 \\ 0.30054 \\ 0.20037 \end{bmatrix} \tag{4.94}$$

Note that, unlike (4.89), the initial condition here sums to 1.

Finally, using the re-indexed $\mathbf{T}_{nd}$ and the re-indexed $\mathbf{D}$, the reduced transition matrix is

$$\hat{\mathbf{D}}_{dd} = \mathbf{T}_{nd}^{T}\begin{bmatrix} \mathbf{D}_{dd} \\ \mathbf{D}_{md} \end{bmatrix} = \begin{bmatrix} -11.0117 & -4.74 \times 10^{-7} & 1.37527 & -1.78 \times 10^{-3} \\ 2.92 \times 10^{-6} & -8.98713 & -1.63 \times 10^{-3} & 4.28886 \\ 11.0168 & -5.14 \times 10^{-3} & -9.48826 & 4.27880 \\ -5.10 \times 10^{-3} & 8.99228 & 8.11462 & -8.56588 \end{bmatrix} \tag{4.95}$$

Note that, as expected, column sum of $\hat{\mathbf{D}}_{dd}$ is 0 preserving probability conservation.

Next, the solution of the reduced IVP

$$\dot{\mathbf{X}}(t) = \hat{\mathbf{D}}_{dd}\mathbf{X}(t)\,, \qquad \mathbf{X}(0) = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{md} \end{bmatrix}^{T}\mathbf{T}\mathbf{P}_0 \tag{4.96}$$

is computed using a non-stiff ODE solver (`ode45`) or other methods.

Next, compute $\hat{\mathbf{M}}_{nd} = \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix}$ by finding the RREF of $\mathbf{TM}_{nd}$ as

$$\hat{\mathbf{M}}_{nd} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0.130263 & -3.40 \times 10^{-11} & 2.82 \times 10^{-5} & 5.80 \times 10^{-8} \\ 0.520189 & 9.70 \times 10^{-12} & 2.20 \times 10^{-4} & 1.79 \times 10^{-7} \\ 0.345869 & 9.66 \times 10^{-11} & 2.62 \times 10^{-4} & 1.55 \times 10^{-7} \\ -5.40 \times 10^{-4} & 1.21 \times 10^{-7} & 0.325295 & -4.50 \times 10^{-5} \\ -3.05 \times 10^{-3} & 2.20 \times 10^{-8} & 0.649603 & 5.60 \times 10^{-4} \\ 6.83 \times 10^{-6} & 3.68 \times 10^{-4} & -2.32 \times 10^{-3} & 0.857667 \end{bmatrix} \tag{4.97}$$

The transformation $\mathbf{T}$ is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^{T} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.98}$$

Note that the top d rows form the matrix $\mathbf{T}_{nd}^{T}$. Due to the structure of $\mathbf{T}$, the inverse of $\mathbf{T}$ can be easily computed by taking the negative of the top right block. i.e.

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{L}_{dd}^{-T}\mathbf{L}_{dm}^{T} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{4.99}$$

where the top right block is the negative of the non-identity part of $\mathbf{T}_{nd}$.

Then the approximate solution $\tilde{\mathbf{P}}$ is computed as

$$\tilde{\mathbf{P}} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) = \mathbf{T}^{-1}\hat{\mathbf{M}}_{nd}\mathbf{X}(t) \tag{4.100}$$

The approximate solution from this reduced system is essentially identical to the Figure 4.1B. The error in the approximation is shown in the Figure 4.3.

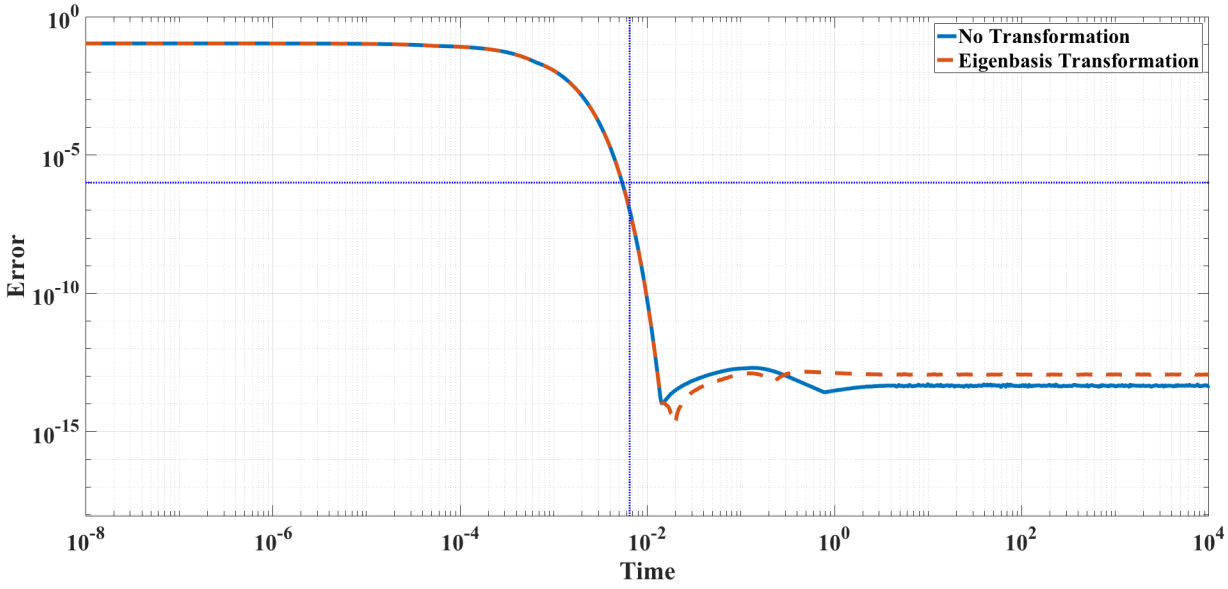

Figure 4.3: Error in the approximation due to eigenbasis transformation approach for the MM mechanism (4.76) with $\mathbf{N}_1 = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$. Blue dotted lines represent the time lower-bound $t_\epsilon = 6.5 \times 10^{-3}$ (updated according to exact value of λ_{d+1}) and error tolerance $\epsilon = 10^{-6}$.

Using Transition Matrix

Using

$$\mathbf{F} = 10^2 \left\lfloor \frac{\mathbf{D}}{10^2} \right\rfloor, \qquad i \in \{1, 2, \ldots, m\} \tag{4.101}$$

we found the matrix $\mathbf{F}$ satisfying $\mathbf{F} = \mathbf{D} - \Delta$ as

$$\mathbf{F} = \begin{bmatrix} -9000 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9000 & -4500 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4000 & -2000 & 1500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & -1500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6000 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6000 & -2500 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2000 & -1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3000 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3000 & -500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.102}$$

Next, we constructed an index $\mathbf{I_T}$ such that the bottom right $n - d \times n - d$ block of $\mathbf{F}$ is identity matrix. As mentioned before, choosing index of one row from each $\mathbf{F}_i$ block as first d indices and combining with rest $n - d$ indices will generate an index with $\mathbf{F}_{mm}$ invertible. One such choice is

$$\mathbf{I_T} = \begin{bmatrix} 1 & 5 & 8 & 10 & 2 & 3 & 4 & 6 & 7 & 9 \end{bmatrix} \tag{4.103}$$

Next, we re-indexed the vector $\mathbf{P}_0$. A corresponding re-indexing for $\mathbf{D}$ and $\mathbf{F}$ changes the index of both rows and columns.

Using the re-indexed $\mathbf{P}_0$, we computed the initial condition as

$$\mathbf{X}(0) = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1} \end{bmatrix} \mathbf{P}_0 = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.2 \\ 0.1 \end{bmatrix} \tag{4.104}$$

Note that the reduced initial condition here sums to 1.

Next, we found the transition matrix $\hat{\mathbf{D}} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}$ which is partitioned as

$$\hat{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix} \tag{4.105}$$

Then the transition matrix of the reduced system is computed as

$$\hat{\mathbf{D}}_{dd} - \hat{\mathbf{D}}_{dm}\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md} = \begin{bmatrix} -11.0072 & 1.37838 & 0 & 0 \\ 11.0072 & -9.48649 & 4.28571 & 0 \\ 0 & 8.10811 & -8.57143 & 9.0 \\ 0 & 0 & 4.28571 & -9.0 \end{bmatrix} \tag{4.106}$$

Note that, this matrix is a proper CME's transition matrix.

Next, the solution of the reduced IVP

$$\dot{\mathbf{X}}(t) = \hat{\mathbf{D}}_{dd} - \hat{\mathbf{D}}_{dm}\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}\mathbf{X}(t)\,, \qquad \mathbf{X}(0) = \begin{bmatrix}\mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1}\end{bmatrix}\mathbf{P}_0 \tag{4.107}$$

is computed using a non-stiff ODE solver (`ode45`) or other methods.

Next, computed $\begin{bmatrix}\mathbf{I}_{dd} \\ -\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}\end{bmatrix}$ as

$$\begin{bmatrix}\mathbf{I}_{dd} \\ -\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}\end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.00719424 & 0 & 0 & 0 \\ 0.129496 & 0 & 0 & 0 \\ 0.517986 & 0 & 0 & 0 \\ 0 & 0.027027 & 0 & 0 \\ 0 & 0.324324 & 0 & 0 \\ 0 & 0 & 0.857143 & 0 \end{bmatrix} \tag{4.108}$$

Then the approximate solution $\tilde{\mathbf{P}}$ is computed as

$$\tilde{\mathbf{P}} = \mathbf{T}^{-1}\begin{bmatrix}\mathbf{I}_{dd} \\ -\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}\end{bmatrix}\mathbf{X}(t) \tag{4.109}$$

The approximate solution from this reduced system is virtually identical to the Figure 4.1(B). Error in the approximation is shown in the Figure 4.4. Also, note that, this approximation and error are identical for the reduced system (2.17).

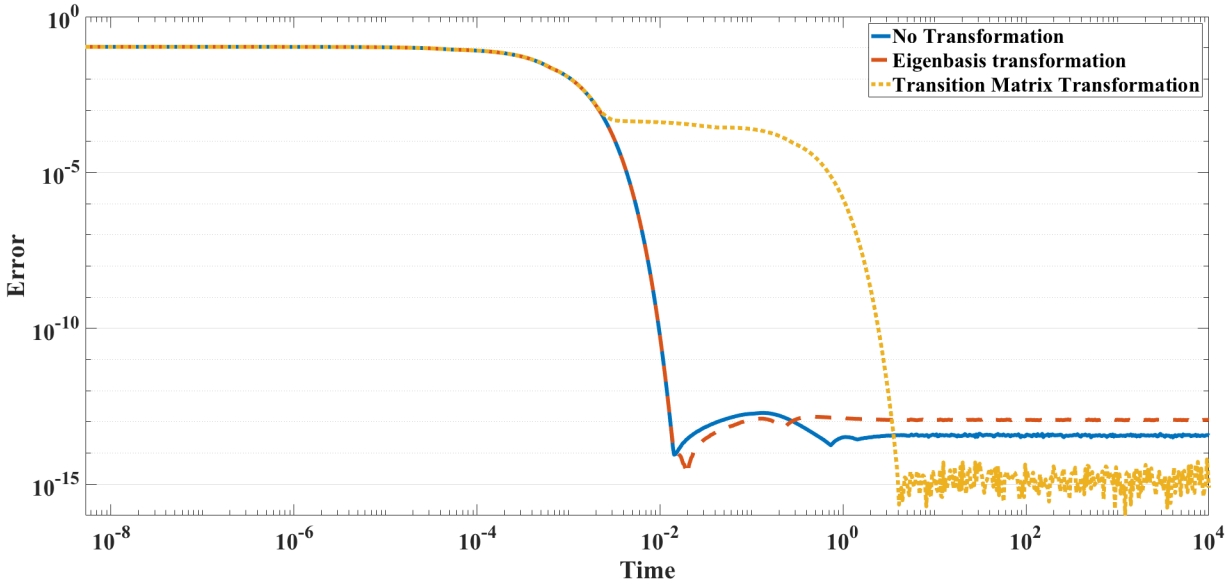

Figure 4.4: Error in the approximation due to transition matrix transformation approach for the MM mechanism (4.76) with $\mathbf{N}_1 = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$.

Reduced Network Generation Using The Reduced System

Note that the reduced transition matrix 4.106 has the properties of a proper CME's transition matrix:

1. Diagonal elements are non-positive
2. Off-diagonal elements are non-negative
3. Each column sums to 0

The reduction achieved here is same as the reduction proposed by Peleš in paper [28]. However, in that paper these structural advantages of the reduced transition matrix were not identified. These structural properties open the possibility of generating a reduced network from the reduced system. We have explored such reduced network for simple systems, as discussed below. A comprehensive analysis is expected to supply a bridge between time-scale lumping approaches and CME-based time-scale separation.

A reduced network reconstruction of the closed reaction chain system is given as follows:

Consider the closed reaction chain system

$$\mathrm{S}_1 \underset{5\times10^2}{\overset{10^3}{\rightleftharpoons}} \mathrm{S}_2 \xrightarrow{1} \mathrm{S}_3 \tag{4.110}$$

with a given state $\mathbf{N}_1 = \begin{bmatrix} S_1 & S_2 & S_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. Then the corresponding state space $\mathbf{S}$ of the system is generated as

$$\mathbf{S} = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{bmatrix} \tag{4.111}$$

Next, the transition matrix $\mathbf{D}$ of the CME is generated as

$$\mathbf{D} = \begin{bmatrix} -3000 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3000 & -2501 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2000 & -2002 & 1500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & -1503 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2000 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2000 & -1501 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 1000 & -1002 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1000 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1000 & -501 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.112}$$

The eigenspectrum of $\mathbf{D}$ is shown in Figure 4.5 Using the eigenspectrum of gap of 10^2, we

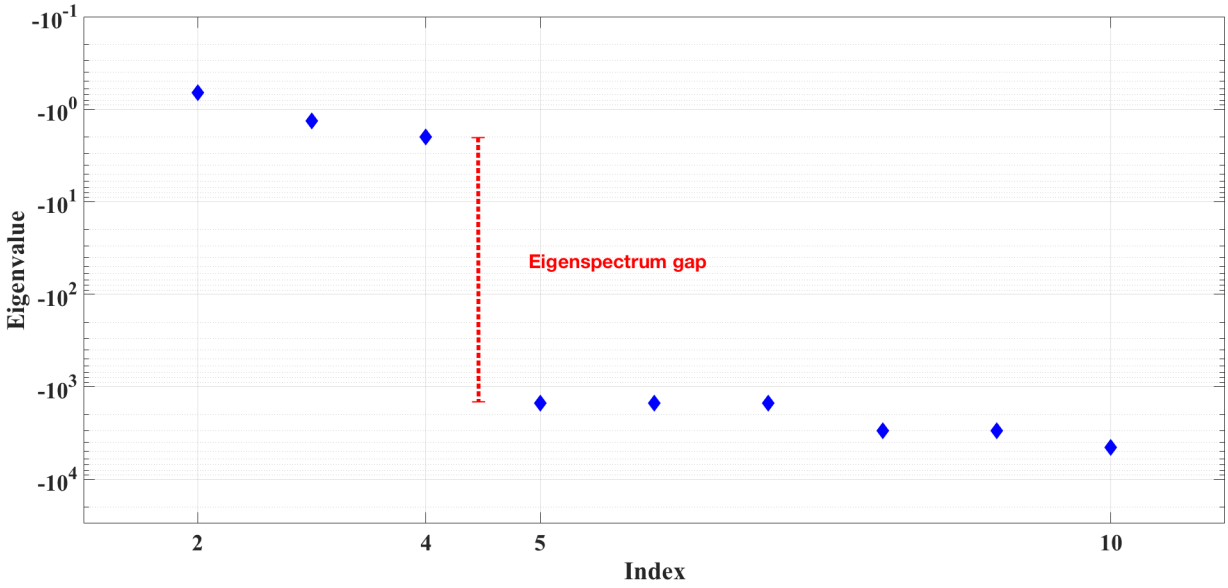

Figure 4.5: Eigenspectrum gap of a closed reaction chain system with 10 states.

split the matrix $\mathbf{F}$ and Δ where $\mathbf{F}$ is given by

$$\mathbf{F} = \begin{bmatrix} -3000 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3000 & -2500 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2000 & -2000 & 1500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & -1500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2000 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2000 & -1500 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & -1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1000 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 & -500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.113}$$

Then using the procedures presented in Section 4.2.2, we generate the reduced transition matrix (4.52) $\mathbf{K}$ as

$$\mathbf{K} = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 2 & -4/3 & 0 & 0 \\ 0 & 4/3 & -2/3 & 0 \\ 0 & 0 & 2/3 & 0 \end{bmatrix} \tag{4.114}$$

Next, states corresponding to each block $\mathbf{F}_i$ are separated as

$$\mathbf{F}_1 = \begin{bmatrix} -3000 & 500 & 0 & 0 \\ 3000 & -2500 & 1000 & 0 \\ 0 & 2000 & -2000 & 1500 \\ 0 & 0 & 1000 & -1500 \end{bmatrix} \quad \rightarrow \quad \mathbf{S}_1 = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 3 & 0 \end{bmatrix} \tag{4.115}$$

$$\mathbf{F}_2 = \begin{bmatrix} -2000 & 500 & 0 \\ 2000 & -1500 & 1000 \\ 0 & 1000 & -1000 \end{bmatrix} \quad \rightarrow \quad \mathbf{S}_2 = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 2 & 1 \end{bmatrix} \tag{4.116}$$

$$\mathbf{F}_3 = \begin{bmatrix} -1000 & 500 \\ 1000 & -500 \end{bmatrix} \quad \rightarrow \quad \mathbf{S}_3 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} \tag{4.117}$$

$$\mathbf{F}_4 = \begin{bmatrix} 0 \end{bmatrix} \quad \rightarrow \quad \mathbf{S}_4 = \begin{bmatrix} 0 & 0 & 3 \end{bmatrix} \tag{4.118}$$

We note that in each group, the total population of S_1 and S_2 is constant. This suggests a lumping. (Also note the abundance of species S_3 is not changing.) We choose the sum of population of species S_1 and S_2 as a lumped quantity, i.e., $X = S_1 + S_2$. Then a new

set of states of the form $\mathbf{S}_i = \begin{bmatrix} X & S_3 \end{bmatrix}$ that represent each group will be

$$\mathbf{S}_1 = \begin{bmatrix} 3 & 0 \end{bmatrix} \tag{4.119}$$

$$\mathbf{S}_2 = \begin{bmatrix} 2 & 1 \end{bmatrix} \tag{4.120}$$

$$\mathbf{S}_3 = \begin{bmatrix} 1 & 2 \end{bmatrix} \tag{4.121}$$

$$\mathbf{S}_4 = \begin{bmatrix} 0 & 3 \end{bmatrix} \tag{4.122}$$

$$\tag{4.123}$$

and the lumped reaction network is

$$\mathrm{X} \xrightarrow{k_x} \mathrm{S}_3 \tag{4.124}$$

where the rate constant k_x needs to be identified. For this, we compare the non-negative reaction propensity components in the reduced matrix with respect to the states. They are

1. $K_{2,1} = 2 \Rightarrow$ Reaction propensity of transition between $\mathbf{S}_1 \rightarrow \mathbf{S}_2$
2. $K_{3,2} = 4/3 \Rightarrow$ Reaction propensity of transition between $\mathbf{S}_2 \rightarrow \mathbf{S}_3$
3. $K_{4,3} = 2/3 \Rightarrow$ Reaction propensity of transition between $\mathbf{S}_3 \rightarrow \mathbf{S}_4$

A transition of $\mathbf{S}_1 \rightarrow \mathbf{S}_2$ occurs when one molecule of three available molecules of X is converted to a S_3 molecule. If this has reaction propensity of $3\,k_x$, we arrive at

$k_x = 2/3$. This is consistent with the other transitions: $2\,k_x = 4/3$ and $1\,k_x = 2/3$ which all gives same value for k_x. Then the reduced network representing the closed reaction chain system (4.110) is

$$\mathrm{X} \xrightarrow{2/3} \mathrm{S}_3 \tag{4.125}$$

We can also confirm this constant using the QSSA assumption that $\frac{10^3}{10^3+5\times10^2} = 2/3$ [2].

We next apply the same approach to the to MM mechanism. Consider the MM system as

$$\mathrm{S}_1 + \mathrm{S}_2 \underset{5\times10^2}{\overset{10^3}{\rightleftharpoons}} \mathrm{S}_3 \xrightarrow{1} \mathrm{S}_2 + \mathrm{S}_4 \tag{4.126}$$

and a state $\mathbf{N}_1 = \begin{bmatrix} S_1 & S_2 & S_4 & S_4 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}$ These produce a state space partitioned into blocks of $\mathbf{S}_i$ as

$$\mathbf{S}_1 = \begin{bmatrix} 3 & 3 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix} \tag{4.127}$$

$$\mathbf{S}_3 = \begin{bmatrix} 2 & 3 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 0 & 1 & 2 & 1 \end{bmatrix} \tag{4.128}$$

$$\mathbf{S}_1 = \begin{bmatrix} 1 & 3 & 0 & 2 \\ 0 & 2 & 1 & 2 \end{bmatrix} \tag{4.129}$$

$$\mathbf{S}_1 = \begin{bmatrix} 0 & 3 & 0 & 3 \end{bmatrix} \tag{4.130}$$

Applying a similar analysis as before, we first note that the abundance of species S_4 is constant in each group. Here, there are two combination of species with constant total abundance: $S_1 + S_3$ and $S_2 + S_3$. This can also be expressed as a single sum $S_1 + S_2 + 2S_3$. Then choosing this single sum as the lumped complex X, we define our reduced network as

$$\mathrm{X} \xrightarrow{k_x} \mathrm{S}_4 \tag{4.131}$$

which gives the representative states $\mathbf{S}_i = \begin{bmatrix} X & S_4 \end{bmatrix}$ as

$$\mathbf{S}_1 = \begin{bmatrix} 6 & 0 \end{bmatrix} \tag{4.132}$$

$$\mathbf{S}_2 = \begin{bmatrix} 5 & 1 \end{bmatrix} \tag{4.133}$$

$$\mathbf{S}_3 = \begin{bmatrix} 4 & 2 \end{bmatrix} \tag{4.134}$$

$$\mathbf{S}_4 = \begin{bmatrix} 3 & 3 \end{bmatrix} \tag{4.135}$$

$$\tag{4.136}$$

Then to identify the rate constant we compare the states $\mathbf{S}_i$s with the reduced reaction

propensities in the reduced transition matrix $\mathbf{K}$ which is generated as

$$\mathbf{K} = \begin{bmatrix} -306/139 & 0 & 0 & 0 \\ 306/139 & -60/37 & 0 & 0 \\ 0 & 60/37 & -6/7 & 0 \\ 0 & 0 & 6/7 & 0 \end{bmatrix} \tag{4.137}$$

1. $\mathbf{K}_{1,2} = 306/139 \Rightarrow \mathbf{S}_1 \rightarrow \mathbf{S}_2 \Rightarrow 6\, k_x \Rightarrow k_x = 306/834$

2. $\mathbf{K}_{2,3} = 60/37 \Rightarrow \mathbf{S}_2 \rightarrow \mathbf{S}_3 \Rightarrow 5\, k_x \Rightarrow k_x = 60/185$

3. $\mathbf{K}_{3,4} = 6/7 \Rightarrow \mathbf{S}_3 \rightarrow \mathbf{S}_4 \Rightarrow 4\, k_x \Rightarrow k_x = 6/28$

Unlike the closed reaction chain system considered previously, these propensities do not correspond to a unique value for k_x. The propensity is a nonlinear function of reactant abundance. To express this function, we explored fitting these values to a known function. To keep this process general, we fit to a polynomial. To improve accuracy of the estimate, we increase the dimension of the original problem from 10 states to 66 states. The analysis then gives data points: Plotting these data points shows a hyperbolic curve as shown in

X	20	19	18	17	16	15	14	13	12	11
k_x	$\frac{281}{684}$	$\frac{207}{511}$	$\frac{323}{825}$	$\frac{311}{839}$	$\frac{294}{857}$	$\frac{96}{311}$	$\frac{82}{307}$	$\frac{298}{1371}$	$\frac{190}{1203}$	$\frac{20}{231}$

Figure 4.6 which we fit with a second order polynomial as

$$k_x = (-3.86 \times 10^{-2} \times X^2) + (0.15516 \times X) - 1.1502 \tag{4.138}$$

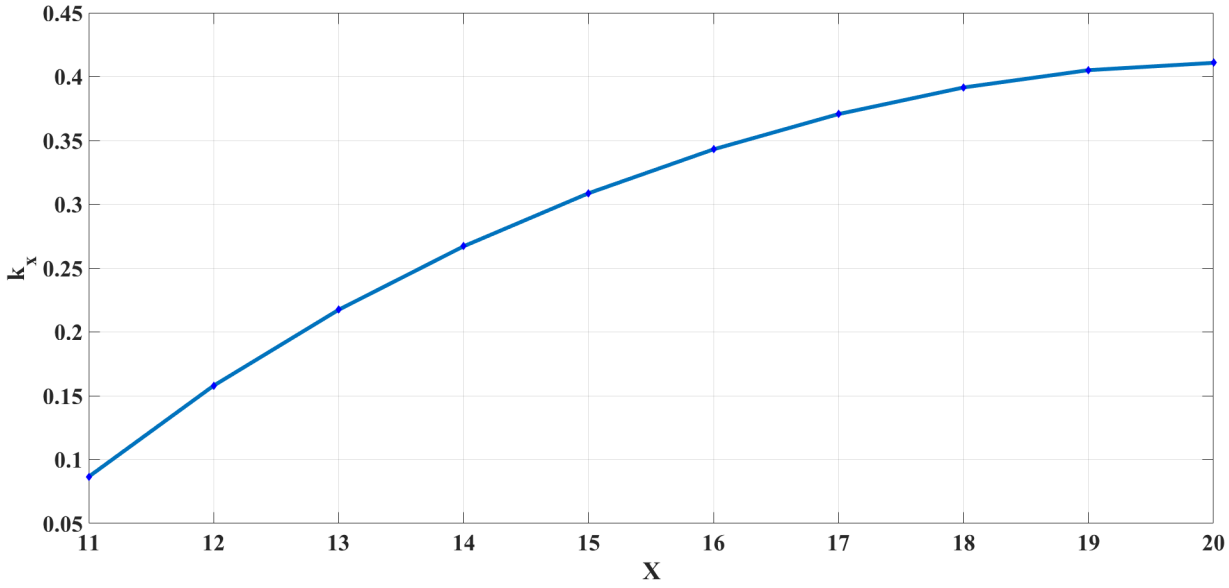

Figure 4.6: Rate of change of stochastic rate constant of reduced MM network

Then the reduced network has the form

$$\mathrm{X} \xrightarrow{(-3.86\times10^{-2}\times X^2)+(0.15516\times X)-1.1502} \mathrm{S}_4 \tag{4.139}$$

As future work, we hope to establish a general procedure for this reconstruction of a reduced network, which should allow efficient representation of large dimensional systems.

4.4.3 Efficiency

Each of the reductions presented in this chapter can be applied to generate approximate solutions to a CME. Table 4.1 shows computational times required when applying the eigenbasis-based reductions to the Michaelis-Menten system (4.76). These approaches are

computational efficient when determining solution behaviour over long times. In particular, these algorithms are well-suited to situations in which a number of initial value problems need to be solved for a particular CME. In this case, the reduction step (i.e. the eigenbasis generation) need only be carried out once.

N	n	d	Exact Solution		Eigenbasis Generation		Approximate Solution	
			$t \in (0,1)$	$t \in (0,100)$	Left	Right	$t \in (0,1)$	$t \in (0,100)$
10	66	11	1.66	2.2	0.77	0.31	0.44	0.47
30	496	31	23.43	65.1	15.0	15.2	0.67	0.72
50	1326	51	252.6	864.2	198.3	185.9	0.80	0.90

Table 4.1: Computation times (in seconds) for the Michaelis-Menten CME (4.76) for molecular populations of size N and corresponding state-space dimension n. Eigenbasis generation is required for the reduction approaches that do not rely on an eigenspectrum gap. The time to then generate the approximate solution (using ode45 on the reduced system) is shown separately. In each case, the reduced dimension d is chosen to achieve an error of order $\varepsilon = 10^{-6}$ for times greater than $t_\varepsilon = 10^{-2}$. Eigenbasis are generated with a tolerance of 10^{-12}. Computer configuration: Windows, Intel i5-6300U CPU, 2.4 GHz and 8GB RAM.

As shown in the previous section, for small molecular populations ($N = 3$), system (4.76) exhibits an eigenspectrum gap. In such cases, the more efficient reduction approach presented in section 4.2.2 (which, recall, was first presented in [28]) can be applied. This approach is not useful as the molecule number increases (past $N = 10$), because the eigenspectrum gap shrinks as the dimension n increases.

Chapter 5

Efficient Time-Step Selection for the Multi-Step Finite State Projection Algorithm

In principle, the solution of a finite dimensional Chemical Master Equation (CME) (2.5) at time t is given by

$$\mathbf{P}(\mathbf{N}, t) = \exp(\mathbf{D}\, t)\, \mathbf{P}(\mathbf{N}, 0) \tag{5.1}$$

However, as discussed in Chapter 1, for most systems of interest, the dimension of the CME is very large or infinite, raising computational challenges to find the matrix exponential term and thus the solution (5.1) of the CME. (Monte-Carlo methods like Gillespie's Stochastic Simulation Algorithm (SSA) reliably provides sample paths of the system. How-

ever, to attain a satisfactory resolution in the approximation, very large number of sample paths needs to be generated. To achieve precise statistics, these methods are computationally very expensive.)

The Finite State Projection (FSP) algorithm, developed by Munsky and Khammash [24], approximates the probability distribution of any CME at a given time t_f by truncating the state space. It is presented here as Algorithm 6.

Algorithm 6: The Finite State Projection algorithm

Input:

Propensity functions and stoichiometry for all reactions.;
$\mathbf{N}_J \leftarrow$ Initial truncated state space;
$\mathbf{P}_J(0) \leftarrow$ Initial probability density vector;
$t_f \leftarrow$ Final time of interest $\in \mathbb{R}_+$;
$\epsilon \leftarrow$ error tolerance $\in \mathbb{R}_+$;

begin

    $flag \leftarrow$ True;
    while *flag* = *True* **do**
        $\mathbf{D}_{J,J} \leftarrow$ Transition matrix of transitions between states in $\mathbf{N}_J$;
        $\mathbf{P}_J^{FSP}(t_f) \leftarrow \exp(\mathbf{D}_{J,J}\, t_f)\, \mathbf{P}_J(0)$;
        if $\mathbf{1}^T \mathbf{P}_J^{FSP}(t_f)_1 \geq 1 - \epsilon$ **then**
            $flag \leftarrow$ False;
        else
            Add more states to $\mathbf{N}_J$;

Output:

$\mathbf{P}_J^{FSP}(t_f) \leftarrow$ Approximate probability distribution at t_f;

For a given network, the FSP algorithm starts with a truncated state space $\mathbf{N}_J$. This truncated state space consist of states that are probable in the interval $(0, t_f)$. Next, the solution of the system of differential equation $\dot{\mathbf{P}}_J^{FSP} = \mathbf{D}_{J,J}\mathbf{P}_J^{FSP}$ is determined at t_f: $\mathbf{P}_J^{FSP}(t_f) = \exp(\mathbf{D}_{J,J}\, t_f)\, \mathbf{P}_J(0)$. Then the FSP algorithm checks whether sum of the

probabilities $\mathbf{P}_J^{FSP}(t_f)$ is sufficiently close to 1:

$$\mathbf{1}^T \mathbf{P}^{FSP}(\mathbf{N}_J, t_f) \geq 1 - \epsilon. \tag{5.2}$$

If so, the components corresponding to states $\mathbf{N}_J$ in the exact solution $\{\mathbf{P}(\mathbf{N}, t_f)\}$ are well-approximated as (proved in [24])

$$\left\| \{\mathbf{P}(\mathbf{N}, t_f)\}_J - \mathbf{P}^{FSP}(\mathbf{N}_J, t_f) \right\|_\infty \leq \epsilon \tag{5.3}$$

Otherwise, the state space $\mathbf{N}_J$ is updated by adding more states and the process is repeated until the condition (5.2) is satisfied.

The FSP algorithm generates an approximation to the solution of a CME at a given time point t_f. Generally, for a small time interval, the number of states that have to be retained in the truncated system is small compared to the total number of states in the system. However, for longer intervals, the system may reach a large number of states and the FSP algorithm is not efficient. An alternate implementation involves splitting the long interval $(0, t_f)$ into a set of short intervals:

$$(0\,, \quad t_f) = \Big((0 = t_1, t_2), \quad (t_2, t_3), \quad \ldots, \quad (t_{n-1}, t_n = t_f) \Big) \tag{5.4}$$

Then to calculate the solution at time t_{i+1} $(i = 1, 2, \ldots, n)$, the solution of the CME at time t_i is used as the initial condition. Efficiency is gained by pruning non-active states in each sub-interval.

Several algorithms for multiple time-step FSP have been presented [26, 27]. In the

paper [27], a multi-step FSP is presented with a fixed time step whereas in paper [26], an adaptive time-stepping method is formulated that has the advantage of limiting the number of states in the truncated system. In addition, Krylov-basis computation of the matrix exponential improved the speed of the algorithm [33, 26, 48] for short time steps (convergence using the Krylov basis is slow over longer intervals).

A concise description of the multiple time-step FSP approach by Munsky and Khammash [27] is presented Algorithm 7. In this algorithm, due to a fixed time-step τ, an

Algorithm 7: Multiple time-step FSP algorithm with fixed time step [27]

Input:
As required in the original FSP algorithm 6;
$n \leftarrow$ Number of intervals;
$\tau \leftarrow$ Fixed time-step such that $t_k = k\tau$ and $t_n = t_f$;
begin
  $k \leftarrow 0$;
  while $t_k < t_f$ **do**
    Update state space and CME;
    $\epsilon_k \leftarrow$ Maximum error in the solution for the interval (t_k, t_{k+1});
    while $1 - \|\mathbf{P}(t_{k+1})\|_1 \leq \epsilon_k$ **do**
      Update state space and CME;
      $\mathbf{P}(t_{k+1}) \leftarrow$ Solution at time t_{k+1};
    $k \leftarrow k + 1$;
Output:
$\mathbf{P}(t) \leftarrow$ Approximate probability distribution of the system at time t_f;

updated state space and the corresponding CME is calculated iteratively until the error tolerance is satisfied. This is an inefficient method because for each time-step the error tolerance condition must be checked iteratively by computing an approximate solution at time t_k. A better approach is by choosing a variable time step, where for a chosen state

space and CME, an optimal time-step will be computed such that the error tolerance is satisfied. Such an approach is implemented in the paper [26] which is presented here in Algorithm 8. Each iteration of this algorithm begins with an assessment of the full remain-

Algorithm 8: Multiple time-step FSP algorithm with variable time step [26]

Input:

As required in the original FSP algorithm 6;

begin

$t_k \leftarrow 0$;

while $t_k < t_f$ **do**

Update system and CME;

$\epsilon_k \leftarrow$ Maximum error;

$\tau \leftarrow t_f - t_k$;

while $\|\mathbf{P}(t_k + \tau)\|_1 < 1 - \epsilon_k$ **do**

$\tau \leftarrow \frac{\tau}{2}$

$k \leftarrow k + 1$;

Output:

$\mathbf{P}(t) \leftarrow$ Approximate probability distribution of the system at time t_f;

ing time-step, which is halved if it is found to be too long. These assessments demands approximation of a solution over the potentially long interval (t_k, t_f), which can be computationally demanding. However, these repeated assessments can be made efficient by exploiting a pre-conditioning of the CME. Because these assessments rely on approximations generated repeatedly from a single version of the system (i.e. truncated state space), such a pre-conditioning can be very efficient because it needs to be applied only once.

In this chapter, we present such a pre-conditioning approach and use it for efficient assessment of the time-step. When applied with the efficient transition-matrix generation algorithm from Chapter 3, this approach provides improved efficiency in the adaptive

multi-step FSP.

5.1 Preconditioned FSP Algorithm

Even with multi-step approach for FSP, the state space of the truncated system can be quite large for some sub-intervals. This challenge is aggravated by the presence of wide range of timescales. Algorithms such as time scale separation[28] and aggregation [40] can be used to precondition the system for treatment of these large systems. However, these algorithms have limitations because the resulting error, which is dictated by the eigenspectrum gap, may be larger than required error tolerance in each sub-interval. In this chapter, We follow a reduction approach using Theorem 2 with eigenbasis transformation that leads to probability conservation. This method is a combination of timescale separation and aggregation methods which is used as a preconditioning for the FSP truncated system in each sub-interval. In contrast to the previous FSP algorithms, our approach generates an approximate transient solution for the whole interval (t_0, t_f). The novel preconditioned FSP approach is presented in Algorithm 9.

The algorithm 9 takes as inputs a description of the network, an initial probability distribution, a time interval (t_0, t_f) on which the transient solution will be determined, and an error tolerance ϵ for the approximate solution at time t_f. It also takes a small time length t_ϵ for generating an interval where the preconditioning is valid.

Using the initial state space and probability distribution, the algorithm updates the state space, initial probability distribution, and transition matrix. Updating of the state

Algorithm 9: Transient multi-time-step finite state projection algorithm with pre-conditioning

Input:

$\{\mathbf{S}_i\}_{i=1}^{h} \leftarrow$ set of Stoichiometry vectors $\in \mathbb{Z}^{k\times h}$;

$\{\mathbf{R}_i\}_{i=1}^{h} \leftarrow$ set of Reactant-Stoichiometry vectors $\in \mathbb{Z}_{\geq 0}^{k\times h}$;

$\mathbf{c} \leftarrow$ Reaction rate vector;

$[\mathbf{N}_0, \mathbf{P}_0] \leftarrow$ Initial state vectors and corresponding probability distribution;

$(t_0, t_f) \leftarrow$ Time interval;

$\epsilon \leftarrow$ error tolerance at time t_f;

$t_\epsilon \leftarrow$ Length of a small time interval;

begin

  $k \leftarrow 0$;

  while $t_k < t_f$ **do**

    $\tau_k \leftarrow 0$;

    while $\tau_k < t_\epsilon$ **do**

      $[\mathbf{N}_k, \mathbf{P}_k] \leftarrow$ Updated state space and probability distribution;

      $\mathbf{D}_k \leftarrow$ Transition matrix using the algorithm 3 for the state space $\mathbf{N}_k$;

      $[\mathbf{D}_k, \mathbf{Z}_k(t), \mathbf{T}_k, \tau_k] \leftarrow$ `Precondition`$(\mathbf{D}_k, \mathbf{P}_k, \epsilon, t_\epsilon)$;

    $t_{k+1} = t_k + \tau_k$;

    $\mathbf{Z}_{k+1} \leftarrow \mathbf{Z}_k(\tau)$;

    $[\mathbf{D}_k, \mathbf{P}_{k+1}, \mathbf{M}_k] \leftarrow$ `Approximation`$(\mathbf{D}_k, \mathbf{T}_k, \mathbf{Z}_{k+1})$;

    InputList(k) $\leftarrow$ A list of variables t_{k+1}, $\mathbf{Z}_k(t)$, $\mathbf{T}_k$, $\mathbf{M}_k$, and $\mathbf{N}_k$;

    $\mathbf{N}_{full} \leftarrow$ Set of all states that the system attained so far;

    $k \leftarrow k + 1$;

Output:

$\mathbf{P}(t) \leftarrow$ `TransientSolution`(InputList, $\mathbf{N}_{full}$), Approximate transient solution of the system in the interval (t_0, t_f);

Algorithm 10: Preconditioning Algorithm

Function `Precondition`($\mathbf{D}$, $\mathbf{P}_0$, ϵ, t_ϵ)**:**

- $d \leftarrow$ Reduced dimension;
- $\mathbf{T}_{nd} \leftarrow$ Left Semi-Orthogonal eigenbasis of $\mathbf{D}$ corresponding to slow time-scale d eigenvalues using Algorithm 5;
- Re-Index $\mathbf{D}$, $\mathbf{P}_0$, $\mathbf{T}_{nd}$ using an index such that top d columns of $\mathbf{T}_{nd}$ is identity matrix.;
- $\mathbf{Q} \leftarrow \begin{bmatrix} \mathbf{T}_{nd}^T \mathbf{D}_{nd} & \mathbf{0} \\ -\mathbf{1}^T \mathbf{Q}_k & 0 \end{bmatrix}$;
- $\mathbf{Z}_0 \leftarrow \begin{bmatrix} \mathbf{T}_{nd}^T \mathbf{P}_0 \\ Z_\epsilon = 1 - \mathbf{1}^T \mathbf{T}_{nd}^T \mathbf{P}_0 \end{bmatrix}$;
- $\mathbf{Z}(t) \leftarrow$ Transient solution of the system $\dot{\mathbf{Z}}(t) = \mathbf{Q}\,\mathbf{Z}(t)\,, \mathbf{Z}(0) = \mathbf{Z}_0$ for times the error component $Z_\epsilon(t) \leq \epsilon$;
- $\tau_k \leftarrow \max(t)$;
- **return** $\mathbf{D}$, $\mathbf{Z}(t)$, τ, and $\mathbf{T}_{nd}$, ;

Algorithm 11: Approximation Algorithm

Function `Approximation`($\mathbf{D}_k$, $\mathbf{T}_k$, $\mathbf{Z}_{k+1}$)**:**

- $\mathbf{T} \leftarrow \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{T}_{md}^T \\ \mathbf{0}_{dm} & \mathbf{I}_{mm} \end{bmatrix}$ where $\mathbf{T}_{md}$ is the last $m = n - d$ columns of $\mathbf{T}_k$;
- $\mathbf{M}_{temp} \leftarrow$ Right Semi-Orthogonal eigenbasis of $\mathbf{D}$ corresponding to slow time-scale d eigenvalues using Algorithm 5;
- $\mathbf{M}_{nd} \leftarrow$ Transpose of `RREF`$((\mathbf{T}\mathbf{M}_{temp})^T)$;
- $\mathbf{P}_{k+1} \leftarrow \mathbf{T}^{-1}\mathbf{M}_{nd}\mathbf{Z}_{k+1}$;
- **return** $\mathbf{P}_{k+1}$ and $\mathbf{M}_{nd}$;

space can involve both addition of new states and removal of previously included states. Then the transition matrix can be generated using the Algorithm 3. This algorithm helps to speed up the iterative process as it can be used to accommodate new state additions and pruning of states efficiently.

Next, we follow the eigenbasis transformation algorithm presented in Chapter 4 for

Algorithm 12: Transient Solution Function

Function `TransientSolution`(InputList, $\mathbf{N}_{full}$)**:**
- $n \leftarrow$ number of sub-intervals;
- **for** $k = 1, 2, \ldots, n$ **do**
  - Get variables $\mathbf{Z}_k(t)$, t_ϵ, $\mathbf{T}_k$, $\mathbf{M}_k$, and $\mathbf{N}_k$ from InputList;
  - $\mathbf{T}^{-1} \leftarrow \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{T}_{md}^T \\ \mathbf{0}_{dm} & \mathbf{I}_{mm} \end{bmatrix}$ where $\mathbf{T}_{md}$ is the last $m = n - d$ columns of $\mathbf{T}_k$;
  - $\mathbf{P}_k(t) \leftarrow \mathbf{T}^{-1}\mathbf{M}_k\mathbf{Z}_k(t)$;
  - $\mathbf{I}_k \leftarrow$ Index of $\mathbf{N}_k$ in $\mathbf{N}_{full}$;
  - $\mathbf{P}(t) \leftarrow$ Re-indexed solution $\mathbf{P}_k(t)$ using $\mathbf{I}_k$ in the interval $(t_k + t_\epsilon, t_{k+1})$;
- **return** $\mathbf{P}(t)$;

the preconditioning because of its implementation as a 2-step process: Reduction and Approximation. In this approach, by only using partial left eigenbasis, we reduce the system as

$$\dot{\mathbf{P}}_k = \mathbf{D}_k\mathbf{P}_k\,,\ \mathbf{P}(0) = \mathbf{P}_0 \qquad \rightarrow \qquad \dot{\mathbf{Z}}_k = \mathbf{Q}_k\mathbf{Z}_k\,,\ \mathbf{Z}(0) = \mathbf{Z}_0. \tag{5.5}$$

This reduced system is then augmented with a state Z_ϵ to captures the dynamics of the error:

$$\begin{bmatrix} \dot{\mathbf{Z}}_k \\ \dot{Z}_\epsilon \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} \\ -\mathbf{1}^T\mathbf{Q}_k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Z}_k \\ Z_\epsilon \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{Z}_0 \\ Z_{\epsilon_0} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_0 \\ 1 - \mathbf{Z}_0 \end{bmatrix} \tag{5.6}$$

The modified system (5.6) is then solved, from which we have the error $Z_\epsilon(t)$ at each time. (Note that, to find the error, there was no need to determine an approximation of the original solution and thus only the left eigenbasis was required for generating the reduced

system. This cuts the computation time by about 50%, as shown in Table 4.1). This error description Z_ϵ is then used to find a time-step τ_k so that the FSP truncated system is valid in the interval $(t_k, t_k + \tau_k)$.

Next, the approximate solution $\mathbf{P}_k(t)$ can be generated following Algorithm 11 (for which the right eigenbasis must be calculated). (Alternatively, any other approximation method could be used at this second step.) Using right eigenbasis ($\mathbf{M}_{nd}$), inverse of the transformation matrix ($\mathbf{T}^{-1}$), and reduced solution ($\mathbf{Z}_{k+1}$), the approximate solution is computed as

$$\mathbf{P}_{k+1} = \mathbf{T}^{-1}\mathbf{M}_{nd}\mathbf{Z}_{k+1} \tag{5.7}$$

Next, after the approximated solutions are identified on each time-step, we can generate the solution over the entire interval (t_0, t_f) as described in Algorithm 12. Using the variables saved in InputList at each step, transient solution can be generated in each interval as

$$\mathbf{P}_k(t) = \mathbf{T}^{-1}\mathbf{M}_k\mathbf{Z}_k(t) \tag{5.8}$$

Finally, finding the index of the corresponding states $\mathbf{N}_k$ in the statespace $\mathbf{N}_{full}$, we assign the probability values and generates $\mathbf{P}(t)$ in the interval $(t_k + t_\epsilon, t_{k+1})$. Iterating over all time intervals, we generates the transient solution for the whole interval (t_0, t_f).

A main contribution of this thesis is an efficient determination of appropriate time-steps using the preconditioning algorithm 10. This time-step determination is a computationally expensive process in state-of-the-art implementations of multi-step FSP [49]. Using this time-step identification, we propose a workflow for approximating solutions to large CME

models is shown in Figure 5.1.

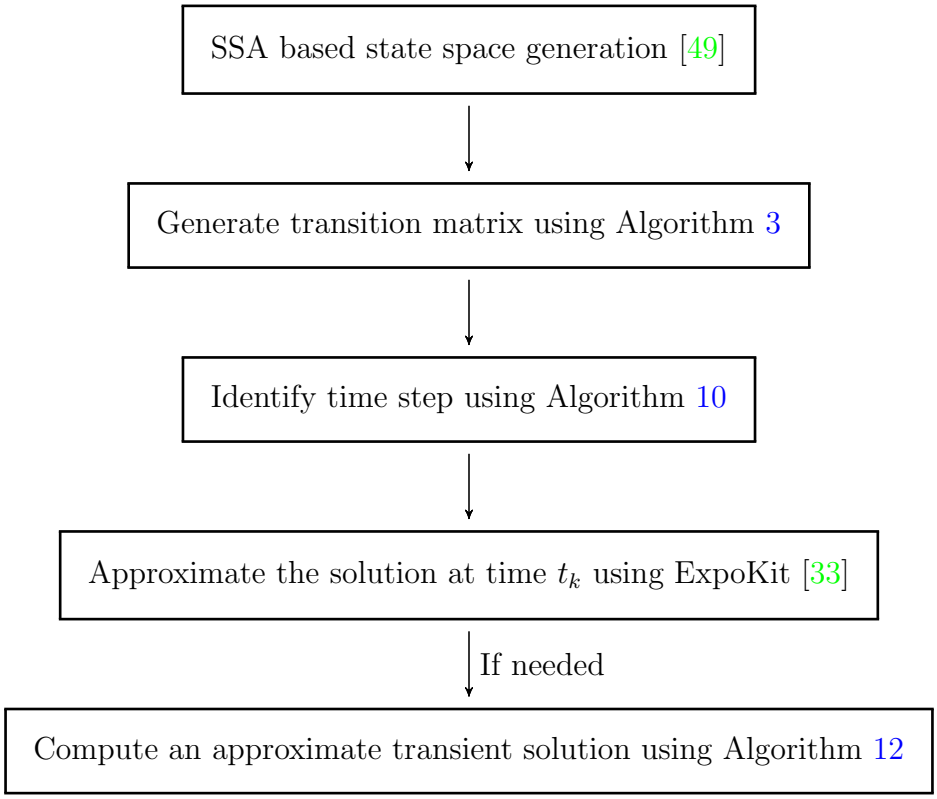

Figure 5.1: Proposed workflow for approximating solutions to large CME models

Chapter 6

Conclusion and Future Directions

In a biochemical system, molecular populations which are in small numbers can produce significant variability in the dynamics of the system. In such cases, deterministic analysis are not favored and stochastic models became a necessity. The Chemical Master Equation (CME) is a standard stochastic modeling approach for this purpose [14, 20, 21].

The task of generating a CME (i.e. state space and transition matrix) from a network is not often considered in the literature. In our experience, we found the published iterative approach to be computationally expensive. The first contribution of this thesis is an efficient generalized algorithm for generating the state space and transition matrix of the CME. When employed within an iterative procedure like Finite State Projection (FSP) [24], this algorithm offers significant speed-up in the computation.

The CME offers a comprehensive analysis of a biochemical system at molecular level, but solving the CME is hindered by the curse of dimensionality and multiple timescales in

the biochemical system. Over the past couple of decades, several algorithms were introduced to approximate the CME. One of the most acknowledged contribution is the FSP algorithm [24]. Improvements published over the past decade significantly improved this approach, allowing it to be applied to complex systems [49]. The time-stepping implementation is a major factor of this improvement [26]. Many optimizations, such as adaptive state space generation through Stochastic Simulation Algorithm (SSA) based approach and fast matrix exponential computation by Krylov approaches, also improved the speed of the FSP algorithm [49, 33]. A final step to improve the FSP further is preconditioning the FSP-truncated CMEs. Efforts are taken to implement preconditioning using well separated timescales and aggregation methods [28, 40]. However, while solving the FSP, the required assumption (e.g. a wide eigenspectrum gap) do not hold for most of the systems of interest. In addition, these methods yield an error which is fixed by the system structure itself. To address this gap, we built on the ideas of Roussel and Zhu to implement a generalized reduction approach based on Semi-Orthogonal Eigenbasis transformation algorithm [4]. This reduction approach results in efficient approximation of the solution of the CME for long times. Moreover, because the computational cost of this approach is paid primarily in the reduction itself, it is particularly well-suited to situations in which many simulations must be made of a single system (with different initial conditions). This would be the case, e.g. when calibrating the initial condition of a model against experimental observations.

Finally, we applied this reduction approach to achieve efficient identification of the variable time step in the multi time-step FSP algorithm, as well as efficient generation an approximate solution over an entire interval $(0, t_f)$, as opposed to the standard multi-step FSP output of an approximation only at discrete time-points.

6.1 Future Directions

The efficiency of the reduction algorithm proposed in Chapter 4 and its application in multi time-step FSP is significantly depend upon the Semi-orthogonal eigenbases. Improvements on the convergence of this algorithm will significantly speed up the process. As a future direction, pursuing on algorithms to generate fast converging Semi-orthogonal eigenbases will have tremendous improvement in the CME's dynamical analysis.

Moreover, a comprehensive timing comparison for the multi-step FSP will establish the gain of efficiency achieved by the reduction-based time-step selection. Table 4.1 gives an idea of the timing, but a proper comparison will take into account all of the variable aspects of multi-step FSP approaches (including state-space generation, preconditioning, approximation, and time-step selection).

APPENDIX

Proof of Theorem 1. From (4.12), we have

$$\mathbf{P}(t) = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \exp\left(\begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} t \right) \begin{bmatrix} \mathbf{A}_d \\ \mathbf{A}_m \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \begin{bmatrix} \exp(\Lambda_{dd}\, t)\mathbf{A}_d \\ \exp(\Lambda_{mm}\, t)\mathbf{A}_m \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \exp(\Lambda_{dd}\, t)\mathbf{A}_d + \begin{bmatrix} \mathbf{R}_{dm} \\ \mathbf{R}_{mm} \end{bmatrix} \exp(\Lambda_{mm}\, t)\mathbf{A}_m \tag{3}$$

$$= \tilde{\mathbf{P}}(t) + \begin{bmatrix} \mathbf{R}_{dm} \\ \mathbf{R}_{mm} \end{bmatrix} \exp(\Lambda_{mm}\, t)\mathbf{A}_m \tag{4}$$

where $\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \exp(\Lambda_{dd}\, t)\mathbf{A}_d$ from (4.13). Then

$$\mathbf{P}(t) - \tilde{\mathbf{P}}(t) = \begin{bmatrix} \mathbf{R}_{dm} \\ \mathbf{R}_{mm} \end{bmatrix} \exp(\Lambda_{mm}\, t)\mathbf{A}_m \tag{5}$$

Taking max-norm on both sides gives

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty = \left\| \begin{bmatrix} \mathbf{R}_{dm} \\ \mathbf{R}_{mm} \end{bmatrix} \exp(\Lambda_{mm}\, t)\mathbf{A}_m \right\|_\infty \tag{6}$$

In summation form,

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty = \left\| \sum_{i=d+1}^{n} a_i \exp(\lambda_i\, t) R_i \right\|_\infty \tag{7}$$

$$\leq \sum_{i=d+1}^{n} |a_i| \exp(\lambda_i\, t)\, \|R_i\|_\infty \tag{8}$$

Since eigenvectors are normalized $\|R_i\|_\infty \leq 1$. Then

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty \leq \sum_{i=d+1}^{n} |a_i| \exp(\lambda_i\, t) \tag{9}$$

Since $\lambda_{d+1} \geq \lambda_i$, $\forall i \in d+2, \ldots, n$, then

$$\exp(\lambda_{d+1}\, t) \geq \exp(\lambda_i\, t) \qquad \forall t \text{ and } i \in d+2, \ldots, n \tag{10}$$

Then

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty \leq \exp(\lambda_{d+1}\, t) \sum_{i=d+1}^{n} |a_i| \tag{11}$$

$$= \exp(\lambda_{d+1}\, t)\, \|\mathbf{A}_m\|_1 \tag{12}$$

For complex eignvalues, because $|\exp(a+ib)| = \exp(a)\,|\cos(b) + i\,\sin(b)| \leq \exp(a))$, we have

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty \leq \exp(\Re(\lambda_{d+1})\, t)\, \|\mathbf{A}_m\|_1 \tag{13}$$

Finally, for a given epsilon, set $t_\epsilon = \frac{1}{\Re(\lambda_{d+1})} \ln(\epsilon)$. Then

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty \leq \exp\left(\Re(\lambda_{d+1})\, \tfrac{1}{\Re(\lambda_{d+1})} \ln(\epsilon)\right) \|\mathbf{A}_m\|_1 \tag{14}$$

$$= \epsilon\, \|\mathbf{A}_m\|_1 \qquad \forall t \geq t_\epsilon \tag{15}$$

$$= \mathcal{O}(\epsilon) \qquad \forall t \geq t_\epsilon \tag{16}$$

as required. □

Lemma 1. *Suppose the assumptions in InfoBox 4.2 is true.*

$$\mathbf{L}_{dd}^T \mathbf{R}_{dm} + \mathbf{L}_{md}^T \mathbf{R}_{mm} = \mathbf{0}_{dm} \tag{17}$$

Proof. Let $\mathbf{L}_i$ and $\mathbf{R}_j$ be left and right eigenvectors of $\mathbf{D}$ corresponding to two distinct

eigenvalues λ_i and λ_j respectively. Then

$$\mathbf{D}\mathbf{R}_j = \lambda_j \mathbf{R}_j \tag{18}$$

$$\mathbf{L}_i^T \mathbf{D} = \lambda_i \mathbf{L}_i^T \tag{19}$$

Left multiplying eq. (18) with $\mathbf{L}_i^T$ and right multiplying eq. (19) with $\mathbf{R}_j$ gives

$$\mathbf{L}_i^T \mathbf{D}\mathbf{R}_j = \lambda_j \mathbf{L}_i^T \mathbf{R}_j \tag{20}$$

$$\mathbf{L}_i^T \mathbf{D}\mathbf{R}_j = \lambda_i \mathbf{L}_i^T \mathbf{R}_j \tag{21}$$

Then subtracting eq. (20) from eq. (21) gives

$$0 = (\lambda_i - \lambda_j)\mathbf{L}_i^T \mathbf{R}_j \tag{22}$$

Because $\lambda_i \neq \lambda_j$, we have

$$\mathbf{L}_i^T \mathbf{R}_j = 0 \tag{23}$$

Then, because $\lambda_d > \lambda_{d+1}$, the eigenvalues in Λ_{dd} are distinct from those in Λ_{mm}. Then

$$\begin{bmatrix} \mathbf{L}_{dd}^T & \mathbf{L}_{md}^T \\ \mathbf{L}_{dm}^T & \mathbf{L}_{mm}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} = \begin{bmatrix} *_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & *_{mm} \end{bmatrix} \tag{24}$$

where $*$ is a non-zero square matrix. In particular, the upper right block gives

$$\mathbf{L}_{dd}^T \mathbf{R}_{dm} + \mathbf{L}_{md}^T \mathbf{R}_{mm} = \mathbf{0}_{dm} \tag{25}$$

□

Lemma 2. *Suppose the assumptions in InfoBox 4.2 is true. Consider an n-dimensional initial value problem*

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t) \qquad \mathbf{P}(0) = \mathbf{P}_0 \tag{26}$$

and define $\mathbf{A} = \mathbf{R}^{-1}\mathbf{P}_0$. *Then the first d components of* $\mathbf{A}$ *can be expressed as*

$$\mathbf{A}_d = \left(\begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \mathbf{P}_0 \tag{27}$$

Proof. Partition $\mathbf{A}$ and write the initial condition as

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{A}_d \\ \mathbf{A}_m \end{bmatrix} \tag{28}$$

Left multiplying with $\begin{bmatrix}\mathbf{L}_{dd}^T & \mathbf{L}_{md}^T\end{bmatrix}$ gives

$$\begin{bmatrix}\mathbf{L}_{dd}^T & \mathbf{L}_{md}^T\end{bmatrix}\mathbf{P}_0 = \begin{bmatrix}\mathbf{L}_{dd}^T & \mathbf{L}_{md}^T\end{bmatrix}\begin{bmatrix}\mathbf{R}_{dd} & \mathbf{R}_{dm}\\ \mathbf{R}_{md} & \mathbf{R}_{mm}\end{bmatrix}\begin{bmatrix}\mathbf{A}_d\\ \mathbf{A}_m\end{bmatrix} \tag{29}$$

$$= \begin{bmatrix}\mathbf{L}_{dd}^T\mathbf{R}_{dd} + \mathbf{L}_{md}^T\mathbf{R}_{md} & \mathbf{L}_{dd}^T\mathbf{R}_{dm} + \mathbf{L}_{md}^T\mathbf{R}_{mm}\end{bmatrix}\begin{bmatrix}\mathbf{A}_d\\ \mathbf{A}_m\end{bmatrix} \tag{30}$$

Using Lemma 1, we have $\mathbf{L}_{dd}^T\mathbf{R}_{dm} + \mathbf{L}_{md}^T\mathbf{R}_{mm} = \mathbf{0}_{dm}$. Then

$$\begin{bmatrix}\mathbf{L}_{dd}^T & \mathbf{L}_{md}^T\end{bmatrix}\mathbf{P}_0 = \left(\mathbf{L}_{dd}^T\mathbf{R}_{dd} + \mathbf{L}_{md}^T\mathbf{R}_{md}\right)\mathbf{A}_d \tag{31}$$

$$\implies \begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T\mathbf{P}_0 = \left(\begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_{dd}\\ \mathbf{R}_{md}\end{bmatrix}\right)\mathbf{A}_d \tag{32}$$

Because $\begin{bmatrix}\mathbf{R}_{dd}\\ \mathbf{R}_{md}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T$ are full rank d, $\left(\begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_{dd}\\ \mathbf{R}_{md}\end{bmatrix}\right)$ is invertible. Then

$$\mathbf{A}_d = \left(\begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_{dd}\\ \mathbf{R}_{md}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{L}_{dd}\\ \mathbf{L}_{md}\end{bmatrix}^T\mathbf{P}_0 \tag{33}$$

□

Lemma 3. *Follow InfoBox 4.1 and assume $\hat{\mathbf{R}}_{dd}$ is invertible. Then $\hat{\mathbf{R}}_{dd}$ is an eigenvector*

matrix of $\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}$. *i.e.,*

$$\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} = \hat{\mathbf{R}}_{dd}\Lambda_{dd}\hat{\mathbf{R}}_{dd}^{-1} \tag{34}$$

Proof. Partition the eigenvalue relation of $\hat{\mathbf{D}}$ as

$$\begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} \begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} \tag{35}$$

Comparing the top left-hand blocks of each product, we have

$$\hat{\mathbf{D}}_{dd}\hat{\mathbf{R}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md} = \hat{\mathbf{R}}_{dd}\Lambda_{dd} \tag{36}$$

Because $\hat{\mathbf{R}}_{dd}$ is invertible, right multiplying with $\hat{\mathbf{R}}_{dd}^{-1}$ gives

$$\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} = \hat{\mathbf{R}}_{dd}\Lambda_{dd}\hat{\mathbf{R}}_{dd}^{-1} \tag{37}$$

which is the eigenvalue relation of the matrix $\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}$. □

Proof of Theorem 2. Partition the eigenvalue relation of $\hat{\mathbf{D}}$ as

$$\begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} \begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} \tag{38}$$

Then using Lemma 3, we have the eigenrelation of $\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}$ as

$$\hat{\mathbf{D}}_{dd} + \hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} = \hat{\mathbf{R}}_{dd}\Lambda_{dd}\hat{\mathbf{R}}_{dd}^{-1} \tag{39}$$

Then the solution of the IVP (4.16), in terms of eigenvectors and eigenvalues, is given by

$$\mathbf{X}(t) = \hat{\mathbf{R}}_{dd}\, \exp(\Lambda_{dd}t)\, \hat{\mathbf{R}}_{dd}^{-1}\, \mathbf{X}(0) \qquad \forall t \geq 0 \tag{40}$$

where

$$\hat{\mathbf{R}}_{dd}^{-1}\mathbf{X}(0) = \hat{\mathbf{R}}_{dd}^{-1}\hat{\mathbf{R}}_{dd} \left(\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \hat{\mathbf{P}}_0 \tag{41}$$

$$= \left(\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T \hat{\mathbf{P}}_0 \tag{42}$$

Using $\begin{bmatrix} \hat{\mathbf{L}}_{dd} \\ \hat{\mathbf{L}}_{md} \end{bmatrix}^T = \begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \mathbf{T}^{-1}$, $\begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix}$, and $\hat{\mathbf{P}}_0 = \mathbf{T}\mathbf{P}_0$, we have

$$\hat{\mathbf{R}}_{dd}^{-1}\mathbf{X}(0) = \left(\begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \mathbf{T}^{-1}\mathbf{T} \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \mathbf{T}^{-1}\mathbf{T}\mathbf{P}_0 \tag{43}$$

$$= \left(\begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{L}_{dd} \\ \mathbf{L}_{md} \end{bmatrix}^T \mathbf{P}_0 \tag{44}$$

$$= \mathbf{A}_d \qquad \text{(using Lemma 2)} \tag{45}$$

Then the solution (40) can be written as

$$\mathbf{X}(t) = \hat{\mathbf{R}}_{dd} \exp(\Lambda_{dd} t)\mathbf{A}_d \qquad \forall t \geq 0 \tag{46}$$

Then

$$\mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \hat{\mathbf{R}}_{dd} \exp(\Lambda_{dd} t)\mathbf{A}_d \tag{47}$$

$$= \mathbf{T}^{-1} \begin{bmatrix} \hat{\mathbf{R}}_{dd} \\ \hat{\mathbf{R}}_{md} \end{bmatrix} \exp(\Lambda_{dd} t)\mathbf{A}_d \tag{48}$$

$$= \begin{bmatrix} \mathbf{R}_{dd} \\ \mathbf{R}_{md} \end{bmatrix} \exp(\Lambda_{dd} t)\mathbf{A}_d \tag{49}$$

$$= \tilde{\mathbf{P}} \tag{50}$$

Then using Theorem 1, for a given ϵ with $t_\epsilon = \frac{1}{\Re(\lambda_{d+1})}\ln(\epsilon)$, we have

$$\left\|\mathbf{P}(t) - \tilde{\mathbf{P}}(t)\right\|_\infty = \left\|\mathbf{P} - \mathbf{T}^{-1}\begin{bmatrix}\mathbf{I}_{dd}\\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}\end{bmatrix}\mathbf{X}(t)\right\|_\infty = \mathcal{O}(\epsilon)\,, \qquad \forall t \geq t_\epsilon \tag{51}$$

□

Lemma 4. *Consider the system*

$$\dot{\mathbf{P}}(t) = \mathbf{D}\mathbf{P}(t) \tag{52}$$

Suppose $\mathbf{1}_n^T\mathbf{P}(t) = c$ *for all time where* c *is any constant. Then*

$$\mathbf{1}_n^T\mathbf{D} = \mathbf{0}_n^T \tag{53}$$

Proof. We have the conservation relation

$$\mathbf{1}_n^T\mathbf{P}(t) = c \qquad\qquad \forall t \geq 0 \tag{54}$$

Taking the time derivative gives

$$\frac{d}{dt}\mathbf{1}_n^T\mathbf{P}(t) = \mathbf{1}_n^T\frac{d}{dt}\mathbf{P}(t) = 0 \qquad\qquad \forall t \geq 0 \tag{55}$$

Using $\frac{d}{dt}\mathbf{P}(t) = \mathbf{D}\mathbf{P}(t)$,

$$\mathbf{1}_n^T \mathbf{D}\mathbf{P}(t) = 0 \qquad \forall t \geq 0 \tag{56}$$

Because $\mathbf{P}$ is an n-dimensional vector, there are n linearly independent solutions possible. Assume $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$ are such n linearly independent solutions, Then

$$\mathbf{1}_n^T \mathbf{D} \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \ldots & \mathbf{P}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \ldots & 0 \end{bmatrix} = \mathbf{0}_n^T \tag{57}$$

Since $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$ are linearly independent,

$$\mathbf{1}_n^T \mathbf{D} = \mathbf{0}_n^T \tag{58}$$

□

Proof of Corollary 1. Consider the eigen-relation of $\mathbf{D}$ as

$$\mathbf{D}\mathbf{R}_i = \lambda_i \mathbf{R}_i \tag{59}$$

Taking one norm on both sides gives

$$\mathbf{1}_n^T \mathbf{D}\mathbf{R}_i = \lambda_i \mathbf{1}_n^T \mathbf{R}_i \tag{60}$$

Using Lemma 4, we have $\mathbf{1}_n^T\mathbf{D} = \mathbf{0}_n^T$. Then

$$\mathbf{0}_n^T\mathbf{R}_i = \lambda_i\mathbf{1}_n^T\mathbf{R}_i \tag{61}$$

$$\implies \quad \lambda_i\mathbf{1}_n^T\mathbf{R}_i = 0 \tag{62}$$

$$\implies \quad \mathbf{1}_n^T\mathbf{R}_i = 0\,, \qquad \text{if } \lambda_i \neq 0 \tag{63}$$

Using equations (47 - 49) from the proof of Theorem 2, we have

$$\mathbf{1}_n^T\tilde{\mathbf{P}}(t) = \mathbf{1}_n^T\mathbf{T}^{-1}\begin{bmatrix}\mathbf{I}_{dd}\\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1}\end{bmatrix}\mathbf{X}(t) = \mathbf{1}_n^T\begin{bmatrix}\mathbf{R}_{dd}\\ \mathbf{R}_{md}\end{bmatrix}\exp(\Lambda_{dd}t)\mathbf{A}_d \tag{64}$$

Then using (63), and assuming $k \leq d$ eigenvalues are equal to zero,

$$\mathbf{1}_n^T\tilde{\mathbf{P}}(t) = \begin{bmatrix}\mathbf{1}_k^T & \mathbf{0}_{d-k}^T\end{bmatrix}\left(\begin{bmatrix}\exp(\mathbf{0}_{k,k}\,t) & \mathbf{0}_{k,d-k}\\ \mathbf{0}_{d-k,k} & \exp(\Lambda_{d-k,d-k}\,t)\end{bmatrix}\right)\begin{bmatrix}\mathbf{A}_k\\ \mathbf{A}_{d-k}\end{bmatrix} \tag{65}$$

$$= \begin{bmatrix}\mathbf{1}_k^T & \mathbf{0}_{d-k}^T\end{bmatrix}\left(\begin{bmatrix}\mathbf{I}_{k,k} & \mathbf{0}_{k,d-k}\\ \mathbf{0}_{d-k,k} & \exp(\Lambda_{d-k,d-k}\,t)\end{bmatrix}\right)\begin{bmatrix}\mathbf{A}_k\\ \mathbf{A}_{d-k}\end{bmatrix} \tag{66}$$

$$= \begin{bmatrix}\mathbf{1}_k^T & \mathbf{0}_{d-k}^T\end{bmatrix}\begin{bmatrix}\mathbf{A}_k\\ \exp(\Lambda_{d-k,d-k}\,t)\mathbf{A}_{d-k}\end{bmatrix} \tag{67}$$

$$= \mathbf{1}_k^T\mathbf{A}_k\,, \qquad \forall t \tag{68}$$

Applying a similar argument to the exact solution $\mathbf{P}$ gives

$$\mathbf{1}_n^T \mathbf{P}(t) = \mathbf{1}_n^T \begin{bmatrix} \mathbf{R}_{k,k} & \mathbf{R}_{k,n-k} \\ \mathbf{R}_{n-k,k} & \mathbf{R}_{n-k,n-k} \end{bmatrix} \exp(\Lambda_{nn} t) \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_{n-k} \end{bmatrix} \tag{69}$$

$$= \begin{bmatrix} \mathbf{1}_k^T & \mathbf{0}_{n-k}^T \end{bmatrix} \left(\begin{bmatrix} \exp(\mathbf{0}_{k,k}\, t) & \mathbf{0}_{k,n-k} \\ \mathbf{0}_{n-k,k} & \exp(\Lambda_{n-k,n-k}\, t) \end{bmatrix} \right) \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_{n-k} \end{bmatrix} \tag{70}$$

$$= \begin{bmatrix} \mathbf{1}_k^T & \mathbf{0}_{n-k}^T \end{bmatrix} \left(\begin{bmatrix} \mathbf{I}_{k,k} & \mathbf{0}_{k,n-k} \\ \mathbf{0}_{n-k,k} & \exp(\Lambda_{n-k,n-k}\, t) \end{bmatrix} \right) \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_{n-k} \end{bmatrix} \tag{71}$$

$$= \begin{bmatrix} \mathbf{1}_k^T & \mathbf{0}_{n-k}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_k \\ \exp(\Lambda_{n-k,n-k}\, t) \mathbf{A}_{n-k} \end{bmatrix} \tag{72}$$

$$= \mathbf{1}_k^T \mathbf{A}_k\,, \qquad \forall t \tag{73}$$

Using (68) and (73), we have

$$\mathbf{1}_n^T \tilde{\mathbf{P}}(t) = \mathbf{1}^T \mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md} \hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) = \mathbf{1}_n^T \mathbf{P}(t) = \mathbf{1}^T \mathbf{P}_0\,, \qquad \forall t \tag{74}$$

□

Proof of Corollary 2. We have

$$\tilde{\mathbf{P}} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md} \hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) \tag{75}$$

Multiplying $\mathbf{1}_d^T \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{dm} \end{bmatrix} \mathbf{T}$ on both side gives

$$\mathbf{1}_d^T \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{dm} \end{bmatrix} \mathbf{T}\tilde{\mathbf{P}} = \mathbf{1}_d^T \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{0}_{dm} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{dd} \\ \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \end{bmatrix} \mathbf{X}(t) \tag{76}$$

$$\implies \mathbf{1}_n^T \tilde{\mathbf{P}}(t) = \mathbf{1}_d^T \mathbf{I}_{dd} \mathbf{X}(t) = \mathbf{1}_d^T \mathbf{X}(t) \tag{77}$$

Using Corollary 1, we have

$$\mathbf{1}_n^T \tilde{\mathbf{P}}(t) = \mathbf{1}_n^T \mathbf{P}_0 = \mathbf{1}_d^T \mathbf{X}(t) \tag{78}$$

□

Lemma 5. *Consider an invertible matrix* $\mathbf{E}$ *partitioned as*

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_{dd} & \mathbf{0}_{dm} \\ \mathbf{E}_{md} & \mathbf{E}_{mm} \end{bmatrix} \tag{79}$$

Then $\mathbf{E}_{dd}$ *and* $\mathbf{E}_{mm}$ *are invertible.*

Proof. Because $\mathbf{E}$ is a triangular block matrix, the determinant of $\mathbf{E}$ is equal to

$$\det(\mathbf{E}) = \det(\mathbf{E}_{dd})\det(\mathbf{E}_{mm}) \tag{80}$$

Because $\mathbf{E}$ has full rank

$$\det(\mathbf{E}) = \det(\mathbf{E}_{dd})\det(\mathbf{E}_{mm}) \neq 0 \tag{81}$$

Then

$$\det(\mathbf{E}_{dd}) \neq 0 \qquad \text{and} \qquad \det(\mathbf{E}_{mm}) \neq 0 \tag{82}$$

Then $\mathbf{E}_{dd}$ and $\mathbf{E}_{mm}$ are invertible. □

Corollary 3. *Suppose the assumptions in InfoBox 4.2 is true. Follow notations in InfoBox 4.1, assume* $\mathbf{L}_{dd}$ *is invertible. Consider an invertible matrix* $\mathbf{T}$ *defined (in partitioned form) as*

$$\mathbf{T} = \begin{bmatrix} \mathbf{B}_{dd}\mathbf{L}_{dd}^T & \mathbf{B}_{dd}\mathbf{L}_{dm}^T \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \quad . \tag{83}$$

where $\mathbf{B}_{dd}$ *is any invertible matrix. Let* $\hat{\mathbf{D}} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}$. *Note that* $\hat{\mathbf{R}} = \mathbf{T}\mathbf{R}$ *is a right eigenmatrix of* $\hat{\mathbf{D}}$. *, and left eigenmatrix of* $\hat{\mathbf{D}}$ *is defined as* $\hat{\mathbf{L}} = \mathbf{T}^{-T}\mathbf{L}$ *Partition these as*

$$\hat{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix} \qquad \hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} \qquad \hat{\mathbf{L}} = \begin{bmatrix} \hat{\mathbf{L}}_{dd} & \hat{\mathbf{L}}_{dm} \\ \hat{\mathbf{L}}_{md} & \hat{\mathbf{L}}_{mm} \end{bmatrix} \tag{84}$$

Then

$$\hat{\mathbf{D}}_{dm} = \mathbf{0}_{dm} \qquad \hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm} \qquad \hat{\mathbf{L}}_{md} = \mathbf{0}_{md} \tag{85}$$

Proof. By definition, we have

$$\begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{dd}\mathbf{L}_{dd}^T & \mathbf{B}_{dd}\mathbf{L}_{dm}^T \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \tag{86}$$

Comparing top left hand blocks, we have

$$\hat{\mathbf{R}}_{dm} = \mathbf{B}_{dd}\mathbf{L}_{dd}^T\mathbf{R}_{dm} + \mathbf{B}_{dd}\mathbf{L}_{md}^T\mathbf{R}_{mm} \tag{87}$$

$$= \mathbf{B}_{dd}(\mathbf{L}_{dd}^T\mathbf{R}_{dm} + \mathbf{L}_{md}^T\mathbf{R}_{mm}) \tag{88}$$

Using Lemma 1, we have $\mathbf{L}_{dd}^T\mathbf{R}_{dm} + \mathbf{L}_{md}^T\mathbf{R}_{mm} = \mathbf{0}_{dm}$. Then

$$\hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm} \tag{89}$$

Because $\mathbf{T}$ and $\mathbf{R}$ are invertible matrices, $\hat{\mathbf{R}} = \mathbf{TR}$ is also invertible. Then from Lemma 5 using (89), the blocks $\hat{\mathbf{R}}_{dd}$ and $\hat{\mathbf{R}}_{mm}$ are invertible.

Next, using the Lemma 1 on the left ($\hat{\mathbf{L}}$) and right ($\hat{\mathbf{R}}$) eigenvectors of $\hat{\mathbf{D}}$, we have

$$\hat{\mathbf{L}}_{dd}^T\hat{\mathbf{R}}_{dm} + \hat{\mathbf{L}}_{md}^T\hat{\mathbf{R}}_{mm} = \mathbf{0}_{dm} \tag{90}$$

$$\hat{\mathbf{L}}_{md}^T\hat{\mathbf{R}}_{mm} = \mathbf{0}_{dm} \qquad (\because\ \hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm}) \tag{91}$$

$$\hat{\mathbf{L}}_{md}^T = \mathbf{0}_{dm} \qquad (\because\ \hat{\mathbf{R}}_{mm} \text{ is invertible}) \tag{92}$$

Since $\mathbf{L}$ and $\mathbf{T}$ are invertible matrices, $\hat{\mathbf{L}}$ is also invertible. Then using Lemma 5, we have $\hat{\mathbf{L}}_{dd}$ and $\hat{\mathbf{L}}_{mm}$ are invertible.

Next, by definition, $\mathbf{DR} = \mathbf{R}\Lambda$. Applying transformation $\mathbf{T}$, we have $\hat{\mathbf{D}}\hat{\mathbf{R}} = \hat{\mathbf{R}}\Lambda$. In the partitioned form, and using (89), this is

$$\begin{bmatrix} \hat{\mathbf{D}}_{dd} & \hat{\mathbf{D}}_{dm} \\ \hat{\mathbf{D}}_{md} & \hat{\mathbf{D}}_{mm} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \mathbf{0}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \mathbf{0}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} \begin{bmatrix} \Lambda_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} \tag{93}$$

Comparing top right blocks of the product gives

$$\hat{\mathbf{D}}_{dm}\hat{\mathbf{R}}_{mm} = \mathbf{0}_{dm} \tag{94}$$

$$\implies \quad \hat{\mathbf{D}}_{dm} = \mathbf{0}_{dm} \qquad \text{(because } \hat{\mathbf{R}}_{mm} \text{ is invertible)} \tag{95}$$

□

Lemma 6. *Suppose the assumptions in InfoBox 4.2 are true and the matrix* $\mathbf{D}$ *is re-indexed such that the sub-matrix* $\mathbf{L}_{dd}$ *in* $\mathbf{L}$ *is invertible . Then*

$$\mathbf{1}_d^T \mathbf{L}_{dd}^{-T} \mathbf{L}_{md}^T = -\mathbf{1}_d^T \mathbf{R}_{dm} \mathbf{R}_{mm}^{-1} = \mathbf{1}_m^T \tag{96}$$

Proof. Consider an invertible matrix constructed in the partitioned form as $\begin{bmatrix} \mathbf{L}_{dd}^T & \mathbf{L}_{md}^T \\ \mathbf{0}_{dm}^T & \mathbf{I}_{mm} \end{bmatrix}$. Then

$$\begin{bmatrix} \mathbf{L}_{dd}^T & \mathbf{L}_{md}^T \\ \mathbf{0}_{dm}^T & \mathbf{I}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{dd}^T \mathbf{R}_{dd} + \mathbf{L}_{dm}^T \mathbf{R}_{md} & \mathbf{L}_{dd}^T \mathbf{R}_{dm} + \mathbf{L}_{dm}^T \mathbf{R}_{mm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \tag{97}$$

Using Lemma 1, we have

$$\mathbf{L}_{dd}^T \mathbf{R}_{dm} + \mathbf{L}_{dm}^T \mathbf{R}_{mm} = \mathbf{0}_{dm} \tag{98}$$

Since the top right block is equal to $\mathbf{0}_{dm}$, using Lemma 5, we have $\mathbf{R}_{mm}$ invertible. Then

$$\mathbf{L}_{dm}^T \mathbf{R}_{mm} = -\mathbf{L}_{dd}^T \mathbf{R}_{dm} \tag{99}$$

$$\implies \quad \mathbf{L}_{dd}^{-T} \mathbf{L}_{md}^T = -\mathbf{R}_{dm} \mathbf{R}_{mm}^{-1} \tag{100}$$

Next, consider the right eigenvalue relation

$$\mathbf{D}\mathbf{R} = \mathbf{R}\Lambda \tag{101}$$

Left multiplying $\mathbf{1}_n^T$ both sides of the right eigenvalue relation gives

$$\mathbf{1}_n^T \mathbf{D}\mathbf{R} = \mathbf{1}_n^T \mathbf{R}\Lambda \tag{102}$$

Using Lemma 4, we have $\mathbf{1}^T\mathbf{D} = \mathbf{0}^T$. Then

$$\mathbf{0}_n^T = \mathbf{1}_n^T \mathbf{R}\Lambda \tag{103}$$

Partitioning $\mathbf{R}$ and Λ gives

$$\begin{bmatrix} \mathbf{1}_d^T & \mathbf{1}_m^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dm} \\ \mathbf{R}_{md} & \mathbf{R}_{mm} \end{bmatrix} \begin{bmatrix} \Lambda_{dd} & 0 \\ 0 & \Lambda_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_d^T & \mathbf{0}_m^T \end{bmatrix} \tag{104}$$

$$\implies \quad \begin{bmatrix} (\mathbf{1}_d^T \mathbf{R}_{dd} + \mathbf{1}_m^T \mathbf{R}_{md})\Lambda_{dd} & (\mathbf{1}_d^T \mathbf{R}_{dm} + \mathbf{1}_m^T \mathbf{R}_{mm})\Lambda_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_d^T & \mathbf{0}_m^T \end{bmatrix} \tag{105}$$

Then

$$(\mathbf{1}_d^T \mathbf{R}_{dm} + \mathbf{1}_m^T \mathbf{R}_{mm})\Lambda_{mm} = \mathbf{0}_m^T \tag{106}$$

Using the invertibility of Λ_{mm}, we have

$$\mathbf{1}_d^T \mathbf{R}_{dm} + \mathbf{1}_m^T \mathbf{R}_{mm} = \mathbf{0}_m^T \tag{107}$$

$$\implies \quad -\mathbf{1}_d^T \mathbf{R}_{dm} = \mathbf{1}_m^T \mathbf{R}_{mm} \tag{108}$$

Using the invertibility of $\mathbf{R}_{mm}$, we have

$$-\mathbf{1}_d^T \mathbf{R}_{dm} \mathbf{R}_{mm}^{-1} = \mathbf{1}_m^T \tag{109}$$

Then using eq. (100)

$$\mathbf{1}_d^T \mathbf{L}_{dd}^{-T} \mathbf{L}_{md}^T = -\mathbf{1}_d^T \mathbf{R}_{dm} \mathbf{R}_{mm}^{-1} = \mathbf{1}_m^T \tag{110}$$

□

Lemma 7. *Let* $\mathbf{D}$ *be a matrix partitioned as*

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{dd} & \mathbf{D}_{dm} \\ \mathbf{D}_{md} & \mathbf{D}_{mm} \end{bmatrix}. \tag{111}$$

Suppose $\mathbf{D}_{mm}$ *is invertible and* $\mathbf{1}^T\mathbf{D} = \mathbf{0}^T$*, then*

$$-\mathbf{1}_d^T\mathbf{D}_{dm}\mathbf{D}_{mm}^{-1} = \mathbf{1}_m^T \tag{112}$$

Proof. We have

$$\begin{bmatrix} \mathbf{1}_d^T & \mathbf{1}_m^T \end{bmatrix} \begin{bmatrix} \mathbf{D}_{dd} & \mathbf{D}_{dm} \\ \mathbf{D}_{md} & \mathbf{D}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_d^T & \mathbf{0}_m^T \end{bmatrix} \tag{113}$$

Then comparing both sides gives

$$\mathbf{1}_d^T\mathbf{D}_{dm} + \mathbf{1}_m^T\mathbf{D}_{mm} = \mathbf{0}_m^T \tag{114}$$

$$\mathbf{1}_d^T\mathbf{D}_{dm} = -\mathbf{1}_m^T\mathbf{D}_{mm} \tag{115}$$

Using the invertibility of $\mathbf{D}_{mm}$, we have

$$-\mathbf{1}_d^T\mathbf{D}_{dm}\mathbf{D}_{mm}^{-1} = \mathbf{1}_m^T \tag{116}$$

□

Lemma 8. *Consider a matrix* $\mathbf{F}$ *and its invertible right and left eigenmatrix* $\mathbf{L}$ *and* $\mathbf{R}$

satisfying the condition

$$\mathbf{F}^T\mathbf{L} = \mathbf{L}\Lambda \tag{117}$$

Also consider an invertible matrix $\mathbf{T}$ *such that* $\hat{\mathbf{F}} = \mathbf{T}\mathbf{F}\mathbf{T}^{-1}$ *and right eigenmatrix satisfying*

$$\hat{\mathbf{F}}\hat{\mathbf{R}} = \hat{\mathbf{R}}\Lambda \tag{118}$$

Consider these matrices in partitioned form as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{dd} & \mathbf{F}_{dm} \\ \mathbf{F}_{md} & \mathbf{F}_{mm} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix}, \quad \hat{\mathbf{F}} = \begin{bmatrix} \hat{\mathbf{F}}_{dd} & \hat{\mathbf{F}}_{dm} \\ \hat{\mathbf{F}}_{md} & \hat{\mathbf{F}}_{mm} \end{bmatrix}, \tag{119}$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{dd} & \mathbf{L}_{dm} \\ \mathbf{L}_{md} & \mathbf{L}_{mm} \end{bmatrix}, \quad \hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \hat{\mathbf{R}}_{dm} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \mathbf{0}_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} \quad . \tag{120}$$

where $\mathbf{F}_{mm}$, $\mathbf{R}_{dd}$ *and* $\mathbf{L}_{dd}$ *are invertible. Then the following are true:*

1. $\mathbf{L}_{dd}^{-T}\mathbf{L}_{md}^{T} = -\mathbf{F}_{dm}\mathbf{F}_{mm}^{-1}$
2. $\hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} = \hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md}$
3. $\hat{\mathbf{F}}_{dm} = \mathbf{0}_{dm}$
4. $\hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm}$
5. $\hat{\mathbf{L}}_{md} = \mathbf{0}_{md}$
6. $\hat{\mathbf{R}}_{dd}$, $\hat{\mathbf{R}}_{mm}$, $\hat{\mathbf{L}}_{dd}$, *and* $\hat{\mathbf{L}}_{mm}$ *are invertible*

Proof. Consider the partial left eigenvalue relation of $\mathbf{F}$ in partitioned form as

$$\begin{bmatrix} \mathbf{L}_{dd}^T & \mathbf{L}_{md}^T \end{bmatrix} \begin{bmatrix} \mathbf{F}_{dd} & \mathbf{F}_{dm} \\ \mathbf{F}_{md} & \mathbf{F}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{dd} & \mathbf{0}_{md} \end{bmatrix} \tag{121}$$

Then

$$\mathbf{L}_{dd}^T \mathbf{F}_{dm} + \mathbf{L}_{md}^T \mathbf{F}_{mm} = \mathbf{0}_{md} \tag{122}$$

$$-\mathbf{L}_{dd}^T \mathbf{F}_{dm} = \mathbf{L}_{md}^T \mathbf{F}_{mm} \tag{123}$$

Using invertibility of $\mathbf{F}_{mm}$ and $\mathbf{L}_{dd}$, we have

$$-\mathbf{F}_{dm} \mathbf{F}_{mm}^{-1} = \mathbf{L}_{dd}^T \mathbf{L}_{md}^T \tag{124}$$

Then the transformation matrix $\mathbf{T}$ can be written in partitioned form as

$$\begin{bmatrix} \mathbf{I}_{dd} & -\mathbf{F}_{dm} \mathbf{F}_{mm}^{-1} \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{dd} & \mathbf{L}_{dd}^{-T} \mathbf{L}_{md}^T \\ \mathbf{0}_{md} & \mathbf{I}_{mm} \end{bmatrix} \tag{125}$$

Then using Corollary 3 for $\hat{\mathbf{F}}$, we have

$$\hat{\mathbf{F}}_{dm} = \mathbf{0}_{dm} \qquad \hat{\mathbf{R}}_{dm} = \mathbf{0}_{dm} \qquad \hat{\mathbf{L}}_{md} = \mathbf{0}_{md} \tag{126}$$

Then using the invertibility of $\mathbf{L}$, $\mathbf{R}$, and $\mathbf{T}$ and Lemma 5, $\hat{\mathbf{R}}_{dd}$, $\hat{\mathbf{R}}_{mm}$, $\hat{\mathbf{L}}_{dd}$, and $\hat{\mathbf{L}}_{mm}$ are invertible.

Next, consider the right eigenvalue relationship of $\hat{\mathbf{F}}$ in partitioned form as

$$\begin{bmatrix} \hat{\mathbf{F}}_{dd} & \hat{\mathbf{F}}_{dm} \\ \hat{\mathbf{F}}_{md} & \hat{\mathbf{F}}_{mm} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \mathbf{0}_{md} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}}_{dd} & \mathbf{0}_{md} \\ \hat{\mathbf{R}}_{md} & \hat{\mathbf{R}}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{dd} & \mathbf{0}_{dm} \\ \mathbf{0}_{md} & \Lambda_{mm} \end{bmatrix} \tag{127}$$

Then $\hat{\mathbf{F}}_{mm}\hat{\mathbf{R}}_{mm} = \hat{\mathbf{R}}_{mm}\Lambda_{mm}$. Since Λ_{mm} is a diagonal matrix with non-zero values on the diagonal, Λ_{mm} is invertible. Then using invertibility of $\hat{\mathbf{R}}_{mm}$, we have $\hat{\mathbf{F}}_{mm}$ invertible.

Next, comparing bottom left block of the product 127, we have

$$\hat{\mathbf{F}}_{md}\hat{\mathbf{R}}_{dd} + \hat{\mathbf{F}}_{mm}\hat{\mathbf{R}}_{md} = \mathbf{0}_{md} \tag{128}$$

$$\implies \quad -\hat{\mathbf{F}}_{md}\hat{\mathbf{R}}_{dd} = \hat{\mathbf{F}}_{mm}\hat{\mathbf{R}}_{md} \tag{129}$$

Using invertibility of $\hat{\mathbf{F}}_{mm}$ and $\hat{\mathbf{R}}_{dd}$, we have

$$-\hat{\mathbf{F}}_{mm}^{-1}\hat{\mathbf{F}}_{md} = \hat{\mathbf{R}}_{md}\hat{\mathbf{R}}_{dd}^{-1} \tag{130}$$

□

References

[1] Eberhard Voit. *A first course in systems biology*. Garland Science, 2017.

[2] BP Ingalls. Mathematical modelling in systems biology: An introduction. *J Chem Inf Model*, 53(9):17–42, 2014.

[3] Edda Klipp, Wolfram Liebermeister, Christoph Wierling, Axel Kowald, and Ralf Herwig. *Systems biology: a textbook*. John Wiley & Sons, 2016.

[4] MR Roussel and R Zhu. Exactly reduced chemical master equations. In *Model Reduction and Coarse-Graining Approaches for Multiscale Phenomena*, pages 295–315. Springer, 2006.

[5] Nesma ElKalaawy and Amr Wassal. Methodologies for the modeling and simulation of biochemical networks, illustrated for signal transduction pathways: A primer. *Biosystems*, 129:1–18, 2015.

[6] Daniel T Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.

[7] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[8] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003.

[9] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Efficient step size selection for the tau-leaping simulation method. *The Journal of chemical physics*, 124(4):044109, 2006.

[10] Stewart N Ethier and Thomas G Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.

[11] Daniel T Gillespie. The chemical langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[12] Joao Pedro Hespanha and Abhyudai Singh. Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 15(15):669–689, 2005.

[13] Carlos A Gomez-Uribe and George C Verghese. Mass fluctuation kinetics: Capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. *The Journal of chemical physics*, 126(2):024109, 2007.

[14] NG Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 2011.

[15] David Schnoerr, Guido Sanguinetti, and Ramon Grima. Approximation and inference methods for stochastic biochemical kinetics-a tutorial review. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093001, 2017.

[16] William J Blake, Mads Kærn, Charles R Cantor, and James J Collins. Noise in eukaryotic gene expression. *Nature*, 422(6932):633, 2003.

[17] Jonathan M Raser and Erin K O'shea. Control of stochasticity in eukaryotic gene expression. *science*, 304(5678):1811–1814, 2004.

[18] Leor S Weinberger, John C Burnett, Jared E Toettcher, Adam P Arkin, and David V Schaffer. Stochastic gene expression in a lentiviral positive-feedback loop: Hiv-1 tat fluctuations drive phenotypic diversity. *Cell*, 122(2):169–182, 2005.

[19] Michael Samoilov, Sergey Plyasunov, and Adam P Arkin. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences*, 102(7):2310–2315, 2005.

[20] Crispin Gardiner. *Stochastic methods*, volume 4. springer Berlin, 2009.

[21] Daniel T Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425, 1992.

[22] Werner Dubitzky, Olaf Wolkenhauer, Hiroki Yokota, and Kwang-Hyun Cho. *Encyclopedia of systems biology*. Springer Publishing Company, Incorporated, 2013.

[23] Darren J Wilkinson. *Stochastic Modelling for Systems Biology*. CRC Press, 2011.

[24] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of chemical physics*, 124(4):044104, 2006.

[25] Khanh N Dinh and Roger B Sidje. Understanding the finite state projection and related methods for solving the chemical master equation. *Physical biology*, 13(3):035003, 2016.

[26] Kevin Burrage, MARKUS Hegland, Shev Macnamara, Roger Sidje, et al. A krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In *Proc. of The AA Markov 150th Anniversary Meeting*, 2006. Paper no. 21-37.

[27] Brian Munsky and Mustafa Khammash. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *Journal of Computational Physics*, 226(1):818–835, 2007.

[28] Slaven Peleš, Brian Munsky, and Mustafa Khammash. Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of chemical physics*, 125(20):204104, 2006.

[29] I Oppenheim, KE Shuler, and GH Weiss. Stochastic and deterministic formulation of chemical rate equations. *The Journal of Chemical Physics*, 50(1):460–466, 1969.

[30] Thomas G Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972.

[31] Donald A McQuarrie. Stochastic approach to chemical kinetics. *Journal of applied probability*, 4(3):413–478, 1967.

[32] DT Gillespie. Markov processes: An introduction for physical scientists (academic, new york, 1992). *Google Scholar*, pages 111–122, 2009.

[33] Roger B Sidje. Expokit: a software package for computing matrix exponentials. *ACM Transactions on Mathematical Software (TOMS)*, 24(1):130–156, 1998.

[34] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[35] Awad H Al-Mohy and Nicholas J Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing*, 33(2):488–511, 2011.

[36] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*, volume 66. Siam, 2011.

[37] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[38] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of chemical physics*, 122(1):014116, 2005.

[39] Yang Cao and Linda Petzold. Slow-scale tau-leaping method. *Computer methods in applied mechanics and engineering*, 197(43-44):3472–3479, 2008.

[40] Markus Hegland, Conrad Burden, Lucia Santoso, Shev MacNamara, and Hilary Booth. A solver for the stochastic master equation applied to gene regulatory networks. *Journal of computational and applied mathematics*, 205(2):708–724, 2007.

[41] Xingye Kan, Chang Hyeong Lee, and Hans G Othmer. A multi-time-scale analysis of chemical reaction networks: Ii. stochastic systems. *Journal of mathematical biology*, 73(5):1081–1129, 2016.

[42] Stefan Schuster and Thomas Höfer. Determining all extreme semi-positive conservation relations in chemical reaction systems: a test criterion for conservativity. *Journal of the Chemical Society, Faraday Transactions*, 87(16):2561–2566, 1991.

[43] Fernando López-Caamal and Tatiana T Marquez-Lago. Order reduction of the chemical master equation via balanced realisation. *PloS one*, 9(8):e103521, 2014.

[44] RB Lehoucq, K Maschhoff, D Sorensen, and C Yang. Arpack software package. *Rice University*, 1996.

[45] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. Siam, 1998.

[46] RN Mohan. On orthogonalities in matrices. *arXiv preprint cs/0605045*, 2006.

[47] N Trefethen Lloyd. Numerical linear algebra/lloyd n. trefethen, david bau. *Philadelphia, USA: Society for Industrial and Applied Mathematics*, 263, 1997.

[48] Shev MacNamara, Kevin Burrage, and Roger B Sidje. Multiscale modeling of chemical kinetics via the master equation. *Multiscale Modeling & Simulation*, 6(4):1146–1168, 2008.

[49] Roger B Sidje and Huy D Vo. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. *Mathematical biosciences*, 269:10–16, 2015.