Explicit Time Integration Methods Based on Simplified Stochastic Computational Singular Perturbation

by

Yung-Chieh Chang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 5, 2018

Keywords: chemical Langevin equations (CLEs), explicit time integration, stochastic computational singular perturbation (SCSP), stiff chemical reaction systems

Approved by

Xiaoying Han, Chair, Professor of Mathematics and Statistics
Yanzhao Cao, Co-chair, Professor of Mathematics and Statistics
Junshan Lin, Assistant Professor of Mathematics and Statistics
Hans-Werner van Wyk, Assistant Professor of Mathematics and Statistics
Habib Najm, Sandia National Laboratories

Abstract

The research presented in this dissertation focuses on the development of efficient explicit time integration schemes for the chemical Langevin equations (CLEs). Due to the presence of multiple time scales in complex chemical reaction networks, CLEs often involve stiffness and hence classical explicit time integration methods require extremely small step sizes to maintain numerical stability. The methodology proposed in this research is based on the concept of stochastic computational singular perturbation, which separates the fast and slow dynamics of an underlying stiff stochastic differential equation system by projecting the drift and diffusion onto appropriate sets of basis. The CLE system can then be integrated forward in two steps, one for the slow dynamics and the other for the fast dynamics which can be approximated by a stochastic algebraic relation, and thus admits much larger step sizes than classical explicit schemes. The schemes are applied to a stiff chemical reaction system involving 3 species and 6 reaction channels. Numerical experiments show significant improvement in computational efficiency compared to classical explicit methods while maintaining numerical stability.

Acknowledgments

It is not easy task to me to complete this dissertation on every level. I would like to take this opportunity to thank everyone who have supported and helped me so much throughout this period.

I would first like to thank my advisors, Dr. Xiaoying Han and Dr. Yanzhou Cao. Thank you for your wise guidance, direction, kindness and encouragement to help me and inspire me throughout these years. I will be forever grateful.

To the members of my committee, Dr. Junshan Lin, Dr. Habib Najm, Dr. Hans Werner Van Wyk, and Dr. Yang Zhou, thank you for helping and providing suggestions.

Next, I would like to thank all my Auburn friends. Thank you for making my time so delightful. I will cherish the memory forever.

I would also like to thank my family in Taiwan. Thanks for your support and always believing in me.

To my husband Bin, thank you for your love and unconditional support. Without you, I wouldn't be me. To my three-year-old son Ryan, thank you for teaching me to accept joy and let go of fear. You are simply the best.

Table of Contents

Abstract ii

Acknowledgments iii

1 Introduction: Chemical Reactions 1

1.1 Reaction Rate Equation about Concentration 3

1.1.1 Example 1: One Unimolecular Reaction 5

1.1.2 Example 2: One Bimolecular Reaction on the Same Species 6

1.1.3 Example 3: One Bimolecular Reaction on Different Species 6

1.1.4 Example 4: Multiple Reactions 7

1.1.5 Note 7

1.2 Propensity Function 8

1.2.1 Unimolecular reaction 8

1.2.2 Bimolecular reaction on different species 9

1.2.3 Same species bimolecular reaction 9

1.2.4 Note 9

1.3 Chemical Master Equation 10

1.3.1 Example 10

1.3.2 Note 12

1.4 Reaction Rate Equation Deduced by Propensity Functions 12

1.5 Stochastic Simulation Algorithm 13

1.6 Tau-Leaping Method 14

1.7 Chemical Langevin Equation 16

1.8 Remarks 18

2 Computational Singular Perturbation and Stochastic Computational Singular Perturbation 20

2.1 Stiff Problems and Singular Perturbation 20

2.2 Computational Singular Perturbation 21

2.3 Stochastic Computational Singular Perturbation 22

2.4 Remarks 26

3 Simplified Stochastic Computational Singular Perturbation 27

3.1 Simplified Stochastic Computational Singular Perturbation (SSCSP) 27

3.2 Time integration: Special Case with No Noise 29

3.3 Time Integration: Stiff SDE System 32

3.3.1 Stratonovich SDE with Only One Noise Term 32

3.3.2 Stratonovich SDE with Multiple Noise Terms 36

3.3.3 Two-Step Time Integration Scheme 37

3.3.4 The Simplified Stochastic Computational Singular Perturbation Algorithm for Stratonovich SDE 38

3.3.5 Modification on SSCSP to Test Variance 40

3.3.6 The SSCSP for Stratonovich SDE with Contribution from Fast Stochastic Modes 40

3.3.7 Remarks 41

3.4 SSCSP for Itô SDE 41

3.4.1 SSCSP Algorithm for Itô SDE 43

3.4.2 Note 44

4 Numerical Experiment: The VG model 46

4.1 Reation Rate Equation 48

4.2 Chemical Langevin Equation of VG Model (Itô SDE) 49
4.3 Chemical Langevin Equation of VG Model (Stratonovich SDE) 52
4.4 Jacobian matrices for SSCSP 53
4.5 SSCSP for VG Model 55
4.5.1 Matrix Form of the Algorithm 56
4.6 Numerical Simulation 59
4.6.1 Stratonovich SDE 59
4.6.2 Itô SDE 60

5 Summary 66

Appendices 75

A Newton's Method on Performing Stratonovich Integration 76
A.1 Multivariate Newton's Method (Newton-Raphson Method) 76
A.2 SDE in Stratonovich Sense 77
A.3 Note 78

B Recalculation on Updating Numbers of Species Due to Reaching Unrealistic Values . 79
B.1 The Algorithm 79
B.2 Remark 80

List of Figures

1.1 Logical structure of stochastic chemical kinetics [1] 18

4.1 This graph shows the slow manifold (yellow curve) which is the intersection of two surfaces: $Z = XY$ (red) and $2X = Y(Y-1)$ (green). 49

4.2 The numerical simmulations (colorful curves) of (4.19) compare with the slow manifold. 50

4.3 X, Y, Z values by Euler and SSCSP for Stratonovich SDE. 61

4.4 X, Y, Z values by Euler and SSCSP for Stratonovich SDE. 62

4.5 X, Y, Z values by Euler and SSCSP for Itô SDE. 64

List of Tables

Chapter 1

Introduction: Chemical Reactions

Consider a chemical reaction system of $N \geq 1$ chemical species $\{S_1, \cdots, S_N\}$ which interact through $M \geq 1$ chemical reactions $\{R_1, \cdots, R_M\}$ inside some fixed volume Ω which is well-stirred and at a constant temperature uniform over Ω. For each chemical reation R_j with chemical equation

$$R_j : \sum_{i=1}^{N} \alpha_{ij} S_i \to \sum_{i=1}^{N} \beta_{ij} S_i, \quad j = 1 \cdots M, \tag{1.1}$$

and stoichiometric coefficients $\alpha_{ij}, \beta_{ij} \in \mathbb{Z}^+$, define the *state-change vector* (or *stoichiometric vector*) $\mathbf{v}_j = \{v_{j1}, \cdots, v_{jN}\}$ with its ith component represented by

$$\begin{aligned} v_{ji} = \beta_{ij} - \alpha_{ij} \equiv & \text{ the change in the number of } S_i \\ & \text{molecules produced by one } R_j \text{ reaction} \\ & (j = 1, \cdots, M; i = 1, \cdots, N). \end{aligned} \tag{1.2}$$

We denote the concentration of species S_i at time t by $X_i(t)$. Our goal is to estimate the evolution of $\mathbf{X}(t) = \{X_1, \cdots, X_N\}$, given the initial state of the system $\mathbf{X}(t_0) = \{X_1(t_0), \cdots, X_N(t_0)\}$ at some time t_0.

To study a chemical reaction system along time without tracking positions and velocities of all the molecules, we introduce the *reaction rate* of a chemical reaction [1, 2, 3, 4].

The reaction rate can be used to describe the trajectories of the concentration X_i of each chemical species S_i evolving in time by a set of coupled ordinary differential equations (ODEs) of the form

$$\frac{dX_i}{dt} = f_i(X_1, \cdots, X_N), \quad i = 1, \cdots, N, \tag{1.3}$$

which is called *reaction rate equations (RREs)* with the function f_i defined by the chemical reactions in the system.

In general, some reactions are naturally faster than others in the same chemical system, resulting in a system of *stiff* ODEs/RREs. The behaviour of stiff ODEs is like differential-algebraic equations (DAEs), in the sense that their solution dynamics exhibit underlying algebraic manifolds. Fast time-scales are "exhausted" rapidly, and solutions move on lower-dimensional eventually, attracting, and invariant "slow manifolds" that characterize the long-term process dynamics. These manifolds are an expression of the partial-equilibration of fast time scales, being characterized by algebraic relationships among a subset of the species. Thus, a smaller *differential* "slow" sub-system is necessary to approximate the full system state when the dynamics approach this manifold.

While in general a stiff chemical reaction system evolves many different time scales [5, 6, 7, 8, 9, 10], theoretical approaches usually assume a stiff system consists of two time-scales: *rapid* and *slow*. The rapid time-scales decay quickly, and the solutions tend to be attracted to a lower-dimensional "slow-manifold" which describes the process in the long-term. To understand stiff systems, identifying slow-manifolds by mathematical analysis is an effective first step to study the system dynamics [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. Moreover, these manifolds can also lead to study the projection of ODEs onto rapid and slow subdomains, and inspire for explicit time integration methods for stiff ODE systems [29, 13].

Computational Singular Perturbation (CSP), which uses automated computational procedures to obtain physical insights on massively complex reaction systems, is an effective method to discover low-dimensional manifolds of underlying ODE systems [14, 11, 29, 30, 2, 31]. It is widely used for chemical kinetics problems. CSP can be used to identify chemical species controlled by different reaction rates and to obtain a reduced system and better approximations.

The CSP method is a powerful tool for analyzing deterministic stiff ODE systems; it serves as an effective model reduction technique and inspires explicit time integration strategies. However, applications in chemical kinetics at small scales are usually presented by the interpretation of a stochastic process governed by random jumps, and ODE systems only represent the average results [32]. Fundamentally, chemical reactions are stochastic, as they are the result of random collisions among particles (molecules, atoms). However, the relevance of stochastic motions depends on the scale of the system being considered. When the system size and particle counts are at the macro-scale, stochastic effects are averaged out and the system can be approximated by a family of coupled deterministic ODEs, which are also RREs. An intermediate meso-scale regime exists when particle counts are large, *e.g., O*(10^3*)*, but not sufficiently so to make the continuum approximation viable. In this regime, the Fokker-Planck equation can be used to simulate the evolution of the probability density function (PDF) of states, or the chemical Langevin equation (CLE) can be used to simulate state trajectories [32].

More precisely, when the size of an chemical reaction system is small, e.g., it is governed by the chemical Master equation (CME).

For chemical reaction systems based on jump Markov processes, fast occurring reaction can be eliminated by quasi-steady-state approximation, and further proceed from reduced CME [33, 34]. However, CME can only be described or solved analytically except for some simple cases. Thus, for a chemical system that consists of fast and slow time-scales, we are interested in the SDEs which also interpret stochastic chemical kinetics problems properly.

1.1 Reaction Rate Equation about Concentration

As an introductory example, consider a chemical reaction with three species X, Y, and Z:

$$2X + Y \xrightarrow{k} Z, \tag{1.4}$$

where k represents the reaction rate of this chemical reaction. It means that two molecules of X combining with one molecule of Y can form one molecule of Z. The rate of decrease of

concentration of X is twice the rate of decrease of concentration of Y. We can describe the relation as follows [4]:

$$\frac{d[X]}{dt} = 2 \cdot \frac{d[Y]}{dt}, \tag{1.5}$$

note that the $[X]$ and $[Y]$ in (1.5) mean the concentration of species X and Y, respectively.

Moreover, the absolute rate of change of concentrations of Y and Z are the same, but Y is decreasing and Z is increasing. Thus, we got the following relation:

$$\frac{d[Z]}{dt} = -\frac{d[Y]}{dt}, \tag{1.6}$$

where $[Z]$ and $[Y]$ are the concentrations of species Z and Y, respectively. Summarizing (1.5) and (1.6), we get

$$-\frac{1}{2}\frac{d[X]}{dt} = -\frac{d[Y]}{dt} = \frac{d[Z]}{dt}. \tag{1.7}$$

Next consider a more general chemical reaction with species X, Y, Z of the form

$$\alpha X + \beta Y \xrightarrow{k} \gamma Z, \tag{1.8}$$

where $\alpha, \beta,$ and γ are positive stoichiometric coefficients describing the number of species involved in one reaction. The governing equations for (1.8) read:

$$v(t) = -\frac{1}{\alpha}\frac{d[X]}{dt} = -\frac{1}{\beta}\frac{d[Y]}{dt} = \frac{1}{\gamma}\frac{d[Z]}{dt}, \tag{1.9}$$

where the function $v(t)$ is called *reaction rate* of the chemical reaction (1.8).

An explicit formulation of the function $v(t)$ can be obtained from the *Kinetic Law* of chemical reaction. For a chemical reaction (1.8) with reaction rate $v(t)$, the *Kinetic Law* suggests

$$v(t) = k[X]^{\alpha}[Y]^{\beta}, \tag{1.10}$$

where the constant k for proportionality is the *rate constant* of the reaction (1.8), α and β are the *order* of the reaction in terms of X and Y.

Putting (1.9) and (1.10) together gives the following system of RREs for the reaction (1.8):

$$\begin{aligned} \frac{d[X]}{dt} &= -\alpha k \cdot [X]^{\alpha} \cdot [Y]^{\beta}, \\ \frac{d[Y]}{dt} &= -\beta k \cdot [X]^{\alpha} \cdot [Y]^{\beta}, \\ \frac{d[Z]}{dt} &= \gamma k \cdot [X]^{\alpha} \cdot [Y]^{\beta}. \end{aligned} \tag{1.11}$$

Note that for a chemical reaction system with M reactions and N species as in (1.1), the RREs will be an ordinary differential equation system with N equations, and there will be M terms at most on the RHS of each equation, as N differential equations describe the rate of changes of species concentration, and each reaction contributes one term to affect the rate of change of species, if the species is involving in the chemical reaction. In general, in macro-scale systems, the RREs are a system of ordinary differential equations describing the time evolution of concentrations of each species.

1.1.1 Example 1: One Unimolecular Reaction

Consider the chemical reaction,

$$X \xrightarrow{k_1} Y, \tag{1.12}$$

which means one molecule of X forms one molecule of Y through this reaction with rate constant k_1. According to *Kinetic Law* (1.10) and relation of concentration changes, the RREs for reaction (1.12) can be derived as

$$\begin{aligned} \frac{d[X]}{dt} &= -k_1 \cdot [X], \\ \frac{d[Y]}{dt} &= k_1 \cdot [X]. \end{aligned} \tag{1.13}$$

1.1.2 Example 2: One Bimolecular Reaction on the Same Species

Consider a bimolecular reaction on the same species such as

$$2X \xrightarrow{k_2} Y, \tag{1.14}$$

with rate constant k_2 and the reaction rate relation

$$v_2(t) = -\frac{1}{2}\frac{d[X]}{dt} = \frac{d[Y]}{dt}. \tag{1.15}$$

According to *Kinetic Law* (1.10), the RRE for (1.14) are

$$\begin{aligned} \frac{d[X]}{dt} &= -2k_2 \cdot [X] \cdot [X], \\ \frac{d[Y]}{dt} &= k_2 \cdot [X] \cdot [X]. \end{aligned} \tag{1.16}$$

1.1.3 Example 3: One Bimolecular Reaction on Different Species

Given a bimolecular reaction on different species with the rate constant k_3 such as

$$X + Y \xrightarrow{k_3} Z, \tag{1.17}$$

the reaction rate relation can be derived as

$$v_3(t) = -\frac{d[X]}{dt} = -\frac{d[Y]}{dt} = \frac{d[Z]}{dt}. \tag{1.18}$$

It leads to the RREs for (1.18):

$$\begin{aligned} \frac{d[X]}{dt} &= -k_3 \cdot [X] \cdot [Y], \\ \frac{d[Y]}{dt} &= -k_3 \cdot [X] \cdot [Y], \\ \frac{d[Z]}{dt} &= k_3 \cdot [X] \cdot [Y]. \end{aligned} \tag{1.19}$$

1.1.4 Example 4: Multiple Reactions

Consider a chemical reaction system with multiple chemical species which interact through more than one chemical reaction in some fixed volume Ω. For example,

$$X + Y \underset{k_5}{\overset{k_4}{\rightleftharpoons}} Z. \tag{1.20}$$

The number of species X, Y, Z may change through either forward or backward reactions. Therefore, the reaction rate relations are

$$\begin{aligned} v_4(t) = &-\frac{d[X]}{dt} = -\frac{d[Y]}{dt} = \frac{d[Z]}{dt}, \\ v_5(t) = &\frac{d[X]}{dt} = \frac{d[Y]}{dt} = -\frac{d[Z]}{dt}. \end{aligned} \tag{1.21}$$

This shows that the amount of species are determined by two reactions, thus there are two terms on the right hand side of each RRE:

$$\begin{aligned} \frac{d[X]}{dt} &= -k_4 \cdot [X] \cdot [Y] + k_5[Z], \\ \frac{d[Y]}{dt} &= -k_4 \cdot [X] \cdot [Y] + k_5[Z], \\ \frac{d[Z]}{dt} &= k_4 \cdot [X] \cdot [Y] - k_5[Z]. \end{aligned} \tag{1.22}$$

1.1.5 Note

In this section, we introduced the rate of change of *concentration* of chemical species in the system. While it is applied at a constant volume, closed system with steady temperature simple environment, the concentration $[X]$ is related to the number of molecules X by

$$[X] = \frac{X}{N_0 V}, \tag{1.23}$$

where the constant N_0 is the *Avogadro constant*, and V is the constant of fixed volume.

1.2 Propensity Function

When the size of a chemical reaction system is at micro-scale, stochasticity becomes dominant and hence the stoichiometry information is no longer enough to characterize the system. In addition to the amount of change of each molecule caused by reactions, we are also interested in when and how each reaction takes place. The information of each reaction R_j is described by the propensity function defined as

$$\begin{aligned} a_j(\mathbf{x})dt : &\text{ the probability, given } \mathbf{X}(t) = \mathbf{x}, \text{ that one } R_j \\ &\text{reaction will occur inside } \Omega \text{ in the next infinitesimal} \\ &\text{time interval } [t, t + dt). \end{aligned} \tag{1.24}$$

The propensity functions can be described by the ignition of chemical reactions between chemical particles in the system, which depends on the number of possible combinations of reactant molecules involved in reaction R_j [1]. Underlying the logical sense, equation (1.24) describes the behavior of stochastic chemical kinetics represented by CME [35, 36] and stochastic simulation algorithm (SSA, also called Gillespie algorithm) [37]. Three of the simplest cases for deriving propensity functions are unimolecular reaction, bimolecular reaction on different species, and bimolecular reaction on the same species.

1.2.1 Unimolecular reaction

For a unimolecular reaction

$$R_j : S_i \rightarrow \text{product(s)},$$

suppose there exists a constant k_j such that the probability of one R_j reaction happens somewhere in the chemical system during the infinitesimal time period $[t, t + dt]$ is $k_j dt$, and k_j is called *specific probability rate constant* for reaction channel R_j [32]. While the population of S_i is X_j at time t, the probability that one of the S_i molecular particles undergoing the reaction R_j in $[t, t + dt)$ is $X_i \cdot k_j dt$. Thus the propensity function of R_j can be written as

$$a_j(\mathbf{x}) = k_j X_i. \tag{1.25}$$

1.2.2 Bimolecular reaction on different species

Consider a bimolecular reaction on different species

$$R_j : S_i + S_k \rightarrow \text{product(s)}.$$

Since the system is well-stirred in some fixed volume and at a constant temperature, the probability rate constant k_j also exists in such bimolecular systems based on chemical kinetics. While the number of S_i and S_k are X_i and X_k, repectively, the propensity function for reaction R_j is

$$a_j(\mathbf{x}) = k_j X_i X_k, \tag{1.26}$$

since the variety of combination of a pair "S_i and S_k" is $X_i X_k$.

1.2.3 Same species bimolecular reaction

For a same species bimolecular chemical reaction

$$R_j : S_i + S_i \rightarrow \text{product(s)},$$

the combination of randomly selecting two S_i particles is $\frac{1}{2}X_i(X_i - 1)$, which leads the propensity function to be

$$a_j(\mathbf{x}) = k_j \cdot \frac{1}{2} X_i(X_i - 1). \tag{1.27}$$

1.2.4 Note

Overall, the propensity function can be denoted as

$$a_j(\mathbf{x}) = k_j \cdot h(\mathbf{x}), \tag{1.28}$$

where $h(\mathbf{x})$ represents the combinations of selected reactants in the chemical system, and k_j is a *probability rate constant* for such chemical reaction.

To estimate k_j, it is necessary to assume the reaction R_j occurs under certain conditions. For bimolecular cases, the probability of two particles meeting in the system depends on the size of the environment, which is inversely proportional to the volume Ω; on the other hand, the volume does not affect the probability of ignition in unimolecular cases [37]. Since we assume the chemical system is with fixed volume, we can omit the influence of volume size on k_j.

1.3 Chemical Master Equation

Chemical Master equations (CME) [38] are used to described the evolution of a system modelled by the probability functions of combinations of the numbers of chemical species. It is a set of partial differential equations (PDEs) on probabilities of different states of chemical species.

As the population $\mathbf{X}(t)$ of species evolve, the goal is to infer the probability

$$\mathbb{P}(\mathbf{x}, t|\mathbf{x}_0, t_0) := \text{Prob}\{\mathbf{X}(t) = \mathbf{x}, \text{given } \mathbf{X}(t_0) = \mathbf{x}_0\}. \tag{1.29}$$

Consider in an infinitesimal time interval $[t, t + dt)$, the development of the probability through time yields the CME :

$$\frac{\partial \mathbb{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\partial t} = \sum_{j=1}^{M} [a_j(\mathbf{x} - \mathbf{v}_j)\mathbb{P}(\mathbf{x} - \mathbf{v}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x})\mathbb{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)] \tag{1.30}$$

where $\mathbf{v}_j$ is the state-change vector of reaction R_j, and $a_j(\mathbf{x})$ is the propensity function of R_j, $j = 1, \cdots, M$. Exact trajectories for CME may be computed by Gillespie's algorithm, also called the stochastic simulation algorithm (SSA). However, SSA is usually computationally expensive, see the note at the end of this section.

1.3.1 Example

Consider a chemical kinetic system of three species X, Y and Z with two reactions:

$$X + Y \underset{k_2}{\overset{k_1}{\rightleftharpoons}} Z \tag{1.31}$$

with the forward reaction rate constants k_1 and backward reaction rate constants k_2 and denote the number of molecules to be $\mathbf{U}(t) = [X(t), Y(t), Z(t)]^T = [X, Y, Z]^T$. According to (1.26) and (1.25), the propensity function of forward reaction is $a_1(\mathbf{U}) = k_1 XY$, and the propensity function of backward reaction is $a_2(\mathbf{U}) = k_2 Z$. Therefore, the chemical master equation of the system (1.31) at some time t can be derived as:

$$\begin{aligned}\frac{\partial \mathbb{P}(\mathbf{U})}{\partial t} &= k_1(X+1)(Y+1)\mathbb{P}(X+1, Y+1, Z-1) \\ &+ k_2(Z+1)\mathbb{P}(X-1, Y-1, Z+1) \\ &- (k_1 XY + k_2 Z)\mathbb{P}(X, Y, Z),\end{aligned} \tag{1.32}$$

where $\mathbb{P}(\mathbf{U}) = \mathbb{P}(\mathbf{U}, t|\mathbf{u}_0, t_0)$.

Suppose we start out with three molecules of X, five molecules of Y and none of Z. Then all the possible combinations of reactant molecules are

$$\mathbf{U}(t) = \{[3,5,0]^T, [2,4,1]^T, [1,3,2]^T, [0,2,3]^T\}. \tag{1.33}$$

Since there are four possibilities of combinations, the chemical Master equation of system (1.31) contains four equations:

$$\frac{\partial \mathbb{P}([3,5,0]^T)}{\partial t} = k_2 \mathbb{P}([2,4,1]^T) - 15k_1 \mathbb{P}([3,5,0]^T), \tag{1.34}$$

$$\begin{aligned}\frac{\partial \mathbb{P}([2,4,1]^T)}{\partial t} &= 15k_1 \mathbb{P}([3,5,0]^T) + 2k_2 \mathbb{P}([1,3,2]^T) \\ &- (8k_1 + k_2)\mathbb{P}([2,4,1]^T),\end{aligned} \tag{1.35}$$

$$\begin{aligned}\frac{\partial \mathbb{P}([1,3,2]^T)}{\partial t} &= 8k_1 \mathbb{P}([2,4,1]^T) + 3k_2 \mathbb{P}([0,2,3]^T) \\ &- (3k_1 + 2k_2)\mathbb{P}([1,3,2]^T),\end{aligned} \tag{1.36}$$

$$\frac{\partial \mathbb{P}([0,2,3]^T)}{\partial t} = 3k_1 \mathbb{P}([1,3,2]^T) - 3k_2 \mathbb{P}([0,2,3]^T). \tag{1.37}$$

In this CME system, there are *four* probability differential equations since it has *four* possibilities of combinations of chemical reactants, and it is numerically computable by solving the equations. However, it is unrealistic to assume the number of chemical species to be less than 10 in a real world chemical system.

1.3.2 Note

Although the CME determines the probability function $\mathbb{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)$ completely, it is almost impossible to solve analytically except for some simple cases, or even very challenging to compute numerically since it needs enormous storage capacity due to the fact that it is essentially an infinite dimensional system of ODEs involving every possible combination of states at any given time.

It is unrealistic to solve CME if there are hundreds or even thousands possibilities on the combination of number of reactants. Unless the chemical system is simple and with few particular properties, then the CME could describe some distinguishing features to let us know the number of species exactly, like the examples and methods in [33, 39, 40].

Therefore, instead of solving analytically the probability density function $\mathbb{P}(\mathbf{U}, t|\mathbf{u}_0, t_0)$, the *stochastic simulation algorithm (SSA)* was introduced in [3, 37] by Gillespie in order to simulate numerically the evolution of number of species during the stochastic process in such chemical kinetic system.

1.4 Reaction Rate Equation Deduced by Propensity Functions

Consider the CME (1.30). When (1.30) is multiplied by $\mathbf{x}$ and summed over all $\mathbf{x}$, we get

$$\frac{d\langle \mathbf{X}(t)\rangle}{dt} = \sum_{j=1}^{M} \mathbf{v}_j \langle a_j(\mathbf{X}(t))\rangle. \tag{1.38}$$

Suppose $\mathbf{X}(t)$ is a deterministic process hypothetically, it leads to

$$\langle h(\mathbf{X}(t))\rangle = h(\mathbf{X}(t)) \tag{1.39}$$

for all function h. Then (1.38) can be reduced to

$$\frac{d\mathbf{X}(t)}{dt} = \sum_{j=1}^{M} \mathbf{v}_j a_j(\mathbf{X}(t)). \tag{1.40}$$

This is the reaction rate equations (RREs) derived by propensity functions, which is a set of coupled ODEs describing a continuous, deterministic process. However, all the fluctuations are ignored in RRE (1.40), which may not be legitimate for chemical kinetics in real world. Therefore, let's consider stochastic simulation algorithm to understand the trajectories of $\mathbf{X}(t)$ in probability sense.

1.5 Stochastic Simulation Algorithm

Due to the challenges of inferring the probability density function of the population $\mathbf{X}(t)$, we look at the trajectories of $\mathbf{X}(t)$ constructed versus t by using Monte Carlo methods, i.e., at each time t, we randomly choose the next reaction R_j to occur, then generate another random number τ to determine the timing when R_j fires and update the molecular populations, then repeat from generating j for next reaction to happen. This generating method is called *stochastic simulation algorithm (SSA)* or *Gillespie algorithm*.

To understant the generated trajectories of $\mathbf{X}(t)$, we are interested in the new probability function $p(\tau, j|\mathbf{x}, t)$. The key idea of SSA is to introduce a new probability function, which is defined as:

$$\begin{aligned} p(\tau, j|\mathbf{x}, t) := &\text{ the probability, given } \mathbf{X}(t) = \mathbf{x}, \text{ that the next reaction in the} \\ &\text{system will fire in the infinitesimal time interval} \\ &[t+\tau, t+\tau+d\tau), \text{ and will be an } R_j \text{ reaction.} \end{aligned} \tag{1.41}$$

Since the probability of some reaction occurs in the time interval $[t, t+dt)$ is $\sum_j a_j(\mathbf{x})dt$, it is shown in [32] that the probability that a time τ will run out without any reaction firing is $\exp(\sum_j a_j(\mathbf{x})\tau)$. Multiplying by the propensity function (1.24), we obtain the probability in

(1.41) as follows:

$$p(\tau, j|\mathbf{x}, t) = a_j(\mathbf{x})\exp(-a_0(\mathbf{x})\tau) \tag{1.42}$$

where

$$a_0(\mathbf{x}) := \sum_{j'=1}^{M} a_{j'}(\mathbf{x}). \tag{1.43}$$

The above mathematical approach assumes that τ is an exponential random variable with mean and standard deviation $1/a_0(\mathbf{x})$, and j is a statistical independent integer random variable with point probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$.

Although the SSA provides exact prediction of $\mathbf{X}(t)$, futher improvements or adaptions need to be made and exactness might be compromised since the direct SSA is computationally expensive.

A few methods have been created to modify for SSA, like the *next reaction method* [41, 42], or an efficient simmulation approach we are going to introduce in the following, the *tau-leaping method*.

1.6 Tau-Leaping Method

While the proceeding time steps in direct SSA generate random variables, instead we consider time increment to be a fixed number τ in tau-leaping or $\tau-$leaping method, for approximation and simulation of a stochastic system, which is introduced by Gillespie [43], as a bridge to *Reaction Rate Equation (RRE)*. Basically, it performs all reactions in a time interval before updating propensity functions.

In what follows, we made the following assumption on the time interval $[t, t+\tau)$,

1. Leap-condition: $a_j(\mathbf{x})$ has no significant changes or is essentially a constant in $[t, t+\tau)$, i.e.,

$$a_j(\mathbf{x}(t)) \approx a_j(\mathbf{x}(s)) \text{ for any } t \leq s \leq t+\tau. \tag{1.44}$$

2. The number of reactions R_j that occur in $[t, t+\tau)$ is a Poisson distributed random variable with mean and variance $a_j(\mathbf{x})\tau$.

The number τ is called the *leap*. It has to be chosen carefully since it needs to be *small enough* to satisfy the leap-condition, and *large enough* that the expected number of firings of each reaction R_j during time τ is $\gg 1$, i.e.,

$$a_j(\mathbf{x})\tau \gg 1 \text{ for all } j = 1, \cdots, M. \tag{1.45}$$

More discussion for efficient step size τ selection is described in [44].

To construct the numerical simulation based on tau-leaping method with a fixed time increment τ, the processes are as follows:

1. Choose a time step τ, which satisfies the leap condition.

2. Given initial state $\mathbf{x} = \mathbf{X}(t)$ at the time t, with the reactions R_j to fire k_j times in the time interval $[t, t+\tau)$, along with its propensity function $a_j(\mathbf{x})$ and state-change vector $\mathbf{v}_j$.

3. For each R_j, compute the Poisson random variable $\mathcal{P}_j(a_j(\mathbf{x})\tau)$.

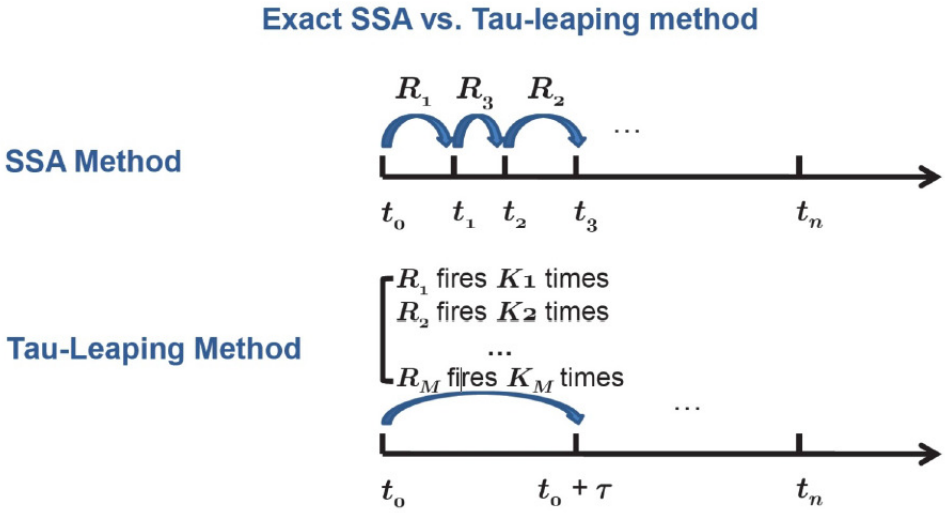
4. Update the number of species by

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{M} \mathcal{P}_j(a_j(\mathbf{x})\tau)\mathbf{v}_j, \tag{1.46}$$

   where $\mathbf{v}_j$ is the state-change vector of reaction R_j. While updating $\mathbf{X}(t)$, it may be necessary to check that no population of species reaches unrealistic values, for example, becoming negative due to the unbounded nature of Poisson random variable $\mathcal{P}_j(a_j(\mathbf{x})\tau)$.

5. Repeat from Step 1 as desired or end the simulation.

Comparing the Tau-Leaping method with the Stochastic Simulation Algorithm (SSA), the main difference is that the time increments of tau-leaping is fixed but the randomness of the system is on the times of firings of reactions, where time increments between reactions in SSA are random numbers. The meanings of them can be interpreted by the following graphs:



1.7 Chemical Langevin Equation

Since no significant modification of propensity functions is performed in the time interval $[t, t+\tau)$, it is computationally more efficient than the direct SSA. However, we need to be careful in choosing the leap τ:

(a) τ has to be *small enough* that the propensity functions $a_j(\mathbf{x})$ do not change during $[t, t+\tau)$.

Thus, the propensity functions satisfy

$$a_j(\mathbf{X}(\hat{t})) \cong a_j(\mathbf{x}), \quad \text{for all } \hat{t} \text{ in } [t, t+\tau). \tag{1.47}$$

There are more details addressed in [32]. So the Poisson random variable $\mathcal{P}(a_j(\mathbf{x})\tau)$ represents the number of times that reaction channel R_j ignites in a duration τ while the propensity function remains essentially a constant at value $a_j(\mathbf{x}(t))$. Therefore, the population $\mathbf{X} = [X_1, \cdots, X_N]^T$ at time $t+\tau$ is allowed to be approximated by

$$X_i(t+\tau) = X_i(t) + \sum_{j=1}^{M} v_{ji} \mathcal{P}_j(a_j(\mathbf{x}), \tau), \quad i = 1, \cdots, N, \tag{1.48}$$

where v_{ji} is the $i-$th entry of the state-change vector $\mathbf{v}_j$.

(b) τ has to be *large enough* that the expected number of occurrences of each reaction R_j in $[t, t+\tau)$ is much greater than 1.

Note that for the Poisson random variable $\mathcal{N}(m, \sigma^2)$ with sufficiently large values of the mean m and variance σ^2 (standard deviation σ) can be approximated excellently by a normal random variable with the same mean and variance $a_j(\mathbf{x})\tau$,

$$\mathcal{P}_j(a_j(\mathbf{x})\tau) \approx \mathcal{N}_j(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau), \tag{1.49}$$

where $\mathcal{N}(m, \sigma^2)$ denotes the normal random variable with mean m and variance σ^2. It leads equation (1.48) into the form

$$X_i(t+\tau) = X_i(t) + \sum_{j=1}^{M} v_{ji}\mathcal{N}_j(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau), \quad i = 1, \cdots, N. \tag{1.50}$$

Using the linear combination theorem for normal random variables

$$\mathcal{N}(m, \sigma^2) = m + \sigma\mathcal{N}(0, 1),$$

equation (1.50) can be written as

$$X_i(t+\tau) = X_i(t) + \sum_{j=1}^{M} v_{ji}a_j(\mathbf{x})\tau + \sum_{j=1}^{M} v_{ji}\sqrt{a_j(\mathbf{x})\tau}\cdot\mathcal{N}_j(0, 1), \quad i = 1, \cdots, N. \tag{1.51}$$

Let's consider any time interval τ that satisfies conditions (a) and (b) as a macroscopic infinitesimal time increment and denote it by dt, equation (1.51) becomes the following "white-noise form" Langevin equation

$$dX_i(t) = \sum_{j=1}^{M} v_{ji}a_j(\mathbf{x})dt + \sum_{j=1}^{M} v_{ji}\sqrt{a_j(\mathbf{x})}dW_j(t), \quad i = 1, \cdots, N, \tag{1.52}$$

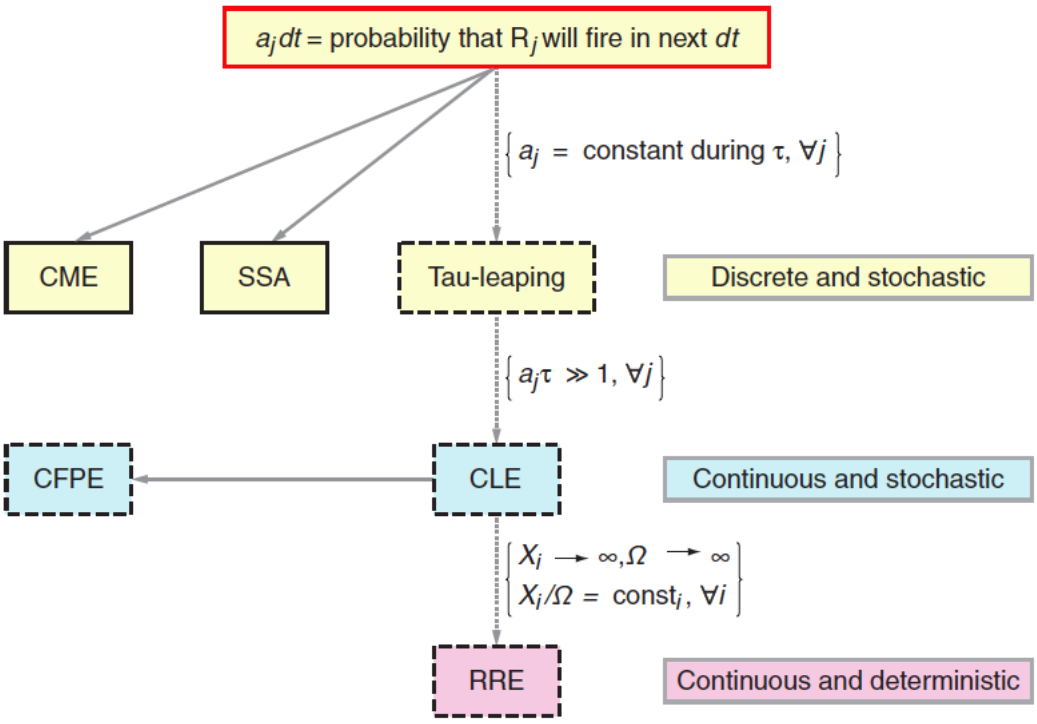


Figure 1.1: Logical structure of stochastic chemical kinetics [1]

where the $W_j(t)$ are temporally uncorrelated, statistically independent Gaussian white noises. Furthermore, equation (1.52) can be written rigorously as the SDE

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} v_{ji} a_j(\mathbf{x}) + \sum_{j=1}^{M} v_{ji} \sqrt{a_j(\mathbf{x})} dW_j(t), \quad i = 1, \cdots, N, \tag{1.53}$$

where $W_j(t)$ are independent Wiener processes.

1.8 Remarks

The logical structure of stochastic chemical kinetics is schematized in Figure 1.1. The propensity function $a_j(\mathbf{x})dt$ leads to the CME and SSA; and the tau-leaping method is developed if $a_j(\mathbf{x})$ is treated as a constant during the infinitesimal time interval $[t, t+\tau)$. As the thermodynamic limit is approached, the second term of CLE (1.53) on the right grows more slowly as

the square root of the system size. When the last term becomes negligibly small compared with other terms such as in full thermodynamic limit, the CLE (1.53) reduces to the RRE (1.40).

Chapter 2

Computational Singular Perturbation and Stochastic Computational Singular Perturbation

2.1 Stiff Problems and Singular Perturbation

While we are interested in the evolution on chemical Langevin equations (1.53), which are stochastic differential equations (SDEs), it shows that the propensity function $a_j(\mathbf{x})$ plays an important role on controlling the speed of processes. It is common that a chemical reaction system involves dissimilar reaction rates, i.e., a chemical system may consists of *probability rate constant* k_j in (1.28) with different magnitudes.

Due to the fact of different reaction rates causing some reactions perform much frequent than the others, mathematically we can classify chemical reaction channels to be *rapid* and *slow* reactions. It is called a "stiff problem" or "stiff system" for such ODEs in multiple time-scales with well-separated rapid and slow dynamical modes.

Solving stiff problems are numerically challenging. Due to rapid variation in solution involving in some of the domain, the numerical simulation may have to take extremely small step size to maintain stability, which is computationally expensive. In [45], the authors developed techniques to deal with extremely stiff ordinary differential equations in dynamic systems based on implicit Runge-Kutta procedure.

Mathematically an ordinary differential equation system with stiffness can be expressed in singular perturbed form [46]:

$$\begin{aligned} d\mathbf{F}(t) &= p(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dt, \\ \varepsilon d\mathbf{G}(t) &= q(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dt. \end{aligned} \tag{2.1}$$

This is a typical sinularly perturbed initial value problem (IVP) with a small positive parameter ε. It is important on applications in chemistry, physics, engineering and biology. Denotes that ε affects the solutions to be dependent on different time scales.

2.2 Computational Singular Perturbation

The Computational Singular Perturbation (CSP) method provides the concepts of an algorithm specifically designed to solve and analyze stiff ordinary differential equation systems [47, 11]. First, identify the contributions of the slow and fast time-scales, then integrate numerically the slow time-scales only to generate the next point; at the end of each integration step, separate and identify the contribution of the fast modes by means of an algebraic amendment to improve computational efficiency.

Consider an ODE system

$$\frac{d\mathbf{X}}{dt} = \mu(\mathbf{X}), \tag{2.2}$$

where $\mathbf{X}$ is a column vector with N entries. Next, represent $\mu(\mathbf{X})$ by some spanning sets $\mathbf{a}_j$,

$$\mu(\mathbf{X}) = \sum_{j=1}^{N} \mathbf{a}_j f^j, \tag{2.3}$$

and denotes that

$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]. \tag{2.4}$$

f^j is called the amplitude corresponding to $\mathbf{a}_j$, which is given by

$$f^j = \mathbf{b}^j \cdot \mu, \tag{2.5}$$

while $\mathbf{b}^j$ can be obtained by the orthogonal sets of $\mathbf{a}_j$, i.e.,

$$\mathbf{B} = [\mathbf{b}^1; \cdots; \mathbf{b}^N] = \begin{bmatrix} — & \mathbf{b}^1 & — \\ & \vdots & \\ — & \mathbf{b}^N & — \end{bmatrix} = \mathbf{A}^{-1}, \tag{2.6}$$

and

$$\mathbf{b}^j \cdot \mathbf{a}_i = \delta_{ji}. \tag{2.7}$$

One of the aims of CSP is to construct a set of "ideal basis vectors" $\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$, such that the representation (2.3) can be well-separated into two rate relative subdomains: fast and slow [11]. In order to find out the proper spanning sets, CSP method studies the evolution of amplitudes of each mode by differentiating (2.2) with respect to time to obtain

$$\frac{d\mu}{dt} = \mathbf{J} \cdot \mu \tag{2.8}$$

where $\mathbf{J}$ denotes the $N \times N$ Jacobian matrix of μ with respect to $\mathbf{X}$. At a given time t, the state $\mathbf{X}(t)$ is known, so is the Jacobian $\mathbf{J}(t)$. Applying inner product of $\mathbf{b}_j$ with (2.8) and using the spanning representation (2.3), the evolution of amplitudes is derived to follow

$$\frac{df^i}{dt} = \sum_{k=1}^{N} \Lambda_k^j f^k, \tag{2.9}$$

where

$$\Lambda_k^j = \left[\frac{d\mathbf{b}^i}{dt} + \mathbf{b}^i \cdot \mathbf{J} \right] \cdot \mathbf{a}_j, \tag{2.10}$$

and

$$\frac{d\mathbf{b}^i}{dt} = \left[\frac{db_1^i}{dt}, \cdots, \frac{db_N^i}{dt} \right]. \tag{2.11}$$

To construct an appropriate set of basis vectors $\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$ such that the fast and slow dynamics are separated while projecting onto $\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$, CSP monitors the behaviour of the square matrix $\Lambda = [\Lambda_k^j]$ which describes the influence to the evolution of $\mathbf{f}^k$ by other magnitudes. CSP performs two steps refinements to let Λ approach a nearly block diagonoal matrix to achieve the goal.

2.3 Stochastic Computational Singular Perturbation

Although the CSP method is a powerful tool to analyze and simulate stiff ODE systems, it is not directly applicable to Chemical Langevin equation since there are non-negligible stochastic

parts. In [48], a stochastic computational singular perturbation method (SCSP) is developed to deal with stochastic ordinary differential equation (SDE) systems with similar tactics.

Consider the stochastic singular perturbation system

$$\begin{aligned} d\mathbf{F}(t) &= p(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dt + \quad P(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dW(t), \\ \varepsilon d\mathbf{G}(t) &= q(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dt + \sqrt{\varepsilon}Q(\mathbf{F}(t), \mathbf{G}(t), \varepsilon)dW(t), \end{aligned} \tag{2.12}$$

where $\mathbf{F}(t)$ and $\mathbf{G}(t)$ are stochastic processes, ε is a small positive number, and $W(t)$ is a one-dimensional Wiener process.

Using the same strategies as CSP, we are interested in finding proper vector sets to span deterministic and stochastic vector fields of SDEs

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sum_{j=1}^{M} \sigma_j(\mathbf{X})dW_j(t). \tag{2.13}$$

For any two given sets of basis vectors, $\mathbf{a}_i$ and $\boldsymbol{\alpha}_i$ for $i = 1, \cdots, N$, we denote

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_1, \cdots, \mathbf{a}_N] \text{ for } \mu(\mathbf{X}), \\ \mathcal{A}^j &= [\boldsymbol{\alpha}_1^j, \cdots, \boldsymbol{\alpha}_N^j] \text{ for each } \sigma_j(\mathbf{X}), \quad j = 1, \cdots, M. \end{aligned} \tag{2.14}$$

Now we span the vector fields $\mu(\mathbf{X})$ and $\sigma_j(\mathbf{X})$ such that

$$\mu(\mathbf{X}) = \sum_{r=1}^{L} \mathbf{a}_r f^r + \sum_{s=L+1}^{N} \mathbf{a}_s f^s, \tag{2.15}$$

$$\sigma_j(\mathbf{X}) = \sum_{r=1}^{L} \boldsymbol{\alpha}_r^j g_j^r + \sum_{s=L+1}^{N} \boldsymbol{\alpha}_s^j g_j^s, \tag{2.16}$$

where L denotes the dimension of fast subdomain, $\mathbf{a}_r$, $\boldsymbol{\alpha}_r^j$ span rapid subdomains, $\mathbf{a}_s$ and $\boldsymbol{\alpha}_s^j$ span slow subdomains, and f^i, g_j^i denotes the amplitudes, respectively. Also let

$$\mathbf{B} = [\mathbf{b}^1; \cdots; \mathbf{b}^N] = \begin{bmatrix} — & \mathbf{b}^1 & — \\ & \vdots & \\ — & \mathbf{b}^N & — \end{bmatrix} \tag{2.17}$$

and

$$\mathcal{B}^j = [\boldsymbol{\beta}_1^j; \cdots; \boldsymbol{\beta}_N^j] = \begin{bmatrix} — & \boldsymbol{\beta}_1^j & — \\ & \vdots & \\ — & \boldsymbol{\beta}_N^j & — \end{bmatrix} \tag{2.18}$$

to be the corresponding orthogonal row vectors set of $\mathbf{A}$ and $\mathcal{A}^j$, respectively, i.e.,

$$\mathbf{B} = \mathbf{A}^{-1}, \quad \mathcal{B}^j = (\mathcal{A}^j)^{-1}. \tag{2.19}$$

Thus, the amplitudes corresponding to $\mathbf{A}$ and $\mathcal{A}^j$ are

$$f^i = \mathbf{b}_i \cdot \mu, \tag{2.20}$$

$$g_j^i = \boldsymbol{\beta}_i^j \cdot \sigma_i. \tag{2.21}$$

Denote $[f] = [f^1, \cdots, f^N]^T$, $[g_j] = [g_j^1, \cdots, g_j^N]^T$ as the N-dimensional column vectors composed of deterministic and stochastic amplitudes.

For simplicity of exposition, first consider a Stratonovich SDE system consisting of only one noise term, which means, $M = 1$ in (2.13) the SDE can thus be reduced to

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sigma(\mathbf{X}) \circ dW(t), \tag{2.22}$$

where $\circ$ represents the Stratonovich product. Here, we can take advantage of its preservation of the chain rule like ordinary calulus. While performing SCSP [48], an arbitrary initial set of basis vectors are chosen to span the vector fields of the deterministic and stochastic parts respectively, then perform iterations of refinements are performed in order to find an ideal spanning vector set by getting optimized amplitudes f and g. Another procedure in SCSP is as follows: Eigenvectors of the Jacobian of $\mu(\mathbf{X})$ and $\sigma(\mathbf{X})$ in (2.22) are chosen as the basis $\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$ and $\{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_N\}$ to span the vector fields of deterministic and stochastic parts

respectively, which leads the time evolution of all modes to be

$$d\begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} \Lambda & V \\ \Gamma & T \end{bmatrix} \cdot d\omega(t), \tag{2.23}$$

where $d\omega(t) = \text{Diag}[f \cdot I_N dt, g \cdot I_N dW(t)]$, J_μ, J_σ are the Jacobians of $\mu(\mathbf{X})$ and $\sigma(\mathbf{X})$ correspondingly, and Λ, V, Γ, T are $N \times N$ matrices that

$$\Lambda = \mathbf{B}J_\mu\mathbf{A}, \quad V = \mathbf{B}J_\mu\mathcal{A}, \quad \Gamma = \mathcal{B}J_\sigma\mathbf{A}, \quad T = \mathcal{B}J_\sigma\mathcal{A}. \tag{2.24}$$

After k iterative refinements are performed, the "drift" f and the "volatility" g will be split into rapid components $\mathbf{f}^r_{(k)}, \mathbf{g}^r_{(k)}$ and slow components $\mathbf{f}^s_{(k)}, \mathbf{g}^s_{(k)}$, respectively. Therefore, the fast and slow deterministic subdomains are spanned by $\mathbf{A}^r_k$ and $\mathbf{A}^s_k$, respectively, and fast and slow stochastic subdomains are spanned by $\mathcal{A}^r$ and $\mathcal{A}^s$, respectively. It leads us to define formally the *k-th stochastic slow manifold* by a stochastic relation that captures the near-equilibration of fast processes. The dynamics of the originial SDE (2.22) are then further reduced to

$$d\mathbf{X} \approx \mathbf{A}^s_k f^s_{(k)} + \mathcal{A}^s_k g^s_{(j)} \circ dW(t) \tag{2.25}$$

while approaching the "k-th stochastic slow manifold". This idea is used to build a new time-scale splitting, explicit algorithm to integrate stiff SDEs that consists of three steps.

1. Identify the number m of exhausted modes at a given time.

2. Starting from a trial set of basis vectors, construct SCSP basis vectors iteratively by the refinement procedures.

3. After $k \geq 0$ iterations of refinements, integrate the SDE system (2.22) from t_j to t_{j+1} in two steps as follows:

$$\hat{\mathbf{X}}_j = \mathbf{X}_j + \int_{t_j}^{t_{j+1}} A^r_k f^r_{(k)} + \mathcal{A}^r_k g^r_{(k)} \circ dW(t), \tag{2.26}$$

$$\mathbf{X}_{j+1} = \hat{\mathbf{X}}_j + \int_{t_j}^{t_{j+1}} A_k^s f_{(k)}^s + \mathcal{A}_k^s g_{(k)}^s \circ dW(t). \tag{2.27}$$

While using the Euler method for $\int_{t_j}^{t_j+\tau} A_k^r f_{(k)}^r$ and the midpoint rule for $\int_{t_j}^{t_j+\tau} \mathcal{A}_k^r g_{(k)}^r \circ dW(t)$, (2.26) becomes

$$\hat{\mathbf{X}}_j = \mathbf{X}_j + A_k^r(\mathbf{X}_j) f_{(k)}^r(\mathbf{X}_j)\tau + \frac{\xi\sqrt{\tau}}{2}\left(\mathcal{A}_k^r(\mathbf{X}_j) g_{(k)}^r(\mathbf{X}_j) + \mathcal{A}_k^r(\hat{\mathbf{X}}_j) g_{(k)}^r(\hat{\mathbf{X}}_j)\right), \tag{2.28}$$

where ξ is a standard normal random variable, and τ is the expected elapsed time of the fast modes. Then integrating along the $k-$th stochastic slow manifold according to the simplified system (2.25) to obtain $\mathbf{X}_{j+1}$ similar to previous strategy,

$$\begin{aligned}\mathbf{X}_{j+1} \approx & \hat{\mathbf{X}}_j + A_k^s(\mathbf{X}_j) f_{(k)}^s(\mathbf{X}_j) h_j \\ & + \frac{\xi\sqrt{h_j}}{2}\left(\mathcal{A}_k^s(\mathbf{X}_j) g_{(k)}^s(\mathbf{X}_j) + \mathcal{A}_k^s(\mathbf{X}_{j+1}) g_{(k)}^s(\mathbf{X}_{j+1})\right),\end{aligned} \tag{2.29}$$

where ξ is a standard normal random variable, and $h_j := t_{j+1} - t_j$.

2.4 Remarks

For stochastic systems, SCSP allows the iterative construction of two sets of basis vectors spanning the fast and slow dynamical subspaces for both the drift and diffusion of an SDE system. With this construction, the fast and slow dynamics of a stiff stochastic differential equation (SDE) can be decoupled, while projecting onto the tensor product of the associated vector spaces. An operator-split explicit time integration algorithm is developed for solving stiff SDEs using SCSP. Since the stochastic algebraic relation describing the fast dynamics does not depend on the step length of time integration, and the subsequent time integration of the slow processes is non-stiff, the construction allows stable integration with large explicit time steps. Numerical experiments highlight the accuracy of computed first order statistics of the state variables, and that of the second order statistics of the slow variables, relative to stiff time integration of the original SDE system.

Chapter 3

Simplified Stochastic Computational Singular Perturbation

3.1 Simplified Stochastic Computational Singular Perturbation (SSCSP)

The governing CLE for a chemical reaction system with multiple reactions involves multiple of noises and thus multiple diffusion vectors. Each additional noise requires one more application of eigen analysis and SCSP. Though this is not a major problem providing the advancement of parallel computing nowadays, it is still a drawback of the algorithm. Moreover, the Jacobian matrices of most of these diffusion vectors, due to the special structure of chemical reaction network, are singular, and hence create more numerical complexity.

Noticing the special relation between the drift and diffusion terms of CLEs, the main goal of this chapter and also of this research is to develop a simplified SCSP-based algorithm that can simulate stiff CLEs more efficiently while maintaining numerical stability.

According to (1.53), the deterministic part and the coefficient of the noise term in (2.13) are

$$\mu(\mathbf{X}) = \begin{bmatrix} \sum_{j=1}^{M} v_{j1} a_j(\mathbf{X}) \\ \vdots \\ \sum_{j=1}^{M} v_{jN} a_j(\mathbf{X}) \end{bmatrix}, \tag{3.1}$$

$$\sigma_j(\mathbf{X}) = \begin{bmatrix} v_{j1}\sqrt{a_j(\mathbf{X})} \\ \vdots \\ v_{jN}\sqrt{a_j(\mathbf{X})} \end{bmatrix}, \tag{3.2}$$

respectively. The Jacobian matrices for $\mu(\mathbf{X})$ and σ_j are

$$J_\mu(\mathbf{X}) = \begin{bmatrix} \sum_{j=1}^{M} v_{j1}\frac{\partial a_j}{\partial x_1} & \sum_{j=1}^{M} v_{j1}\frac{\partial a_j}{\partial x_2} & \cdots & \sum_{j=1}^{M} v_{j1}\frac{\partial a_j}{\partial x_N} \\ \sum_{j=1}^{M} v_{j2}\frac{\partial a_j}{\partial x_1} & \sum_{j=1}^{M} v_{j2}\frac{\partial a_j}{\partial x_2} & \cdots & \sum_{j=1}^{M} v_{j2}\frac{\partial a_j}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{M} v_{jN}\frac{\partial a_j}{\partial x_1} & \sum_{j=1}^{M} v_{jN}\frac{\partial a_j}{\partial x_2} & \cdots & \sum_{j=1}^{M} v_{jN}\frac{\partial a_j}{\partial x_N} \end{bmatrix}, \tag{3.3}$$

$$J_{\sigma_j}(\mathbf{X}) = \frac{1}{2\sqrt{a_j(\mathbf{X})}} \begin{bmatrix} v_{j1}\frac{\partial a_j}{\partial x_1} & v_{j1}\frac{\partial a_j}{\partial x_2} & \cdots & v_{j1}\frac{\partial a_j}{\partial x_N} \\ v_{j2}\frac{\partial a_j}{\partial x_1} & v_{j2}\frac{\partial a_j}{\partial x_2} & \cdots & v_{j2}\frac{\partial a_j}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{jN}\frac{\partial a_j}{\partial x_1} & v_{jN}\frac{\partial a_j}{\partial x_2} & \cdots & v_{jN}\frac{\partial a_j}{\partial x_N} \end{bmatrix}. \tag{3.4}$$

Due to the similarity in structures of J_μ and J_{σ_i}'s, we propose to use the same basis vector set

$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N], \tag{3.5}$$

for both deterministic and stochastic parts. Following the SCSP, let $\mathbf{A}$ to be the eigenvectors of $\mu(\mathbf{X})$, which leads the evolution of magnitudes to be

$$d[f] = d[g] = \Lambda f \cdot dt + \Lambda g \cdot dW,$$

where $\Lambda = \mathbf{B}J_\mu\mathbf{A}$.

3.2 Time integration: Special Case with No Noise

Consider the ODE

$$d\mathbf{X} = \mu(\mathbf{X})dt, \tag{3.6}$$

where the stochastic term from the SDE (2.22) is ignored. The function $\mu(\mathbf{X})$ can be projected onto any set of basis vectors:

$$\mu(\mathbf{X}) = \sum_{j=1}^{N} \mathbf{a}_j f^j. \tag{3.7}$$

It is important to study the evolution of amplitudes f^j. The bigger the magnitude of the amplitudes f^j is, the greater significance on the corresponding vector $\mathbf{a}_j$. We follow the strategy in [30], differentiate (3.7) with respect to time and use (3.6) to obtain

$$\frac{d\mu}{dt} = \mathbf{J} \cdot \frac{d\mathbf{X}}{dt} = \mathbf{J} \cdot \mu, \tag{3.8}$$

where $\mathbf{J} = [J_{ij}]$ is the Jacobian of μ with dimension $N \times N$ and

$$J_{ij} = \frac{\partial \mu_i}{\partial x_j}. \tag{3.9}$$

Differentiating (2.20) and using (3.8), the evolution of f^i can be described as an expansion of $f^1, f^2, \cdots, f^N$ as

$$\frac{df^i}{dt} = \sum_{j=1}^{N} \Lambda_j^i f^j, \quad i = 1, \cdots, N, \tag{3.10}$$

where

$$\Lambda_j^i \equiv \left(\frac{d\mathbf{b}^i}{dt} + \mathbf{b}^i \cdot \mathbf{J}\right) \cdot \mathbf{a}_j, \tag{3.11}$$

and $\mathbf{b}^i$'s are defined in (2.17) as an orthogonal vector sets with respect to $\mathbf{a}_j$. The derivative of a row vector $\mathbf{b}^i = [b_1^i, \cdots, b_N^i]$ are defined to be the row vector where the entries are the derivatives of entries of $\mathbf{b}^i$, i.e.,

$$\frac{d\mathbf{b}^i}{dt} = \left[\frac{db_1^i}{dt}, \cdots, \frac{db_N^i}{dt}\right]. \tag{3.12}$$

At any fixed time t, the matrix $\mathbf{J}$ and vector $\mathbf{X}(t)$ are known. Choose the column vector set $\mathbf{a}_j$ to be the eigenvectors of $\mathbf{J}$ corresponding to eigenvalues λ_j with descending magnitudes, i.e., $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_N|$. Since $\mathbf{a}_j$ is a constant initial trial set of time-independent basis vectors at time $t = t_0$, we can assume

$$\frac{d\mathbf{b}^i}{dt} = \mathbf{0}. \tag{3.13}$$

Thus, the formula (3.11) is further updated

$$\Lambda^i_j \equiv \mathbf{b}^i \cdot \mathbf{J} \cdot \mathbf{a}_j = \lambda_j(\mathbf{b}^i \cdot \mathbf{a}_j) = \begin{cases} \lambda_i & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}, \tag{3.14}$$

which indicates Λ^i_j is an $N \times N$ diagonal matrix and its diagonal elements are the eigenvalues of the Jacobian $\mathbf{J}$, i.e.,

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix} \tag{3.15}$$

Moreover, it simplifies (3.10) and the evolution of amplitude f^i can be expressed as a linear ODE

$$\frac{df^i}{dt} = \lambda_i f^i. \tag{3.16}$$

Assume that after certain time T, the fast mode with dimension M becomes stable and the remaining $N - M$ dimensions slow subdomain dominates the behavior of $\mathbf{X}$. Based on the idea for CSP, the change of $\mathbf{X}$ during the time interval $[T, T + \Delta t)$ is expressed as

$$\mathbf{X}(T + \Delta t) - \mathbf{X}(T) = \int_T^{T+\Delta t} \left(\sum_{r=1}^{M} \mathbf{a}_r f^r \right) dt + \int_T^{T+\Delta t} P\mu \, dt, \tag{3.17}$$

where P is an $N \times N$ matrix representing the projection of the slow subdomain. To derive the matrix P, write the ODE system (3.6) as

$$d\mathbf{X} = \mu(\mathbf{X})dt = \Big(\sum_{r=1}^{M} \mathbf{a}_r f^r + P\mu\Big)dt, \tag{3.18}$$

therefore,

$$\begin{aligned} P\mu &= \mu - \sum_{r=1}^{M} \mathbf{a}_r f^r \\ &= \mu - \sum_{r=1}^{M} \mathbf{a}_r \mathbf{b}^r \mu \\ &= \Big(I - \sum_{r=1}^{M} \mathbf{a}_r \mathbf{b}^r\Big)\mu. \end{aligned} \tag{3.19}$$

From this we get the projection matrix of the slow subdomain:

$$P = I - \sum_{r=1}^{M} \mathbf{a}_r \mathbf{b}^r. \tag{3.20}$$

Dividing by λ_i to the equation (3.16), and substituting

$$f^i = \frac{1}{\lambda_i}\frac{df^i}{dt} \tag{3.21}$$

into the first integral of (3.17), yields

$$\mathbf{X}(T+\Delta t) - \mathbf{X}(T) = \sum_{r=1}^{M} \frac{1}{\lambda_r} \int_T^{T+\Delta t} \mathbf{a}_r \frac{df^r}{dt}\, dt + \int_T^{T+\Delta t} P\mu\, dt. \tag{3.22}$$

Applying integration by parts, yields

$$\begin{aligned} \mathbf{X}(T+\Delta t) - \mathbf{X}(T) = \sum_{r=1}^{M} \frac{1}{\lambda_r} \left[\mathbf{a}_r f^r|_{t=T+\Delta t} - \mathbf{a}_r f^r|_{t=T} - \int_T^{T+\Delta t} \frac{d\mathbf{a}_r}{dt} f^r\, dt\right] \\ + \int_T^{T+\Delta t} P\mu\, dt. \end{aligned} \tag{3.23}$$

Due to the fact that the projection on fast subdomain has been exhausted, the quantity $\mathbf{a}_r f^r$ at $t = T + \Delta t$ becomes much smaller than at $t = T$. Thus, the first term in the bracket on the

RHS can be omitted, i.e.,

$$\mathbf{a}_r f^r|_{t=T+\Delta t} = \mathbf{0}. \tag{3.24}$$

Moreover, the integral in the bracket can also be ignored since $\mathbf{a}_j$ is a constant initial trial set of time-independent basis and it yields

$$\frac{d\mathbf{a}_r}{dt} = \mathbf{0}. \tag{3.25}$$

Thus, the change of $\mathbf{X}$ as (3.17) can be rewritten as

$$\mathbf{X}(T+\Delta t) - \mathbf{X}(T) = -\sum_{r=1}^{M} \frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right] + \int_T^{T+\Delta t} P\mu \, dt + Remainder, \tag{3.26}$$

where

$$Remainder = \sum_{r=1}^{M} \frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T+\Delta t} - \int_T^{T+\Delta t} \frac{d\mathbf{a}_r}{dt} f^r \, dt\right], \tag{3.27}$$

and it represents the small terms which can be neglected in approximation.

3.3 Time Integration: Stiff SDE System

3.3.1 Stratonovich SDE with Only One Noise Term

First, consider the Stratonovich SDE of stochastic column process $\mathbf{X}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]^T$ with only one diffusion term σ,

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sigma(\mathbf{X}) \circ dW(t), \tag{3.28}$$

where the drift (deterministic) term μ and diffusion (stochastic) term σ denote $N \times 1$ column vectors, the operator $\circ$ denotes the Stratonovich product, and $W(t)$ is a Wiener process. Here we use the same basis $\mathbf{a}_j$ to span both drift and diffusion parts,

$$\mu(\mathbf{X}) = \sum_{j=1}^{N} \mathbf{a}_j f^j, \tag{3.29}$$

$$\sigma(\mathbf{X}) = \sum_{j=1}^{N} \mathbf{a}_j g^j, \tag{3.30}$$

which is unlike (2.15) and (2.16) using different basis to span deterministic and stochastic parts respectively. It is easier to study and analyze the evolution of the corresponding magnitudes f^j and g^j.

In order to study the evolution of amplitudes f^j, use similar tactic as [30], differentiate (3.29) to obtain

$$d\mu = \mathbf{J} \cdot d\mathbf{X}, \tag{3.31}$$

where $\mathbf{J} = [J_{ij}]$ is the Jacobian of μ with dimension $N \times N$ and

$$J_{ij} = \frac{\partial \mu_i}{\partial x_j}. \tag{3.32}$$

Differentiating

$$f^i = \mathbf{b}^i \cdot \mu, \tag{3.33}$$

where $\mathbf{b}^i$ is from the orthogonal row vector sets to $\mathbf{a}_j$ and use (3.8), the evolution of f^i can be written as

$$\begin{aligned} df^i &= d\mathbf{b}^i \cdot \mu + \mathbf{b}^i \cdot \mathbf{J} \cdot d\mathbf{X} \\ &= d\mathbf{b}^i \cdot \mu + \mathbf{b}^i \cdot \mathbf{J} \cdot \left[\left(\sum_{j=1}^{N} \mathbf{a}_j f^j \right) dt + \left(\sum_{j=1}^{N} \mathbf{a}_j g^j \right) \circ dW(t) \right] \end{aligned} \tag{3.34}$$

Given the time $t = T$, the Jacobian matrix $\mathbf{J}$ and vector $\mathbf{X}(t)$ are known and let $\mathbf{a}_j$ be the eigenvectors of $\mathbf{J}$ corresponding to eigenvalues λ_j with descending magnitudes, i.e.,

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_N|. \tag{3.35}$$

Since $\mathbf{a}_j$ is a constant initial trial set of time-independent basis vectors, $d\mathbf{b}^i$ can be neglected, and (3.34) can be expressed as

$$df^i = \sum_{j=1}^{N} \Lambda_j^i f^j dt + \sum_{j=1}^{N} \Lambda_j^i g^j \circ dW(t), \tag{3.36}$$

where

$$\Lambda_j^i \equiv \mathbf{b}^i \cdot \mathbf{J} \cdot \mathbf{a}_j = \lambda_j (\mathbf{b}^i \cdot \mathbf{a}_j) = \begin{cases} \lambda_i & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}. \tag{3.37}$$

Similar to (3.15), Λ_j^i is a diagonal matrix and its diagonal elements are the eigenvalues of the Jacobian $\mathbf{J}$. Moreover, (3.36) can be simplified and the evolution of amplitude f^i can be expressed as a SDE

$$df^i = \lambda_i [f^i dt + g^i \circ dW(t)]. \tag{3.38}$$

Assume that after a short period of certain time, the fast mode with dimension M are exhausted and the remaining $N - M$ dimensions slow subdomain dominates the behavior of $\mathbf{X}$. Based on the concept of CSP, the integration of $\mathbf{X}$ during the time interval $[T, T + \Delta t)$ can be expressed as

$$\begin{aligned} \mathbf{X}(T + \Delta t) - \mathbf{X}(T) = & \int_T^{T+\Delta t} \left(\sum_{r=1}^{M} \mathbf{a}_r f^r \right) dt \\ & + \int_T^{T+\Delta t} \left(\sum_{r=1}^{M} \mathbf{a}_r g^r \right) \circ \, dW(t) \\ & + \int_T^{T+\Delta t} P \left(\mu \, dt + \sigma \circ \, dW(t) \right), \end{aligned} \tag{3.39}$$

where P is the projection matrix of the slow subdomain and can be derived as

$$P = I - \sum_{r=1}^{M} a_r b^r. \tag{3.40}$$

From equation (3.38), substituting

$$f^i dt = \frac{1}{\lambda_i} df^i - g^i \circ dW(t) \tag{3.41}$$

into the first integral of (3.39) yields

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &\sum_{r=1}^{M}\int_{T}^{T+\Delta t}\mathbf{a}_r\left(\frac{1}{\lambda_r}df^r - g^r\circ dW(t)\right)\\ &+\int_{T}^{T+\Delta t}\sum_{r=1}^{M}\mathbf{a}_r g^r\circ dW(t)\\ &+\int_{T}^{T+\Delta t}P\left(\mu\,dt+\sigma\circ\,dW(t)\right).\end{aligned} \tag{3.42}$$

Note that the second integral of (3.42) can be cancelled with the latter part of the first integral, we obtain

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &\sum_{r=1}^{M}\int_{T}^{T+\Delta t}\frac{1}{\lambda_r}\left(\mathbf{a}_r df^r\right)\\ &+\int_{T}^{T+\Delta t}P\left(\mu\,dt+\sigma\circ\,dW(t)\right).\end{aligned} \tag{3.43}$$

Applying the same strategy with formula (3.23) by using integration by parts on the first integral, yields

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T+\Delta t}-\mathbf{a}_r f^r|_{t=T}-\int_{T}^{T+\Delta t}\frac{d[\mathbf{a}_r]}{dt}f^r\,dt\right]\\ &+\int_{T}^{T+\Delta t}P\left(\mu\,dt+\sigma\circ\,dW(t)\right).\end{aligned} \tag{3.44}$$

Due to the fact that the projection on the fast domain has been exhausted, the quantity $\mathbf{a}_r f^r$ at $t=T+\Delta t$ becomes much smaller than at $t=T$, i.e., $\mathbf{a}_r f^r|_{t=T+\Delta t}$ can be omitted. Moreover, the integral in the bracket can also be neglected since $\mathbf{a}_j$ is a constant initial trial set of time-independent basis vectors and it yields

$$\frac{d\mathbf{a}_r}{dt}=\mathbf{0}. \tag{3.45}$$

Thus, the change of $\mathbf{X}$ as (3.17) can be rewritten as

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &-\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]+\int_T^{T+\Delta t} P\left(\mu\, dt+\sigma\circ\, dW(t)\right)\\ &+Remainder,\end{aligned} \tag{3.46}$$

where

$$Remainder=\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T+\Delta t}\right], \tag{3.47}$$

represents the small term which can be neglected in approximation.

3.3.2 Stratonovich SDE with Multiple Noise Terms

Consider the Stratonovich SDE of stochastic column process $\mathbf{X}(t)=[x_1(t),x_2(t),\cdots,x_N(t)]^T$ with multiple diffusion terms σ_k,

$$d\mathbf{X}=\mu(\mathbf{X})dt+\sum_{k=1}^{M}\sigma_k(\mathbf{X})\circ dW_k(t), \tag{3.48}$$

where the (deterministic) drift term μ and (stochastic) diffusion term σ_k denote $N\times 1$ column vectors, $\circ$ denotes the Stratonovich product, and $W_k(t)$ is Wiener process. Similar to (3.30), denote each diffusion terms σ_k spanned by column vectors $\mathbf{a}_j$,

$$\sigma_k(\mathbf{X})=\sum_{j=1}^{N}\mathbf{a}_j g_k^j. \tag{3.49}$$

where g_k^j represents the corresponding magnitude with respect to vector $\mathbf{a}_j$ on spanning the function σ_k. Similar to (3.46) and with the same setting of matrix P, the change of $\mathbf{X}$ in (3.48) can be expressed as

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &-\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]\\ &+\sum_{j=1}^{N}\int_T^{T+\Delta t} P\left(\mu\, dt+\sigma_j\circ\, dW_j(t)\right)+Remainder,\end{aligned} \tag{3.50}$$

where *Remainder* represents the negligible small terms.

Although the derivation of (3.50) is mathematically correct, it disobeys the intuition on the variance of $\mathbf{X}(t)$ which is contributed from the diffusion parts. In (3.50), the stochastic parts are only provided by the slow subdomain, while the contribution from the fast subdomain is cancelled (see (3.43)).

However, the result is still valuable in providing insights for improving the scheme. From the numerical experiments, it is clearly shown that the contribution from the fast subdomain may be neglected. In the next few sections, we will develop a modification following the idea from (3.50) but without omitting the contribution from the fast subdomain, which turns out to preserve the variance to a certain degree.

3.3.3 Two-Step Time Integration Scheme

Since the fast and slow modes are well seperated and can be interpreted in different time integration form, next we write a two-time-scale scheme of time integration including two steps.

First, compute the contribution from slow modes by the expression

$$\tilde{\mathbf{X}}(T+\Delta t) = \mathbf{X}(T) + \sum_{k=1}^{M} \int_{T}^{T+\Delta t} P\left(\mu\, dt + \sigma_j \circ dW_j(t)\right), \tag{3.51}$$

where P is an $N \times N$ matrix and can be derived as (3.40). During the time elapsed from T to $T+\Delta t$, P is considered as a constant $P(T)$ and it can be computed in terms of the column basis $\mathbf{a}_j$ and corresponding row basis $\mathbf{b}_j$, which are known at time T.

Second, we substract the adjustment term contributed by the fast mode to complete the time integration:

$$\mathbf{X}(T+\Delta t) = \tilde{\mathbf{X}}(T+\Delta t) - \sum_{r=1}^{M} \frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]. \tag{3.52}$$

We express the last term on the RHS to be

$$\Delta \mathbf{X}_{fast}(T+\Delta t) \approx \sum_{r=1}^{M} \frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right], \tag{3.53}$$

which is similar to the "radical correction" referred in [47].

3.3.4 The Simplified Stochastic Computational Singular Perturbation Algorithm for Stratonovich SDE

The *simplified stochastic computational singular perturbation (SSCSP) algorithm* for Stratonovich SDE (3.28) is developed in two substeps to deal with fast and slow modes by (3.51) and (3.52).

Begin

Step I: Time integration on slow mode

1. Evaluate the source term $\mu(T) = \mu(\mathbf{X}(T))$ and $\sigma_j(T) = \sigma_j(\mathbf{X}(T))$.

2. Compute the Jacobian of μ at time T.

   (a) Find the eivenvalues of J_μ, and sort them in descending order by magnitudes,

   $$|\lambda_1| > \cdots > |\lambda_N|, \tag{3.54}$$

   and the corresponding eigenvector set

   $$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N], \tag{3.55}$$

   (b) Find the corresponding orthogonal row vectors set

   $$\mathbf{B} = [\mathbf{b}^1; \cdots; \mathbf{b}^N] = \mathbf{A}^{-1}. \tag{3.56}$$

3. Determine the dimension M of the exhausted fast mode at time T. The number M is the greatest integer between 1 and N that satisfies the inequality

   $$|\tau_{M+1} \sum_{j=1}^{M} \mathbf{a}_j f^j| < \mathbf{X}_{error}, \tag{3.57}$$

   where $\mathbf{X}_{error}$ is an error vector, and τ_{M+1} is the time scale with CSP concept and defined as

   $$\tau_{M+1} = \frac{1}{\lambda_{M+1}}. \tag{3.58}$$

4. Generate the $N \times N$ projection matrix

$$P(\mathbf{X}(T)) = I - \sum_{r=1}^{M} \mathbf{a}_r \mathbf{b}^r, \tag{3.59}$$

where $\mathbf{a}_r$ and $\mathbf{b}_r$ represent the first M vectors of (3.55) and (3.56) and they span the fast subdomain.

5. Perform time integration on the slow modes during time T to $T + \Delta t$,

$$\begin{aligned} \tilde{\mathbf{X}}(T + \Delta t) = \mathbf{X}(T) + \Delta t \cdot P \cdot (\mu(\mathbf{X}(T))) \\ + \frac{1}{2} \sum_{j=1}^{N} \mathcal{N}_j \sqrt{\Delta t} \cdot P \cdot (\sigma_j(\mathbf{X}(T)) + \sigma_j(\tilde{\mathbf{X}}(T + \Delta t))), \end{aligned} \tag{3.60}$$

where $\mathcal{N}_j$ denote standard normal variables. Forward Euler method is used on the deterministic part. $\tilde{\mathbf{X}}(T+\Delta t)$ is solved implicitly by Newton's method due to the Stratonovich integration on the stochastic terms. Significant computational cost is consumed here for performing Newton's method and for computing the Jacobian matrices of σ_j and their inverses. See more details at Appendix A.

Step II: Adjustment contributed from fast mode

According to the formula (3.52), the contribution from the fast mode can be evaluated as follows:

1. Evaluate the amplitudes contributed from the fast subdomain

$$f^r(T) = b^r(\mathbf{X}) \cdot \mu(\mathbf{X}); \tag{3.61}$$

2. Calculate the quantity contributed from the fast mode:

$$\Delta \mathbf{X}_{fast} = \sum_{r=1}^{M} \frac{1}{\lambda_r} \left[\mathbf{a}_r(\mathbf{X}(T)) \cdot f^r(T) \right]; \tag{3.62}$$

3. Finish the time integration of $\mathbf{X}$ by Modifing the state by the correction from fast time-scales:

$$\mathbf{X}(T+\Delta t) = \hat{\mathbf{X}}(T+\Delta t) - \Delta \mathbf{X}_{fast}. \tag{3.63}$$

End

3.3.5 Modification on SSCSP to Test Variance

Some concerns are addressed in Section 3.3.2 regarding the variance of simulation. Although there is no rigorous mathematical analysis supporting such modifications, the contributions from fast modes on stochastic parts are added heuristically, based on the SSCSP in previous section.

Similar to (3.50), suppose the time integration for (3.48) can be approximated by

$$\begin{aligned}\mathbf{X}(T+\Delta t) - \mathbf{X}(T) = &-\sum_{r=1}^{M} \frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right] \\ &+ \frac{1}{2}\sum_{j=1}^{N}\sum_{r=1}^{M} \frac{\mathcal{N}_r}{\sqrt{|\lambda_r|}}\left[\mathbf{a}_r g_j^r|_{t=T} + \mathbf{a}_r g_j^r|_{t=T+\Delta t}\right] \\ &+ \sum_{j=1}^{N}\int_{T}^{T+\Delta t} P\left(\mu\, dt + \sigma_j \circ dW_j(t)\right) + Remainder,\end{aligned} \tag{3.64}$$

where the second term on RHS represents the contribution from fast mode on stochastic parts. Neither the quantities $\mathbf{a}_r g^r$ at $t = T$ nor $t = T + \Delta t$ can be neglected in order to deal with the Stratonovich integration by midpoint rule.

3.3.6 The SSCSP for Stratonovich SDE with Contribution from Fast Stochastic Modes

The algorithm to perform (3.64) is similar to Section 3.3.4 with a slight modification to deal with the term $\mathbf{a}_r g_i^r|_{t=T+\Delta t}$. The amplitudes g^r at time $t = T + \Delta t$ are approximated by

$$g_i^r(T+\Delta t) = b^r(\mathbf{X}(T)) \cdot \sigma_i(\mathbf{X}(T+\Delta t)), \tag{3.65}$$

where $b^r(\mathbf{X}(T))$ is the orthogonal row vector set computed from $\mathbf{a}_r$, and $\sigma_i(\mathbf{X}(T+\Delta t))$ has to be computed implicitly. For the purpose of saving computational cost and avoiding solving $\mathbf{X}_{i+1} = \mathbf{X}(T+\Delta t)$ implicitly twice, the scheme is designed as follows:

$$\begin{aligned}\mathbf{X}(T+\Delta t) = \mathbf{X}(T) &+ \Delta t \cdot P \cdot (\mu(\mathbf{X}(T))) \\ &+ \frac{1}{2}\sum_{j=1}^{N} \mathcal{N}_j \sqrt{\Delta t} \cdot P \cdot \Big(\sigma_j(\mathbf{X}(T)) + \sigma_j(\mathbf{X}(T+\Delta t))\Big) \\ &- \sum_{r=1}^{M}\left(\frac{1}{\lambda_r}\left[\mathbf{a}_r(\mathbf{X}(T)) \cdot f^r(T)\right] - \sum_{j=1}^{N}\frac{1}{\sqrt{|\lambda_r|}}\mathcal{N}_j \mathbf{a}_r\left[g_i^r(T) + g_i^r(T+\Delta t)\right]\right),\end{aligned} \tag{3.66}$$

where the last line is the contribution from fast mode.

3.3.7 Remarks

Even after the contribution from the fast subdomain is taken into account for the stochastic parts in (3.66), the variance of $\mathbf{X}$ still shows inconsistency comparing with the the result by Euler-Maruyama method with small step size.

This inspires us to test the schemes on an Itô SDE system to monitor the behavior of $\mathbf{X}$ and its variance.

3.4 SSCSP for Itô SDE

Consider the Itô SDE of the stochastic column process $\mathbf{X}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]^T$ with multiple diffusion terms σ_k,

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sum_{k=1}^{M}\sigma_k(\mathbf{X})dW_k(t), \tag{3.67}$$

where the setting is similar to (3.48). The main difference between (3.48) and (3.67) is the latter SDE system is in the Itô sense, instead of the Stratonovich sense as (3.48).

The significant advantage of the Stratonovich SDE is on mathematical analysis, as it follows the usual Calculus rules and we can differentiate (3.33) to get (3.34), which leads to the time integration formula (3.50).

However, since Stratonovich SDE defines the sum as the average of left-hand and right hand sums, it is inevitable to solve the SDE *implicitly*, which results in a much higher computational cost.

Thus, we consider the SDE in the Itô sense as (3.67) to alleviate the efforts on solving $\mathbf{X}$ implicitly caused by Stratonovich SDE, and also consider the contribution from fast modes to the stochastic part. The previous time integration formula (3.50) is modified as

$$\begin{aligned}\mathbf{X}(T+\Delta t)-\mathbf{X}(T) = &-\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]+\sum_{r=1}^{M}\sum_{j=1}^{N}\frac{1}{\sqrt{|\lambda_r|}}\left[\mathbf{a}_r g_j^r|_{t=T}\right]\\ &+\sum_{j=1}^{N}\int_T^{T+\Delta t} P\Big(\mu\,dt+\sigma_j\,dW_j(t)\Big)+Remainder.\end{aligned} \tag{3.68}$$

To build up the simulation based on SSCSP to project the rapid and slow subdomains properly, the first step of the process is computing the contribution from slow modes by the follows:

$$\tilde{\mathbf{X}}(T+\Delta t)=\mathbf{X}(T)+\sum_{k=1}^{M}\int_T^{T+\Delta t} P\Big(\mu\,dt+\sigma_j\,dW_j(t)\Big), \tag{3.69}$$

while the setting of P is as (3.51), it is a constant matrix at $t=T$ with dimension is $N\times N$ and can be computed by $\mathbf{a}_j$ and $\mathbf{b}_j$.

Second, subtracting the adjustment term contributed by the fast modes to complete the time integration:

$$\mathbf{X}(T+\Delta t)=\tilde{\mathbf{X}}(T+\Delta t)-\left(\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]+\sum_{r=1}^{M}\sum_{j=1}^{N}\frac{1}{\sqrt{|\lambda_r|}}\left[\mathbf{a}_r g_j^r|_{t=T}\right]\right). \tag{3.70}$$

The adjustment by fast mode on the RHS is expressed as

$$\Delta\mathbf{X}_{fast}(T+\Delta t)=\sum_{r=1}^{M}\frac{1}{\lambda_r}\left[\mathbf{a}_r f^r|_{t=T}\right]+\sum_{r=1}^{M}\sum_{j=1}^{N}\frac{1}{\sqrt{|\lambda_r|}}\left[\mathbf{a}_r g_j^r|_{t=T}\right]. \tag{3.71}$$

3.4.1 SSCSP Algorithm for Itô SDE

The SSCSP algorithm for Itô SDE is similar with the setting for Stratonovich SDE. The only differences between the two are

1. Using the Euler-Maruyama scheme to handle the time integration on stochastic parts, instead of the mid-point rule for the diffusion term in the Stratonovich SDE.

2. The adjustment from fast subdomain also includes the contribution from stochastic terms, i.e., taking into account of the influence from $\mathbf{a_r}$ by amplitudes g_j^r.

It is developed in two substeps, one for slow modes by (3.69), the other for fast modes by (3.70).

Begin

Step I: Time integration on the slow mode

1. Given the initial state $\mathbf{x} = \mathbf{X}(t)$, evalutate the source term $\mu(\mathbf{x})$ and $\sigma_j(\mathbf{x}), j = 1, \cdots, N$ at time $t = T$.

2. Compute the Jacobian of $\mu(\mathbf{x})$ and its eigenvectors $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$, where the absolute values of corresponding eigenvalues are in descending order, i.e.,

$$|\lambda_1| > |\cdots > |\lambda_N|. \tag{3.72}$$

3. Calculate the corresponding orthogonal row vectors set $\mathbf{B} = [\mathbf{b}^1(t); \cdots; \mathbf{b}^N(t)] = \mathbf{A}^{-1}$.

4. Determine the dimension M of exhausted fast mode at time T. The number M is the greatest integer between 1 and N that satisfies the inequality

$$|\tau_{M+1} \sum_{j=1}^{M} \mathbf{a}_j f^j| < \mathbf{X}_{error}, \tag{3.73}$$

where $\mathbf{X}_{error}$ is an error vector, and τ_{M+1} is the time scale with CSP concept [29] and defined as

$$\tau_{M+1} = \frac{1}{\lambda_{M+1}}. \tag{3.74}$$

5. Evaluate projection matrix for the slow modes

$$P(\mathbf{X}(T)) = I - \sum_{r=1}^{M} \mathbf{a}_r \mathbf{b}^r. \tag{3.75}$$

6. Perform time integration on slow modes during time T to $T + \Delta t$,

$$\begin{aligned} \hat{\mathbf{X}}(T + \Delta t) = \mathbf{X}(T) + \Delta t \cdot P \cdot (\mu(\mathbf{X}(T))) \\ + \sum_{j=1}^{M} \mathcal{N}_j \sqrt{\Delta t} \cdot P \cdot \sigma_j(\mathbf{X}(T)). \end{aligned} \tag{3.76}$$

where $\mathcal{N}_j$ denote standard normal variables.

Step II: Adjustment contributed from fast modes

1. Compute the amplitudes of $\mu(\mathbf{x})$ and $\sigma_j(\mathbf{x})$ for the fast subdomain corresponding to $\mathbf{A^r}$ = $[\mathbf{a}_1, \cdots, \mathbf{a}_M]$ by

$$\begin{aligned} f^r &= \mathbf{b}^r(\mathbf{X}) \cdot \mu(\mathbf{X}), \\ g_j^r &= \mathbf{b}^r(\mathbf{X}) \cdot \sigma_j(\mathbf{X}), \quad j = 1, \cdots, N. \end{aligned} \tag{3.77}$$

2. Calculate the quantity contributed by the fast modes:

$$\Delta \mathbf{X}_{fast} = \sum_{r=1}^{M} \frac{1}{\lambda_r} \left[\mathbf{a}_r(\mathbf{X}(T)) \cdot f^r \right] + \sum_{r=1}^{M} \sum_{j=1}^{N} \frac{1}{\sqrt{|\lambda_r|}} \left[\mathbf{a}_r(\mathbf{X}) \cdot g_j^r \right]. \tag{3.78}$$

3. Finish the time integration of $\mathbf{X}$ by Modifing the state by the correction from fast time-scales:

$$\mathbf{X}(T + \Delta t) = \hat{\mathbf{X}}(T + \Delta t) - \Delta \mathbf{X}_{fast}. \tag{3.79}$$

End

3.4.2 Note

Unlike the SSCSP algorithm for Stratonovich SDE, the SSCSP for Itô SDE maintain the proper variance as expected since the contributions from fast subdomain on diffusion parts are not

neglected. The computation performance is better than SSCSP for Stratonovich SDE because it is basically a modification from forward Euler-Maruyama scheme, and it doesn't need to be solved implicitly as in Stratonovich SDE. It also improves the computational efficiency significantly since the contribution from fast modes is an algebraic amendment, and the selected time step Δt can be much bigger than the time step in ordinary numerical schemes like Euler-Maruyama scheme and Milstein scheme. In next chapter, we shall demonstrate that the SSCSP has better efficiency while maintaining the accuracy.

Chapter 4

Numerical Experiment: The VG model

A model referred to as the VG problem with relavant features of a stiff chemical kinetics mechanism was introduced in [49]. We are going to apply the SSCSP method on the VG model and show its maintenance of stability and improvement of efficiency compared to other numerical methods. The model was characterized by having a constant, explicitly defined small parameter and non-prescribed slow and fast variables. In particular, the following set of six symbolic reactions modeling the dynamics of three species X, Y and Z was considered:

$$X \underset{k_2}{\overset{k_1}{\rightleftharpoons}} 2Y, \tag{4.1}$$

$$X + Y \underset{k_4}{\overset{k_3}{\rightleftharpoons}} Z, \tag{4.2}$$

$$Y + Z \underset{k_6}{\overset{k_5}{\rightleftharpoons}} X, \tag{4.3}$$

with the forward reaction rate constant k^f and reverse reaction rate constants k^b of the three reactions are defined as

$$k^f = [k_1, k_3, k_5] = [\frac{5}{\varepsilon}, \frac{1}{\varepsilon}, 1], \tag{4.4}$$

$$k^b = [k_2, k_4, k_5] = [\frac{5}{\varepsilon}, \frac{1}{\varepsilon}, 1], \tag{4.5}$$

where $\varepsilon > 0$ is an explicit small constant. Note that R_1, R_4, R_6 are unimolecular reactions with propensity functions derived as (1.25), R_3, R_5 are bimolecular reaction on different species with propensity functions in the form of (1.26), and R_2 is a bimolecular reaction on the same species, where the propensity function can be derived by (1.27).

Consider a well-stirred system of molecules of chemical species X, Y and Z that interact through chemical reaction channels $\{R_1, ..., R_6\}$. Assume that the system is confined to a fixed volume Ω and is in thermal equilibrium at a constant temperature. Denote by $X(t)$, $Y(t)$ and $Z(t)$ the concentration of molecules of species X, Y and Z in the system at time t. Let's write $\mathbf{U}(t) \equiv (X(t), Y(t), Z(t))$. Our goal is to estimate the state vector $\mathbf{U}(t)$, given that the system was in state $\mathbf{U}(t_0) = x_0 = (x_0, y_0, z_0)$ at some initial time t_0.

For each reaction channel R_j, the propensity function a_j is defined so that

$a_j(\mathbf{x})dt \equiv$ the probability given $\mathbf{U}(t) = \mathbf{x}$, that one R_j reaction will occur somewhere inside Ω in the next infinitesimal time interval $[t, t + dt)$.

According to [37, 38], the propensity function is in the form

$$a_j(x) = k_j h_j(\mathbf{x}),$$

where k_j is the specific probability rate constant for reaction channel R_j, and $h_j(\mathbf{x})$ is the number of distinct combinations of R_j reactant molecules in the system, $j = 1, \cdots, 6$.

According to [38], the functions $h_j(\mathbf{x})$ for the reaction chanels (4.1) - (4.3) are:

$$a_1(\mathbf{x}) = k_1 h_1(\mathbf{x}) = \frac{5}{\varepsilon} \cdot X, \tag{4.6}$$

$$a_2(\mathbf{x}) = k_2 h_2(\mathbf{x}) = \frac{5}{\varepsilon} \cdot \frac{1}{2} Y(Y-1), \tag{4.7}$$

$$a_3(\mathbf{x}) = k_3 h_3(\mathbf{x}) = \frac{1}{\varepsilon} \cdot XY, \tag{4.8}$$

$$a_4(\mathbf{x}) = k_4 h_4(\mathbf{x}), = \frac{1}{\varepsilon} \cdot Z, \tag{4.9}$$

$$a_5(\mathbf{x}) = k_5 h_5(\mathbf{x}) = YZ, \tag{4.10}$$

$$a_6(\mathbf{x}) = k_6 h_6(\mathbf{x}) = X. \tag{4.11}$$

The *state-change vector* $\mathbf{v}_j$ of system (4.1) - (4.3), which are defined as (1.2), are:

$$\mathbf{v}_1 = [-1, 2, 0]^T \tag{4.12}$$

$$\mathbf{v}_2 = [1, -2, 0]^T, \tag{4.13}$$

$$\mathbf{v}_3 = [-1, -1, 1]^T \tag{4.14}$$

$$\mathbf{v}_4 = [1, 1, -1]^T, \tag{4.15}$$

$$\mathbf{v}_5 = [1, -1, -1]^T, \tag{4.16}$$

$$\mathbf{v}_6 = [-1, 1, 1]^T, \tag{4.17}$$

where $\mathbf{v}_{ij} = [\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}]^T$ indicates the number of species $\{X, Y, Z\}$ changed through the reaction R_i.

4.1 Reation Rate Equation

Following the above construction, the ODE for the mean of each X_i can be deduced as

$$\frac{d\mathbb{E}[X_i]}{dt} = \sum_j v_{ji}\mathbb{E}[a_j(\mathbf{x})], \quad i = 1, 2, 3, \tag{4.18}$$

where the $\mathbf{x} = [X_1(t), X_2(t), X_3(t)]^T = [X, Y, Z]^T$. This gives us the reaction rate equation for the VG system:

$$\begin{aligned}
\frac{dX}{dt} &= -\frac{5}{\varepsilon}X + \frac{5}{2\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z + YZ - X, \\
\frac{dY}{dt} &= \frac{10}{\varepsilon}X - \frac{5}{\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z - YZ + X, \\
\frac{dZ}{dt} &= \frac{1}{\varepsilon}XY - \frac{1}{\varepsilon}Z - YZ + X.
\end{aligned} \tag{4.19}$$

Multiplying ε and then setting $\varepsilon = 0$, we obtain a slow manifold defined by the intersection of two surfaces (See Figure 4.1):

$$Z = XY, \quad 2X = Y(Y-1). \tag{4.20}$$

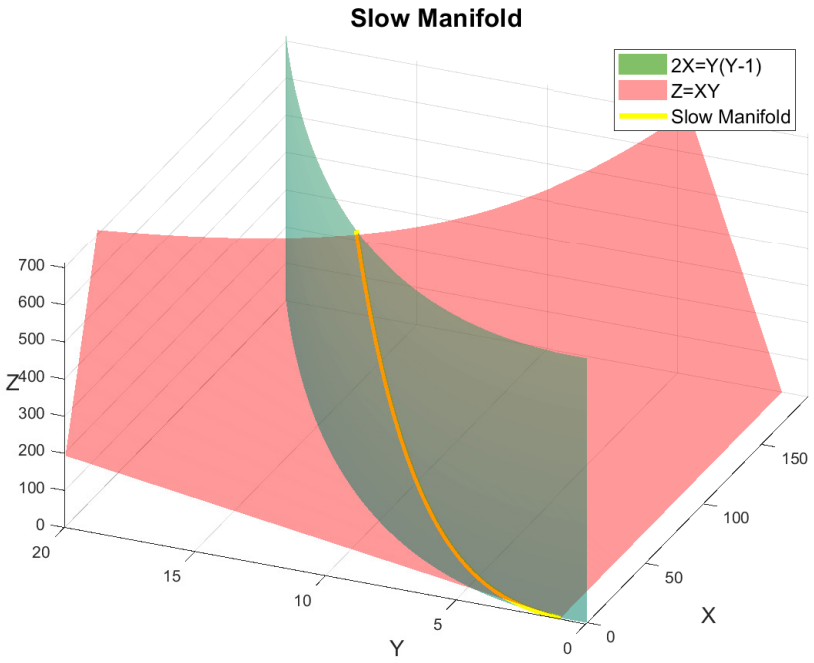


Figure 4.1: This graph shows the slow manifold (yellow curve) which is the intersection of two surfaces: $Z = XY$ (red) and $2X = Y(Y-1)$ (green).

The numerical experiments show that the slow manifold attracts all trajectories of $[X, Y, Z]^T$ starting from an initial value off the slow manifold, and all solutions approach the origin $[X, Y, Z]^T = [0, 0, 0]^T$ as time passes by. We also notice all trajectories are attracted to the surface $2X = Y(Y-1)$ before approaching to the slow manifold (4.20), even if it is close to the surface $Z = XY$ in the beginning.

4.2 Chemical Langevin Equation of VG Model (Itô SDE)

Following the formula (2.13) with propensity functions (4.6) - (4.11) and state-change vectors (4.12) - (4.17), the chemical Langevin equation of VG model can be derived as:

$$\begin{aligned} dX(t) = &\left(-\frac{5}{\varepsilon}X + \frac{5}{2\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z + YZ - X\right)dt \\ &- \sqrt{\frac{5}{\varepsilon}X}dW_1(t) + \sqrt{\frac{5}{2\varepsilon}Y(Y-1)}dW_2(t) - \sqrt{\frac{1}{\varepsilon}XY}dW_3(t), \\ &+ \sqrt{\frac{1}{\varepsilon}Z}dW_4(t) + \sqrt{YZ}dW_5(t) - \sqrt{X}dW_6(t) \end{aligned} \tag{4.21}$$

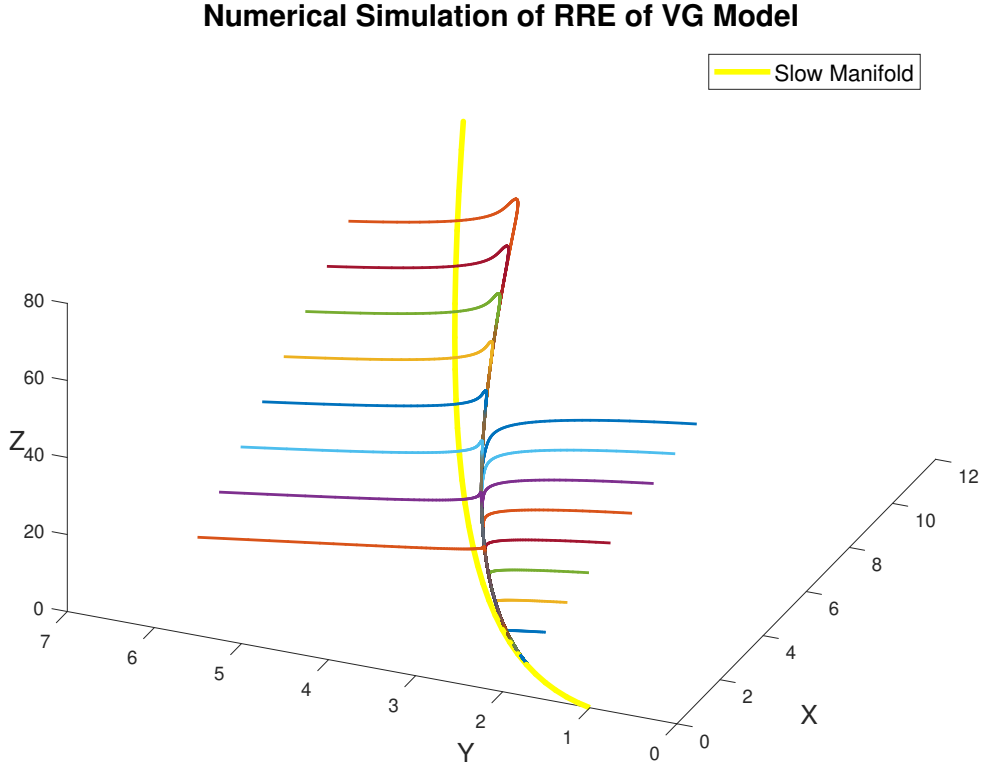


Figure 4.2: The numerical simmulations (colorful curves) of (4.19) compare with the slow manifold.

$$
\begin{aligned}
dY(t) = &\left(\frac{10}{\varepsilon}X - \frac{5}{\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z - YZ + X\right)dt \\
&+ 2\sqrt{\frac{5}{\varepsilon}X}dW_1(t) - 2\sqrt{\frac{5}{2\varepsilon}Y(Y-1)}dW_2(t) - \sqrt{\frac{1}{\varepsilon}XY}dW_3(t) \\
&+ \sqrt{\frac{1}{\varepsilon}Z}dW_4(t) - \sqrt{YZ}dW_5(t) + \sqrt{X}dW_6(t),
\end{aligned} \tag{4.22}
$$

$$
\begin{aligned}
dZ(t) = &\left(\frac{1}{\varepsilon}XY - \frac{1}{\varepsilon}Z - YZ + X\right)dt + \sqrt{\frac{1}{\varepsilon}XY}dW_3(t) \\
&- \sqrt{\frac{1}{\varepsilon}Z}dW_4(t) - \sqrt{YZ}dW_5(t) + \sqrt{X}dW_6(t),
\end{aligned} \tag{4.23}
$$

where $\mathbf{X}(t) = [X(t), Y(t), Z(t)]^T$ denotes the population of species X, Y, Z at time t, ε is a given small parameter, and W_i are independent two-sided Brownian motions. It is an SDE system in the form of (3.48), and the respective column-vector functions $\mu, \sigma_1, \cdots, \sigma_6$ of $\mathbf{X} =$

$[X, Y, Z]^T$ each with three entries are

$$\mu(\mathbf{X}) = \mu([X, Y, Z]^T) = \begin{bmatrix} -\frac{5}{\varepsilon}X + \frac{5}{2\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z + YZ - X \\ \frac{10}{\varepsilon}X - \frac{5}{\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z - YZ + X \\ \frac{1}{\varepsilon}XY - \frac{1}{\varepsilon}Z - YZ + X \end{bmatrix}, \quad (4.24)$$

$$\sigma_1(\mathbf{X}) = \sigma_1([X, Y, Z]^T) = \begin{bmatrix} -\sqrt{\frac{5}{\varepsilon}X} \\ 2\sqrt{\frac{5}{\varepsilon}X} \\ 0 \end{bmatrix}, \quad (4.25)$$

$$\sigma_2(\mathbf{X}) = \sigma_2([X, Y, Z]^T) = \begin{bmatrix} \sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \\ -2\sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \\ 0 \end{bmatrix}, \quad (4.26)$$

$$\sigma_3(\mathbf{X}) = \sigma_3([X, Y, Z]^T) = \begin{bmatrix} -\sqrt{\frac{1}{\varepsilon}XY} \\ -\sqrt{\frac{1}{\varepsilon}XY} \\ \sqrt{\frac{1}{\varepsilon}XY} \end{bmatrix}, \quad (4.27)$$

$$\sigma_4(\mathbf{X}) = \sigma_4([X, Y, Z]^T) = \begin{bmatrix} \sqrt{\frac{1}{\varepsilon}Z} \\ \sqrt{\frac{1}{\varepsilon}Z} \\ -\sqrt{\frac{1}{\varepsilon}Z} \end{bmatrix}, \quad (4.28)$$

$$\sigma_5(\mathbf{X}) = \sigma_5([X,Y,Z]^T) = \begin{bmatrix} \sqrt{YZ} \\ -\sqrt{YZ} \\ -\sqrt{YZ} \end{bmatrix}, \tag{4.29}$$

$$\sigma_6(\mathbf{X}) = \sigma_6([X,Y,Z]^T) = \begin{bmatrix} -\sqrt{X} \\ \sqrt{X} \\ \sqrt{X} \end{bmatrix}. \tag{4.30}$$

4.3 Chemical Langevin Equation of VG Model (Stratonovich SDE)

We may also consider the Chemical Langevin Equation in the Stratonovich sense to take advantage of the mathematical analysis in Chapter 3. Similar to (4.21) - (4.23), the CLEs of the VG model as a Stratonovich SDE system are as follows:

$$\begin{aligned} dX(t) = &\left(-\frac{5}{\varepsilon}X + \frac{5}{2\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z + YZ - X\right)dt \\ &- \sqrt{\frac{5}{\varepsilon}X} \circ dW_1(t) + \sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \circ dW_2(t) - \sqrt{\frac{1}{\varepsilon}XY} \circ dW_3(t), \\ &+ \sqrt{\frac{1}{\varepsilon}Z} \circ dW_4(t) + \sqrt{YZ} \circ dW_5(t) - \sqrt{X} \circ dW_6(t) \end{aligned} \tag{4.31}$$

$$\begin{aligned} dY(t) = &\left(\frac{10}{\varepsilon}X - \frac{5}{\varepsilon}Y(Y-1) - \frac{1}{\varepsilon}XY + \frac{1}{\varepsilon}Z - YZ + X\right)dt \\ &+ 2\sqrt{\frac{5}{\varepsilon}X} \circ dW_1(t) - 2\sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \circ dW_2(t) - \sqrt{\frac{1}{\varepsilon}XY} \circ dW_3(t) \\ &+ \sqrt{\frac{1}{\varepsilon}Z} \circ dW_4(t) - \sqrt{YZ} \circ dW_5(t) + \sqrt{X} \circ dW_6(t), \end{aligned} \tag{4.32}$$

$$\begin{aligned} dZ(t) = &\left(\frac{1}{\varepsilon}XY - \frac{1}{\varepsilon}Z - YZ + X\right)dt + \sqrt{\frac{1}{\varepsilon}XY} \circ dW_3(t) \\ &- \sqrt{\frac{1}{\varepsilon}Z} \circ dW_4(t) - \sqrt{YZ} \circ dW_5(t) + \sqrt{X} \circ dW_6(t), \end{aligned} \tag{4.33}$$

where the $\mu(\mathbf{X})$ and $\sigma_1(\mathbf{X}), \cdots, \sigma_6(\mathbf{X})$ are listed as (4.24) - (4.30).

4.4 Jacobian matrices for SSCSP

While implementing SSCSP, we need to calculate the Jacobian of $\mu(\mathbf{x})$ at given time t to find eigenvectors as spanning basis. The Jacobian of $\mu(\mathbf{x})$ is a 3×3 matrix which is determined by given $\mathbf{X}(t) = \mathbf{x} = [X, Y, Z]^T$:

$$\mathbf{J}_\mu(\mathbf{X}) = \begin{bmatrix} -\frac{5}{\varepsilon} - \frac{Y}{\varepsilon} - 1 & \frac{5}{2\varepsilon}(2Y-1) - \frac{1}{\varepsilon}X + Z & \frac{1}{\varepsilon} + Y \\ \frac{10}{\varepsilon} - \frac{Y}{\varepsilon} + 1 & -\frac{5}{\varepsilon}(2Y-1) - \frac{1}{\varepsilon}X - Z & \frac{1}{\varepsilon} - Y \\ \frac{Y}{\varepsilon} + 1 & \frac{1}{\varepsilon}X - Z & -\frac{1}{\varepsilon} - Y \end{bmatrix}. \tag{4.34}$$

Moreover, we also need to calculate the Jacobians at given time t of vector-functions $\sigma_j(\mathbf{x})$ on diffusion terms in order to

1. perform Newton-Raphson method as (A.14) derived for solving Stratonovich SDE system implicitly,

2. implement Explicit and Implicit Milstein schemes,

since the derivatives of the diffusion terms are involved in the process.

According to (4.25) - (4.30), the Jacobian matrices of $\sigma_1, \cdots, \sigma_6$ have dimension 3×3 which are determined by the given $\mathbf{X}(t) = \mathbf{x} = [X, Y, Z]^T$, and they can be calculated as follows:

$$\mathbf{J}_{\sigma_1}(\mathbf{X}) = \begin{bmatrix} -\frac{1}{2}\sqrt{\frac{5}{\varepsilon X}} & 0 & 0 \\ \sqrt{\frac{5}{\varepsilon X}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.35}$$

$$\mathbf{J}_{\sigma_2}(\mathbf{X}) = \begin{bmatrix} 0 & \sqrt{\frac{5}{8\varepsilon}}\frac{2Y-1}{\sqrt{Y^2-Y}} & 0 \\ 0 & -\sqrt{\frac{5}{2\varepsilon}}\frac{2Y-1}{\sqrt{Y^2-Y}} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.36}$$

$$\mathbf{J}_{\sigma_3}(\mathbf{X}) = \begin{bmatrix} -\frac{1}{2}\sqrt{\frac{Y}{\varepsilon X}} & -\frac{1}{2}\sqrt{\frac{X}{\varepsilon Y}} & 0 \\ -\frac{1}{2}\sqrt{\frac{Y}{\varepsilon X}} & -\frac{1}{2}\sqrt{\frac{X}{\varepsilon Y}} & 0 \\ \frac{1}{2}\sqrt{\frac{Y}{\varepsilon X}} & \frac{1}{2}\sqrt{\frac{X}{\varepsilon Y}} & 0 \end{bmatrix}, \tag{4.37}$$

$$\mathbf{J}_{\sigma_4}(\mathbf{X}) = \begin{bmatrix} 0 & 0 & \frac{1}{2}\sqrt{\frac{1}{\varepsilon Z}} \\ 0 & 0 & \frac{1}{2}\sqrt{\frac{1}{\varepsilon Z}} \\ 0 & 0 & -\frac{1}{2}\sqrt{\frac{1}{\varepsilon Z}} \end{bmatrix}, \tag{4.38}$$

$$\mathbf{J}_{\sigma_5}(\mathbf{X}) = \begin{bmatrix} 0 & \frac{1}{2}\sqrt{\frac{Z}{Y}} & \frac{1}{2}\sqrt{\frac{Y}{Z}} \\ 0 & -\frac{1}{2}\sqrt{\frac{Z}{Y}} & -\frac{1}{2}\sqrt{\frac{Y}{Z}} \\ 0 & -\frac{1}{2}\sqrt{\frac{Z}{Y}} & -\frac{1}{2}\sqrt{\frac{Y}{Z}} \end{bmatrix}, \tag{4.39}$$

$$\mathbf{J}_{\sigma_6}(\mathbf{X}) = \begin{bmatrix} -\frac{1}{2}\sqrt{\frac{1}{X}} & 0 & 0 \\ \frac{1}{2}\sqrt{\frac{1}{X}} & 0 & 0 \\ \frac{1}{2}\sqrt{\frac{1}{X}} & 0 & 0 \end{bmatrix}. \tag{4.40}$$

4.5 SSCSP for VG Model

First, we need to calculate the Jacobian of $\mu(\mathbf{x})$ at given time t for later use. Next, compute the eigenvalues and eigenvectors of J_μ, sort the eigenvalues by theire absolute values from the greatest to the least, i.e., $|\lambda_1| > |\lambda_2| > |\lambda_3|$.

From the observation of the model, the dimension of the rapid mode is set to be $L = 2$ and the time-step needs to be chosen carefully. Select the time step h to satisfy the condition $h > \dfrac{1}{|\lambda_1|}$, which could improve the efficiency over some ordinary numerical methods since they need the numerical time increment to be less than the greatest absolute eigenvalue. Also, $h > \frac{1}{|\lambda_1|}$ allows the fastest mode to exhaust within one time step. Then calculate the middle step

$$\begin{aligned}\hat{\mathbf{X}}_i = \mathbf{X}_i + \mathbf{a}_1 f^1 \cdot \frac{1}{|\lambda_1|} + \mathbf{a}_2 f^2 \cdot \tilde{h} + \frac{1}{\sqrt{|\lambda_1|}} \sum_{j=1}^{6} \mathbf{a}_1 g_j^1 \mathcal{N}_j \\ + \sqrt{\tilde{h}} \sum_{j=1}^{6} \mathbf{a}_2 g_j^2 \mathcal{N}_j\end{aligned} \tag{4.41}$$

where

$$\tilde{h} = \alpha \frac{1}{|\lambda_1|} + (1-\alpha)\frac{1}{|\lambda_2|}, \quad 0 < \alpha < 1. \tag{4.42}$$

Note that the middle step only involves the integration in the rapid subdomain. Then finish the scheme by calculating

$$\mathbf{X}_{i+1} = \hat{\mathbf{X}}_i + \mathbf{a}_3 f^3 \cdot h + \sqrt{h} \sum_{j=1}^{6} \mathbf{a}_3 g_j^3 \mathcal{N}_j, \tag{4.43}$$

where the $\mathbf{a}_k, f^k, g_j^k$ are depending on $\mathbf{X}_i$ instead of $\hat{\mathbf{X}}_i$.

The other way to understand the scheme is in terms of the following expression:

$$\begin{aligned}\mathbf{X}_{i+1} = \mathbf{X}_i + \sum_{k=1}^{3} h_k \cdot \mathbf{a}_k f^k \\ + \sum_{k=1}^{3} \sqrt{h_k} (g_1^k \quad g_2^k \quad \cdots \quad g_6^k) \cdot \begin{pmatrix} \mathcal{N}_1 \\ \vdots \\ \mathcal{N}_6 \end{pmatrix} \mathbf{a}_k,\end{aligned} \tag{4.44}$$

where $\mathbf{a}_k$ are the eigenvectors corresponding to the sorted absolute eigenvalues $|\lambda_k|$, $\mathcal{N}_j$ as standard normal random variables, $f^k = \mathbf{b}^k \cdot \mu, \quad g_j^k = \mathbf{b}^k \cdot \sigma_j$, and

$$\begin{aligned} h_1 &= \frac{1}{|\lambda_1|}, \\ h_2 &= \alpha\frac{1}{|\lambda_1|} + (1-\alpha)\frac{1}{|\lambda_2|}, \qquad \text{where } 0 < \alpha < 1, \\ h_3 &= h \qquad\qquad\qquad\qquad \text{which satisfies } h > \frac{1}{|\lambda_1|}. \end{aligned} \tag{4.45}$$

4.5.1 Matrix Form of the Algorithm

For algorithms applied to multi-dimensional SDEs, it is feasible to represent the steps by matrix operations, which is very helpful especially for implementing the algorithm in the programming stage. The steps for computing the matrices and performing the algorithm are as follows:

1. Given the intial state
$$\begin{bmatrix} | \\ \mathbf{X}_i \\ | \end{bmatrix} = \begin{bmatrix} | \\ \mathbf{X}(t) \\ | \end{bmatrix} = \begin{bmatrix} X(t) \\ Y(t) \\ X(t) \end{bmatrix} \tag{4.46}$$
which is a 3×1 column vector in VG model.

2. Evaluate $\mu(\mathbf{X}_i)$ and $\sigma_1(\mathbf{X}_1), \cdots, \sigma_6(\mathbf{X}_6)$. Each of them are 3×1 column vectors. Denote
$$\mathbf{P} = \begin{bmatrix} | \\ \mu(\mathbf{X}_i) \\ | \end{bmatrix}, \tag{4.47}$$
$$\mathbf{Q} = \begin{bmatrix} | & | & & | \\ \sigma_1(\mathbf{X}_i) & \sigma_2(\mathbf{X}_i) & \dots & \sigma_6(\mathbf{X}_i) \\ | & | & & | \end{bmatrix}, \tag{4.48}$$
where $\mathbf{P} = \mu(\mathbf{X}_i)$ is 3×1 column vector, and $\mathbf{Q}$ is a 3×6 rectangular matrix.

3. Compute the Jacobian matrix of $\mu(\mathbf{X})$, then evaluate the eigenvalues $|\lambda_1| > |\lambda_2| > |\lambda_3|$ and their corresponding eigenvector $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. Denote

$$\mathbf{A} = \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix}, \tag{4.49}$$

where $\mathbf{A}$ is a 3×3 square matrix. Then find the orthogonal row vector sets $\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3$ and denote

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} — & \mathbf{b}^1 & — \\ — & \mathbf{b}^2 & — \\ — & \mathbf{b}^3 & — \end{bmatrix}. \tag{4.50}$$

4. Evaluate the amplitude of μ and $\sigma_1, \cdots, \sigma_6$ with respect to $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ by

$$\mathbf{F} = \begin{bmatrix} f^1 \\ f^2 \\ f^3 \end{bmatrix} = \begin{bmatrix} — & \mathbf{b}^1 & — \\ — & \mathbf{b}^2 & — \\ — & \mathbf{b}^3 & — \end{bmatrix} \cdot \begin{bmatrix} | \\ \mu(\mathbf{X}_i) \\ | \end{bmatrix} = \mathbf{B} \cdot \mathbf{P}, \tag{4.51}$$

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} g_1^1 & \cdots\cdots\cdots & g_6^1 \\ g_1^2 & \cdots\cdots\cdots & g_6^2 \\ g_1^3 & \cdots\cdots\cdots & g_6^3 \end{bmatrix} = \begin{bmatrix} ——— & \mathbf{g}^1 & ——— \\ ——— & \mathbf{g}^2 & ——— \\ ——— & \mathbf{g}^3 & ——— \end{bmatrix} \\ &= \begin{bmatrix} — & \mathbf{b}^1 & — \\ — & \mathbf{b}^2 & — \\ — & \mathbf{b}^3 & — \end{bmatrix} \cdot \begin{bmatrix} | & | & & | \\ \sigma_1(\mathbf{X}_i) & \sigma_2(\mathbf{X}_i) & \dots & \sigma_6(\mathbf{X}_i) \\ | & | & & | \end{bmatrix} = \mathbf{B} \cdot \mathbf{Q}, \end{aligned} \tag{4.52}$$

5. Generate a column vector with six standard normal variables

$$\begin{bmatrix} | \\ dW \\ | \end{bmatrix} = \begin{bmatrix} \mathcal{N}_1 \\ \vdots \\ \mathcal{N}_6 \end{bmatrix}. \tag{4.53}$$

6. Compute the column vector $[h\mathbf{F}]$ with the $i-$th entry is $h_i \cdot f^i$, i.e.,

$$\begin{bmatrix} | \\ h\mathbf{F} \\ | \end{bmatrix} = \begin{bmatrix} h_1 \cdot f^1 \\ h_2 \cdot f^2 \\ h_3 \cdot f^3 \end{bmatrix}, \tag{4.54}$$

and the 3×6 rectangular matrix $[\sqrt{h}\mathbf{G}]$ with the $i-$th row equals to $\sqrt{h_i} \cdot \mathbf{g}^i$:

$$\begin{bmatrix} | \\ \sqrt{h}\mathbf{G} \\ | \end{bmatrix} = \begin{bmatrix} \text{———} & \sqrt{h_1} \cdot \mathbf{g}^1 & \text{———} \\ \text{———} & \sqrt{h_2} \cdot \mathbf{g}^2 & \text{———} \\ \text{———} & \sqrt{h_3} \cdot \mathbf{g}^3 & \text{———} \end{bmatrix}. \tag{4.55}$$

7. Finally, get the updated state $\mathbf{X}_{i+1} = \mathbf{X}(T + \Delta t)$ by matrix operation:

$$\begin{aligned} \mathbf{X}_{i+1} &= \mathbf{X}_i + \mathbf{A} \cdot [h\mathbf{F}] + \mathbf{A} \cdot [\sqrt{h}\mathbf{G}] \cdot dW \\ &= \mathbf{X}_i + \mathbf{A} \cdot \left([h\mathbf{F}] + [\sqrt{h}\mathbf{G}] \cdot dW \right) \end{aligned} \tag{4.56}$$

or

$$\begin{aligned} \begin{bmatrix} | \\ \mathbf{X}_{i+1} \\ | \end{bmatrix} = \begin{bmatrix} | \\ \mathbf{X}_i \\ | \end{bmatrix} + \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} \cdot \begin{bmatrix} h_1 \cdot f^1 \\ h_2 \cdot f^2 \\ h_3 \cdot f^3 \end{bmatrix} \\ + \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} \cdot \begin{bmatrix} — & \sqrt{h_1} \cdot \mathbf{g}^1 & — \\ — & \sqrt{h_2} \cdot \mathbf{g}^2 & — \\ — & \sqrt{h_3} \cdot \mathbf{g}^3 & — \end{bmatrix} \cdot \begin{bmatrix} | \\ dW \\ | \end{bmatrix}. \end{aligned} \tag{4.57}$$

As the matrix set up abow, it is the forward Euler-Maruyama scheme if $h_1 = h_2 = h_3$.

4.6 Numerical Simulation

For the purpose of comparison, we test both the Stratonovich SDE and Itô SDE systems for the VG model. We are interested in the circumstance when α is set to be close to 1, i.e., when h_2 approaches to $\frac{1}{|\lambda_1|}$ as α close to 1. Set $\varepsilon = 1000$ and the initial condition

$$X_0 = 100, Y_0 = 200, Z_0 = 300.$$

Some known numerical schemes are used to demonstrate the stability and performance of SSCSP algorithm.

4.6.1 Stratonovich SDE

For a Stratonovich SDE system

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sum_{k=1}^{M} \sigma_k(\mathbf{X}) \circ dW_k(t), \tag{4.58}$$

and step size Δt, the **Euler-Maruyama scheme** is used to compare with SSCSP method for $i = 0, 1, \cdots N$,

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mu(\mathbf{X}_i)\Delta t + \frac{1}{2}\sum_{k=1}^{M}\left(\sigma_k(\mathbf{X}_i) + \sigma_k(\mathbf{X}_{i+1})\right)\Delta W_{k,i}, \tag{4.59}$$

where

$$\Delta W_{k,i} := W_{k,i}(T + \Delta t) - W_{k,i}(T) \tag{4.60}$$

are independent and identically distributed normal random variables with expected value zero and variance Δt. Equivelently

$$\Delta W_{k,i} = \mathcal{N}_j\sqrt{\Delta t}, \tag{4.61}$$

where $\mathcal{N}_j$ denotes the normal random variable. The midpoint rule is applied (see e.g., [50, 51]) to handle the Stratonovich integral.

While solving the Stratonovich SDE system using midpoint rule involves implicit computations, the computational efficiency is affected regardless of the numerical scheme. Newton-Raphson Method is used to solve the multidimensional equation implicitly (see Appendix A).

We tested two schemes, one follows (3.63) which does not include the contribution from fast subdomain on stochastic part (See Figure 4.3), which shows reasonable variance from the results implementing SSCSP schemes; the other follows (3.66) where the contributions are included, however, the variance is greater than expected obviously from the Figure 4.4.

4.6.2 Itô SDE

Many widely used numerical schemes for the numeric solution of Langevin equations requires the equation to be in Itô form [52]. For a Itô SDE

$$d\mathbf{X} = \mu(\mathbf{X})dt + \sum_{k=1}^{M} \sigma_k(\mathbf{X})dW_k(t), \tag{4.62}$$

with step size Δt, these shemes are:

1. **Euler-Maruyama Scheme:**

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mu(\mathbf{X}_i)\Delta t + \sum_{k=1}^{M} \sigma_k(\mathbf{X}_i)\Delta W_{k,i}, \tag{4.63}$$

   where

$$\Delta W_{k,i} := W_{k,i}(T + \Delta t) - W_{k,i}(T) \tag{4.64}$$

   are independent and identically distributed normal random variables with expected value zero and variance Δt.

2. **Implicit Euler-Maruyama Scheme**

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \left[(1-\beta)\cdot\mu(\mathbf{X}_i) + \beta\cdot\mu(\mathbf{X}_{i+1})\right]\Delta t + \sum_{k=1}^{M} \sigma_k(\mathbf{X}_i)\Delta W_{k,i}, \tag{4.65}$$

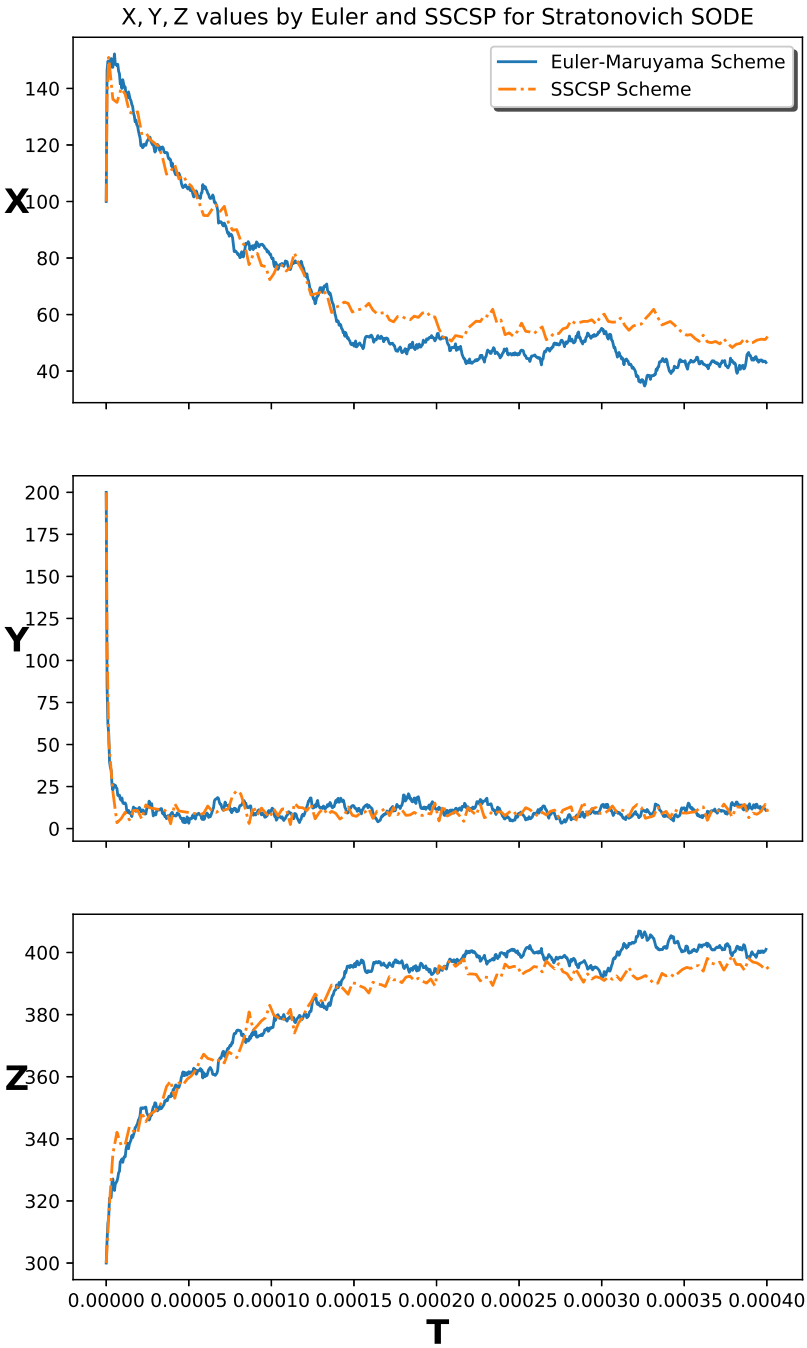


Figure 4.3: X, Y, Z values by Euler and SSCSP for Stratonovich SDE.

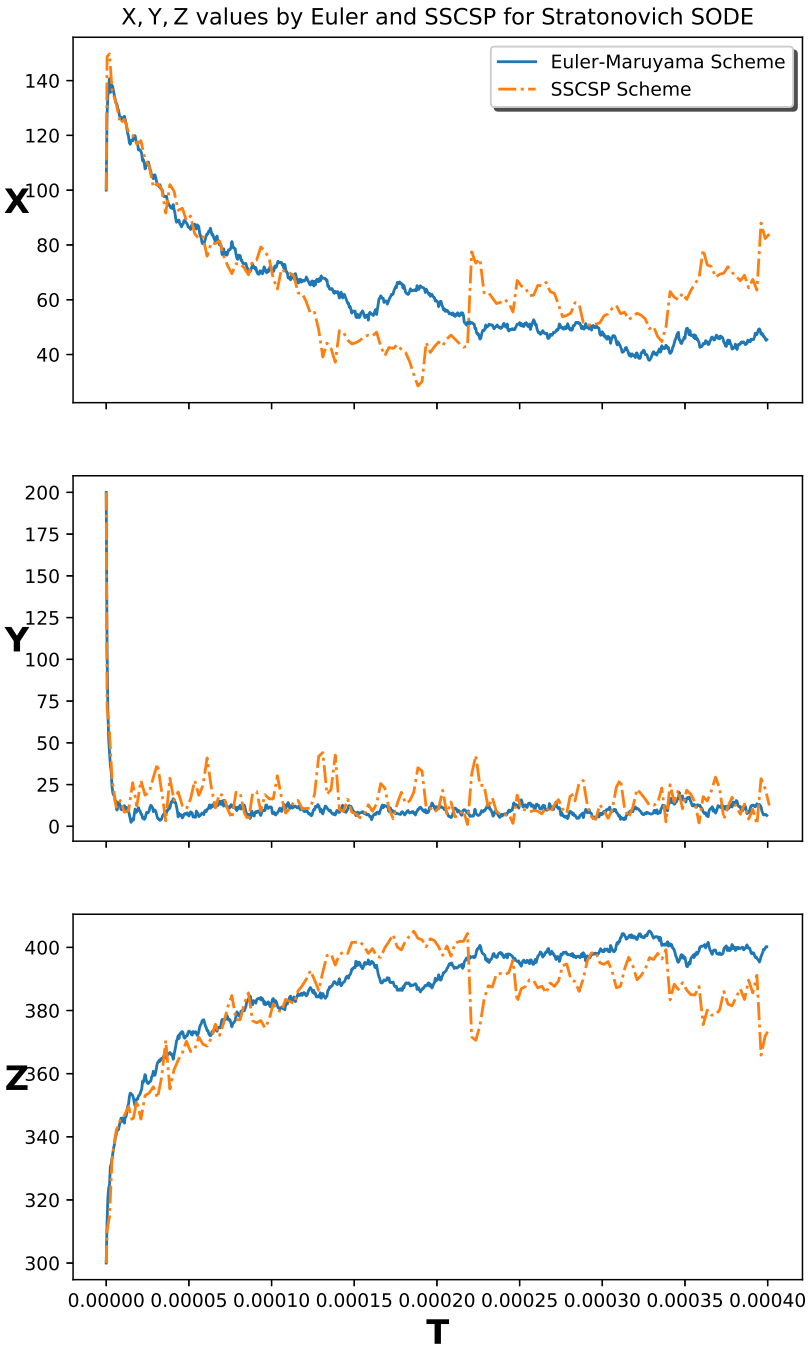


Figure 4.4: X, Y, Z values by Euler and SSCSP for Stratonovich SDE.

where β is a parameter with $0 \leq \beta \leq 1$ which characterizes the degree of implicitness. (When $\beta = 0$, it reduces to the Euler-Maruyama scheme (4.63)).

3. **Explicit Milstein Scheme:**

   Introduced in [50], the Milstein scheme is associated with Itô-Taylor expansion. The Milstein method for multi-dimensional stochastic system is given by

$$\begin{aligned} \mathbf{X}_{i+1}^j = \mathbf{X}_i^j + \mu(\mathbf{X}_i)\Delta t + \sum_{k=1}^{M} \sigma_k^j(\mathbf{X_i})\Delta W_{k,i} \\ + \sum_{k=1}^{M} \frac{1}{2}\sigma_k^j(\mathbf{X}_i)\frac{\partial}{\partial x_j}\sigma_k^j(\mathbf{X}_i)\Bigg((\Delta W_{k,i})^2 - \Delta t\Bigg), \end{aligned} \tag{4.66}$$

   for $\mathbf{X}_i^j$ represent the $j-$th component of $\mathbf{X}_i$ at the $i-$th time step.

   The main difference between Euler-Maruyama and Milstein scheme is that Milstein carries one extra term. The Itô-Taylor expansion is used in order to derive this method, hence providing a componentwise order 1.0 strong Taylor scheme. See more discussions about strong Taylor scheme in [51, 53].

4. **Implicit Milstein Scheme:**

   The order 1.0 implicit strong Taylor scheme given in [51] is a drift-implicit version of the Milstein scheme, also called Implicit Milstein scheme:

$$\begin{aligned} \mathbf{X}_{i+1}^j = \mathbf{X}_i^j + \Bigg[(1-\beta)\cdot\mu(\mathbf{X}_i) + \beta\cdot\mu(\mathbf{X}_{i+1})\Bigg]\Delta t + \sum_{k=1}^{M} \sigma_k^j(\mathbf{X_i})\Delta W_{k,i} \\ + \sum_{k=1}^{M} \frac{1}{2}\sigma_k^j(\mathbf{X}_i)\frac{\partial}{\partial x_j}\sigma_k^j(\mathbf{X}_i)\Bigg((\Delta W_{k,i})^2 - \Delta t\Bigg), \end{aligned} \tag{4.67}$$

   where β is a parameter with $0 \leq \beta \leq 1$ which characterises the degree of implicitness. (When $\beta = 0$, it reduces to the Explicit Milstein scheme (4.66)).

Figure 4.5 is the comparison of SSCSP with Explicit Euler-Maruyama (4.63) and Explicit Milstein (4.66), which shows the stability of SSCSP method.

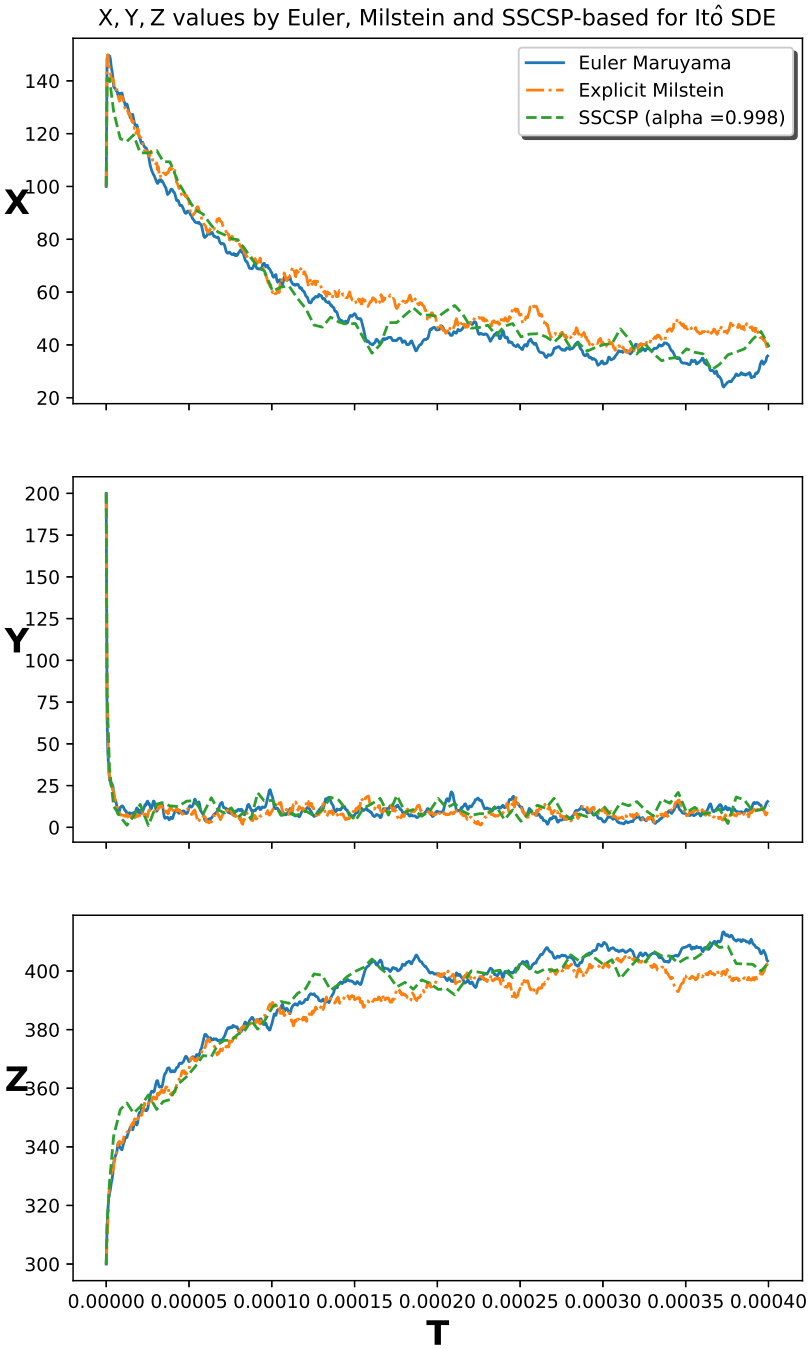


Figure 4.5: X, Y, Z values by Euler and SSCSP for Itô SDE.

It also improves the efficiency tremendously. Even though we need to compute the Jacobian and its eigenvalues and eigenvectors at each state $\mathbf{X}(t)$, the SSCSP still saves the running time greatly since bigger time steps are used, which depends on the eigenvalues of $\mathbf{J}\mu$.

Numerical Scheme	Running Time	Δt
Euler-Maruyama (EM)	23 min	$5 \cdot 10^{-8}$
Implicit EM ($\beta = 0.5$)	71 min	$1 \cdot 10^{-7}$
Explicit Milstein	93 min	$5 \cdot 10^{-8}$
Implicit Milstein ($\beta = 0.5$)	111 min	$1 \cdot 10^{-7}$
SSCSP ($\alpha = 0.99$)	5 min	$2.5 \cdot 10^{-6} (T > 0.0001)$
SSCSP ($\alpha = 0.998$)	6 min	$2.5 \cdot 10^{-6} (T > 0.00013)$

Chapter 5

Summary

The goal of this research is to develop efficient and accurate numerical schemes for chemical Langevin equations with stiffness. In the first part of this thesis, we introduced the mathematical framework for modeling chemical reaction networks under different regimes. In particular, we first introduced the reaction rate equation, which are systems of continuous and deterministic ordinary differential equations on concentration of chemical species, that can be used to model chemical reaction systems at macro scale. We then introduced the chemical Master equations, which are partial differential equations or essentially infinite dimensional ordinary differential equations on probability density functions of discrete states for number of chemical species. While chemical Master equations provide complete information on an underlying chemical reaction systems, they are almost analytically intractable and also numerically expensive to simulate, and hence have been used mainly for micro-scale systems. To bridge the gap between RREs and CMEs, chemical Langevin equations are introduced as the result of the tau-leaping method. CLEs are continuous stochastic differential or difference equations on concentrations of chemical species, and are the focus of this research.

With the consideration that most chemical reaction systems involve multiple time scales, their governing CLEs genuinely involve stiffness. Numerical simulations of stochastic differential equations with stiffness have always been an important and challenging topics. It is well known that due to rapid changes in some of the state variables, explicit schemes have to take extremely small step sizes to ensure numerical stability. Alternatively, implicit schemes achieve

better numerical stability, but both types of scheme are computationally expensive. In this research we developed explicit time integration schemes for stiff CLEs, based on the concept of stochastic computational singular perturbation.

Stochastic computational singular perturbation was generalized from computational singular perturbation for deterministic ODEs with stiffness, but incorporates stochasticity in the system. The idea for SCSP is to project the drift and diffusion terms onto a carefully chosen set of basis vectors, such that the evolution of fast and slow modes can be separated. The stochastic system can then be integrated in two steps, one for the slow dynamics which can be integrated completely on the slow subdomain, and one for the fast dynamics which are approximated by sophisticatedly derived algebraic relations. One of the challenges to integrate according to full SCSP schemes lies in the multiple noise structures, that lead to multiple diffusion matrices which are usually singular. Noticing the special similarity of the drift and diffusion terms in CLEs, in this research we developed numerical schemes based on a simplified SCSP, in which only one set of basis vectors is chosen according to the drift part of the underlying CLE.

We developed SSCSP numerical schemes for both CLEs with Itô and Stratonovich noise and compared their difference. We have also tested all for our schemes by a real stiff chemical reaction system involving three species and six reactions channels, referred to as the VG system.

In particular, we first derived the governing CLEs for the VG system and conducted preliminary eigen analysis of the system. We then applied our proposed SSCSP-based explicit time integration schemes to the CLE system with either Stratonovich or Itô noise. To facilitate the coding process, matrix formulation of the schemes was also developed.

Our numerical experiments show that

(i) SSCSP-based explicit time integration schemes applied to CLEs with Stratonovich noise give correct mean but misses the correct variance. Although no theoritical proof is provided, we conjecture this is due to the implicit algorithms used to handle the Stratonovich noise terms.

(ii) SSCSP-based explicit time integration schemes applied to CLEs with Itô noise captures correctly both the mean and variance of the system.

(iii) The computing times of SSCSP-based explicit time integration schemes are significantly lower than those of the classical schemes Euler-Maruyama, Implicit Euler-Maruyama, Explicit Milstein, Implicit Milstein for stiff systems.

Note that several schemes proposed in the research are heuristic with partial analytical support, though their performance was demonstrated by numerical experiments. Future work includes (i) completing the mathematical framework of explicit time integration schemes based on SCSP and SSCSP; (ii) investigating how the choice of step sizes affect accuracy and efficiency of the schemes; (iii) proving convergence of SCSP and SSCSP algorithms and finding convergence rates if possible; and (iv) developing precise description of random slow manifold.

Reference

[1] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annu. Rev. Phys. Chem.*, vol. 58, pp. 35–55, 2007.

[2] L. S. H. and G. D. A., "The csp method for simplifying kinetics," *International Journal of Chemical Kinetics*, vol. 26, no. 4, pp. 461–486, 1994.

[3] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[4] D. of Chemical Nomenclature, S. R. I. U. of Pure, A. Chemistry, M. Nic, J. Jirat, and B. Kosata, *IUPAC goldbook*. IUPAC, 2006.

[5] A. Fersht, *Enzyme structure and mechnisms, 2nd Edition.* W H Freeman and Co (Sd), 1975.

[6] A. L. Kuharsky and A. L. Fogelson, "Surface-mediated control of blood coagulation: The role of binding site densities and platelet deposition," *Biophysical Journal*, vol. 80, no. 3, pp. 1050 – 1074, 2001.

[7] N. Peters and B. Rogg, *Reduced Kinetic Mechanisms for Applications in Combustion Systems*. Springer-Verlag Berlin Heidelberg, 1993.

[8] J. Seinfeld and S. Pandis, *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. 1998.

[9] R. C. Anderson, "Combustion theory: The fundamental theory of chemically reacting flow systems (williams, forman a.)," *Journal of Chemical Education*, vol. 42, no. 7, p. A548, 1965.

[10] J. Warnatz, U. Maas, R. Dibble, and R. Dibble, *Combustion: Physical and Chemical Fundamentals, Modelling and Simulation, Experiments, Pollutant Formation.* Springer, 2001.

[11] S. Lam and D. Goussis, "Understanding complex chemical kinetics with computational singular perturbation," *Symposium (International) on Combustion*, vol. 22, no. 1, pp. 931 – 941, 1989.

[12] A. Kazakov, M. Chaos, Z. Zhao, and F. L. Dryer, "Computational singular perturbation analysis of two-stage ignition of large hydrocarbons," *The Journal of Physical Chemistry A*, vol. 110, no. 21, pp. 7003–7009, 2006. PMID: 16722715.

[13] M. Valorani and S. Paolucci, "The g-scheme: A framework for multi-scale adaptive model reduction," *J. Comput. Physics*, vol. 228, pp. 4665–4701, 2009.

[14] D. Goussis and H. Najm, "Model reduction and physical understanding of slowly oscillating processes: The circadian cycle," vol. 5, 01 2006.

[15] H. N. Najm, M. Valorani, D. A. Goussis, and J. Prager, "Analysis of methane–air edge flame structure," *Combustion Theory and Modelling*, vol. 14, no. 2, pp. 257–294, 2010.

[16] J. Prager, H. Najm, M. Valorani, and D. Goussis, "Structure of n-heptane/air triple flames in partially-premixed mixing layers," vol. 158, pp. 2128–2144, 11 2011.

[17] M. Valorani, H. Najm, and D. Goussis, "Csp analysis of a transient flame-vortex interaction," vol. 134, pp. 35–53, 07 2003.

[18] V. Bykov and V. Goladshtein, "Fast and slow invariant manifolds in chemical kinetics," *Computers and Mathematics with Applications*, vol. 65, no. 10, pp. 1502 – 1515, 2013. Grasping Complexity.

[19] C. Gear, T. Kaper, I. Kevrekidis, and A. Zagaris, "Projecting to a slow manifold: Singularly perturbed systems and legacy codes," *SIAM Journal on Applied Dynamical Systems*, vol. 4, no. 3, pp. 711–732, 2005.

[20] D. Goussis and S. Lam, "A study of homogeneous methanol oxidation kinetics using csp," *Symposium (International) on Combustion*, vol. 24, no. 1, pp. 113 – 120, 1992. Twenty-Fourth Symposium on Combustion.

[21] D. A. Goussis and M. Valorani, "An efficient iterative algorithm for the approximation of the fast and slow dynamics of stiff systems," *J. Comput. Phys.*, vol. 214, pp. 316–346, May 2006.

[22] M. Hadjinicolaou and D. Goussis, "Asymptotic solution of stiff pdes with the csp method: The reaction diffusion equation," vol. 20, 10 1998.

[23] J. C. Lee, H. N. Najm, S. Lefantzi, J. Ray, M. Frenklach, M. Valorani, and D. A. Goussis, "A csp and tabulation-based adaptive chemistry model," *Combustion Theory and Modelling*, vol. 11, no. 1, pp. 73–102, 2007.

[24] T. Lu, Y. Ju, and C. K. Law, "Complex csp for chemistry reduction and analysis," *Combustion and Flame*, vol. 126, no. 1, pp. 1445 – 1455, 2001.

[25] A. Massias, D. Diamantis, E. Mastorakos, and D. Goussis, "Global reduced mechanisms for methane and hydrogen combustion with nitric oxide formation constructed with csp data," *Combustion Theory and Modelling*, vol. 3, no. 2, pp. 233–257, 1999.

[26] J. Prager, H. N. Najm, M. Valorani, and D. A. Goussis, "Skeletal mechanism generation with csp and validation for premixed n-heptane flames," *Proceedings of the Combustion Institute*, vol. 32, no. 1, pp. 509 – 517, 2009.

[27] M. Valorani, F. Creta, F. Donato, H. N. Najm, and D. A. Goussis, "Skeletal mechanism generation and analysis for n-heptane with csp," *Proceedings of the Combustion Institute*, vol. 31, no. 1, pp. 483 – 490, 2007.

[28] M. Valorani, H. N. Najm, and D. A. Goussis, "Csp analysis of a transient flame-vortex interaction: time scales and manifolds," *Combustion and Flame*, vol. 134, no. 1, pp. 35 – 53, 2003.

[29] M. Valorani and D. A. Goussis, "Explicit Time-Scale Splitting Algorithm for Stiff Problems: Auto-ignition of Gaseous Mixtures behind a Steady Shock," *Journal of Computational Physics*, vol. 169, pp. 44–79, May 2001.

[30] S. H. Lam, "Using csp to understand complex chemical kinetics," *Combustion Science and Technology*, vol. 89, no. 5-6, pp. 375–404, 1993.

[31] B. J. Debusschere, Y. M. Marzouk, H. N. Najm, B. Rhoads, D. A. Goussis, and M. Valorani, "Computational singular perturbation with non-parametric tabulation of slow manifolds for time integration of stiff chemical kinetics," *Combustion Theory and Modelling*, vol. 16, no. 1, pp. 173–198, 2012.

[32] D. T. Gillespie, "The chemical langevin equation," *The Journal of Chemical Physics*, vol. 113, no. 1, pp. 297–306, 2000.

[33] Y. Cao, D. T. Gillespie, and L. R. Petzold, "The slow-scale stochastic simulation algorithm," *The Journal of Chemical Physics*, vol. 122, no. 1, p. 014116, 2005.

[34] C. V. Rao and A. P. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm," *The Journal of Chemical Physics*, vol. 118, no. 11, pp. 4999–5010, 2003.

[35] D. McQuarrie, *Stochastic Approach to Chemical Kinetics*. Methuen's monographs on applied probability and statistics, Methuen, 1968.

[36] N. V. KAMPEN, *Stochastic Processes in Physics and Chemistry (Third Edition)*. North-Holland Personal Library, Amsterdam: Elsevier, third edition ed., 2007.

[37] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Computational Phys.*, vol. 22, no. 4, pp. 403–434, 1976.

[38] D. T. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A: Statistical Mechanics and its Applications*, vol. 188, no. 1, pp. 404 – 425, 1992.

[39] S. MacNamara, A. M. Bersani, K. Burrage, and R. B. Sidje, "Stochastic chemical kinetics and the total quasi-steady-state assumption: Application to the stochastic simulation algorithm and chemical master equation," *The Journal of Chemical Physics*, vol. 129, no. 9, p. 095105, 2008.

[40] R. Erban, J. Chapman, and P. Maini, "A practical guide to stochastic simulations of reaction-diffusion processes," 05 2007.

[41] D. F. Anderson, "A modified next reaction method for simulating chemical systems with time dependent propensities and delays," *The Journal of Chemical Physics*, vol. 127, no. 21, p. 214107, 2007.

[42] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *The Journal of Physical Chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.

[43] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001.

[44] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Efficient step size selection for the tau-leaping simulation method," *The Journal of Chemical Physics*, vol. 124, no. 4, p. 044109, 2006.

[45] T. Bui and T. Bui, "Numerical methods for extremely stiff systems of ordinary differential equations," *Applied Mathematical Modelling*, vol. 3, no. 5, pp. 355 – 358, 1979.

[46] K. Nipp, "Numerical integration of stiff ode's of singular perturbation type," *Zeitschrift für angewandte Mathematik und Physik ZAMP*, vol. 42, pp. 53–79, Jan 1991.

[47] S. H. Lam, *Singular Perturbation for Stiff Equations Using Numerical Methods*, pp. 3–19. Boston, MA: Springer US, 1985.

[48] L. Wang, X. Han, Y. Cao, and H. Najm, "Computational singular perturbation analysis of stochastic chemical systems with stiffness," vol. 335, 01 2017.

[49] M. Valorani, D. A. Goussis, F. Creta, and H. N. Najm, "Higher order corrections in the approximation of low-dimensional manifolds and the construction of simplified problems with the CSP method," *J. Comput. Phys.*, vol. 209, no. 2, pp. 754–786, 2005.

[50] M. V. T. Grigori Noah Milstein, *Stochastic Numerics for Mathematical Physics*. Springer-Verlag Berlin Heidelberg, 2004.

[51] E. P. Peter E. Kloeden, *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag Berlin Heidelberg, 1992.

[52] R. Perez-Carrasco and J. M. Sancho, "Stochastic algorithms for discontinuous multiplicative white noise," *Phys. Rev. E*, vol. 81, p. 032104, Mar 2010.

[53] G. N. Milstein, *Numerical Integration of Stochastic Differential Equations*. Springer Netherlands, 1995.

[54] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Avoiding negative populations in explicit poisson tau-leaping," *The Journal of Chemical Physics*, vol. 123, no. 5, p. 054104, 2005.

[55] Y. Cao and L. R. Petzold, "Slow scale tau-leaping method.," *Computer methods in applied mechanics and engineering*, vol. 197 43-44, pp. 3472–3479, 2008.

[56] D. F. Anderson, "Incorporating postleap checks in tau-leaping," *The Journal of Chemical Physics*, vol. 128, no. 5, p. 054103, 2008.

[57] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Efficient step size selection for the tau-leaping simulation method," *The Journal of Chemical Physics*, vol. 124, no. 4, p. 044109, 2006.

[58] T. Tian and K. Burrage, "Binomial leap methods for simulating stochastic chemical kinetics," *The Journal of Chemical Physics*, vol. 121, no. 21, pp. 10356–10364, 2004.

[59] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis, "Binomial distribution based tau-leap accelerated stochastic simulation," *The Journal of Chemical Physics*, vol. 122, no. 2, p. 024112, 2005.

Appendices

Appendix A

Newton's Method on Performing Stratonovich Integration

A.1 Multivariate Newton's Method (Newton-Raphson Method)

For solving a multivariate equation system

$$\begin{cases} f_1(x_1, \cdots, x_N) = f_1(\mathbf{x}) = 0 \\ f_2(x_1, \cdots, x_N) = f_2(\mathbf{x}) = 0 \\ \cdots\cdots\cdots\cdots \\ f_N(x_1, \cdots, x_N) = f_N(\mathbf{x}) = 0 \end{cases} \tag{A.1}$$

where the variables are defined $\mathbf{x} = [x_1, \cdots, x_N]^T$, and further defined the vector function

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_N(\mathbf{x})]^T. \tag{A.2}$$

Thus, the equation system can be expressed as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \tag{A.3}$$

Denote $\mathbf{J_f}(\mathbf{x})$ as $N \times N$ Jacobian of the vector-valued function (A.2) as follows,

$$\mathbf{J_f} = \left[\frac{\partial \mathbf{f}}{\partial x_1}, \cdots, \frac{\partial \mathbf{f}}{\partial x_N}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \cdots & \frac{\partial f_N}{\partial x_N}, \end{bmatrix} \tag{A.4}$$

or component-wise,

$$\big(\mathbf{J_f}\big)_{ij} = \frac{\partial f_i}{\partial x_j}. \tag{A.5}$$

Given the initial guess of solution $\mathbf{x}_0$, by Newton-Raphson Method, the solution $\bar{\mathbf{x}}$ of the system (A.1) can be obtained by

$$\bar{\mathbf{x}} = \lim_{n\to\infty} \mathbf{x}_n \tag{A.6}$$

if the limit exists, where $\mathbf{x}_n$, a column vector with N entries, is iterated as follows:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}_\mathbf{f}^{-1} \cdot \mathbf{f}(\mathbf{x}_n), \tag{A.7}$$

and $\mathbf{J}_\mathbf{f}^{-1}$ represents the inverse with dimension $N \times N$ of Jacobian in (A.5).

A.2 SDE in Stratonovich Sense

While the SDE system in Stratonovich sense in (3.48), the evolution of $\mathbf{X}(t)$ needs to be solved implicitly since Stratonovich integral applyed at middle point. In this dissetation, we have different schemes to deal with Stratonovich SDEs with different types of stochastic terms involve. Here we take the Forward Euler Method scheme (3.48) for example. It can be written as follows,

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \Delta t \cdot \mu(\mathbf{X}_i) + \frac{1}{2}\sum_{j=1}^{N} \mathcal{N}_j\sqrt{\Delta t} \cdot \Big(\sigma_j(\mathbf{X}_i) + \sigma_j(\mathbf{X}_{i+1})\Big), \tag{A.8}$$

where $\mathcal{N}_j$ denotes the normal random variable.

For solving $\mathbf{X}_{i+1}$ in (A.8) in the form of (A.1) with $\mathbf{0}$ on RHS, rewrite the scheme formula (A.8) as

$$\begin{aligned} \mathbf{X}_i + \Delta t \cdot \mu(\mathbf{X}_i) + \frac{1}{2}\sum_{j=1}^{N} \mathcal{N}_j\sqrt{\Delta t} \cdot \sigma_j(\mathbf{X}_i) \\ + \frac{1}{2}\sum_{j=1}^{N} \mathcal{N}_j\sqrt{\Delta t} \cdot \sigma_j(\mathbf{X}_{i+1}) - \mathbf{X}_{i+1} = \mathbf{0}. \end{aligned} \tag{A.9}$$

As $\mathbf{X}_i$ are known, the first three terms on the LHS of (A.9) is constant on the system of functions we are trying to solve implicitly. To express our problem within the setting to perform

Newton-Raphson Method, the system of functions $\mathbf{f}$ can be expressed as

$$\mathbf{f}(\mathbf{x}) = \mathbf{C} + \mathbf{G}(\mathbf{x}) = \mathbf{0}, \tag{A.10}$$

which includes the variable to be solved

$$\mathbf{x} = \mathbf{X}_{i+1}, \tag{A.11}$$

a constant term with known variables $\mathbf{X}_i$, Δt, $\mathcal{N}_j$, functions μ and σ_j:

$$\mathbf{C} = \mathbf{X}_i + \Delta t \cdot \mu(\mathbf{X}_i) + \frac{1}{2}\sum_{j=1}^{N} \mathcal{N}_j \sqrt{\Delta t} \cdot \sigma_j(\mathbf{X}_i), \tag{A.12}$$

and a vector-function of $\mathbf{x}$:

$$\mathbf{G}(\mathbf{x}) = \frac{1}{2}\sqrt{\Delta t} \cdot \sum_{j=1}^{N} \mathcal{N}_j \cdot \sigma_j(\mathbf{x}) - \mathbf{x}. \tag{A.13}$$

Thus, the $N \times N$ Jacobian matrix involves in the Newton-Raphson Method formula (A.7) can be derived as

$$\mathbf{J}_{\mathbf{f}}(\mathbf{x}) = \mathbf{J}_{\mathbf{G}}(\mathbf{x}) = \frac{1}{2}\sqrt{\Delta t} \cdot \sum_{j=1}^{N} \mathcal{N}_j \cdot \mathbf{J}_{\sigma_j}(\mathbf{x}) - \mathbf{I}, \tag{A.14}$$

where $\mathbf{J}_{\mathbf{G}}$ denotes the Jacobian matrix of vector-function (A.13), which happens to be equal to the Jacobian of $\mathbf{f}$, consists $\mathbf{J}_{\sigma_j}$ representing the Jacobian matrix of vector-function σ_j, and $\mathbf{I}$, the last term on the RHS, is the $N \times N$ identity matrix.

A.3 Note

One of the advantages of the Stratonovich SDE is that it preserves the normal Calculus chain rule using a middle point selection scheme, which is helpful in mathematical analysis. However, it inevitably causes the numerical scheme to become implicit, which depreciates the computational efficiency since we need to perform the Newton-Raphson method to solve, which involves calculating Jacobian matrices for every function in stochastic terms.

Appendix B

Recalculation on Updating Numbers of Species Due to Reaching Unrealistic Values

B.1 The Algorithm

Due to the nature of chemical systems, the number of species must remain non-negative in the simulation. However, it is indicated in (1.46) that numerical simulation for $\mathbf{X}(t)$ by tau-leaping method may make the numbers become unrealistic due to the unbounded nature of Poisson random variable, such as negative numbers [43, 54, 55, 56, 57]. Thus, it is legitimate to make further intervention on modeling.

Regarding the VG model (4.21) - (4.23), the restriction has to be stronger than nonnegativeness to cope with the diffusion terms (4.25) - (4.30). According to the second diffusion function

$$\sigma_2(\mathbf{X}) = \begin{bmatrix} \sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \\ -2\sqrt{\frac{5}{2\varepsilon}Y(Y-1)} \\ 0 \end{bmatrix}, \tag{B.1}$$

the number Y must be equal to or greater than 1 to guarantee the positiveness in the square root. On the other five diffusion functions, it induces that all of X, Y, Z must be nonnegative to perform square root reasonably. Therefore, the following condition needs to be imposed on VG model:

$$X \geq 0, Y \geq 1, Z \geq 0. \tag{B.2}$$

If the unrealistic values are disregarded and move on while performing the numerical simulation, it will be troublesome on the next update of $\mathbf{X}(t) = [X, Y, Z]^T$ and it may consist one or more complex numbers with negative real part and a small imaginary part.

Therefore, we choose not to move forward and recalculate from previous step if the updated numbers do not satisfy the condition (B.2). A condition statement in code is added to check the rationality of updated $\mathbf{X}(t)$.

Algorithm 1 Checking Updated $\mathbf{X}$(t)

```
1: procedure CHECKING RATIONALITY OF X
2: top:
3:     X_temp = X_i
4:     X_{i+1} = [updated from X_i by some algorithms]
5:     if X_{i+1} <= [0,1,0]^T then
6:         X_i = X_temp
7:         goto top.
```

B.2 Remark

Some numerical methods are also developed to avoid the possibility to produce negative populations, such as binomial tau-leaping method in [58, 59].