MULTIPLE TIME SCALE ALGORITHMS FOR GENE NETWORK SIMULATIONS

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

VASSILIOS SOTIROPOULOS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

YIANNIS N. KAZNESSIS, ADVISOR

OCTOBER 2009

Acknowledgements

This dissertation would not have been possible if it were not for the support of many dear people. First, I would like to thank my family, my father Jannis and my brother Thanassis, for the continues support and inspiration throughout the length of my PhD studies. Especially, I would like to extend my deep gratitude to my father, who has devoted many years of his life to ensure that I would embrace the proper principles and knowledge, without which I would not have been a part of the great Minnesota tradition.

My advisor, Yiannis N. Kaznessis has been a mentor and a impeccable teacher. His efforts, support and guidance were the spark for this dissertation. He has been an advisor both in academic matters but also during the ups and downs of my PhD program and graduate life in general. For that I am deeply grateful to him.

I would also like to thank my committee members, Professor Frank S. Bates, Professor Aditya Bhan, Professor Prodromos Daoutidis and Professor Georgios B. Giannakis for their valuable time and advice. I would especially like to thank Professor Prodromos Daoutidis who over the years has been more like a second advisor, spending time to guide and help me.

I am in debt to my friends and colleagues, who made my time in Minnesota a unique and enjoyable experience. I wish to thank my former and current lab members, Howard, Jon, Tony, Allison, Spyros, Abdallah, Kat, John, Dan, Anushree, Emma, Kostas and Ben, for their timely help and support and for providing a fun working environment. I am thankful for being surrounded by many loyal friends, Dimitris (×3), Tassos, Maria, Hlias, George, Spyros (×2), Vassia, Vassilis (×2), Vaggelis, Evgenia, Yiannis (×3), Kostas, Sofia, Nikos (×2), Mina, Triantafyllos and all those I am forgetting at this moment, and glad that they were by my side through the good and bad times of my

PhD.

Last but not least, I would like to acknowledge financial support from the National Institute of Health (NIH), the Nationals Science Foundation (NSF) and the University of Minnesota Doctoral Dissertation Fellowship. Computational support from the University of Minnesota Digital Technology Center (DTC), Minnesota Supercomputing Institute (MSI) and the NSF Teragrid Supercomputers is also gratefully acknowledged.

This dissertation is dedicated to my father, Jannis, who lovingly supported me over the years.

Abstract

The objective of this dissertation is the development and implementation of multiple time scale stochastic models necessary for analysis, design and construction of novel synthetic biological systems, such as gene networks.

At the dawn of the 21$^{\text{st}}$ century, scientists and engineers turned into engineering new biological systems. Synthetic biology emerged as a distinct discipline, combining biology and engineering towards the design and construction of new biological parts, devices and systems with useful applications. This ambitious endeavor would not have been possible, were it not for the recent, impressive discoveries in biology and the equally remarkable advances in biotechnology. Indeed, we can now literally ”cut” and ”paste” DNA at will.

The impact in everyday life may be significant, with wide-ranging applications: from medicine, where gene regulatory networks can be used for gene therapy applications, to the production of biopolymers, to the removal of environmental pollutants, and to clean energy alternatives.

Even though wet lab experiments have provided ample proof of concept, the challenge facing the scientific and engineering communities is how to rationally design novel biological systems. An answer lies with mathematical models and sophisticated algorithms. It is the same philosophy used to design many of the modern marvels of technology, such as airplanes. Analogously, sophisticated computer-aided design (CAD) algorithms, alongside with a minimal number of experiments, could be the standard in constructing novel biological systems, devices, even entire organisms, alleviating the need for expensive trial and error approaches.

There are primarily three types of challenges in developing new CAD tools for synthetic biological systems, such as gene networks. First, the number of molecular components in biological systems is overwhelming. Second, all living microorganisms are impacted by thermal noise and on occasion behave randomly. Third, the time scales at which many of the biological phenomena occur can differ by many orders of magnitude, resulting in stiff mathematical descriptions.

The aim of this thesis is the CAD of synthetic gene networks addressing these challenges. For that to be accomplished an important step is the development of multiple time scale methods for the efficient and accurate integration of stiff chemical Langevin equations. These describe the dynamics of many biological systems. Methods developed also include the description of noise through classical mathematical descriptions instead of the more demanding stochastic formulation. Algorithms developed as part of this dissertation are incorporated into CAD software tools built by our group. In the last part of this dissertation we discuss how such tools are employed for CAD of novel synthetic gene networks.

Contents

List of Tables **x**

List of Figures **xi**

1 Introduction **1**

1.1 Dissertation Outline 2

2 Background Information on Synthetic Biology **7**

2.1 Synthetic Biology and Applications 7

2.2 Synthetic Biology Examples 9

2.2.1 Experimental Perspective 9

2.2.2 Computational Approach 12

2.3 Computer-Aided Design of Synthetic Constructs 13

2.4 Mechanistic Models 15

2.4.1 Gene Expression and Regulation 15

2.4.2 Chemical Kinetics Models 18

2.5 ”It’s a Noisy Business!”: Importance of Intrinsic Noise 21

3 Background Information on Computational Models and Algorithms **23**

3.1 Stochastic Chemical Kinetics Models 23

3.1.1 Markov Processes and Chemical Kinetics 24

3.1.2 Stochastic Simulation Algorithm 26

3.1.3 Progress in Stochastic Algorithms for Chemical Kinetics 27

3.2 A Hybrid Stochastic Algorithm 30

3.2.1 System Partitioning 30

3.2.2 Propagation of Fast Subsystem - Chemical Langevin Equation . 31
3.2.3 Propagation of Slow Subsystem - Jump Equations 31
3.3 Multiple Time Scales in Gene Expression 32

4 An Adaptive Time Step Scheme for Systems of SDEs with Multiple Multiplicative Noise: Chemical Langevin Equation, A Proof of Concept 35

4.1 Introduction 35
4.2 Theory 39
4.2.1 Strong and Weak Convergence 40
4.2.2 Milstein Method 41
4.2.3 Adaptive Time Stepping: Brownian Trees 43
4.2.4 Error Criteria 47
4.2.5 Chemical Langevin Equations 48
4.2.6 Implementation Details 50
4.3 Examples 52
4.3.1 A Reversible Dimerization Reaction 52
4.3.2 A System of Stiff, Nonlinear CLEs 60
4.3.3 An Actual Stiff Biochemical Network 66
4.4 Summary 73

5 Model Reduction of Multiscale Chemical Langevin Equations: A Numerical Case Study 75

5.1 Introduction 75
5.2 Continuous Markov Processes: Chemical Langevin Equations 77
5.3 Motivating Example 78
5.4 Problem Formulation 81
5.5 Model Reduction Framework 82
5.5.1 Identifying Fast and Slow Variables 82
5.5.2 Decoupling Fast and Slow Variables 84
5.5.3 Approximated Probability Distribution 85
5.6 Examples 86
5.6.1 Example Revisited 86

5.6.2 Second Example 94

5.6.3 Sugar Cataract Development Model 101

5.7 Summary 106

6 Analytical Derivation of Moment Equations in Stochastic Chemical Kinetics 108

6.1 Introduction 108

6.2 Theory 110

6.2.1 Markov Processes 110

6.2.2 Moments and Jump Moments 111

6.2.3 Derivation of Moment Equations 112

6.2.4 Jump Moments and Stochastic Chemical Kinetics 117

6.3 Examples 120

6.3.1 Linear Kinetics 120

6.3.2 Non-Linear Kinetics 123

6.4 Discussion 125

6.4.1 Infinite Dimensional Moment Equations 125

6.4.2 Moment Closure Schemes 125

6.4.3 Importance of Higher Moments in Non-linear Chemical Kinetics 127

6.5 Summary 130

7 Software Tools for Computer-Aided Design of Synthetic Gene Networks 132

7.1 Guided User Interfaces for Gene Network Engineering 132

7.2 Hybrid Stochastic Simulation for Supercomputers 133

7.3 Synthetic Biology Software Suite 134

7.3.1 Synthetic Biology Software Suite: Desktop Simulator 134

7.3.2 Synthetic Biology Software Suite: Wiki 135

7.3.3 Synthetic Biology Software Suite: Designer 135

8 Synthetic Tetracycline-Inducible Regulatory Networks: Computer Aided Design of Dynamic Phenotypes 137

8.1 Introduction 137

8.2 Four Novel Networks Based on the Tetracycline Resistance Operon of *E. coli* 140
8.3 Methods 143
8.3.1 Chemical Kinetics Models 143
8.3.2 Model Assumptions and Initial Conditions 153
8.3.3 Computer-aided Design Software Tools 154
8.4 Network I 155
8.4.1 Dynamical Behavior Based on Wild Type Kinetics 155
8.4.2 Fine Tuning Using Mutated TetR and TetO Variants 156
8.5 Network II 158
8.5.1 Dynamical Behavior Based on Wild Type Kinetics 158
8.5.2 Fine Tuning Using Mutated TetR and TetO Variants 159
8.6 Network III 162
8.6.1 Dynamical Behavior Based on Wild Type Kinetics 162
8.6.2 Fine Tuning Using Mutated TetR and TetO Variants 163
8.7 Network IV 165
8.7.1 Dynamical Behavior Based on Wild Type Kinetics 165
8.7.2 Fine Tuning Using Mutated TetR and TetO Variants 166
8.8 Summary 168

9 Concluding Remarks 170
9.1 Summary of Contributions 170
9.2 Future Research Directions 173

References 176

Appendix A. Results for Chapter 6 191
A.1 Derivation of Third Order Moment Equations 191
A.2 Derivation of Fourth Order Moment Equations 193

List of Tables

2.1 A General Chemical Kinetics Representation of Gene Expression & Regulation. 19

4.1 Reactions and Parameters for a Reversible Dimerization Reaction 53

4.2 Reactions and Parameters for a Stiff, Nonlinear Network of Reactions . 61

4.3 Reactions that Quantify the Heat Shock Response of *E. coli.* 67

4.4 Initial Values of Species Involved in the Heat Shock Model of *E. coli.* . . 69

5.1 Reactions and Parameters for the Motivating Example 79

5.2 Mean and Standard Deviation Errors for the Slow Dynamics of the Motivating Example 92

5.3 Reactions and Parameters for the Second Example 95

5.4 Mean and Standard Deviation Errors for the Slow Dynamics of the Second Example 100

5.5 Reactions and Parameters for the Sugar Cataract Development Model . 103

5.6 Mean and Standard Deviation Errors for the Slow Dynamics of the Sugar Cataract Development Model 105

6.1 Reactions and Parameters for the Schlögl Model 127

6.2 First Eight Moment Values for the Schlögl Model 128

8.1 A Chemical Kinetics Representation of Network III 145

List of Figures

2.1 Synthetic biology milestones. 9

2.2 Response of a synthetic bistable switch. 10

2.3 Oscillating fluorescence in an *E. coli* cell. 11

2.4 Multiple time scales in biology 14

2.5 Schematic representation of key biological interactions. 17

2.6 A synthetic oscillating gene circuit where *E. coli* cells flash green and red and everything in between. 21

3.1 Appropriate Markov process regimes depending on the number of molecules and reaction propensities. 33

4.1 A schematic representation of a Brownian tree. 44

4.2 Two distinct representations of a Brownian tree. 46

4.3 Solution of the dimerization reaction system, depicted in Table 4.1, using SSA in the interval [0, 0.01] s. 54

4.4 Solution of Eq. (4.25) using the fixed step Milstein method ($p = 10$). . . 55

4.5 Solution of Eq. (4.25) using the adaptive scheme with variable SDE tolerance (SDE coeff = 1000, p=10 and $\Delta t_{\text{ini}} = 10^{-4}$ s). 57

4.6 Solution of Eq. (4.25) using the adaptive scheme with variable p values for constant initial time step, $\Delta t_{\text{ini}} = 10^{-4}$ s 58

4.7 Comparison of execution times between the adaptive and fixed step methods for a single trial of the dimerization reaction system depicted in Table 4.1. 59

4.8 Solution of Eq. (4.26) using SSA in the interval [0, 0.01] s. 62

4.9 Solution of Eq. (4.26) using fixed step Milstein method ($p = 10$). 63

4.10 Solution of Eq. (4.26)) using the adaptive scheme with variable SDE tolerance (SDE coeff = 1,000, $p = 10$ and $\Delta t_{\text{ini}} = 10^{-4}$ s). 64

4.11 Comparison of execution times between the adaptive and fixed step methods for a single trial of the stiff, nonlinear reaction system depicted in Table 4.2. 65

4.12 Solution of the heat shock response model using SSA in the interval [0, 100] s. 70

4.13 Solution of the heat shock response model using the fixed step Milstein method ($p = 10$). 71

4.14 Solution of the heat shock response model using the adaptive scheme with variable SDE tolerance (SDE coeff = 1,000, $p = 10$ and $\Delta t_{\text{ini}} = 0.1$ s). . 72

4.15 Comparison of execution times between the adaptive and fixed step methods for a single trial of the heat shock response model depicted in Table 4.3. 73

5.1 A simulated trajectory of system (5.5) using the fixed step Euler-Maruyama method with time step 5×10^{-5} s. 80

5.2 The pseudo-stationary distribution that the fast variable, Y_5, relaxes into obtained as the solution of Eq. (5.23). 88

5.3 Probability distribution of the fast variable $Y_5 = X_A$ at two different time points, $t = 0.01$ s and $t = 180$ s, using 2,000 independent trials of the original set of CLEs (cf. Eq. (5.5)). 88

5.4 Probability distribution of the slow species, Y_1, at different time points. (a) Using the reduced model. (b) Using the full model. 90

5.5 Joint probability distribution $t = 180$ s. (a) Using the reduced model. (b) Using the full model. 93

5.6 The pseudo-stationary distribution that the fast variables, Y_4 and Y_5, relax into obtained as the solution of Eq. (5.15). 97

5.7 Probability distribution of the slow species, Y_1, at different time points. 98

5.8 The pseudo-stationary distribution that the fast variables, X_{SDH} and X_{NADH}, relax into obtained as the solution of Eq. (5.15) as it applies to the SCD model. 104

6.1 Actual and reconstructed probability distributions for the Schlögl model, depicted in Table 6.1, using the moment data in Table 6.2. 129

7.1 Screenshot of the main GUI window of Hy3S. 133
7.2 Screenshot of the main GUI window of SynBioSS Desktop Simulator. . . 135
8.1 A schematic representation of four novel regulatory gene networks . . . 141
8.2 A schematic representation of the network connectivities 142
8.3 Dynamical behavior of Network I 157
8.4 Dynamical behavior of Network II 160
8.5 Dynamical behavior of Network III 164
8.6 Dynamical behavior of Network IV 167

Chapter 1

Introduction

During the 20$^{\text{th}}$ century scientists unveiled many of the secrets of the biological world by studying the cell and its functions, identifying key molecules and molecular structures within cells, and most importantly discovering DNA. The pace of discoveries appears ever increasing in biology. Powerful technologies have been developed that probe biological systems, from molecules to cells to tissues to organisms. With all this gained knowledge, at the dawn of the 21$^{\text{st}}$ century, scientists and engineers turned into engineering new biological systems. Synthetic biology emerged as a distinct discipline combining biology and engineering. With synthetic biology scientists set their goals higher, than previous genetic engineering goals, constructing biological systems, devices, even entire organisms.

The promise of synthetic biology is the design and construction of biological factories, cells that will perform specific tasks or generate products similar to modern factories. The idea is analogous to programming robots to carry out certain tasks, but instead of dealing with electrical circuits it would require reprogramming the cell's DNA; introducing new or modifying existing gene circuits (i.e. DNA segments). The objective is not just to utilize existing cell functions to our benefit, such as the biodegradation of treated wastes for methane production, but also create new ones, which cells normally would not perform. The realization of such an undertaking will impact significantly everyday life, with wide-ranging applications: from medicine, where gene regulatory networks can be used for gene therapy purposes, to the production of biopolymers, to the removal of environmental pollutants, and to clean energy alternatives where microbes can be engineered to produce hydrogen from sunlight and water. A number of synthetic gene constructs has been proposed and implemented *in vivo*. Notable examples are bistable switches, oscillators, and logic gates. Several of these synthetic constructs have been applied towards useful applications.

Even though wet lab experiments have provided ample proof of concept, the challenge facing the scientific and engineering communities is how to rationally design novel biological systems. The picture of an airplane comes to mind. The goal of an engineer is to build the airplane from very many interacting components in a rational way, that is without many rounds of expensive trial and error. A solution lies with mathematical models. Mathematical models assist engineers to build marvelously complex entities, from huge refineries to aircrafts and from supercomputers to tiny phones that can surf the web.

In our work we set out to develop models of biological systems that can assist us in rationally designing synthetic ones. There are chiefly three types of challenges: first, the number of molecular components in biological systems (DNA sequences, proteins, small organic molecules, etc.) is large. Second, all living microorganisms behave largely in what can be summed up as a random behavior, which results in crucial parameters, like the size of the cell, to vary considerably in a population. Third, the time scales at which many of the biological phenomena occur can differ by many orders of magnitude, resulting in stiff mathematical descriptions.

The need arises then for sophisticated models and algorithms that run on supercomputers. With the help of supercomputers we can run our models to quickly shift through very many alternative synthetic designs. In the aircraft design example, engineers use nowadays sophisticated simulation models to design components such as wings and the fuselage of an aircraft. We are doing the same with synthetic biological systems.

1.1 Dissertation Outline

The work in this dissertation builds on existing mathematical models and algorithms developed in the Kaznessis group. It improves the state of the art algorithms for efficient and accurate integration of multiple time scale models for gene networks. The doctoral dissertation has produced five peer-revied journal publications [1, 2, 3, 4, 5]. There are more currently in preparation.

A brief overview of the key points presented and discussed in the current dissertation follows.

- Chapter 2 introduces synthetic biology and discusses the recently emerged multidisciplinary field while also highlighting its extremely promising applications. A number of wet lab and simulation experiments are briefly described in order to demonstrate the initial successes of the field. Next the methodology to create detailed mechanistic models of gene expression is presented. Chemical kinetics models promise to become the foundation for a rational computer-aided design approach for gene networks, similar to how we design many of the modern marvels of technology. A short review on gene expression and regulation is presented since a sound understanding of the central dogma of molecular biology is imperative for any mechanistic model to accurately capture all interactions at the molecular level. The effects of noise on the observed biological behavior is discussed through pioneering work on single cell experiments.

- Chapter 3 discusses the implications of intrinsic noise effects in the modeling approaches of gene regulatory networks. This is the last introductory chapter as it communicates the recent advances in the field of stochastic chemical kinetics models. A brief literature review highlights much of the state of the art algorithms. Among them, a hybrid stochastic algorithm devised in the Kaznessis group is presented in more depth as in many aspects it is the starting point for much of the work in this dissertation. We discuss how the existence of different time scales in a range of biological functions results in stiff mathematical descriptions, i.e. numerical integration methods become numerically unstable, for many chemical kinetics models. This observation generated the spark for our effort and led to the work presented in the following two chapters.

- Chapter 4 presents an integration method for a subclass of stiff stochastic differential equations (SDEs). Models involving SDEs play a prominent role in a wide range of applications where systems are not at the thermodynamic limit, for example biological population dynamics. Therefore there is a need for numerical schemes that are capable of accurately and efficiently integrating systems of SDEs.

In this work we introduce a variable size step algorithm and apply it to systems of stiff SDEs with multiple multiplicative noise. The algorithm is validated using a subclass of SDEs called chemical Langevin equations that appear in the description of dilute chemical kinetics models, with important applications mainly in biology. Three representative examples are used to test and report on the behavior of the proposed scheme. We demonstrate the advantages and disadvantages over fixed time step integration schemes of the proposed method, showing that the adaptive time step method is considerably more stable than fixed step methods with no excessive additional computational overhead.

- Chapter 5 considers an alternative approach for the problem discussed in Chapter 4. The scheme of the previous chapter is more general while the method in hand is restricted on the dynamics of reaction sets governed by stiff chemical Langevin equations, i.e. stiff stochastic differential equations. These are particularly challenging systems to model, requiring prohibitively small integration step sizes. We describe and illustrate the application of a semi-analytical reduction framework for chemical Langevin equations that results in significant gains in computational cost. We illustrate this through a number of different instructive examples. The framework, similarly to the previous chapter, handles successfully the two very important characteristics of biological reaction networks, namely it applies to models that account for the inherent probabilistic nature of systems far from the thermodynamic limit and takes into consideration the disparate spectrum of time scales observed in biological phenomena, such as slow transcription events and fast dimerization reactions.

- In Chapter 6 we take a few steps back and go to the root of the modeling methodology. We start from the defining equation of any stochastic chemical kinetics model, the master probability equation. We note that the master probability equation is very general and chemical kinetic models represent a subclass for which it applies. The equation helps us capture and understand the dynamic behavior of a variety of stochastic phenomena that can be modeled as Markov processes. Analytical solutions to the master equation are hard to come by though because they require the enumeration of all possible states and the determination of the transition probabilities between any two states. These two tasks quickly become intractable for all but the simplest of systems. The master probability distribution can be expressed as a function of its moments. Instead of determining how the probability distribution changes in time, we can then write transient equations for the probability distribution moments. In this work we present a general scheme for deriving analytical moment equations for any N-dimensional Markov process as a function of the jump moments. Jump moments are measures of the rate of change in the probability distribution moment values, i.e. what is the impact of any given transition between states on the moment values. Additionally, we propose a scheme to derive analytical expressions for the jump moments for any N-dimensional Markov process. We then focus on stochastic chemical kinetics models for which we derive analytical relations for jump moments of arbitrary order. The elements in the jump moment expressions are a function of the stoichiometric matrix and the reaction propensities, i.e the probabilistic reaction rates. Then we use two toy examples, a linear and a non-linear set of reactions, to demonstrate the applicability and limitations of the scheme. Finally we estimate the minimum number of moments necessary to obtain statistically significant data that would uniquely determine the dynamics of the underlying stochastic chemical kinetic system. Contrary to broad belief, the first two moments only provide limited information, especially when complex, non-linear dynamics are involved.

- In Chapter 7 two software tools, Hybrid Stochastic Simulation for Supercomputers (Hy3S) and Synthetic Biology Software Suite (SynBioSS), developed by the Kaznessis group are showcased. Both are the epitome of the group's research efforts. These tools embrace the key aspiration and motivation of the present dissertation and implement the developed algorithms for use in computer-aided design of synthetic gene construct. Our objective was and still is, for algorithms to be publicly available and most importantly in a user friendly form, where even those with limited programming knowledge can put them to good use.

- Chapter 8 present a computer-aided design example. In this chapter we use Hy3S and SynBioSS in order to design novel synthetic tetracycline-inducible regulatory gene networks. In recent years tightly regulated gene networks, precisely controlling the expression of protein molecules, have received considerable interest by the biomedical community due to their promising applications. Among the most well studied inducible transcription systems are the tetracycline regulatory expression systems based on the tetracycline resistance operon of *Escherichia coli*, Tet-Off (tTA) and Tet-On (rtTA). Despite their initial success and improved designs, limitations still persist, such as low inducer sensitivity. Instead of looking at these networks statically, and simply changing or mutating the promoter and operator regions with trial and error, a systematic investigation of the dynamic behavior of the network can result in rational design of regulatory gene expression systems. With computer-aided design, we aim to improve the synthesis of regulatory networks and propose new designs that enable tighter control of expression. In this work we engineer novel networks by recombining existing genes or part of genes. We synthesize four novel regulatory networks based on the Tet-Off and Tet-On systems. We model all the known individual biomolecular interactions involved in transcription, translation, regulation and induction. Important biomolecular interactions are identified and the strength of the interactions engineered to satisfy design criteria. A set of clear design rules is developed and appropriate mutants of regulatory proteins and operator sites are proposed. We propose, test and accept or reject design principles for each network.

- Chapter 9 conveys a final discussion and synopsis of the key results in the present thesis and discuses interesting avenues of future research.

Chapter 2

Background Information on Synthetic Biology

2.1 Synthetic Biology and Applications

In the last half of the 20$^{\text{th}}$ century our perspective and understanding for the biological world has grown significantly. The first big milestone was the discovery of the DNA and its structure [6]. Since then many key molecules and molecular structures within cells have been identified along with their key functionalities. Moreover, the identification and characterization of complex pathways and interactions have drastically contributed in understanding how information is propagated within any living cell, how cells utilize the available energy sources, interact with their environment, grow and populate among many other cell functions [7]. At the same time technological advances have made feasible and affordable DNA sequencing, i.e. the decryption of the genetic information encoded within DNA molecules, for many organisms [7]. Projects such as the sequencing of *Escherichia coli* (*E. coli*), *yeast* and human genomes, led to the characterization of multiple genes in miniscule amount of times, unveiling even more secrets of the biological world.

The closer we study cells, the more we realize that they resemble modern factories. Instead of workers, molecules are the workforce, responsible for communication, protein production, biodegration of nutrients through complicated enzymatic pathways, that resemble assemble lines. Furthermore certain molecules function like administrative personnel by controlling or supervising entire cell functions. One can claim that DNA corresponds to the organization chart and business plan where all orders stem from. Moreover the remarkable organization present inside cells resembles the organization inside factories where every department functions on its own but also would not exist independently. Cells are sophisticated, well organized and optimized miniature factories.

These findings are exciting, because they provide a bridge between the physical world of molecules to the biological world of complex behaviors, or phenotypes as biologists call them. What is perhaps more exciting is that we now have the ability to construct such bridges synthetically, that did not exist in the natural world. *Synthetic biology* is what we call the efforts that stem from this ability. A combination of biology and engineering that promises the design and construction of biological factories, cells that will perform specific tasks or generate products similar to modern factories. The idea is analogus to programming robots to carry out certain tasks, but instead of dealing with electrical circuits it would require reprogramming the cell's DNA; introducing new or modifying existing gene circuits (i.e. DNA segments).

In other words, Synthetic biology aims at the design and construction of new biological parts, devices and systems from natural biological systems and the redesign of existing motifs towards useful applications [8, 9]. The objective is similar to the constant strive of chemical engineers to improve chemical processes in modern chemical plants. An important note, is that in contrary to genetic engineering where usually one to two genes are mutated to meet a design objective, synthetic biology proposes the complete redesign of a cell process or a cell module to optimize the phenotypic behavior at hand. Factiously we can say that synthetic biology is genetic engineering on steroids. Synthetic biology is possible now because of recent advances in biotechnolgies, such as DNA synthesis, cloning polymerase chain reaction (PCR) and green fluorescent protein (GFP) monitoring [7].

The realization of such an undertaking will have significant impact in everyday life, with applications in a range of disciplines: from medicine, where gene regulatory networks can be used for gene therapy purposes, to the production of biopolymers, to the removal of environmental pollutants [10, 11, 12], and to clean energy alternatives where microbes can be engineered to produce hydrogen from sunlight and water [13].

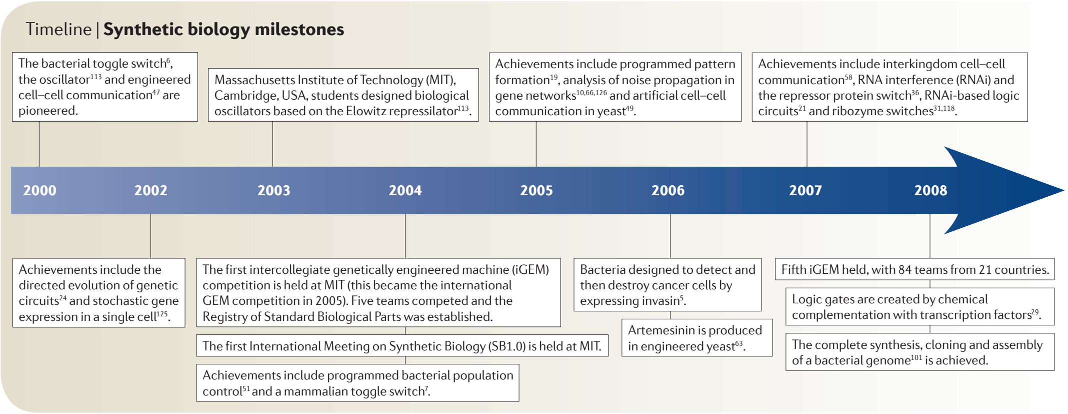

Timeline | **Synthetic biology milestones**
The bacterial toggle switch[6], the oscillator[113] and engineered cell–cell communication[47] are pioneered.
Massachusetts Institute of Technology (MIT), Cambridge, USA, students designed biological oscillators based on the Elowitz repressilator[113].
Achievements include programmed pattern formation[19], analysis of noise propagation in gene networks[10,66,126] and artificial cell–cell communication in yeast[49].
Achievements include interkingdom cell–cell communication[58], RNA interference (RNAi) and the repressor protein switch[36], RNAi-based logic circuits[21] and ribozyme switches[31,118].
2000
2002
2003
2004
2005
2006
2007
2008
Achievements include the directed evolution of genetic circuits[24] and stochastic gene expression in a single cell[125].
The first intercollegiate genetically engineered machine (iGEM) competition is held at MIT (this became the international GEM competition in 2005). Five teams competed and the Registry of Standard Biological Parts was established.
The first International Meeting on Synthetic Biology (SB1.0) is held at MIT.
Achievements include programmed bacterial population control[51] and a mammalian toggle switch[7].
Bacteria designed to detect and then destroy cancer cells by expressing invasin[5].
Artemesinin is produced in engineered yeast[63].
Fifth iGEM held, with 84 teams from 21 countries.
Logic gates are created by chemical complementation with transcription factors[29].
The complete synthesis, cloning and assembly of a bacterial genome[101] is achieved.

Figure 2.1: Synthetic biology milestones. Figure adapted from Ref. [12].

One of the most promising and enticing application of synthetic biology is the design of artificially regulatory gene networks for gene therapy applications. As such, they are an important part of this dissertation (cf. Chapter 8). Artificially regulatory gene networks consist of well-characterized natural gene components that control the expression of a reporter protein at will. Desired phenotypes range from simple on-off systems to oscillating behavior. Such networks may significantly impact our ability for targeted drug delivery. A well studied example of synthetic regulatory gene network is the Tet-Off system, based on the tetracycline resistance operon of *E. coli* [14].

2.2 Synthetic Biology Examples

In its early years, synthetic biology mainly concentrated in strengthening and supporting the rationality of its underlying logic. Since then, the field has grown significantly managing to produce noteworthy results that promise to positively impact everyday life [12]. A brief timeline chart depicting the milestones of synthetic biology is presented in Fig. 2.1.

2.2.1 Experimental Perspective

Early wet lab experiments focused on simple examples that addressed design challenges and developed design principles for simple modules, networks of two to three genes,

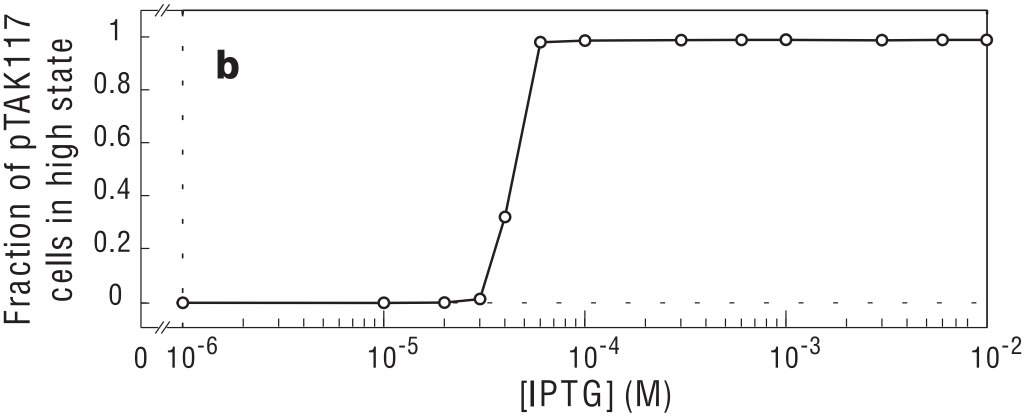


Figure 2.2: Response of a synthetic bistable switch. The toggle behavior is evident as cells move from the ”off” state, i.e. low protein levels, to the ”on” state, i.e. high protein levels. The fraction of cells that are in the ”on” state increases sharply after a threshold inducer (IPTG) concentration is reached. Figure adapted from Ref. [15].

with predefined phenotype. Many of the design objectives have been borrowed from the fields of electrical and computer engineering, where similar to electrical circuits, specific cell molecules were designed to perform oscillations, exhibit on-off behavior, function as filters, produce time delays, compare signal values and even display logic gate behavior [12].

The bistable switch of Gardner and coworkers is the first module successfully designed *in vivo* [15]. The system consists of a two gene constructs where the protein expressed from gene 1 represses the expression of gene 2. In turn, the product of gene 2, protein 1, represses the expression of gene 1. At the same time, both repressor proteins can be induced by small chemical compounds that alleviate their repressing capabilities (cf. Fig. 2.2). This simple genetic construct, which does not occur naturally, when inserted into *E. coli* causes cells to exhibits on-off behavior depending on the relative inducer concentrations. Likewise, synthetic switches that facilitate autoregulation, negative feedback loop [16] or positive feedback loop [17] have been implemented experimentally.

The next successful example, and among the most studied designs, is the repressilator of Elowitz and Leibler [18]. In this module, a network of three genes was built, where each gene produces a repressor that represses the expression of one of the other genes in

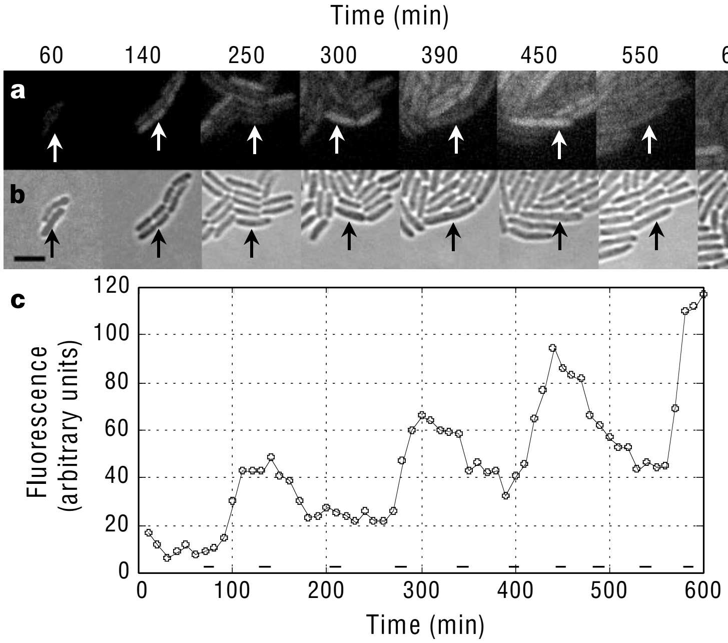


Figure 2.3: Oscillating fluorescence in an *E. coli* cell. Figure adapted from Ref. [18]

a sequential manner, i.e. forming a "vicious" cycle. This gene network results in oscillating behavior when inserted into *E. coli* cells. Oscillations are observed through a single fluorescent protein that is related to one of the three repressor proteins (cf. Fig. 2.3).

In 2003, Atkinson and coworkers used a two gene network to construct a system with dual phenotypes [19]. The system exhibits both toggle switch and oscillatory behavior depending on induction.

Another noteworthy attempt is the implementation of logic gate behavior within bacteria. Recently Cox and Elowitz presented multiple single-promoter motifs that exhibit different logical gate behavior [20]. The alternative promoter design are the result of a powerful combinatorial technique. In a similar approach, in our group, Ramalingam and coworkers obtained AND logic gate behavior starting from a different approach [21]. By combining detailed mechanistic-kinetic models and *in vivo* experiments they studied the potential of a synthetic, single promoter AND gate. The single promoter's topology and weak repression effects are considered and their impact on the AND gate functionality is quantified. Efficient implementation of cascades within cells, composed of different logic gate motifs maybe the forefront of biocomputaion.

Similarly, Guet and coworkers used combinatorial techniques to create and screen a vast variety of regulatory motifs based on the building blocks of three genes, encoding the well known transcriptional regulators LacI, TetR and lambda cI [22]. The possible permutations are increased through the use of five different promoters, regulated by the three proteins. Cells can also be programmed to perform certain tasks during their life cycle by coupling genes switches with cell signaling pathways [23].

Finally, an important part of the synthetic biology community is the international genetically engineered machine (iGEM) competition that is conducted each year since 2004. Undergraduates with the guidance of faculty are given the opportunity to put their imagination to work creating synthetic biological constructs with unique functionalities, such as devices to detect heavy metal markers [24] or a trifold bistable switch [25]. In 2008, the Minnesota iGEM team constructed a comparator *in vivo*, that allows cells to compare the strength of two different external signals and act accordingly.

2.2.2 Computational Approach

Computational tools have been used to assist scientists to understand and explain phenotypic behavior observed in a series of experiments. For instance, the bistable switch and the repressilator examples discussed in Sec. 2.2.1 used simplified lumped models to explain dynamical behavior while also served as tools to characterize responses [18, 15].

More recently mechanistic models combined with sophisticated algorithms have been developed in order to predict dynamical behavior of biological systems and rationally guide engineering of new synthetic biological constructs. The first notable approach is that of Arkin and McAdams who used chemical kinetics models to predict how *E. coli* cells behave when they are infected by the lambda-phage virus [26]. They found that the health status of the infected cell determines which pathway the virus will follow, lysis or lysogeny. Similarly, Wolf and Arkin modeled the fim switch in *E. coli*, which controls the piliation of *E. coli* while it grows inside an animals intestine and determines its strategy against the host immune response [27]. Vilar and coworkers modeled the dynamical behavior of the well known lac operon [28].

Even though the previously mentioned examples are not directly connected to synthetic biology, since they study the behavior of naturally occurring systems rather than artificial ones, they served as modeling prototypes for synthetic biology. Sophisticated algorithms can be used to improve existing designs, predict new ones and handle systems with an ever increasing complexity. Computational work focused on better understanding and improving the design of the first successful wet lab experiments. Salis and Kaznessis modeled the dynamical behavior of a bistable switch based on the Gardner, Cantor and Collins design investigating also parameters that result in fine tuned systems, such as the affinity and placement of the operators [29]. Furthermore, based on the design of the reprissilator, Tuttle and coworkers designed a three gene synthetic network that sustains oscillations under certain conditions, while also exploring parameters that influence the periodicity of the system [30]. Lately, a bottom-up approach for designing biological systems has been tested, where a stochastic model with fitted parameters is used to predict the dynamical behavior of novel networks [31]. Ramalingam and coworkers employed mechanistic models to first design AND logic gates constructs *in silico* and then have them tested *in vivo*. A combination of experimental and computational work is also that of Cox and coworkers, which proposes mathematical formulae to predict promoter activity as a function of inducer concentration [20]. Similarly, a computational approach that combines thermodynamical reasoning has also been used to predict and explain designs with boolean logic [32].

2.3 Computer-Aided Design of Synthetic Constructs

As the design objectives of synthetic biology become even more ambitious, the different design alternatives can multiply exponentially. At the same time the boom of biological knowledge has increased the number of available components, enriching the available toolboxes with thousands new building blocks, i.e. DNA sequences such as proteins, promoters, operators and ribosome binding sites, every year. Technology is at a point where we can literarily "cut" and "paste" DNA sequences, creating our own modified organism. It then becomes evident, that while experiments have helped the field of synthetic biology to mature, they will become intractable and expensive if they were required to explore the vast space of all possible combinations.

Mathematical models, on the other hand, can help reduce the number of different combinations, so that experiments can focus on only the most promising alternatives.

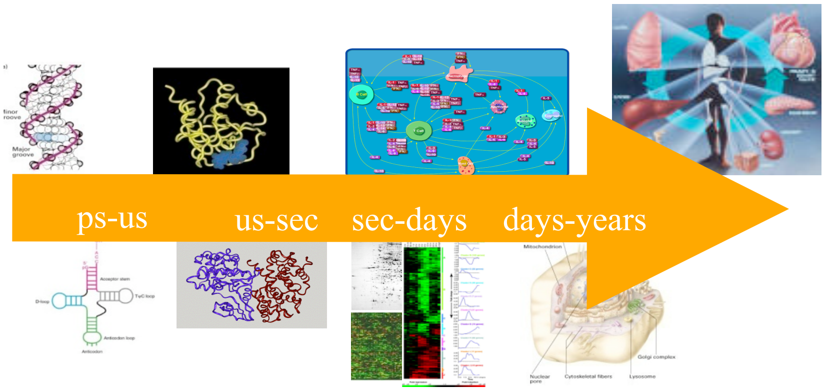


Figure 2.4: Multiple time scales in biology

Instead of a trial and error experimental approach computer-aided design (CAD), based on sophisticated algorithms, becomes a reasonable choice to quickly shift through different designs. The picture of an airplane comes to mind. The engineers goal is to build the airplane from very many interacting components in a rational way, that is without many rounds of expensive trial and error. The solution lies with mathematical models. They are routinely used in aircraft design. In similar fashion sophisticated CAD algorithms alongside with a minimal number of experiments should be the standard in designing novel biological organisms or modules within them.

The work in this dissertation sets out to develop new CAD models for synthetic biology. There are chiefly three types of challenges when modeling biological systems. First, the number of molecular components in biological systems (DNA sequences, proteins, small organic molecules, etc.) is overwhelming. Second, all living microorganisms are impacted by thermal noise (intrinsic noise) and on occasion behave randomly. Third, the time scales at which many of the biological phenomena occur can differ by many orders of magnitude (cf. Fig. 2.4), resulting in stiff mathematical descriptions.

With these difficulties in mind, in the rest of this chapter we describe first how detailed mechanist-kinetic models for gene networks are created. Second we document the importance of intrinsic noise effects in a cell's phenotype and comment on the impact it has on the modeling approach. In Sec. 3.3 we elucidate on the effect that different time scales have on the mathematical description of chemical kinetics models for gene networks.

2.4 Mechanistic Models

A convenient way to represent interactions at the molecular level, such as transcription, translation, induction stages and protein-protein interactions among others, is through chemical kinetics models. Indeed, cell functions can be depicted as cascades of chemical reactions, where the reacting molecules are promoters, operators, proteins and so forth while generation, degradation, binding and conformational changes are the reacting outcomes.

In order to convert cellular functions into simple mechanistic chemical reacting systems it is necessary to understand the central dogma of molecular biology [33]. Translating gene networks interactions into chemical reactions is the key element behind the detailed mechanistic models we pursued.

2.4.1 Gene Expression and Regulation

We briefly outline how prokaryotes go from DNA to proteins. We focus on prokaryotic cells as the findings of this dissertation are build around bacteria cells, *E. coli* in particular. Two well written books discussing the subject of gene expression and regulation are Molecular Biology of The Cell [7] and Genes & Signals [34].

Gene expression, according to the central dogma of molecular biology, takes place in two major stages, transcription and translation. Each of these two steps consists of three phases, initiation, elongation and termination. In transcriptional initiation RNA polymerase (RNAp) first recognizes a specific DNA site, known as the promoter. Recognition is carried out by a certain subunit of RNAp, called sigma factor, by making specific protein-DNA contacts with the promoter bases. RNAp binds with the promoter to form the closed-promoter complex and then the DNA helix unwinds to form the open-promoter complex. Initiation is completed when the first two bases have been transcribed. As soon as the initiation stage is over, elongation starts and involves RNAp moving along the DNA template strand copying the bases according to the Watson-Crick base pairing and producing messenger RNA (mRNA). Similarly to initiation, a specific sequence in the DNA signals RNAp when to stop transcribing. This sequence is referred as the terminator.

After mRNA is produced, it is translated in the ribosome, a two subunit molecule. Ribosomes translate mRNA into amino acids and consequently to proteins based on the genetic code. In similar fashion to transcriptional initiation, signals hidden in the mRNA code dictate translational initiation. During elongation mRNA flows through the ribosome subunits, like the conveyer belt in a factory, where the corresponding amino acids bind to form proteins. Specific codons, i.e. three consecutive nucleotides, on the mRNA template are responsible for signaling termination. The released mRNA can go on and be translated in another ribosome. Both prokaryotes and eukaryotes facilitate polyribosomes, an assembly of ribosomes, where a single mRNA is translated simultaneously by many ribosomes in order to speed up production.

Gene expression is a highly regulated process. Both prokaryotes and eukaryotes use complex regulatory pathways to control the expression of genes. Cell differentiation, environmental adaptation and different protein requirements during cell cycle are some of the reasons explaining why cells use such pathways. As the complexity of the organisms increases so do the layers of regulation. Eukaryotes utilize a far more complex and sophisticated network of regulation than prokaryotes do [7, 34].

Cells have many regulatory pathways in order to control gene expression. Usually regulation occurs in the initiation phase. The main regulatory mechanisms through which prokaryotic cells regulate transcriptional initiation involve proteins that enhance or repress the functionality of a promoter. These proteins bind to specific sequences on the DNA known as operators. Some bacteria promoters are weak; they have poor affinity for RNAp, therefore regulatory proteins, called activators, bind adjacent to the promoter and increase the likelihood for initiation to occur. The result is a gene more actively transcribed; hence this mode is characterized as positive control. Activators either provide an additional surface for RNAp binding or they increase the transition probability from the close to the open promoter complex. On the other hand negative control is observed when a bound protein obstructs the binding of RNAp or the binding of an activator or when it obstructs elongation. These proteins are called repressors and

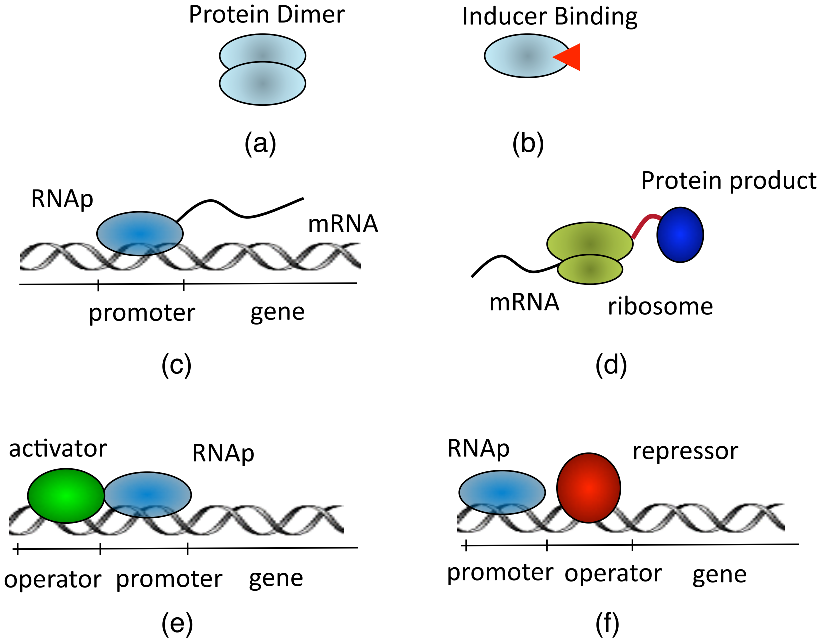


Figure 2.5: Schematic representation of key biological interactions. (a) Protein dimerization. (b) Inducer binding. (c) Transcription. (d) Translation. (e) Activation. (f) Repression.

the obstruction is likely caused due to steric hindrance. Commonly, the binding or unbinding of an activator or a repressor is mediated through small molecules called ligands, adding another layer of control. Ligands binding or unbinding causes conformational changes (allostery) to the protein that either enables or hinders it from binding to the DNA sequence. Furthermore, some bacterial regulatory proteins can function both as an activator or repressor depending on the relative placement of the operator compared to the promoter. The most well studied example of a gene exhibiting both positive and negative regulation is the lac operon of *E. coli* cells. The lac repressor represses expression of beta-galactosidase when glucose is abundant compared to lactose, while CAP (catabolite activator protein) activates expression in the reverse scenario. Unbinding of the lac repressor happens when allolactose (ligand) is present in the medium.

A special case of the above modes is self regulation, where the protein product controls its own production either positively or negatively. An example is the tetracycline resistance operon of E. coli, where the regulatory protein TetR mediates its own production and that of the resistance protein TetA [35]. Another control mechanism that involves transcriptional initiation involves sigma factors. Prokaryotes have different sigma factors, that can be used under different external conditions, e.g. under nitrogen limitation, mediating which genes are expressed under certain environmental stimuli. Regulation can also occur in all the other steps of gene expression, such as translation. For instance, adjustable degradation rates help the cell to control protein levels. A basic schematic representation of some key interactions discussed in the current section is depicted in Fig. 2.5.

2.4.2 Chemical Kinetics Models

Certainly, there are many more hidden aspects of biology and Sec. 2.4.1 only briefly scrapes the surface of the knowledge obtained over the last few decades. Still, that knowledge is enough to create detailed mechanistic models of gene expression and regulation. In what follows we briefly present a chemical kinetics model based on a general gene expression and regulation scheme borrowed from *E. coli* cells. A more detailed description of such a process can be found in Sec. 8.3.1.

The mentality to create chemical kinetics models for biological systems is straightforward. The first step is to identify the interactions at the molecular level that are of interest and next translate those into chemical reaction formulae. In order to avoid confusion all reactions are presented in Table 2.1 with brief descriptive comments. While most of the time our understanding for bacteria cell cycle is sufficient to study their dynamics the usual bottleneck of this approach is to obtain the appropriate kinetic data. Even though many molecular components have been identified, kinetic studies and or thermodynamic properties from which to deduct the necessary kinetic rates are rare. The lack of quantitative description for much of the biological compounds examined makes the need for a continuously expanding database containing kinetic constants of essence.

All in all, mechanistic models of biological functions based on chemical kinetic models constitute a simple and intuitive way for computer-aided design of novel biological systems.

Table 2.1: A General Chemical Kinetics Representation of Gene Expression & Regulation.

#	Reaction	Description
	Protein Interactions	
1	Monomer + Monomer $\rightleftharpoons$ Dimmer	Monomers form a dimer protein
2	Protein + Inducer $\rightleftharpoons$ Protein:Inducer	Inducer binding
	Protein-DNA Interactions	
3	Protein + Operator $\rightleftharpoons$ Protein:Operator	Repressor/Activator binding on DNA
4	Protein:Operator + Inducer $\rightleftharpoons$ Protein:Inducer + Operator	Inducer binding on bound proteins
	RNA-polymerase Recruitment	
5	RNAp + Promoter + Operator $\rightleftharpoons$ RNAp:Promoter:Operator	Open complex formation
	Transcription	
6	RNAp:Promoter:Operator $\longrightarrow$ RNAp*:Promoter:Operator	Transcriptional initiation

Continued on Next Page. . .

Table 2.1: A General Chemical Kinetics Representation of Gene Expression & Regulation. – Continued

#	Reaction	Description
7	RNAp*:Promoter:Operator $\longrightarrow$ RNAp:DNA + Promoter + Operator	Transcriptional elongation
8	RNAp:DNA $\longrightarrow$ RNAp+ mRNA	Transcriptional termination
	Translation	
9	mRNA + Ribosome $\longrightarrow$ mRNA:Ribosome	Translational initiation
10	mRNA:Ribosome $\longrightarrow$ mRNA + mRNA:Rib	Translational elongation
11	mRNA:Rib $\longrightarrow$ Protein + Ribosome	Translational termination
	Degradation	
12	Protein $\longrightarrow \varnothing$	Protein degradation.
13	mRNA $\longrightarrow \varnothing$	mRNA degradation.
14	Inducer $\longrightarrow \varnothing$	Inducer degradation.

The symbol ":" is used to represent complex formation.

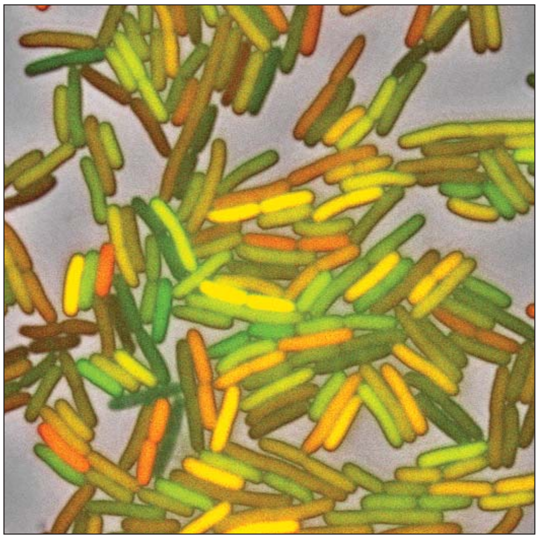


Figure 2.6: A synthetic oscillating gene circuit where *E. coli* cells flash green and red and everything in between. Figure adapted from Ref. [13].

2.5 ”It’s a Noisy Business!” [*]: Importance of Intrinsic Noise

In 2002 Elowitz and coworkers used single-cell experiments to prove that cell expression is an inherent noisy process [37]. They used fluorescent marker proteins to study cell cycles and concluded that cells that initially have identical key components, such as RNA polymerase, ribosomes and protein levels, exhibit different phenotypic behavior. In other words, even though each cell should have demonstrated identical fluorescent levels at the same time intervals, the outcome was cells that revealed a distribution of fluorescent levels. This can be more vividly visualized through Fig. 2.6 where in a similar experiment cells flash in different colors instead of all being in the same ”color” state.

The work of Elowitz and coworkers was the first in a long line showing that cell dynamics are largely stochastic. Intrinsic noise effects play a crucial role in cell differentiation and there are also examples where the cell’s fate is dictated by random effects [26]. Internal noise is the result of fluctuations due to random collision of molecules. Indeed,

[*] Adapted from Ref. [36]

whenever distinct numbers of molecules interact, fluctuations are important and must be considered. Chemical kinetics models far from the thermodynamic limit, which is the limit as the number of molecules (particles) in a system reaches infinity or Avogadros number for practical purposes, experience fluctuations which disappear inversely proportional to the number of molecules in the system [38]. For example the interaction of RNAp with a promoter is a distinct event happening between few numbers of molecules.

Consequently, using mass action formalism to numerical integrate chemical kinetics models may be invalid in many cases. Instead of using ordinary differential equations (ODEs) there is the need for sophisticated stochastic algorithms that incorporate intrinsic noise effects. Among the first to realize that, Arkin and McAdams, used stochastic simulations to describe cell dynamics [26, 39].

Since then, stochastic mathematical models have provided useful insights for many biological processes, leading to a better understanding of the specific interactions involved in gene expression, while quantifying intrinsic noise effects. For instance, the effects of fluctuations in operator-protein binding dynamics on gene expression have been the study in the work of Kepler and Elston [40]. Moreover, the significance of fluctuations of mRNA and protein numbers in gene expression has also been investigated [41]. Bundschuh and coworkers examined how the dimerization of regulatory proteins affects the magnitude of fluctuations when these proteins regulate gene expression [42] and how traditional reduced models fail to capture fluctuations or even averages [43]. Cox and coworkers studied the effects of fluctuations in a quorum sensing system, which is a type of cell-cell signaling pathway [44]. Paulsson used the fluctuation-dissipation theorem to explain and quantify many of the stochastic effects present in the early pioneering single cell experiments [45]. Bar-Evenm and coworkers investigated how noise scales with protein abundance and whether noise levels are affected by global or local pathways, mRNA or promoter levels [46].

Chapter 3

Background Information on Computational Models and Algorithms

3.1 Stochastic Chemical Kinetics Models

Cell functions, such as gene expression, can be depicted through cascades of chemical reactions, where generation, degradation, binding and conformational changes are the reacting outcomes. Due to the importance of intrinsic noise effects in gene expression (cf. Sec. 2.5) models have to account for the stochastic nature of many biological functions. Therefore ordinary differential equations (ODEs) do not represent a valid mathematical description to propagate such systems in time.

A discrete and probabilistic description of chemical kinetics is required, where deterministic rates are substituted with reaction probabilities per unit time. Instead of ODEs, a gain-loss equation for probabilities, known as the chemical master equation (CME), governs the time evolution of the reaction probability density function of the systems chemical population [47, 48].

Consider a well-mixed bacterial-size volume V, i.e. virtually and spatially homogeneous medium, containing N distinct chemical species S_i participating in M chemical reactions. The state vector $\underline{X}(t) = \big(X_1(t), \ldots, X_N(t)\big)$ contains the time evolution of the system, i.e. the number of molecules from each species at a certain time. An $M \times N$ matrix $\underline{\underline{\nu}}$ is defined, containing all stoichiometric coefficients, where ν_{ij} is the change in the number of molecules of the i^{th} species caused by the j^{th} reaction. Reaction propensities, $\underline{\alpha}\big(\underline{X}(t)\big)$, represent the probabilistic reaction rates and form an M^{th} order vector. In particular, $\alpha_j\big(\underline{X}(t)\big)dt$ gives the probability that the j^{th} reaction occurs in a small time interval $\big[t, t + dt\big]$.

Propensities may be calculated using different rate laws such as mass action or Michaelis Menten kinetics. Using mass action kinetics, the probabilistic reaction rates are calculated given the macroscopic (deterministic) rate constant k'_j and the corresponding reaction form and law [48, 29]. In general, the propensity of the j^{th} reaction can be calculated using the equation,

$$\alpha_j\big(\underline{X}(t)\big) = c_j h_j\big(\underline{X}(t)\big) \tag{3.1}$$

where h_j is the number of distinct combinations of the reacting species and c_j is the average specific reaction propensity for the j^{th} reaction, which is also referred to as the mesoscopic reaction rate, designated as k_j. Consider the second order biomolecular reaction,

$$A + B \xrightarrow{k'_j} C \tag{3.2}$$

and the corresponding reaction propensity terms are

$$\begin{aligned} h_j &= \big(\text{number of molecules of A}\big) \times \big(\text{number of molecules of B}\big) \\ c_j \equiv k_j &= \frac{k'_j}{N_A V} \end{aligned} \tag{3.3}$$

where N_A is Avogadro's number.

3.1.1 Markov Processes and Chemical Kinetics

The random walk of a reaction network system in phase space, a collection of possible states, can be described as a Jump Markov process. The Jump Markov process describes how the system moves from one phase point, the current position on the phase space, to others. These transitions are discontinuous and occur with a certain probability, which depends only on the current phase point and not on the system's path. This last characteristic is also the main advantage of using Markov processes to describe stochastic chemical kinetics models.

The time evolution of such systems is described by the chemical master equation (CME), a gain-loss equation for probability [47]. The derivation of the CME is based on the definition of the propensities and uses the "probability balance" axiom, meaning that probabilities of jumping to other phase points from the current one, including the current one too, always add up to one.

The probability of being at a state $\underline{X}(t)$ at time $t+dt$ is the sum of the probabilities of first being in an adjacent state and jumping to state $\underline{X}(t)$ and second the probability of being in state $\underline{X}(t)$ with no transition occurring in $t+dt$. In equation form,

$$\begin{aligned} P\big(\underline{X};t+dt|\underline{X}_0;t_0\big) = &\underbrace{P\big(\underline{X};t|\underline{X}_0;t_0\big)\left[1-\sum_{j=1}^{M}\alpha_j\big(\underline{X}\big)dt\right]}_{\text{no transition occurrence}} \\ &+\underbrace{\sum_{j=1}^{M}\big[P\big(\underline{X}-\underline{\nu}_j;t|\underline{X}_0;t_0\big)\alpha_j\big(\underline{X}-\underline{\nu}_j\big)dt\big]}_{\text{transition from an adjacent state}}, \end{aligned} \tag{3.4}$$

where $P\big(\underline{X};t|\underline{X}_0;t_0\big)$ is the conditional probability of being at state $\underline{X}$ at time t given the fact that the system was at state $\underline{X}$ at time t_0, $\left[1-\sum_{j=1}^{M}\alpha_j\big(\underline{X}\big)dt\right]$ is the probability of no reacting event occurring and $P\big(\underline{X}-\underline{\nu}_j;t|\underline{X}_0;t_0\big)$ is the probability that the system jumps to state $\underline{X}$ from an adjacent state. By doing some algebraic rearrangements in Eq. (3.4) and taking the limit of $dt \rightarrow 0$, yields the CME

$$\frac{\partial P\big(\underline{X};t|\underline{X}_0;t_0\big)}{\partial t} = \sum_{j=1}^{M}\big[\alpha_j\big(\underline{X}-\underline{\nu}_j\big)P\big(\underline{X}-\underline{\nu}_j;t|\underline{X}_0;t_0\big)-\alpha_j\big(\underline{X}\big)P\big(\underline{X};t|\underline{X}_0;t_0\big)\big] \tag{3.5}$$

The CME describes the time evolution of the system. In principle, the CME uniquely determines the probability $P(\underline{X},t)$ of the system being at a state $\underline{X}=\underline{X}(t)$ at time $t>0$. In the Jump Markov process regime the phase space is an ensemble of states with discrete number of molecules in each state, this implies that the solution is a discrete probability distribution.

If instead of a jump or discrete Markov process we were to consider a continuous Markov process then the corresponding CME solution would be a continuous probability distribution. The main difference between the continuous and discrete case is that the state space is no longer a discrete set of states but rather a continuum of states. For a continuous Markov process the CME in its general form and not directly applied to chemical kinetics models has the following form

$$\frac{\partial P(\underline{X},t)}{\partial t} = \int \Big[T\big(\underline{X}/\underline{X}^{'}\big)P\big(\underline{X}^{'},t\big) - T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)\Big]d\underline{X}^{'}, \tag{3.6}$$

where $P(\underline{X},t)$ is the probability of the system being at state $\underline{X}$ at time t. $T\big(\underline{X}/\underline{X}^{'}\big)$ is the transition probability per unit time for the system to jump from state $\underline{X}^{'}$ to state $\underline{X}$ and in the case of chemical kinetics it depends on the reaction propensities.

The solution for any given Markov process is a probability distribution. On the contrary, that for a deterministic description is is a sharp delta function, i.e. a point in the phase space. In Chapter 6 the connection between the stochastic and deterministic description of a chemical kinetics model is established and discussed in depth.

3.1.2 Stochastic Simulation Algorithm

Analytical solutions for the CME exist only for the simplest of cases. For instance, one step Master equations can be solved analytically when the transition rates are constant or linear function of the state of the sytem [47, 49]. On the other hand, numerical solutions of CMEs are discouraged as the computational intensity increases rapidly with increasing system sizes and quickly become intractable.

For complex non-linear systems the CME cannot be analytically solved. Instead numerical techniques are necessary. In the mid 70's Gillespie devised the stochastic stimulation algorithm (SSA) that uses Monte Carlo techniques to accurately sample the underlying probability distribution [48, 50]. The exact form of which can only be obtained if the CME could be solved analytically.

In equation form and for the Direct method variant [48] the algorithm determines

$$\tau = \frac{1}{\alpha} ln \left(\frac{1}{r_1} \right) \tag{3.7}$$

and

$$\sum_{k=1}^{\mu-1} \alpha_k < r_2 \alpha \leq \sum_{k=1}^{\mu} \alpha_k, \tag{3.8}$$

where τ is the time increment at which the next reaction will occur, α is the sum of all reaction propensities and r_1 and r_2 are uniform random numbers. Briefly, the algorithm first determines when the next reaction will occur based on the sum of probabilistic reaction rates (cf. Eq. (3.7)) and then the method establishes which of the M reaction channels will indeed fire, given the relative propensities values. The j^{th} reactions occurs when the cumulative sum of the j first terms becomes greater than $r_2\alpha$ (cf. Eq. (3.8)). Subsequently the reaction propensities are updated, since the system has "jumped" to a neighboring state and the process is repeated until the system reaches the desired end time point.

In fact, the process is repeated multiple times, generating a set of different trials from which the underlying probability distribution can be obtained. The larger the number of trials the more accurate the reconstruction of the distribution will be. There are instances, when the number of trials has to be extremely large for an accurate representation to be obtained rendering the process computational demanding [51, 52]. In general and for most cases, trials in the order of tenth to hundredth of thousands are more than adequate.

3.1.3 Progress in Stochastic Algorithms for Chemical Kinetics

The stochastic simulation algorithm simulates each reacting event, which renders the algorithm accurate, but at the same time computationally demanding. It especially becomes unfavorable when there is a large number of reactions occurring with high frequency. For instance, when fast dimerization reactions are present. There have been numerous attempts to improve the efficiency of SSA and most of them within the last ten years, even though the algorithm exists since the 70's. The effort is parallel to the realization that synthetic biology and gene network engineering in particular can greatly benefit from the existence of sophisticated stochastic algorithms targeted for computer-aided design.

Gibson and Bruck improved the performance of SSA by resourcefully managing the need for random numbers, creating the Next Reaction variant of SSA [53]. Cao and coworkers optimized the Direct reaction variant of the SSA, proving that for certain systems this approach is more efficient than the Next Reaction variant [54]. However the algorithm remained computationally expensive.

A number of mathematically equivalent approximation to the SSA have been proposed aiming to balance efficiency with complexity. In general, such approximations can be classified into two major categories, time-leaping methods and system-partitioning methods. The time-leaping methods depend on the assumption that many reacting events will occur in a time period without significantly changing the reaction probabilities, i.e. the change in the state of the system minimally impacts the reaction propensities. This group includes the explicit [55] and implicit tau-leaping [56] algorithms, which use Poisson random variables to compute the reacting frequencies of each reacting events in a given time interval. The main drawback of the tau-leaping approximation is that it becomes inaccurate when a significant number of critical reactions (reactions where even a single reacting event significantly impacts the reaction propensities) are included in a single leap such that the reaction propensities change excessively or some molecular populations become negative. Concerns have been addressed by adaptively restricting the size of each individual leap [57, 58, 59]. Similarly, Tian and Burrage proposed a tau-leaping method based upon binomial random variables rather than unbounded Poisson random variables [60]. While these recent versions appear to be more vigorous they still are insufficient when small numbers of reacting molecules result in dramatic changes in propensity functions.

The second approach to speeding up the SSA involves separating the system into slow and fast subsets of reactions. In these methods, analytical or numerical approximations to the dynamics of the fast subset are computed and then the slow subset is stochastically simulated. In one of the first such methods, Rao and Arkin applied a quasi-steady-state assumption to the fast reactions and treated the remaining slow reactions as stochastic events [61]. Recently, hybrid methods have received considerable interest. Puchalka and Kierzek partitioned the system into slow and fast reaction subsets, with the first propagated through the Next Reaction variant and the latter through a Poison (tau-leaping) distribution [62]. Haseltine and Rawlings also partition the system into slow and fast reactions, representing them as jump and continuous Markov processes respectively [63]. Both aforementioned hybrid methods suffer when it comes to implementation issues, making them slower or inaccurate. In a similar fashion to Haseltine and Rawlings, Salis and Kaznessis separated the system into slow and fast reactions and managed to overcome the inadequacies and achieve a substantial speed up compared to the SSA while retaining accuracy [64]. Fast reactions are approximated as a continuous Markov process, through Chemical Langevin Equations (CLE) [65] and the slow subset is approximated through jump equations derived by extending the Next Reaction variant approach [53]. This hybrid method is discussed in more detail in Sec. 3.2. Goutsias proposed a quasiequilibrium method [66] based on the work of Rao and Arkin [61] and that of Haseltine and Rawlings [63]. Cao et al. have partitioned the system according to fast and slow species in order to develop the slow-scale SSA (ssSSA) [67]. Finally, Weinan and coworkers studied the use of a two SSA combination scheme, an outer and an inner SSA, where the first simulates slow reactions based on the information received from the latter that propagates the fast subset [68].

Except from those two major categories there are also other approaches that cannot be classified to one or the other group. Such methods include the equation free probabilistic steady state approach of Salis and Kaznessis [69]. In this methodology reactions are partitioned into slow/discrete and fast/discrete subsets and the future states are predicted through the sampling of a quasi steady state marginal distribution. A similar approach is followed in the work of Samant and Vlachos [70]. Recently, Erban and coworkers proposed an equation-free numerical technique in order to speed up SSA [71]. SSA is used to initialize and estimate probability densities and then standard numerical techniques propagate the system. In a different approach, Munsky and Kammash used projection formalism to truncate the state space of the corresponding Markov process and then directly solve or approximate the solution of the CME [72].

3.2 A Hybrid Stochastic Algorithm

Salis and Kaznessis proposed a hybrid stochastic algorithm that is based on a dynamical partitioning of the set of reactions into fast and slow subsets [64]. The fast subset is treated as a continuous Markov process governed by a multidimensional Fokker-Plank equation, while the slow is considered to be a jump or discrete Markov process governed by a CME. The approximation of fast/continuous reactions as a continuous Markov process significantly reduces the computational intensity and introduces a marginal error when compared to the exact jump Markov simulation. This idea becomes very useful in biological systems where reactions with multiple reaction scales are constantly present.

3.2.1 System Partitioning

Given a stochastic chemical kinetics models (cf. Sec 3.1), the set of reactions is dynamically portioned into two subsets, the fast/continuous and slow/discrete reactions. Namely M is now the sum of the fast M^{fast} and slow M^{slow} reactions respectively. Propensities are also designated as fast $\underline{\alpha}^f$ and slow $\underline{\alpha}^s$.

For any reaction to be classified as fast the following two conditions need to be met [64, 65]

- The reaction occurs many times in a small time interval.
- The effect of each reaction on the numbers of reactants and products species is small, when compared to the total numbers of reactant and product species.

Or in equation form, respectively,

$$\begin{aligned} &\text{(i)} \qquad \alpha_j\big(\underline{X}(t)\big) \geq \lambda \gg 1 \\ &\text{(ii)} \qquad X_i(t) > \epsilon|\nu_{ji}|, \end{aligned} \tag{3.9}$$

where the i^{th} species is either a product or a reactant in the j^{th} reaction. The two parameters λ and ϵ define respectively the numbers of reactions occurring within time Δt and what is the upper limit for the effect of a reaction to be negligible in the number of molecules of the reactants and products. This approximation becomes valid when both λ and ϵ become infinite i.e. in the thermodynamic limit. In practice, typical values for λ and ϵ are 10 and 100 respectively. Obviously the conditions must be evaluated multiple times within a simulation since both the propensities and the state of the system change over time. This practically means that it is possible for one reaction to interchange subsets, i.e. fast or slow, within an execution.

3.2.2 Propagation of Fast Subsystem - Chemical Langevin Equation

The fast subset dynamics are assumed to follow a continuous Markov process description and therefore a multidimensional Fokker-Planck equation describes their time evolution [65]. The multidimensional Fokker-Plank equation more accurately describes the evolution of the probability distribution of only the fast reactions. The solution is a distribution, not necessarily Gaussian, depicting the state occupancies. If the interest is in obtaining one of the possible trajectories of the solution, the proper course of action is to solve a system of chemical Langevin Equations (CLEs) [47].

A CLE is an Itô stochastic differential equation (SDE) [73] with multiplicative noise terms and represents one possible solution of the Fokker-Planck equation. From a multidimensional Fokker-Planck equation we end up with a system of CLEs

$$dX_i = \sum_{j=1}^{M^{fast}} \nu_{ji} \alpha_j \big(\underline{X}(t)\big) dt + \sum_{j=1}^{M^{fast}} \nu_{ji} \sqrt{\alpha_j \big(\underline{X}(t)\big)} dW_j, \tag{3.10}$$

where a_j, ν_{ji} are the propensities and the stoichiometric coefficients respectively and W_j is a Wiener process responsible for the Gaussian white noise.

Efficient and accurate integration of CLEs is a significant part of the current dissertation. Chapters 4 and 5 discuss in depth methods that successfully overcome many of the numerical integration challenges that are highlighted in Sec 3.3.

3.2.3 Propagation of Slow Subsystem - Jump Equations

On the other hand, the time evolution of the subset of slow reactions is propagated in time using a slightly modification of the Next Reaction variant of SSA [53]. A system of differential jump equations is used to calculate the next jump of any slow reaction.

The jump equations are defined as follows,

$$\begin{aligned} dR_j(t) &= \alpha_j^s\big(\underline{X}(t)\big)dt \\ R_j\big(t_0\big) &= log\big(r_j\big) \qquad j = 1, \dots, M^{slow}, \end{aligned} \tag{3.11}$$

where R_j denotes the residual of the j^{th} slow reaction, α_j^s are the propensities of only the slow reactions and r_j is a uniform random number in the interval $(0, 1)$. Equation (3.11) depicts the rate at which the reaction residuals change. Note that the initial conditions of all R_j are negative. The next slow reaction occurs when the corresponding residual value makes a zero crossing, from negative to positive values.

Equations (3.11) are also Itô differential equations even though they do not contain any Wiener process, because the propensities of the slow reactions depend on the state of the system, which in turn depends on the system of CLEs. Due to the coupling between the system of CLEs and the differential jump equations, a simultaneous numerical integration is necessary. If there is no coupling between fast and slow subsets or there are only slow reactions the system of differential jump equations (cf. Eq. (3.11)) simplifies to the Next Reaction variant.

The method can be further sped up by allowing more than one zero crossings, i.e. more than one slow reactions to occur in the time it takes the system of CLEs to advance by Δt. Though this is an additional approximation contributing to the error introduced by the approximation of the fast reactions as continuous Markov process, it results in significant decrease in simulation times. The accuracy depends on the number of slow reactions allowed within a Δt and decreases as the number increases.

3.3 Multiple Time Scales in Gene Expression

Biological phenomena, such as gene expression, and in general cell functions are characterized by disparity in time scales. Fast dimerization reactions versus slow transcription events are among the most common encounters in gene network engineering. The different time scales result in stiff mathematical descriptions of the underlying physical problem causing extra challenges in the time integration.

In a sense, the reason for the inefficiency of SSA can be attributed to the existence of

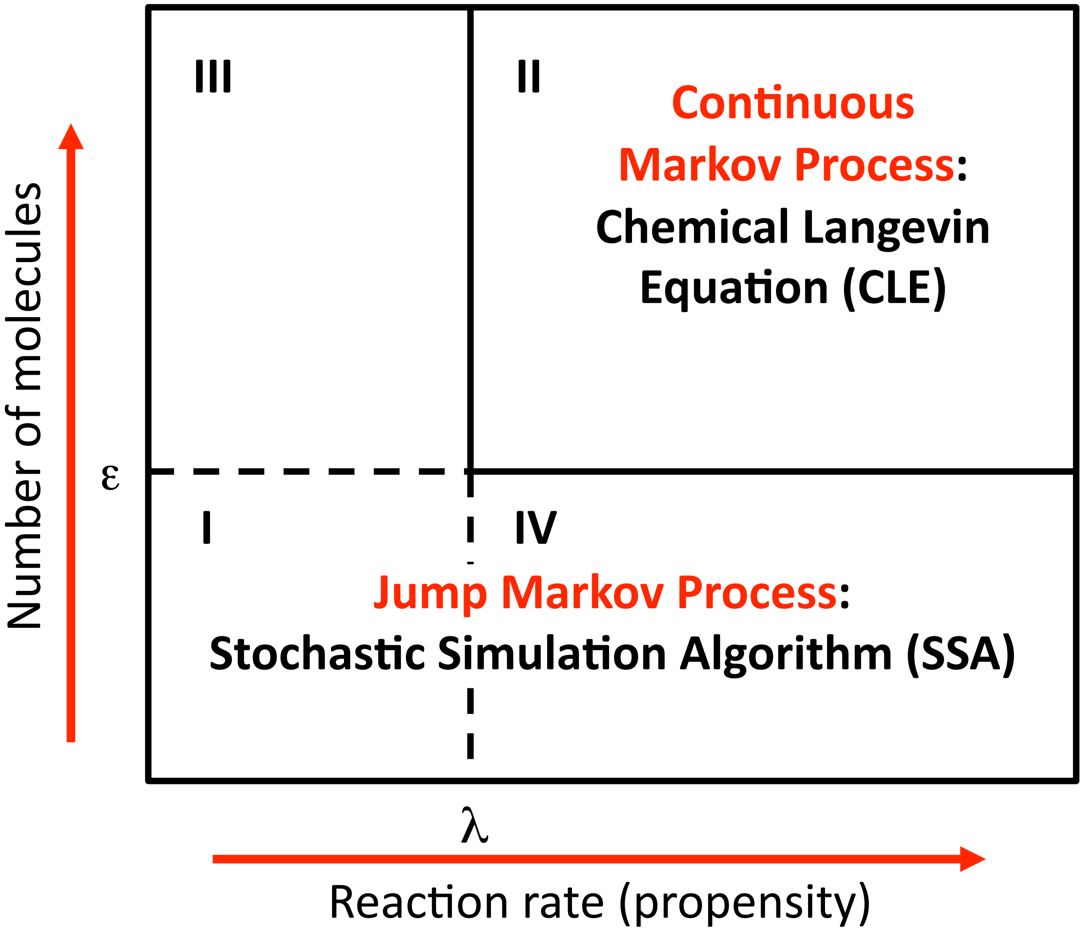


Figure 3.1: Appropriate Markov process regimes depending on the number of molecules and reaction propensities. For even larger values of ϵ and λ the appropriate description falls into ordinary differential equation and intrinsic noise effects can be neglected for practical purposes.

multiple time scales. Therefore much of the effort and approaches discussed in Sec. 3.1.3 aim to increase the efficiency of the algorithms when stiffness is present. A stiff system causes numerical integration methods to become numerically unstable, unless the step size is taken to be extremely small. Figure 3.1 depicts the appropriate Markov process regimes that each of the reactions in a given chemical kinetics model may belong to.

This separation is necessary in order to efficiently and successfully handle stiffness. Many of the methods presented in Sec. 3.1.3 tend to perform better on some of the four regions (I-IV), which are characterized by the relative values of ϵ and λ, or even parts of regions. In particular, the source of stiffness may be dual, either from rather rapidly fluctuating species populations or large kinetic parameter values. Both lead to quickly varying probabilistic reaction rates which in principle are the main reason for numerical instabilities.

At the moment there is not a universal algorithm that can handle the different time scales present in any given chemical kinetics model stemming from a real biological example. A major part of this dissertation focuses on addressing the accurate and efficient numerical integration of stiff chemical Langevin equations. This work builds on the hybrid method presented in Sec. 3.2 studying multiple time scale algorithms that will allow for a greater flexibility in the numerical integration of chemical kinetic systems. To be more concise, Chapters 4 and 5 concentrate on section II of Fig. 3.1 presenting two different methods to overcome stiffness in the continuous Markov process regime.

Examples of biological systems that lie in the continuous Markov process regime may include but are not limited to fast occurring dimerization reactions, inducer-protein interactions and protein binding to non-specific DNA sites.

Chapter 4

An Adaptive Time Step Scheme for Systems of SDEs with Multiple Multiplicative Noise: Chemical Langevin Equation, A Proof of Concept

4.1 Introduction

Studying the effects of the intrinsic fluctuations of cell species in the overall cell phenotype requires the development of sophisticated stochastic models [74], including stochastic-discrete and stochastic-continuous. Both have been introduced by Gillespie [50, 65]. Recent literature has focused on developing algorithms that integrate multiple time scales present in the system kinetics. Building on the work of Gillespie there have been many algorithms that continue and improve its initial work for simulating stochastic chemical systems [53, 55, 60, 54, 67, 69, 68, 61, 63, 64, 70]. The majority of these algorithms have been incorporated in software suites that are able to simulate multiscale models of biological reacting networks, but are not limited to them [1, 75, 76, 77]. In this work we focus on chemical reacting systems that are modeled by a set of stochastic differential or chemical Langevin equations. Importantly we present an adaptive time-step algorithm that numerically integrates stochastic differential equations (SDEs) involving multiple time scales.

The challenge is to develop integration schemes for SDEs that are both accurate and as fast as their deterministic equivalents. While SDEs can be numerically integrated in a similar fashion as ordinary differential equations (ODEs) there are significant differences in the two approaches. The major one stems from the fact that the classic chain rule found in the deterministic case is substituted by the well known Itô formula in stochastic calculus. This complicates the extraction of numerical methods from an Itô-Taylor expansion, since extra terms are introduced. The latter reduces to the chain rule formula only for linear systems. Moreover, the theory behind SDEs becomes complicated and differs from that of ODEs for adaptive and implicit integration methods. However, there is a sufficient number of numerical schemes for SDEs, starting from the simple Euler-Maruyama method, going on with the Milstein method, and continuing with higher order schemes such as Runge Kutta methods. To that we can add the explicit and implicit, partial implicit or fully implicit, versions of the methods [78, 79, 80]. A detailed description and analysis of these methods can be found in the well written book of Kloeden and Platen [73].

Similar to ODEs, multiple time scales in the underlying models cause a system of SDEs to become mathematically stiff. Conventional fixed step methods, in both the stochastic and the deterministic regime, require a small time step for integrating stiff systems. Therefore they become computationally slow. In addition, stiffness may arise during some parts of the simulation allowing for a larger time step in the remaining time interval. Hence the need for an adaptive time step method that will adjust the time step accordingly is evident. To our knowledge, and in contrast to the deterministic cases, adaptive time stepping strategies for SDEs are significantly less developed and limited to special cases. Although recently there has been a considerable effort to develop adaptive time stepping schemes for SDEs the majority of them deals only with cases where there is only a single Wiener process and where SDE terms are commutative [81, 82, 83, 84]. In the literature there are variable step size algorithms which in their majority use higher order methods to numerically integrate the system [81, 82]. This implies multiple Itô integrals have to be approximated increasing the computational cost. An adaptive scheme based on the Euler-Maruyama method is also available, but with a significant implementation cost [85].

In this chapter we present an adaptive time stepping scheme for integrating systems of stiff stochastic differential equations (SDEs) with multiple multiplicative noise. These types of SDEs are the most difficult to numerically integrate due to the intense coupling of the noise terms and the existing stiffness. We use the Milstein method as our numerical method and combine it with local error criteria originating from the work of Lamba [84]. These determine when the adaptive time stepping selection mechanism should be introduced. For the variable step size scheme we choose to use the methodology of Gaines and Lyons [83] that introduced the notion of Brownian trees originated from the work of Lévy [86]. Brownian trees are based on a binary logic; the step can be either halved or doubled.

Finally, the developed framework is applied and tested to a particular class of SDEs, the chemical Langevin equations (CLEs) that arise in the description of dilute chemical reacting systems far from the thermodynamic limit [65]. Such systems can be described as continuous Markov processes governed by a chemical master equation that reduces to a Fokker-Planck equation [64]. The CLE is an Itô stochastic differential equation with multiplicative noise and represents one possible solution of the Fokker-Planck equation. These kinds of systems have recently appeared in the field of computational biology, modeling cell processes and interactions of cell species, where fluctuations play a key role. Three examples are chosen, the first is a system of linear SDEs with multiple multiplicative noise and the second is a nonlinear system. In the final example we use an actual stiff biological example to test the performance of the proposed algorithm in a realistic biological network. Therefore the adaptive scheme is integrated into Hy3S, a collection of multiscale algorithms that use CLEs to propagate system of reactions that belong in the continuous Markov process regime [1]. Hy3S is capable of simulating the stochastic dynamics of networks of biochemical reactions and has been used to study the dynamic behavior of gene regulatory networks [2].

The present chapter is organized as follows. In the Theory section (cf. Sec. 4.2) we formulate the problem and then discuss convergence properties of numerical solutions for SDEs. Next we outline the Milstein method, the numerical scheme used predominantly in CLEs integration. Then we describe the binary adaptive time step selection based on the notion of Brownian trees and continue with the introduction of the local error criteria. Next, we briefly provide some necessary background information about CLEs and finally we discuss implementation issues. In the Examples section (cf. Sec. 4.3) we test and report on the behavior of the proposed scheme and finally in the discussion section (cf. Sec. 4.4) we conclude and argue about the contribution of the present scheme.

Very recently, a new method was published by Lamba *et al.* that demonstrates strong convergence of an adaptive time stepping scheme for SDEs based on the Euler-Maruyama method instead of the Milstein method used in the present work [87]. While their method would be likely faster for a majority of SDEs, since it has less implementation requirements, mainly in that it does not require the approximation of a two dimensional Itô integral, it fails to work for most CLEs, because they do not always satisfy Assumption 5.1 in their manuscript. This assumption requires that the number of reactants be equal to the number of reactions which is not the case in the majority of CLEs and also requires the CLEs to follow a dissipativity condition which is not always necessarily met.

For the first time an adaptive time stepping method is presented for integrating systems of chemical Langevin equations accurately and efficiently. Chemical Langevin equations are stochastic differential equations with multiple multiplicative noises and belong to the subclass of SDEs that are among the hardest to numerically integrate due to the noncommutativity and the multiple noise terms. To our knowledge there are no simple and relative easy to numerically implement schemes that accurately and efficiently overcome noncommutative terms and multiplicative noise. Most previously reported schemes rely on higher order Runge-Kutta algorithms and are harder to implement. Importantly, the scheme is simple to implement and can be potentially applied to any system of SDEs beyond CLEs.

4.2 Theory

We consider a system of Itô SDEs with multiple multiplicative noise,

$$\begin{aligned} dX_i &= f_i\big(\underline{X}(t)\big)dt + \sum_{j=1}^{M} g_{i,j}\big(\underline{X}(t)\big)dW_j(t), \qquad i = 1, \ldots, N \\ X_i(t=0) &= X_{i,0}, \qquad t \in \big[0, T\big], \end{aligned} \tag{4.1}$$

where $\underline{X}(t)$ is a N-dimensional state vector. In the case of biomolecular systems, for example, $\underline{X}(t)$ can be the vector with the concentration of the N species. $\underline{W}(t)$ is an M-dimensional Wiener process. A Wiener process W is a Gaussian process with the following properties

$$E\big(W(t=0)\big) = 0, \qquad E\big(W(t)W(s)\big) = \min(t, s), \tag{4.2}$$

where $E(y)$ is the expectation value of variable y. Additionally, $f_i\big(\underline{X}(t)\big)$, $g_{i,j}\big(\underline{X}(t)\big)$ are scalars with values depending on the state vector of the system. The first part on the right hand side of Eq. (4.1) is called the drift term, while the second is usually referred to as the diffusion part. For chemical reaction networks $f_i\big(\underline{X}(t)\big)$ is associated with the deterministic reaction rates and $g_{i,j}\big(\underline{X}(t)\big)$ are the terms of the fluctuations for a system away from the thermodynamic limit.

There are two ways to write down an SDE, using either the Itô or the Stratonovich formulation. Equation (4.1) uses the Itô formulation. The main difference between the two lies in the way the stochastic integrals are computed. The Itô formulation computes the integral in the beginning of each subinterval while Stratonovich computes it in the middle. The two forms are equivalent and we can obtain the Stratonovich from the Itô form by using a simple formula [73]. From a purely mathematical point of view, both representations are appropriate. From a physical point of view Itô SDEs are more appropriate describing systems in which intrinsic noise is important [73]. In the remainder of the chapter we will consider SDEs in the Itô form, since CLEs are also cast in Itô form.

4.2.1 Strong and Weak Convergence

Before dealing with the actual numerical integration schemes, there is an important concept we have to consider, convergence of the numerical solution. Although these definitions are found in the literature we present these definitions here for completeness. In numerical methods used to integrate SDEs, two definitions of convergence are present, the *weak* and the *strong convergence*. The weak convergence deals with convergence in the probability distribution of the actual and numerical solution. On the other hand, strong convergence deals with convergence between the trajectories of the actual and numerical solution. In equation form the above definitions are formulated as follows [73].

- *Strong convergence*. A time discrete approximation $X_\delta(t)$, of the Itô process X_t, converges strongly to X_t with order $\gamma > 0$ at time T, if there exists a positive constant C, independent of δ and $\delta_0 > 0$, such that,

$$\varepsilon(\delta) = E\big(\big|X_T - X_\delta(T)\big|\big) \leqslant C\delta^\gamma, \qquad \delta \in (0, \delta_0) \tag{4.3}$$

- *Weak convergence*. A time discrete approximation $X_\delta(t)$, of the Itô process X_t, converges weakly to X_t with order $\beta > 0$ at time T, if for each $q \in C_p^{2(\beta+1)}$ [q is $2(\beta+1)$ times differentiable] there exists a positive constant C, independent of δ and $\delta_0 > 0$, such that,

$$\varepsilon(\delta) = \big|E\big(q\big(X_T\big)\big) - E\big(q\big(X_\delta(T)\big)\big)\big| \leqslant C\delta^\beta, \qquad \delta \in (0, \delta_0) \tag{4.4}$$

In practical situations when algorithms are used to numerically compute the solution of SDEs, Eq. (4.3) and (4.4) can be used to calculate the error. Though Eq. (4.3) is more appropriate, its implementation for computing the strong mean and variance errors requires fixing the Brownian paths that the Wiener process follows. Practically this means we have to use the same sequence of random numbers to compute the actual solution, analytically if possible, and the approximate one, which is not always applicable.

4.2.2 Milstein Method

The simplest scheme to numerically integrate Eq. (4.1) is the explicit Euler-Maruyama method. It is derived from first order truncation of the Itô-Taylor expansion [73]. The Euler-Maruyama method has a strong order of 0.5 and weak order 1.0. The scheme for Eq. (4.1) has the form

$$X_i^{k+1} = X_i^k + f_i\big(\underline{X}^k\big)\Delta t + \sum_{j=1}^{M} g_{i,j}\big(\underline{X}^k\big)I(j), \qquad i = 1, \ldots, N, \tag{4.5}$$

where

$$I(j) = \int_t^{t+\Delta t} dW_j(t) \approxeq N_j(0, \Delta t) \tag{4.6}$$

is an one dimensional (1D) Itô integral. In practical applications the 1D Itô integral of Eq. (4.6) can be approximated as a Gaussian random number with zero mean and variance Δt, $N(0, \Delta t)$.

The method we concentrate on is the the explicit Milstein method, which is similar to the Euler-Maruyama method with the only difference being the addition of an extra term containing a two dimensional Itô integral. This extra term is of order $O(\Delta t)$and is responsible for increasing the strong order convergence to 1.0 compared to the 0.5 order of the Euler-Maruyama, while the weak order is also 1.0. The Milstein scheme is derived also from an Itô-Taylor expansion. Applying it directly to Eq. (4.1)

$$\begin{aligned} X_i^{k+1} = X_i^k + f_i\big(\underline{X}^k\big)\Delta t + \sum_{j=1}^{M} g_{i,j}\big(\underline{X}^k\big)I(j) \\ + \underbrace{\sum_{j_1=1}^{M}\sum_{j_2=1}^{M} L_{j_1} g_{i,j_2}\big(\underline{X}^k\big)I(j_1, j_2)}_{\text{extra term}}, \qquad i = 1, \ldots, N, \end{aligned} \tag{4.7}$$

where $I(j_1, j_2)$ is a two dimensional Itô integral:

$$I(j_1, j_2) = \int_t^{t+\Delta t} dW_{j_1}(t)dW_{j_2}(t), \tag{4.8}$$

and L_{j_1} is an operator defined as follows

$$L_{j_1} = \sum_{d}^{M} g_{d,j_1} \frac{\partial}{\partial X_d} \tag{4.9}$$

The difficult part in the implementation of this method is the approximation of the two dimensional (2D) Itô integral (cf. Eq.(4.8)). When $j_1 = j_2$ the expression for the Itô integral simplifies to

$$I(j_1, j_1) = \frac{1}{2}\big\{\big(\Delta W_{j_1}\big)^2 - \Delta t\big\} \tag{4.10}$$

When $j_1 \neq j_2$ with $j_1, j_2 = 1, \ldots, M$ the Itô and Stratonovic versions of the integral are equal and thus we can use the following approximation which is based on the Stratonovic definition of the integral. The approximation is based on a Fourier expansion [73] of the Stratonovic version of Eq.(4.8).

$$\begin{aligned} I(j_1, j_2) = \Delta t\big\{\frac{1}{2}\xi_{j_1}\xi_{j_2} + \sqrt{\rho_p}\big(\mu_{j_1,p}\xi_{j2} - \mu_{j_2,p}\xi_{j1}\big)\big\} \\ + \frac{\Delta t}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\big\{\zeta_{j_1,r}\big(\sqrt{2}\xi_{j_2} + \eta_{j_2,r}\big) - \zeta_{j_2,r}\big(\sqrt{2}\xi_{j_1} + \eta_{j_1,r}\big)\big\}, \end{aligned} \tag{4.11}$$

where

$$\rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2} \tag{4.12}$$

for $j = 1, \ldots, M$, $r = 1, \ldots, p$ and $p = 1, 2, \ldots,$ determines the number of Fourier expansion terms retained in the solution. ξ_j, $\mu_{j,p}$, $\eta_{j,r}$ and $\zeta_{j,r}$ being all independent normal Gaussian random numbers, $N(0,1)$. In particular ξ_j is correlated to ΔW_j by the following relation

$$\xi_j = \frac{1}{\sqrt{\Delta t}}\Delta W_j, \tag{4.13}$$

where ΔW_j is again approximated as a Gaussian random number with zero mean and variance Δt, $N(0, \Delta t)$. The choice of p value determines the accuracy of the 2D Itô integral approximation. In this work it is chosen so it guarantees that the order of strong convergence is $\gamma = 1$. Its actual influence on the solution will be examined in Sec. 4.3.

In a similar fashion, we can extract higher order methods from the Itô-Taylor expansion, such as Runge-Kutta methods [78]. The drawback is that although higher order methods provide a more accurate solution, they also demand the calculation of higher order Itô integrals. Practically this increases the demand for random number generation, hence causing significant slow down in execution times.

Except from the relative small computational overheads, the Milstein method has also another advantage. It is the simplest numerical method with a strong order 1.0. This feature becomes important in our effort to combine the method with a variable size algorithm. According to Gaines and Lyons in order for an adaptive time step regime to converge to the correct solution the numerical method must have at least strong order of convergence one [83]. In conclusion, the Milstein method is the simplest numerical scheme one can use in a variable time stepping procedure by balancing out the required accuracy and computational overhead.

4.2.3 Adaptive Time Stepping: Brownian Trees

Using an adaptive time stepping scheme allows for an integration step size that is dynamically changed depending on convergence criteria. Such schemes increase the efficiency of simulation, especially when dealing with dynamically stiff systems, where the method will decrease the time step of the numerical integration when stiffness exists, but will increase it when the system is no longer stiff.

Adaptive time step methods for SDEs differ significantly from ODEs. In the deterministic case using an adaptive scheme requires the recalculation of all terms using the new time step. In the stochastic case while the recalculation using the new time step is also required, the procedure differs when considering the Wiener increments ΔW_j. While it seems logical to solely draw new ΔW_j that will correspond to the altered time step and continue with the calculation, this would be inappropriate. When integrating SDEs it is important that the solution remains in the proper Brownian path, where the term Brownian path is used for describing the increment of the Wiener process ΔW_j. This means that if we have to alter the time step we cannot simply throw away ΔW_j and draw new ones. In that case we would end up with a biased solution, meaning the

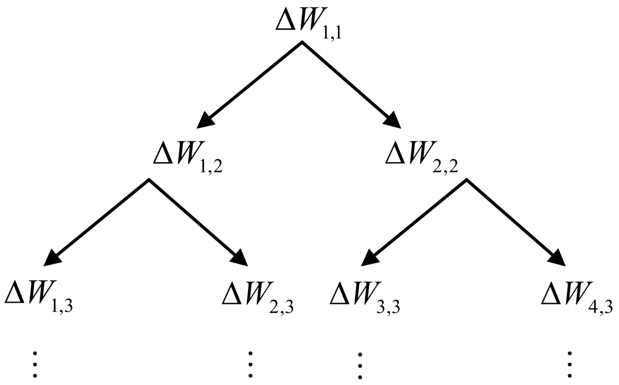


Figure 4.1: A schematic representation of a Brownian tree.

solution would not remain in the correct Brownian path. In order to ensure that the solution remains on the correct path we have to condition the new ΔW_j selection on the previous one. In other words instead of drawing a new ΔW_j we compute the new one using the old one. For example, if $\Delta W_{\Delta t}$ corresponds to time step Δt, then if the time step is halved, $\Delta W_{\Delta t/2}$ has to be computed using $\Delta W_{\Delta t}$. The implementation of this strategy requires the use of binary data structures referred to as Brownian trees.

The notion of Brownian trees introduced by Gaines and Lyons based on results of Lévy becomes very useful in selecting the new Wiener increments (ΔW_j) conditioned on the previous ones [83]. Brownian trees are based on the following binary logic: the time integration step can be either halved or doubled. Generalizations of the Brownian trees where the time step is increased in multiples between 0.0 and 2.0 exist in the literature, but require extra effort to compute the new Wiener increments [81]. A Brownian tree is made up of increments of Wiener processes $\Delta W_{i,j}$, where j indicates the row and i the branch where ΔW is located on the tree. The values of each row are conditioned to the corresponding values of the previous row. Also each $\Delta W_{i,j}$ corresponds to a time step that is equal to the initial time step divided by 2^{j-1}.

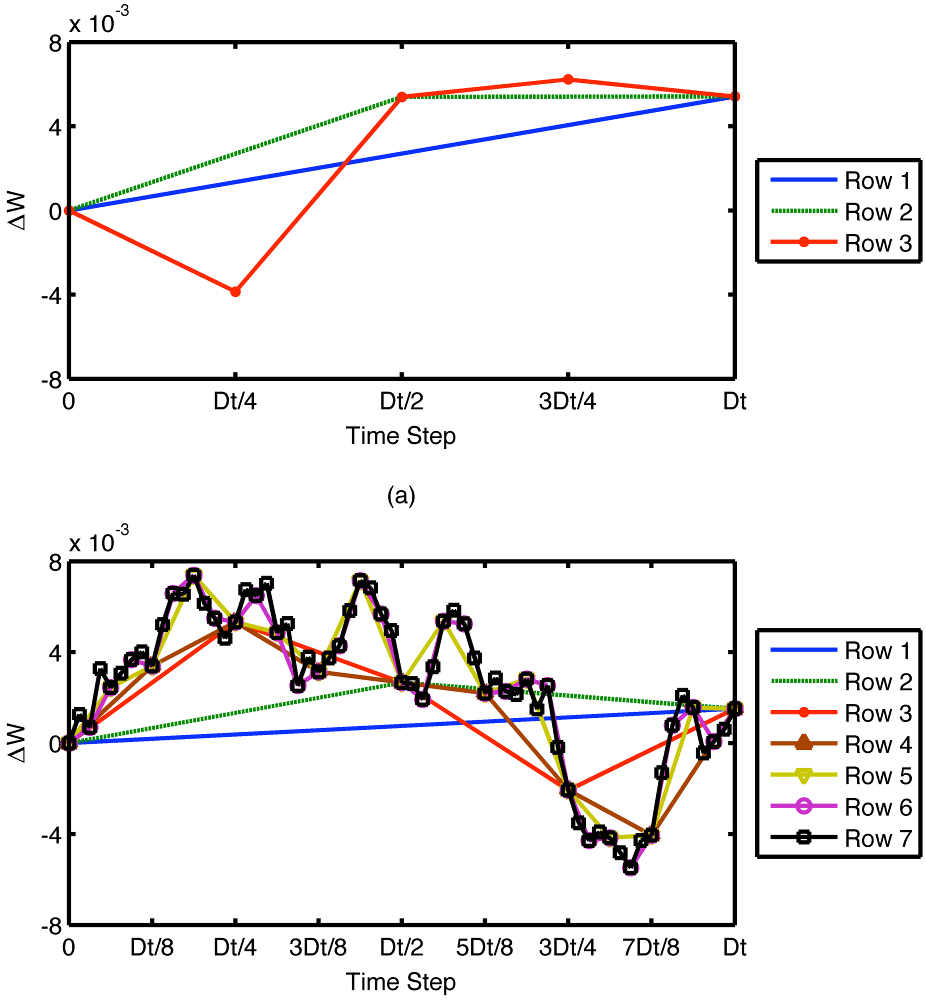
In Fig. 4.1, a schematic representation of a Brownian tree is presented. The top row corresponds to the initial time step of the simulation Δt_{ini} and houses the corresponding Gaussian random numbers (for a single SDE, $\Delta W_{1,1}$ is a scalar value, while for a system of SDEs $\Delta W_{1,1}$ is a vector). Additional branches and rows are created when halving the time step. The number of branches of each row is 2^{j-1} and the corresponding time step for each $\Delta W_{i,j}$ is $\Delta t_{\text{ini}}/2^{j-1}$, where j is the row number. The construction of the Brownian tree, i.e. evaluating the Wiener increments of each branch, utilizes a relationship introduced by Lévy [82, 83]

$$\begin{aligned} \Delta W_{2k-1,j+1} &= \frac{1}{2}\Delta W_{k.j} + y_{k,j}, \qquad && j = 1, 2, \ldots, \#\text{ of rows} \\ \Delta W_{2k,j+1} &= \frac{1}{2}\Delta W_{k.j} - y_{k,j}, \qquad && k = 1, 2, \ldots, 2^{j-1}, \end{aligned} \tag{4.14}$$

where $y_{k,j}$ follows a normal distribution with zero mean and variance 2^{-j} and k denotes the number of nodes with each node having two branches. This procedure assures that the integration will remain on the proper path initially defined by $\Delta W_{1,1}$. It is important to note that for the definition of the Brownian tree only $\Delta W_{1,1}$ is necessary, since all other rows follow from Eq. (4.14). This allows for a convenient way to dynamically generate the tree during simulations.

Going up and down in the Brownian tree or equivalently altering the time step depends on whether or not a set of pre-defined constraints are met. For example, if we try an initial time step Δt_{ini} with $\Delta W_{1,1}$ and one or more of the constraints are not met, the system is rewound and half the initial step size is tried where now $\Delta W_{1,2}$ is used. If that fails then $\Delta t_{\text{ini}}/4$ with $\Delta W_{1,3}$ is tried, but if the criteria are met the system is propagated with the remaining half step and $\Delta W_{2,2}$. The procedure continues until the final time has been reached. Every time the step size is accepted the algorithm checks if it is possible to climb up the tree, otherwise the time step is kept the same. The step can be doubled whenever the branch number is divisible by two. This procedure allows for a flexible time step size that will decrease if necessary and increase when able.

The procedure described above corresponds to an SDE with a single Wiener process. For a system of SDEs with multiple noise terms (M Wiener increments) the above mentioned scheme can be easily generalized. The main difference is that at every branch of the Brownian tree instead of $\Delta W_{i,j}$ being a scalar, $\Delta W_{i,j}$ is a vector containing the M Wiener increments. As in the single Wiener process case, we use Eq. (4.14) to generate the corresponding vector elements of each branch. Equation (4.14) is applied at each vector element. Again we only need to know the elements of vector $\underline{\Delta W}_{1,1}$ in order to construct all subsequent rows and branches of the tree. Finally the strategy for climbing up and down the tree is independent of the number of Wiener processes and therefore



(a)

(b)

Figure 4.2: Two distinct representations of a Brownian tree. (a) A Brownian tree with 3 rows. (b) A Brownian tree with 7 rows.

is identical in either the single or multiple Wiener processes cases.

In Fig. 4.2 we use MATLAB to generate two different visualizations of the Brownian tree. In both subfigures an initial value of the random process $W \sim N(0, \Delta t)$ is chosen and through Eq. (4.14) the underlying rows are created. Δt_{ini} is chosen to be 0.01 s and we assume that we start from $W = 0$ for simplicity, i.e., $\Delta W = W \sim N(0, \Delta t)$. In Fig. 4.2(a) the values between each dot depict a Wiener increment corresponding to $\Delta W_{i,3}$, $i = 1, 2, 3, 4$. In other words, Fig. 4.2(a) is an example realization of Fig. 4.1, where the bold (blue) dashed (green) and bold with dots (red) line depict the first, second and third row respectively. Note that all three lines have the same start and end points which basically summarizes the idea of Brownian trees, remain on the correct Brownian path. From Fig. 4.2(b) we observe that as the row number increases the Wiener increment become finer and finer.

4.2.4 Error Criteria

In order for the adaptive scheme to decide whether or not to alter the time step local error criteria are necessary. Lamba proposed a set of criteria that determine the local error on both the drift and diffusion terms [84]. In that work both criteria are present for an SDE with one dimensional Wiener process. Based on these derivations, we extend the criteria so that they apply to a system of SDEs with multiple multiplicative noise as in Eq. (4.1). The drift local error (E_d) is namely

$$E_d\big(\underline{X}^k, \Delta t\big) = \left\| \frac{\Delta t}{2} \big(\underline{f}(\underline{X}^k + \Delta t \underline{f}(\underline{X}^k)) - \underline{f}(\underline{X}^k)\big) \right\|_\infty \tag{4.15}$$

The infinity norm corresponds to the maximum absolute sum along the row dimension. The drift local error is of order . The local error of the diffusion term E_{df} is

$$E_{df}\big(\underline{X}^k, \Delta t\big) = \frac{1}{6} \,\big|\, \underline{\Delta W}_3 \,\big|\, \big\| \underline{\underline{h}}' \big\|_\infty \big\| \underline{\underline{h}}' \cdot \underline{h} \big\|_\infty \tag{4.16}$$

where $\underline{\Delta W}_3$ is an M-dimensional vector with cubed Gaussian random numbers as elements, h ($N \times 1$ matrix) contains the sum of the corresponding row elements of matrix $\underline{\underline{g}}\big(\underline{X}(t)\big)$, meaning $h_j = \sum_i g_{j,i}$, $\underline{\underline{h}}'$ ($N \times N$ matrix) is the Jacobian of $\underline{h}$ and $\cdot$ symbolizes matrix vector multiplication. Again the infinity norm corresponds to the maximum absolute sum along the row dimension. Both errors correspond to estimates of the leading error component in the drift and diffusion terms.

4.2.5 Chemical Langevin Equations

A subset of SDEs with multiple multiplicative noise terms is the chemical Langevin equations (CLEs) [65]. In what follows, we will briefly outline how CLEs emerge given a chemical kinetics model. Consider a well-mixed volume V, containing N distinct chemical species S_i $(i = 1, \dots, N)$ participating in M chemical reactions. The state vector $\underline{X}(t) = \big(X_1(t), \dots, X_N(t)\big)$ contains the time evolution of the system, i.e. the number of molecules from each species at a certain time. An $M \times N$ matrix $\underline{\underline{\nu}}$ is defined, containing all stoichiometric coefficients, where $\nu_{i,j}$ is the change in the number of S_i molecules caused by the j^{th} reaction. Reaction propensities, $\underline{\alpha}\big(\underline{X}(t)\big)dt$, form an M-vector denoting the probabilistic rates of a reaction. In particular, $\underline{\alpha}\big(\underline{X}(t)\big)dt$ gives the probability that the j^{th} reaction occurs in a small time interval $[t, t + dt]$.

If the following conditions are met the system of reactions can be described as a continuous time Markov process governed by multidimensional Fokker-Planck equation [64].

(i) The reaction occurs many times in a small time interval.

(ii) The effect of each reaction on the numbers of reactants and products species is small, when compared to the total numbers of reactant and product species.

Or in equation form, respectively,

$$\begin{aligned} &\text{(i)} \qquad \alpha_j\big(\underline{X}(t)\big) \geq \lambda \gg 1 \\ &\text{(ii)} \qquad X_i(t) > \epsilon|\nu_{ji}|, \end{aligned} \tag{4.17}$$

where the i^{th} species is either a product or a reactant in the j^{th} reaction.

The two parameters λ and ϵ define respectively the numbers of reactions occurring within time Δt and what is the upper limit for the effect of a reaction to be negligible in the number of molecules of the reactants and products. This approximation becomes valid when both λ and ϵ become infinite i.e. in the thermodynamic limit. In practice, typical values for λ and ϵ are 10 and 100 respectively.

The multidimensional Fokker-Planck equation describes the evolution of the probability distribution of the reactions. The solution is a distribution, not necessarily Gaussian, depicting the state occupancies. If the interest is in obtaining one of the possible trajectories of the solution, the proper course of action is to solve a system of CLEs [47]. The CLE is an Itô stochastic differential equation with multiplicative noise and represents one possible solution of the Fokker-Planck equation. From a multidimensional Fokker-Planck equation we end up with a system of CLEs

$$dX_i = \sum_{j=1}^{M} \nu_{ji} \alpha_j \big(\underline{X}(t)\big) dt + \sum_{j=1}^{M} \nu_{ji} \sqrt{\alpha_j \big(\underline{X}(t)\big)} dW_j, \tag{4.18}$$

where α_j, ν_{ji} are the propensities and the stoichiometric coefficients respectively, M is the number of fast reactions and W is a Wiener Process with dimension M, producing the Gaussian white noise.

In order to validate the proposed algorithm we will need to compare our numerical solution with the actual solution. In general, for systems of SDEs with multiple multiplicative noise, like Eq. (4.18), analytical solutions do not exist. Therefore there is the need to find an alternative way to estimate the error. In the case of chemical kinetics models we are able to compute an accurate numerical solution using the fact that the original system is a discrete time Markov process governed by a chemical master equation [50]. We use the stochastic simulation algorithm (SSA) to obtain the trajectories which we consider as accurate as the actual solution [53]. Note that the trajectories produced by the system of CLEs are an approximation to the solution of SSA when the above mentioned conditions are met with the error being marginal [64].

As a final point we would like to discuss under what conditions systems in the form of Eq. (4.18) are considered to be stiff. While in the deterministic case we can simply judge by looking into the eigenvalues of the system, in the stochastic regime there is not a similar criterion. Nevertheless, one can judge whether or not the system is stiff by looking into the values of the reaction propensities. The larger the absolute ratio of the maximum propensity over the minimum propensity, the larger the stiffness of the system [69]. A measure of stiffness S_f can then be defined as

$$S_f = \frac{\max\big(\underline{\alpha}(t)\big)}{\min\big(\underline{\alpha}(t)\big)} \tag{4.19}$$

where $\underline{\alpha}(t)$ corresponds to the vector of reaction propensities.

4.2.6 Implementation Details

In this section we apply the adaptive time stepping algorithm to the system of CLEs (cf. Eq. (4.18)) and present a brief description of how the algorithm functions.

1. Milstein Method

We start by rewriting the basic relations of the numerical integration scheme as they apply in the system of CLEs. Applying the Milstein method to the system of CLEs,

$$\begin{aligned} X_i^{k+1} &= X_i^k + \sum_{j=1}^{M} \nu_{ji}\alpha_j\big(\underline{X}^k\big)\Delta t + \sum_{j=1}^{M} \nu_{ji}\sqrt{\alpha_j\big(\underline{X}^k\big)}I(j) \\ &\quad + \frac{1}{2}\sum_{j_1=1}^{M}\sum_{j_2=1}^{M}\sum_{n=1}^{M} \nu_{j_1,n}\nu_{j_2,i}\sqrt{\frac{\alpha_{j_1}\big(\underline{X}^k\big)}{\alpha_{j_2}\big(\underline{X}^k\big)}}\frac{\partial \alpha_{j_1}}{\partial X_n}I(j_1,j_2), \qquad i = 1,\ldots,N, \end{aligned} \tag{4.20}$$

where $I(j)$, $I(j_1, j_2)$ represent 1D and 2D Itô integrals, respecively.

2. Local Error Criteria

For the local error criteria, Eq. (4.15) and (4.16), we have for the vector $\underline{f}$ and matrix $\underline{\underline{g}}$, respectively.

$$\underline{f} = \begin{bmatrix} \sum\limits_{j=1}^{M} \nu_{j1}\alpha_j\big(\underline{X}^k\big) \\ \vdots \\ \sum\limits_{j=1}^{M} \nu_{jN}\alpha_j\big(\underline{X}^k\big) \end{bmatrix} \tag{4.21}$$

$$\underline{\underline{g}} = \begin{bmatrix} \sum_{j=1}^{M} \nu_{11} \sqrt{\alpha_1 \left(\underline{X}^k \right)} & \cdots & \sum_{j=1}^{M} \nu_{M1} \sqrt{\alpha_M \left(\underline{X}^k \right)} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{M} \nu_{1N} \sqrt{\alpha_1 \left(\underline{X}^k \right)} & \cdots & \sum_{j=1}^{M} \nu_{M1} \sqrt{\alpha_M \left(\underline{X}^k \right)} \end{bmatrix}, \qquad (4.22)$$

which when substituted in the respective equations yield the local error criteria applicable to our case. In our algorithm we introduce two user defined parameters that are used to control the error tolerance in the adaptive scheme. The first variable called SDE tolerance represents the tolerance in the diffusion error term. The second parameter is a weight coefficient, called SDE coeff, that weights in the importance of the tolerance in the drift term. Basically, the tolerance in the drift term is calculated by multiplying SDE coef with SDE tolerance. The higher the SDE coeff value the less important the drift error becomes and vise versa. While the standard procedure is to use the same error tolerance for both error terms, we found that by using this extra weight coefficient the algorithm can be better tuned to produce both accurate and fast results. Therefore we choose to use a value of 1,000 for the SDE coeff. Of course, the choice of the coefficient is user defined and may vary depending the application. Usually the further the system of SDEs from the diffusion limit the more important the drift terms becomes and the smaller the SDE coeff should be.

3. Convergense

For comparison purposes we will contrast the behavior of the proposed scheme with the solution obtained from SSA. For that we chose to use the definition of the weak convergence. Even though the definition of strong convergence is more appropriate there is no known way we can use the same sequence of random numbers to ensure the same path wise solution. This difficulty stems from the different approach the two algorithms, namely SSA and the proposed one, have. Therefore, the error is computed based on the definition of the weak error. We know that the weak convergence quantifies the convergence of probability distributions of the actual and the numerical solution. The mean and the variance of a distribution are two of its important characteristics. The weak mean error measures the difference in the mean of the actual and numerically obtained distribution while the weak variance error measures the difference in the variance of the two distributions. If we normalize the errors using the actual solution, we obtain the following equations for the weak mean and variance error

$$\begin{aligned}\Delta^i_{\text{mean}}(t) &= \frac{\left|\, E\big[X_i^{\text{actual}}(t)\big] - E\big[X_i^{\text{numerical}}(t)\big] \,\right|}{E\big[X_i^{\text{actual}}(t)\big]} \\[2ex] \Delta^i_{\text{var}}(t) &= \frac{\left|\, \text{var}\big[X_i^{\text{actual}}(t)\big] - \text{var}\big[X_i^{\text{numerical}}(t)\big] \,\right|}{\text{var}\big[X_i^{\text{actual}}(t)\big]}\end{aligned} \tag{4.23}$$

where index i corresponds to the i^{th} component of the state vector, $E\big[X_i(t)\big]$ denotes the mean of $X_i(t)$ and $\text{var}\big[X_i(t)\big]$ stands for the variance of $X_i(t)$.

4.3 Examples

The subject of this section is to validate the adaptive scheme. Three examples are used. The first is a chemical kinetics model that leads to a system of stiff linear CLEs, while the second one leads to a system of stiff non-linear CLEs. In the final example the adaptive scheme is integrated into Hy3S, a multiscale algorithm that uses CLEs to propagate the system of reactions that belong in the continuous Markov process regime [1]. Results are compared with the actual solution obtained through SSA and with the simple Milstein scheme.

4.3.1 A Reversible Dimerization Reaction

Consider the reaction network depicted in Table 4.1, a slight modification from the one used to study the implicit and explicit tau-leap methods [56]. The initial propensities of the two reactions are

$$\begin{aligned}\alpha_1 &= k_1 \times S_1 = 10^8 \text{ molecules s}^{-1} \\ \alpha_2 &= k_2 \times S_2 = 10^8 \text{ molecules s}^{-1} \\ \alpha_3 &= k_3 = 9.998 \times 10^4 \text{ molecules s}^{-1}\end{aligned} \tag{4.24}$$

The system initially and during the course of simulation satisfies Eq. (4.17), which means that its time evolution can be described through a system of CLEs. Moreover, fluctuations and the large values of the propensities, initially and during the course of

Table 4.1: Reactions and Parameters for a Reversible Dimerization Reaction

Set of Reactions	Mesoscopic Reaction Rates[†]	Initial Values[‡]
$S_1 \xrightarrow{k_1} S_2$	$k_1 = 1.0 \times 10^5$	$\left[S_1\right]_0 = 10^3$
$S_2 \xrightarrow{k_2} S_1$	$k_2 = 1.0 \times 10^5$	$\left[S_2\right]_0 = 10^3$
$\xrightarrow{k_3} S_1$	$k_3 = 9.998 \times 10^4$	

[†] for 0^{th} order reactions the units are molecules s^{-1} and for 1^{st} order reactions the units are s^{-1}.

[‡] initial values are in number of molecules.

simulation, are responsible for the mathematical stiffness arising in the system of CLEs.

In order to validate the adaptive scheme we use SSA to simulate the system on the time interval [0, 0.01] s. We use the SSA realization available in Hy3S [1]. As we mentioned in Sec. 4.2 the trajectories generated by both the SSA and the proposed scheme sample the underlying distribution. In order to accurately sample the distribution we need a large sample of trajectories. Here we run 10,000 independent trials. The results obtained through SSA are presented in Fig. 4.3. Figure 4.3(b) presents the probability distribution of the number of molecules of S_1 and S_2 at time $t = 0.005$ s. The computed mean and variance (cf. Fig. 4.3(c) and 4.3(d)) will be used to validate the adaptive Milstein scheme in the remainder of this section.

In what follows we propagate system of reactions (cf. Table 4.1) in time through a system of CLEs and compare the results with SSA so that we can determine the accuracy of the adaptive scheme. The system of CLEs is

$$\begin{aligned} dX_1 &= \left[\nu_{11}\alpha_1\big(\underline{X}(t)\big) + \nu_{21}\alpha_2\big(\underline{X}(t)\big) + \nu_{31}\alpha_3\big(\underline{X}(t)\big)\right]dt \\ &\quad + \nu_{11}\sqrt{\alpha_1\big(\underline{X}(t)\big)}dW_1 + \nu_{21}\sqrt{\alpha_2\big(\underline{X}(t)\big)}dW_2 + \nu_{31}\sqrt{\alpha_3\big(\underline{X}(t)\big)}dW_3 \\ dX_2 &= \left[\nu_{12}\alpha_1\big(\underline{X}(t)\big) + \nu_{22}\alpha_2\big(\underline{X}(t)\big)\right]dt + \nu_{12}\sqrt{\alpha_1\big(\underline{X}(t)\big)}dW_1 \\ &\quad + \nu_{22}\sqrt{\alpha_2\big(\underline{X}(t)\big)}dW_2, \end{aligned} \tag{4.25}$$

where X_1 corresponds to the state of species S_1, meaning the number of molecules of

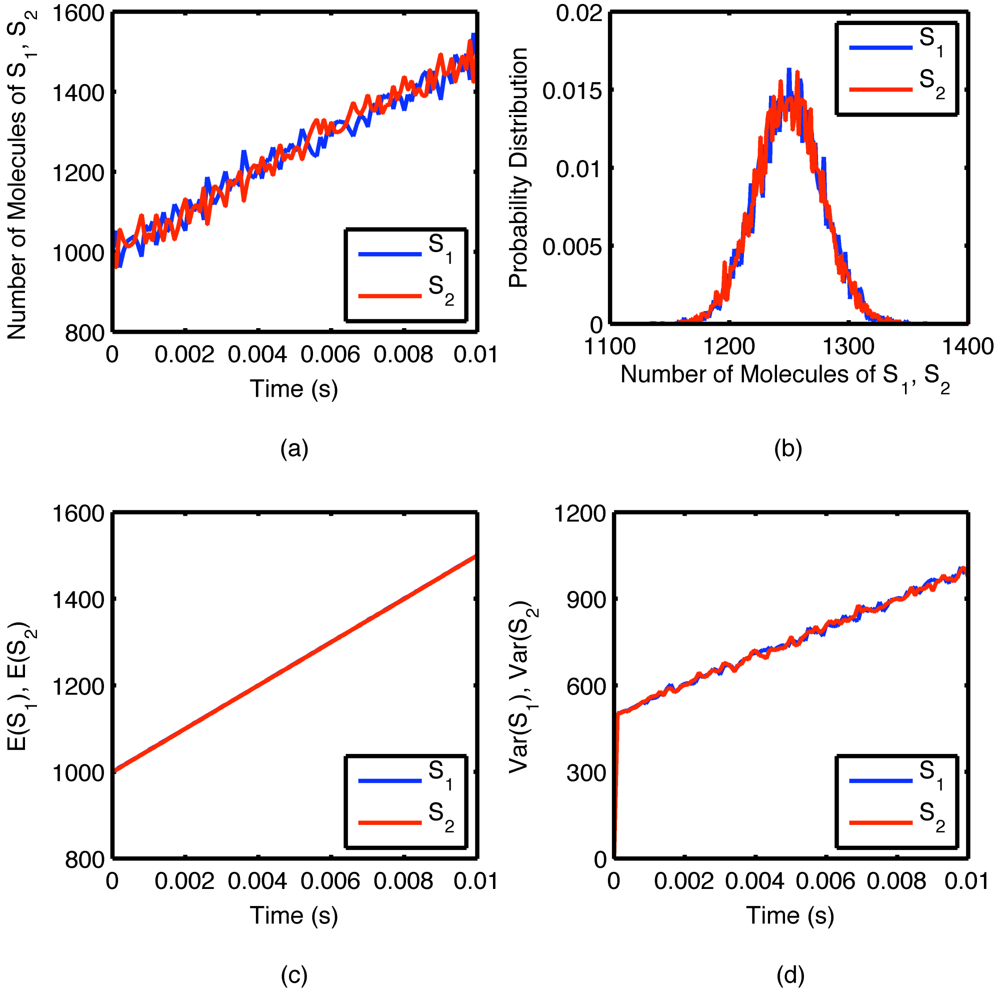


Figure 4.3: Solution of the dimerization reaction system, depicted in Table 4.1, using SSA in the interval [0, 0.01] s. (a) A sample trajectory of the number of S_1 and S_2 molecules. (b) The probability distribution of species S_1 and S_2 at $t = 0.005$ s. (c) Mean number of S_1 and S_2 molecules averaged over 10,000 trials. (d) Variance of number of S_1 and S_2 molecules averaged over 10,000 trials.

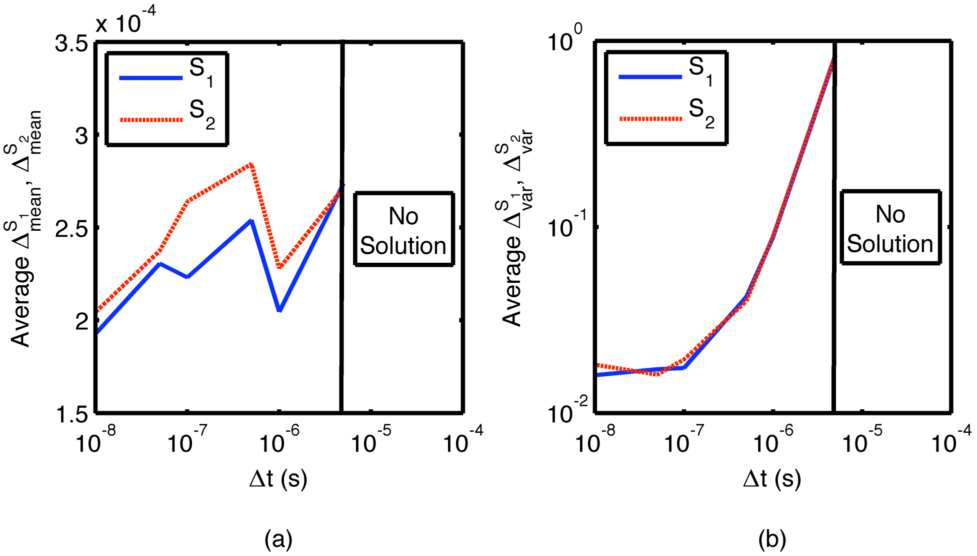


Figure 4.4: Solution of Eq. (4.25) using the fixed step Milstein method ($p = 10$). (a) Comparison of the average normalized weak mean error of S_1 and S_2 generated by the fixed step (log scale) Milstein method. (b) Comparison of the average normalized weak mean variance of S_1 and S_2 (log scale) generated by the fixed step (log scale) Milstein method.

S_1, and X_2 corresponds to the state of species S_2, ν_{ij} is the stoichiometric coefficient of the i^{th} species in the j^{th} reaction, and $\alpha_1\big(\underline{X}(t)\big) = k_1 \times X_1(t)$, $\alpha_2\big(\underline{X}(t)\big) = k_2 \times X_2(t)$, and $\alpha_3\big(\underline{X}(t)\big) = k_3$ are the reaction propensities.

We first examine how well the fixed step Milstein method captures the complex dynamics. The 2D Itô integral is approximated with a parameter p set to 10. Its effect on the error will be discussed below. We run again 10,000 independent trials and calculate the solution in the interval [0, 0.01] s. For comparison with the actual solution (cf. Fig. 4.3) we calculate the normalized weak mean and variance errors of S_1 using Eq. (4.23) and report on their average. The results are depicted in Fig. 4.4.

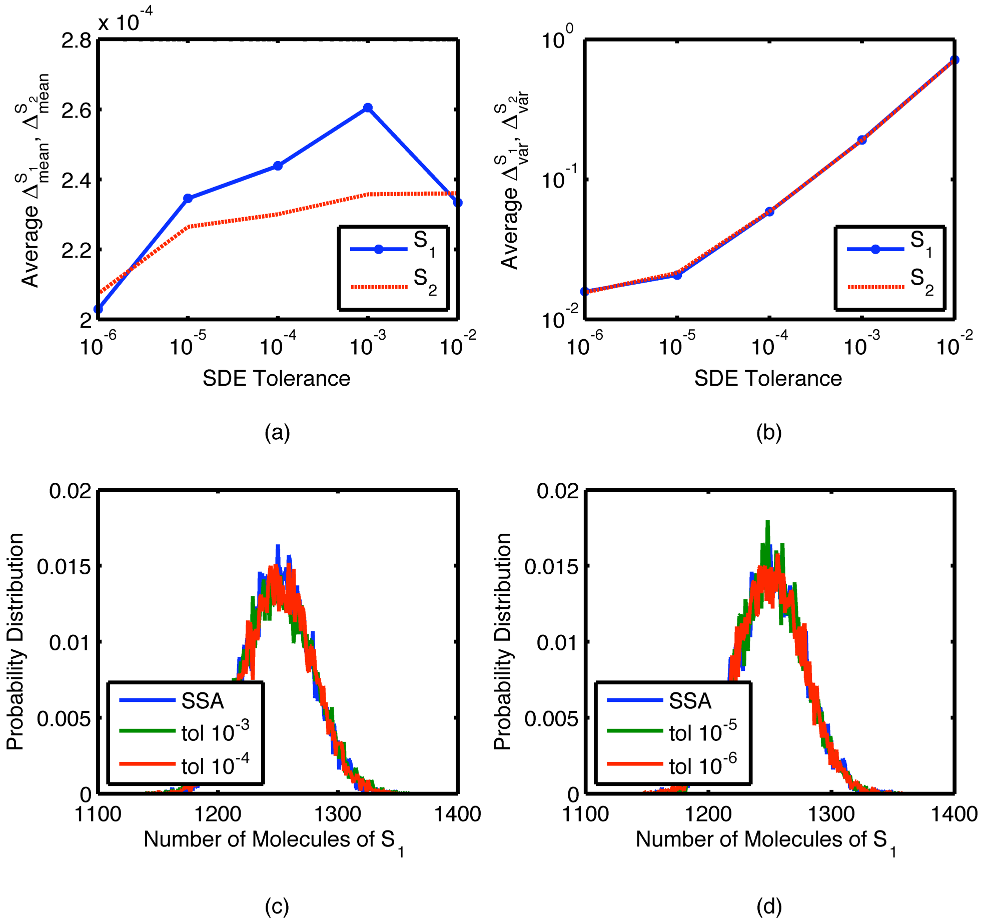
From Fig. 4.4 we observe that the fixed step method fails to produce solutions when the step size is larger than 5.0×10^{-6} s. This happens because during the integration species concentrations attain negative values, which is physically unacceptable. Additionally, there is a slight decrease in the normalized weak mean error as the time step decreases (cf. Fig. 4.4(a)), but still the order of the error remains $O(10^{-4})$. On the other hand, the error in the variance decreases continuously as the time step decreases (cf. Fig. 4.4(b)). The variance error decreases roughly by two orders of magnitude. In summary, as the time step decreases significantly the fixed step method produces accurate solutions. By comparing the corresponding probability distributions we can infer that an acceptable convergence in distribution is observed when Δt is equal to or less than 5.0×10^{-7} s.

Next we study the accuracy of the proposed adaptive scheme. First, we examine how the error generated by the scheme is affected by the user defined error tolerance (SDE tolerance). On the other hand SDE coeff is kept constant at 1,000 during the course of all simulations involving the adaptive scheme. Second, we examine the effect of parameter p on the approximation of the 2-D Itô integral and consequently in the error of the solution.

For analyzing the effect of SDE tolerance we again run 10,000 independent simulations and fix the initial time step to be $\Delta t_{\text{ini}} = 10^{-4}$ s and $p = 10$ for varying SDE tolerance values, ranging from 10^{-2} to 10^{-6}. Subsequently, we calculate the average normalized weak mean and variance errors of S_1 and S_2 and the probability distribution of S_1 at $t = 0.005$ s (cf. Fig. 4.5). It is evident from Fig. 4.5(a) and 4.5(b) that as we decrease the error tolerance the weak mean error decreases slightly, while the decrease is more extreme in the variance where the change is two orders of magnitude between SDE tolerance 10^{-2} to 10^{-6}. The most affected quantity by the variation of SDE tolerance is the variance as was the case in Fig. 4.4. If we further compare Fig. 4.4, 4.5(a) and 4.5(b) we observe that for the fixed step Milstein method to reach the same accuracy as the adaptive step Milstein with SDE tolerance 10^{-4} the needed step size is approximately $\Delta t = 5.0 \times 10^{-7}$ s. Recall that all adaptive time step runs were initialized with $\Delta t_{\text{ini}} = 10^{-4}$ s for which the fixed step method failed.

From the definition of the weak error we know that it quantifies the convergence in the probability distribution of the actual and numerical solution. Figures 5C and 5D compare the distribution of the solution between SSA (actual) and the adaptive Milstein method with varying SDE tolerance. Results show that an acceptable convergence is observed for SDE tolerance values of or less. Concluding, both the mean and variance errors have to be small in order to have convergence in distribution. This fact is not always obvious from figures similar to Figures 5A and 5B.

Now we turn our attention to the effect of parameter p in the solution error. 10,000



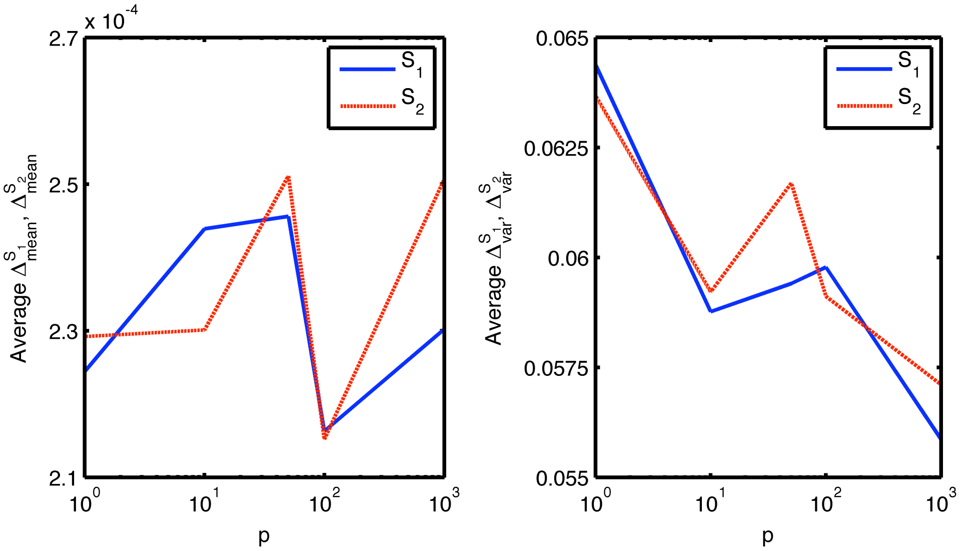
(a)

(b)

(c)

(d)

Figure 4.5: Solution of Eq. (4.25) using the adaptive scheme with variable SDE tolerance (SDE coeff = 1000, p=10 and $\Delta t_{\text{ini}} = 10^{-4}$ s). (a) Average normalized weak mean error of S_1 and S_2 for different error tolerances (log scale). (b) Average normalized weak variance error of S_1 and S_2 (log scale) for different error tolerances (log scale). (c) Comparison of the probability distribution of species S_1 between SSA and solutions with SDE tolerance of 10^{-3} and 10^{-4} at t = 0.005 s. (d) Comparison of the probability distribution of species S_1 between SSA and solutions with SDE tolerance 10^{-5} and 10^{-6} at t = 0.005 s.



(a)

(b)

Figure 4.6: Solution of Eq. (4.25) using the adaptive scheme with variable p values for constant initial time step, $\Delta t_{\text{ini}} = 10^{-4}$ s. (a) Average normalized weak mean error of S_1 and S_2 for varying p (log scale). (b) Average normalized weak variance error of S_1 and S_2 for varying p (log scale).

independent trajectories with initial time step of $\Delta t_{\text{ini}} = 10^{-4}$ s and SDE tolerance of 10^{-4} are used to obtain solutions with varying p. The values of p we tested are 1, 10, 50, 100 and 1,000. Recall that p determines the truncation order of the Fourier expansion of the 2D Itô Integral (cf. Eq. (4.8)). Average normalized weak mean and variance errors of S_1 and S_2 for the different values of p are plotted in Fig. 4.6. Figure 4.6(a) and4.4(b) reveal that the value of p does not have a significant effect on the weak mean and variance error for given initial Δt and given SDE tolerance. The argument is further supported by examining the corresponding distributions which differ only slightly. This result was more or less expected since we examine the behavior of the weak error. Both the Milstein and the Euler-Maruyama methods are of weak order 1.0. On the other hand the Euler-Maruyama is of strong order 0.5, while the Milstein is of strong order 1.0. The inclusion of the 2D Itô integral is the reason why the latter method has a higher strong order. Hence, when we compute the weak error, the value of p should not make any difference, which is indeed obvious, looking at Fig. 4.6. The value of p should

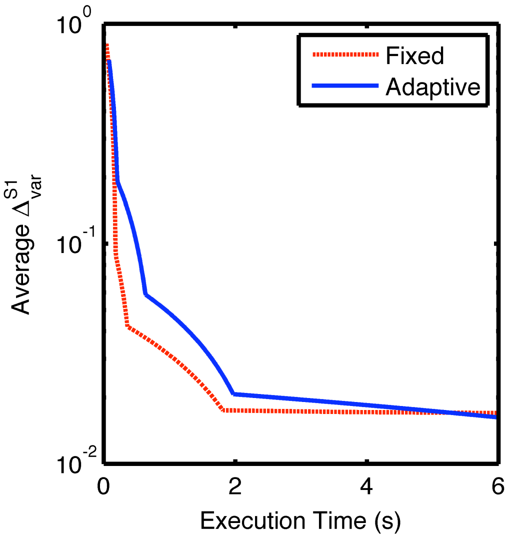


Figure 4.7: Comparison of execution times between the adaptive and fixed step methods for a single trial of the dimerization reaction system depicted in Table 4.1. Average normalized weak variance error of S_1 (log scale) as a function of the execution time. Both methods have p set to 10 and the adaptive scheme uses an initial time step of $\Delta t_{\text{ini}} = 10^{-4}$ s.

play a role when computing the strong error.

Finally, since computational costs are also important we briefly present and compare the simulations times. Note that simulation times refer to the overall time frame of 0.01 s. All realizations were obtained using dual-core 2.6 GHz AMD Opteron processors. In Fig. 4.7 we display how the simulation time of each trial depends on the weak variance error introduced by either the fixed step or adaptive Milstein method. We only display results using the variance since both methods yield similar behaviour and accuracy for the weak mean error and thus the error in the variance is more indicative. For the variable step size method we chose $\Delta t_{\text{ini}} = 10^{-4}$ s and $p = 10$.

From Fig. 4.7 we note that as the desired accuracy increases the two schemes converge in execution time. In general the fixed step method is slightly faster for a given error tolerance. We believe that this is directly correlated with the constant calculation of the two error criteria and the need to reapproximate the 2D Itô integral using a series of random number that add up to the execution time. In our opinion the latter is the main reason for the existing slow down. Obviously the anticipated speed up observed in the case of adaptive schemes in ODEs, where a substantial decrease in computational costs is the usual case, is not present. That was something we could have anticipated given the fact that adaptive methods are more intensive in the stochastic regime. However the adaptive scheme manages to integrate the system regardless of the initial time step, adding stability to the integrator. Finally, for comparison purposes we note that SSA required approximately 2.5 s per trial to simulate the dimerization reaction system depicted in Table 4.1. This makes both the fixed and adaptive step methods faster than SSA except if we use time steps less than 5.0×10^{-8} s for the fixed step method or SDE tolerance less than 10^{-4} for the adaptive method, which will not be necessary since an acceptable convergence is observed for larger values in both cases.

4.3.2 A System of Stiff, Nonlinear CLEs

As a second example we exploit a more elaborate one. Consider the reaction network depicted in Table 4.2. During the length of the simulation interval, of [0, 0.01] s, the stiff, nonlinear network of reactions satisfy the conditions presented in Eq. (4.17), which means that the systems time evolution can be described through a system of CLEs. The corresponding system of CLEs is

$$\begin{aligned} dX_i = {} & \big[\nu_{1i}\alpha_1\big(\underline{X}(t)\big) + \nu_{2i}\alpha_2\big(\underline{X}(t)\big) + \nu_{3i}\alpha_3\big(\underline{X}(t)\big) + \nu_{4i}\alpha_4\big(\underline{X}(t)\big) \\ & + \nu_{5i}\alpha_5\big(\underline{X}(t)\big) + \nu_{6i}\alpha_6\big(\underline{X}(t)\big)\big]dt + \nu_{1i}\sqrt{\alpha_1\big(\underline{X}(t)\big)}dW_1 \\ & + \nu_{2i}\sqrt{\alpha_2\big(\underline{X}(t)\big)}dW_2 + \nu_{3i}\sqrt{\alpha_3\big(\underline{X}(t)\big)}dW_3 + \nu_{4i}\sqrt{\alpha_4\big(\underline{X}(t)\big)}dW_4 \\ & + \nu_{5i}\sqrt{\alpha_5\big(\underline{X}(t)\big)}dW_5 + \nu_{6i}\sqrt{\alpha_6\big(\underline{X}(t)\big)}dW_6, \qquad i = 1, 2, 3, \end{aligned} \tag{4.26}$$

where X_i corresponds to the state of i^{th} species and the corresponding propensities are

$$\begin{aligned} \alpha_1\big(\underline{X}(t)\big) &= k_1X_1(t)X_2(t), & \alpha_2\big(\underline{X}(t)\big) &= k_2X_3(t), \\ \alpha_3\big(\underline{X}(t)\big) &= k_3X_1(t)X_3(t), & \alpha_4\big(\underline{X}(t)\big) &= k_4X_2(t), \\ \alpha_5\big(\underline{X}(t)\big) &= k_5X_2(t)X_3(t), & \alpha_6\big(\underline{X}(t)\big) &= k_6X_1(t), \end{aligned} \tag{4.27}$$

By substituting the kinetic parameters and initial conditions in the above equations we note that there is a five order separation in the reaction scales, denoted by the

Table 4.2: Reactions and Parameters for a Stiff, Nonlinear Network of Reactions

Set of Reactions	Mesoscopic Reaction Rates[†]	Initial Values[‡]
$S_1 + S_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} S_3$	$k_1 = 10^3$ / $k_2 = 10^3$	$\left[S_1\right]_0 = 10^3$
$S_1 + S_3 \underset{k_4}{\overset{k_3}{\rightleftharpoons}} S_2$	$k_3 = 10^{-5}$ / $k_4 = 10$	$\left[S_2\right]_0 = 10^3$
$S_2 + S_3 \underset{k_6}{\overset{k_5}{\rightleftharpoons}} S_1$	$k_5 = 1.0$ / $k_6 = 10^6$	$\left[S_3\right]_0 = 10^6$

[†] for 1^{st} order reactions the units are s^{-1} and for 2^{nd} order reactions the units are $\mathrm{molecules}^{-1}\mathrm{s}^{-1}$.

[‡] initial values are in number of molecules.

propensities values.

Again we use SSA as our actual solution. We simulate the system on the time interval [0, 0.01] s and conduct 10,000 independent trials. For brevity and clarity we only present the results for S_1 (cf. Fig. 4.8). Similar behavior is observed for S_2 and S_3 where both their values fluctuate over time around their initial conditions. Finally, the mean and the variance evaluated through SSA are used for the comparison and the evaluation of the adaptive scheme.

Figure 4.9 examines how well the fixed step method integrates the system of CLEs. The 2D Itô integral is approximated with parameter p set to 10. We run again 10,000 independent trials and calculate the solution in the interval [0, 0.01] s. For comparison with the actual solution (cf. Fig. 4.8) we calculate the average normalized weak mean and variance errors of all species using Eq. (4.23). The results are presented in Fig. 4.9. As in the previous example, the fixed step method fails to integrate the system for large time step values. In particular we could not obtain solutions for step size larger than 5.0×10^{-7} s. This occurs because species concentrations attain negative values. Notice that the behaviour of the scheme is similar to the one of the previous example, meaning that the error in the variance is the one mostly affected by the decrease in the time step (cf. Fig. 4.9(b)), while the normalized weak mean error remains for practical purposes the same (cf. Fig. 4.9(a)). A notable exception is the behavior of S_3 which appears to have a very small error in the mean. This can be explained if we take into the account

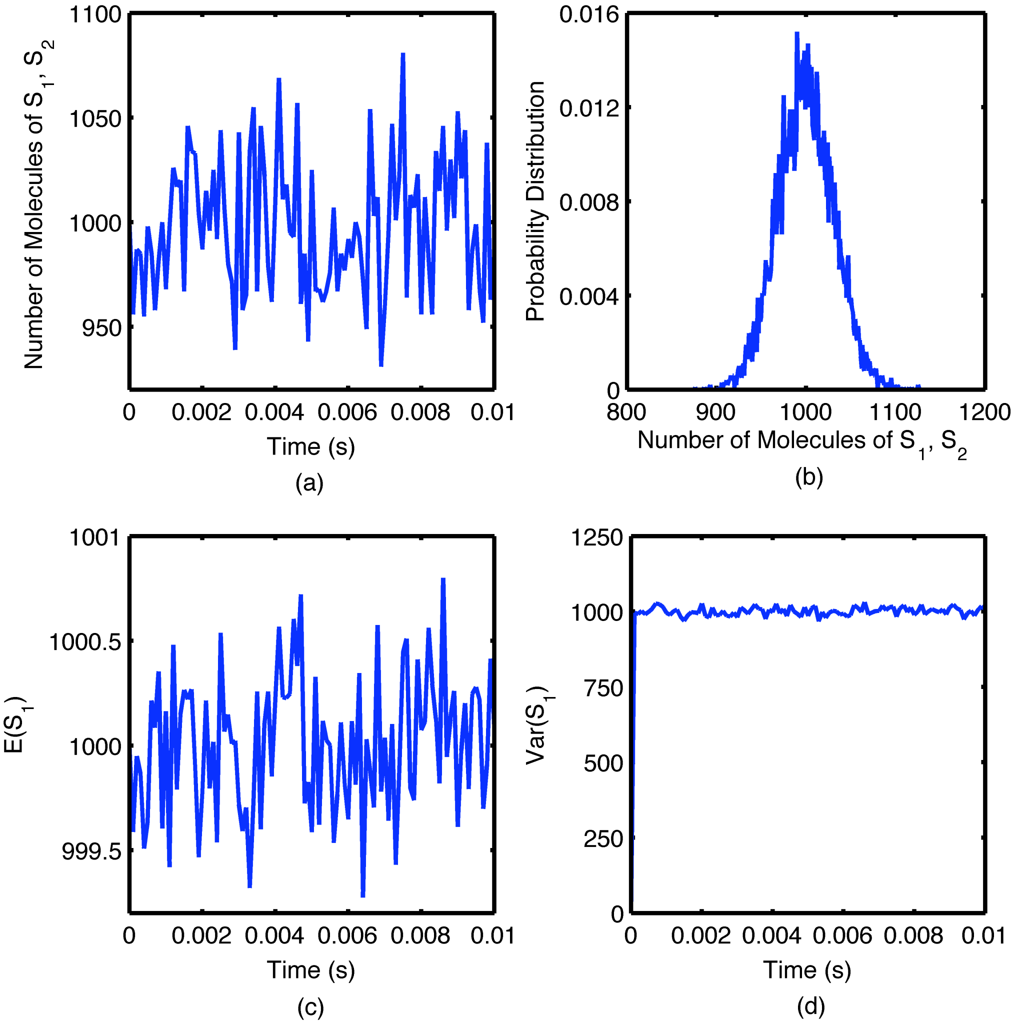
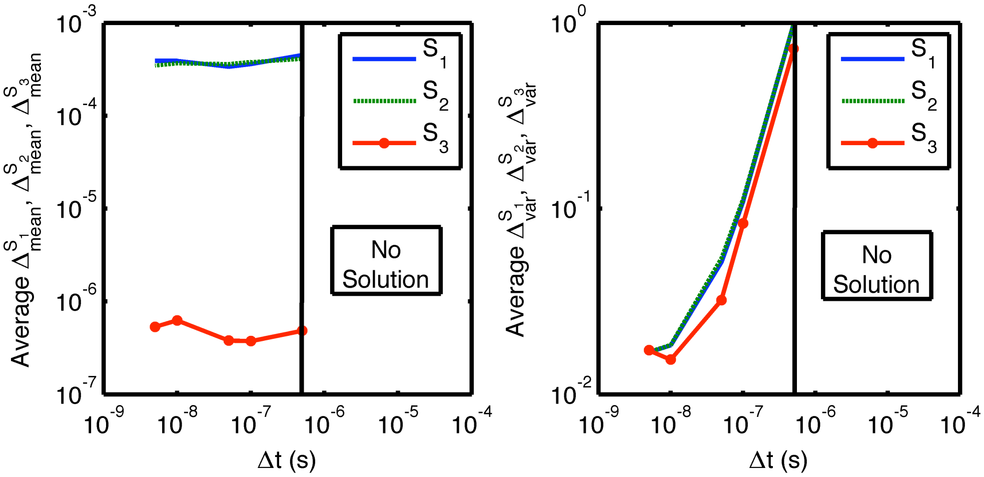


Figure 4.8: Solution of Eq. (4.26) using SSA in the interval [0, 0.01] s.] (a) A sample trajectory of the number of S_1 molecules. (b) The probability distribution of species S_1 at $t = 0.005$ s. (c) Mean number of S_1 molecules averaged over 10,000 trials. (d) Variance of number of S_1 molecules averaged over 10,000 trials.



(a)

(b)

Figure 4.9: Solution of Eq. (4.26) using fixed step Milstein method ($p = 10$). (a) Comparison of the average normalized weak mean error of S_1, S_2 and S_3 (log scale) generated by the fixed step (log scale) Milstein method. (b) Comparison of the average normalized weak mean variance of S_1, S_2 and S_3 (log scale) generated by the fixed step (log scale) Milstein method.

the large, in comparison with the other species, initial and hence equilibrium values. Its large concentrations allow for a small influence by the noisy environment, meaning it is minimally affected by the evolution of the noise terms.

Subsequently we look into the accuracy of the proposed scheme. We examine the effect on the user defined parameter SDE tolerance in the accuracy of the solution. SDE coeff is kept at 1,000. Results are shown in Fig. 4.10, obtained from 10,000 independent trajectories and fixed initial time step of $\Delta t_{\text{ini}} = 10^{-4}$ s and $p = 10$. The values of SDE tolerance range from 10^{-2} to 10^{-5}. We calculate the average normalized weak mean and variance errors of all species and look into the probability distribution of species S_1 at $t = 0.005$ s. Obviously as we decrease the error tolerance the weak variance error decreases. The decrease is approximately one order of magnitude between SDE tolerance 10^{-2} to 10^{-5} (cf. Fig. 4.10(b)). For the weak mean error we observe small if any decrease in its value as SDE tolerance decreases. It is noteworthy to point out that the adaptive scheme achieves to integrate the system even though it uses a high initial time step if compared with the fixed step counterpart. Finally, from Fig. 4.10(c) and

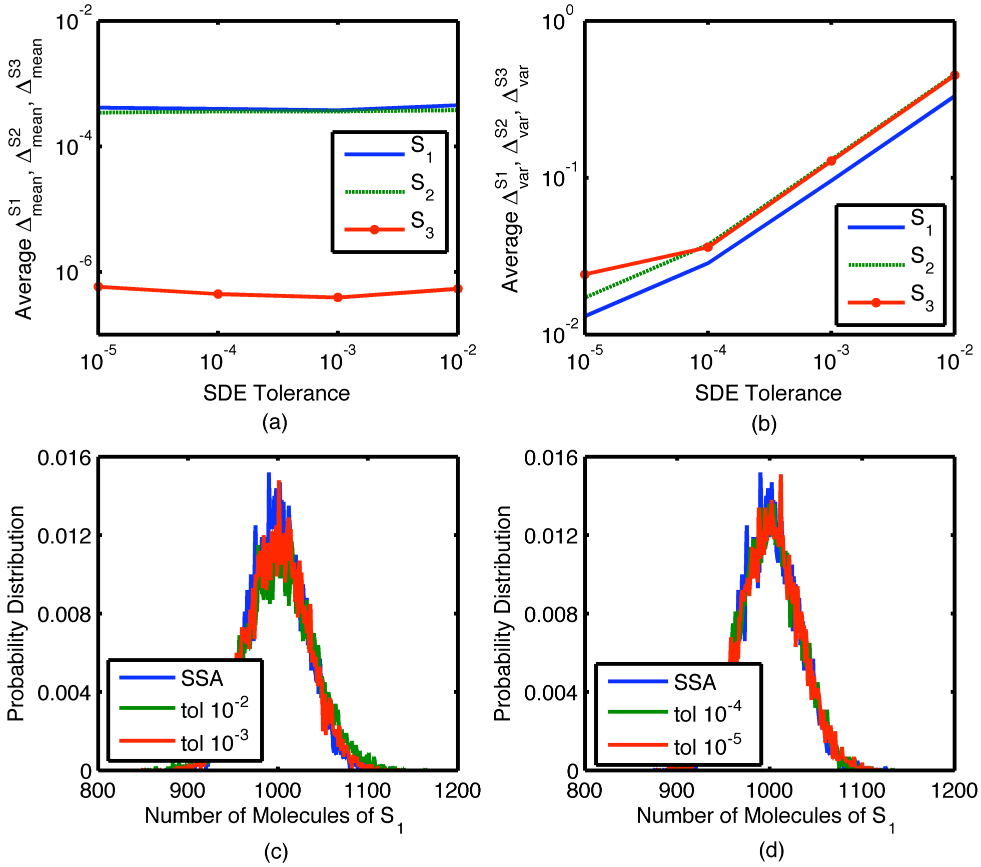


Figure 4.10: Solution of Eq. (4.26)) using the adaptive scheme with variable SDE tolerance (SDE coeff = 1,000, $p = 10$ and $\Delta t_{\text{ini}} = 10^{-4}$ s). (a) Average normalized weak mean error of S_1, S_2 and S_3 (log scale) for different error tolerances (log scale). (b) Average normalized weak variance error of S_1, S_2 and S_3 (log scale) for different error tolerances (log scale). (c) Comparison of the probability distribution of species S_1 between SSA and solutions with SDE tolerance of 10^{-2} and 10^{-3} at $t = 0.005$ s. (d) Comparison of the probability distribution of species S_1 between SSA and solutions with SDE tolerance 10^{-4} and 10^{-5} at $t = 0.005$ s.

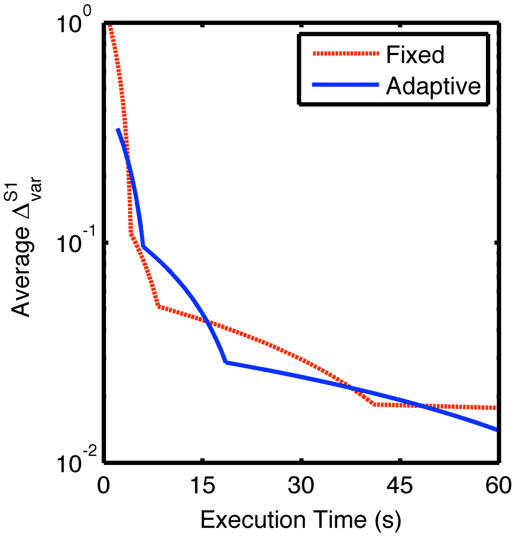


Figure 4.11: Comparison of execution times between the adaptive and fixed step methods for a single trial of the stiff, nonlinear reaction system depicted in Table 4.2. Average normalized weak variance error of S_1 (log scale) as a function of the execution time. Both methods have p set to 10 and the adaptive scheme uses an initial time step of $\Delta t_{\text{ini}} = 10^{-4}$ s.

4.10(d) we infer that an acceptable convergence in distribution is feasible for values of SDE tolerance equal to or less than .

Figure 4.11 shows a different behavior from the one noted in Fig. 4.7. In this example both approaches seem to produce results with the same accuracy while approximately requiring the same computational effort. This means that there is no apparent advantage of the fixed step scheme over the adaptive. We believe that this is the case because from a two variable system in the first example we went to a three variable system which is also nonlinear that allowed the adaptive scheme to gain slightly over the fixed step method. Nonetheless, perhaps the important benefit is larger time steps while retaining stability. Finally, it is interesting to note that SSA required approximately 65.81 s per trial to simulate the system of stiff, nonlinear reactions depicted in Table 4.2. Apparently, both methods are faster except if we use very small values for either the time step of the fixed method or SDE tolerance of the adaptive method.

4.3.3 An Actual Stiff Biochemical Network

The third example involves a larger chemical kinetics network, in particular an actual biochemical network that experiences stiffness. The reason we choose to use this last example is to highlight the advantages of using the present algorithm in conjunction with our Hybrid multiscale algorithm, called Hy3S (Hybrid Stochastic Simulations for Supercomputers) capable of fast and accurately simulate in time biochemical networks that are far from the thermodynamic limit [1]. Hy3S is based on a hybrid approach that separates the reactions into two subsets, fast/continuous and slow/discrete. The first are propagated in time using the chemical Langevin equation (CLE) and the later using differential jump equations. When the underlying biological network is stiff the fixed step integration in Hy3S fails, mainly because species populations become negative. Therefore we incorporated into Hy3S our adaptive time step selection scheme in order to add stability and better error control in the time integration process.

As an example we use a previously studied stochastic Petri model proposed by Srivastava *et al.* that quantifies the heat shock response of *E. coli* [88]. The model involves 17 linear and nonlinear reactions with 14 participating species. The reactions with their corresponding kinetic parameters are presented in Table 4.3, while the initial values of each species are depicted in Table 4.4. The volume of the cell is considered to be $V = 1.5 \times 10^{-15}$ l. All the kinetic and initial data are chosen in accordance with Ref. [89] except for the initial value for species DnaJ where we followed an approach similar to Ref. [68] in order for the system to be further from the equilibrium state. As it becomes apparent from Tables 4.3 and 4.4 not all reactions can be initially (and during the course of the simulation) classified as fast (cf. Eq. (4.17)), hence not all species are propagated using CLEs. Reactions (15)-(17) are the ones classified as fast throughout the course of the simulation. This is the shortcoming of using an actual biological example since in nature slow and fast reactions coexist, interact and control the rates of each other. Still the existence of fast reactions will allow examining the behavior of the present algorithm in the context of Hy3S.

As in the previous two examples we use SSA as our actual solution. We simulate the system on the time interval [0, 100] s and conduct 1,000 independent trials. Again the mean and the variance evaluated through SSA are used for the comparison and the evaluation of the adaptive scheme. Note that we conduct 10 times fewer trials than in

Table 4.3: Reactions that Quantify the Heat Shock Response of *E. coli* According to Ref. [89].

#	Reaction	Kinetic Parameter †
1	$DNA \cdot \sigma^{32} \longrightarrow mRNA \cdot \sigma^{32}$	1.4×10^{-3}
2	$mRNA \cdot \sigma^{32} \longrightarrow \sigma^{32} + mRNA \cdot \sigma^{32}$	7.0×10^{-2}
3	$mRNA \cdot \sigma^{32} \longrightarrow \varnothing$	1.4×10^{-6}
4	$\sigma^{32} \longrightarrow RNAP\sigma^{32}$	0.7
5	$RNAP\sigma^{32} \longrightarrow \sigma^{32}$	0.13
6	$DNA \cdot DnaJ + RNAP\sigma^{32} \longrightarrow DnaJ + DNA \cdot DnaJ + \sigma^{32}$	4.41×10^{-6}
7	$DnaJ \longrightarrow \varnothing$	6.4×10^{-10}
8	$\sigma^{32} + DnaJ \longrightarrow \sigma^{32} \cdot DnaJ$	3.27×10^{-5}
9	$\sigma^{32} \cdot DnaJ \longrightarrow \sigma^{32} + DnaJ$	4.4×10^{-4}
10	$DNA \cdot FtsH + RNAP\sigma^{32} \longrightarrow FtsH + DNA \cdot FtsH + \sigma^{32}$	4.41×10^{-6}
11	$FtsH \longrightarrow \varnothing$	7.4×10^{-11}
12	$\sigma^{32} \cdot DnaJ + FtsH \longrightarrow DnaJ + FtsH$	1.28×10^{3}
13	$DNA \cdot GroEL + RNAP\sigma^{32} \longrightarrow GroEL + DNA \cdot GroEL + \sigma^{32}$	5.69×10^{-6}
14	$GroEL \longrightarrow \varnothing$	1.8×10^{-8}
15	$Protein \longrightarrow Unfolded\ Protein$	0.2

Continued on Next Page...

Table 4.3: Reactions that quantify the heat shock response of *E. coli* according to Ref. [89]. – Continued

#	Reaction	Kinetic Parameter †
16	$DnaJ + Unfolded\ Protein \longrightarrow DnaJ \cdot Unfolded\ Protein$	9.7256×10^6
17	$GroEL \longrightarrow \varnothing$	0.2

† for 1^{st} order reactions the units are s^{-1} and for 2^{nd} order reactions the units are $M^{-1}s^{-1}$

Table 4.4: Initial Values of Species Involved in the Heat Shock Model of *E. coli* according to Ref. [89].

#	Species	Initial Values†
1	$DNA \cdot \sigma^{32}$	1
2	$mRNA \cdot \sigma^{32}$	17
3	σ^{32}	15
4	$RNAP\sigma^{32}$	76
5	$DNA \cdot DnaJ$	1
6	$DNA \cdot FtsH$	0
7	$DNA \cdot GroEL$	1
8	$DnaJ$	4640
9	$FtsH$	200
10	$GroEL$	4314
11	$DnaJ \cdot Unfolded\ Protein$	5.0×10^6
12	$Protein$	5.0×10^6
13	$\sigma^{32} \cdot DnaJ$	2959
14	$Unfolded\ Protein$	2.0×10^6

† initial values are in number of molecules.

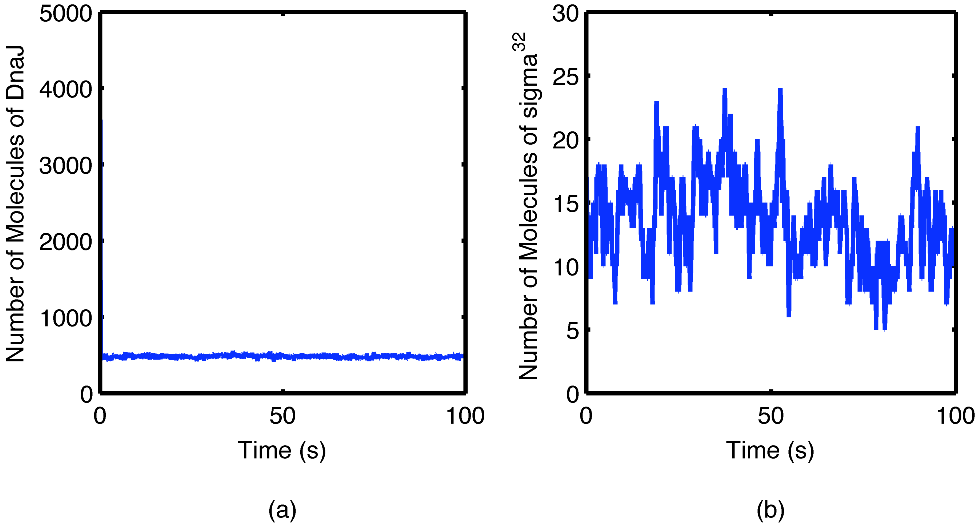


Figure 4.12: Solution of the heat shock response model using SSA in the interval [0, 100] s. (a) A sample trajectory of the number of DnaJ molecules. (b) A sample trajectory of the number of σ^{32} molecules.

the previous two examples. In order to study the effect of the algorithm on a bigger time frame, 100 s compared to 0.01 s, we had to reduce the number of total trials. Still the number of trials is enough to produce an accurate sampling of the underlying distribution. Our results will be reported based on two species, DnaJ and σ^{32}. The first is chosen because it is involved in fast reactions and the second because it is only affected by slow reactions, hence we will be able to judge whether or not the algorithm introduces error in both the fast and slow subspaces. A sample trajectory for both species is shown in Fig. 4.12. Note in Fig. 4.12(a) the steep initial decrease in the concentration of DnaJ.

Similar to the previous two examples the fixed step Milstein method fails to integrate the system when the integration time step is larger than 10^{-4} s. This is shown in Fig. 4.13 where the average normalized weak mean and variance errors for different time steps are compared for the two species. All results were obtained through 1,000 independent trials with the parameter p set at 10. Focusing on the results for DnaJ we note that as the time step decreases the error in the mean remains practically unaltered

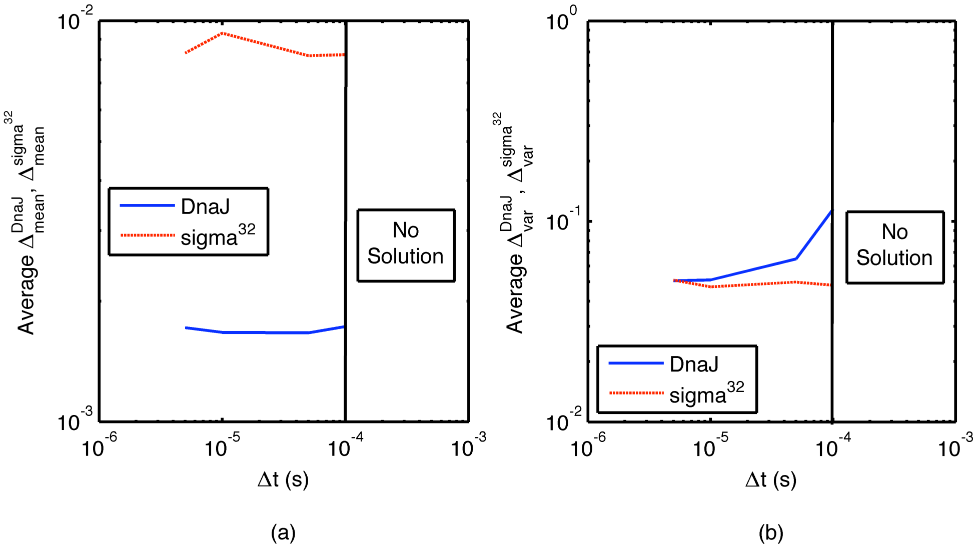
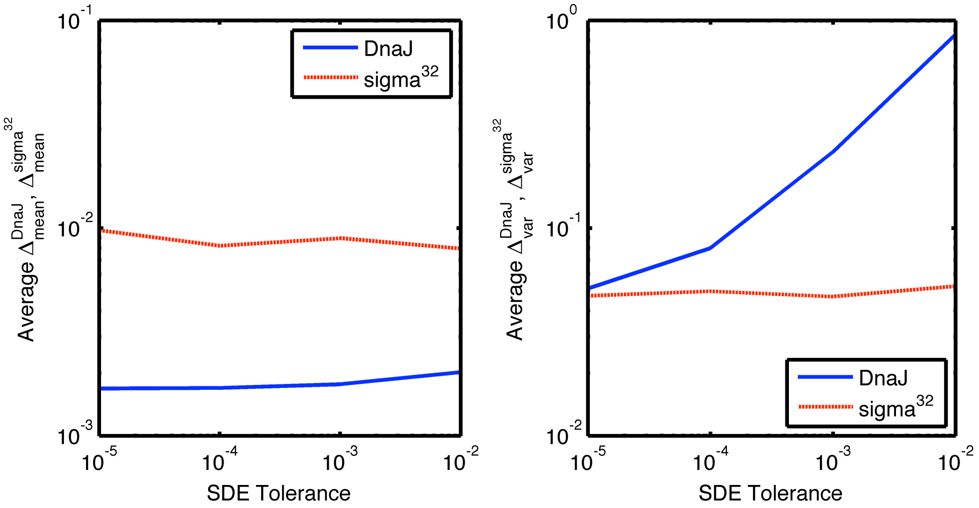


Figure 4.13: Solution of the heat shock response model using the fixed step Milstein method ($p = 10$). (a) Comparison of the average normalized weak mean error of DnaJ and σ^{32} (log scale) generated by the fixed step (log scale) Milstein method. (b) Comparison of the average normalized weak mean variance of DnaJ and σ^{32} (log scale) generated by the fixed step (log scale) Milstein method.

while the error in the variance decreases, a scenario observed in the previous two examples. On the other hand the error in both the mean and variance for σ^{32} remain the same as the time step changes. This implies that the time step used in the integration of the fast species does not affect the propagation of the slow species. The last observation is something we anticipated and also expect to see when instead of the fixed step we use the adaptive method.

By substituting the fixed step integrator with the adaptive one the results report a similar performance but instead of the varying time step we vary the SDE tolerance. Figure 4.14 shows the results obtained from 1,000 independent trials using a fixed initial time step of $\Delta t_{\text{ini}} = 0.1$ s, $p = 10$ and SDE coeff $= 1{,}000$. We calculate the average normalized weak mean and variance errors of the two species of interest using values of SDE tolerance that range from to 10^{-2} to 10^{-5}. As in Fig. 4.13 the error in the mean is mainly unaffected for both DnaJ and σ^{32}, while a decrease is only observed in the variance error for σ^{32} as SDE tolerance decreases. Importantly we note that first the



(a)

(b)

Figure 4.14: Solution of the heat shock response model using the adaptive scheme with variable SDE tolerance (SDE coeff $= 1000$, $p = 10$ and $\Delta t_{\text{ini}} = 0.1$ s). (a) Average normalized weak mean error of DnaJ and σ^{32} (log scale) for different error tolerances (log scale). (b) Average normalized weak variance error of DnaJ and σ^{32} (log scale) for different error tolerances (log scale).

adaptive scheme does not affect the propagation of the slow species, since σ^{32} mean and variance errors are not affected. And second the adaptive algorithm manages to integrate the system starting from an initial time step of $\Delta t_{\text{ini}} = 0.1$ s. Comparing the probability distributions obtained through SSA and Hy3S we report that an acceptable convergence is obtained for SDE tolerance equal or less to 10^{-4}.

Finally reporting on the execution times we note that the adaptive scheme requires more time to produce accurate results. Results shown in Fig. 4.15 resemble those in Fig. 4.7. We think this is because the complexity of the system of CLEs in this example, two linear and one nonlinear reactions, is closer to that of the first example and hence the same reasons apply. Still the adaptive scheme retained its stability by integrating the system even though the starting time step was large. As an additional comment we want to point out the use of the variance error of DnaJ to report the execution times, as it is the most indicative.

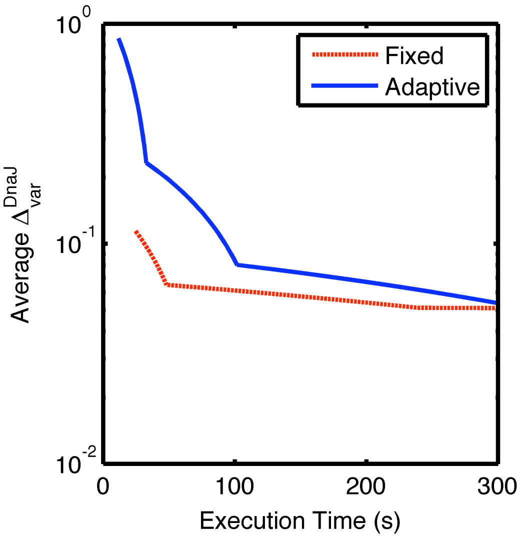


Figure 4.15: Comparison of execution times between the adaptive and fixed step methods for a single trial of the heat shock response model depicted in Table 4.3. Average normalized weak variance error of DnaJ (log scale) as a function of the execution time. Both methods have p set to 10 and the adaptive scheme uses an initial time step of $\Delta t_{\text{ini}} = 0.1$ s.

4.4 Summary

We have demonstrated that the proposed adaptive scheme can produce accurate results when stiff systems of Itô SDEs with multiple multiplicative noise arise, such as the system of CLEs in Eq. (4.25) or (4.26). The SDE tolerance, a user defined parameter, can be used to tune the scheme and balance precision with simulation times. The use of the weight coefficient (SDE coeff) allows balancing the importance between the drift and diffusion error controls. The higher the SDE coeff value the less important the drift error becomes and vise versa.

We also noted that the proposed scheme is more stable than its fixed step counterpart. While the fixed step method fails to produce results, because of numerical instabilities, the adaptive scheme is able to produce stable solutions even when the initial time step is large. This feature becomes important when trying to integrate systems of SDEs for which we do not know *a priori* whether or not they are stiff or if they become during the course of integration. The adaptive scheme adds the necessary stability in the integration scheme allowing the method to avoid incorrect integration paths.

In terms of computational efficiency the developed adaptive time step method slightly underperforms compared to the fixed time step method. In the first and third examples the margin of the fixed step method was present but not significant while in the second example, fixed and adaptive scheme produced the same accuracy in the same amount of computational time. This means that an adaptive time step scheme is considerably more stable than fixed step analogues with no excessive additional computational overhead. It is interesting to note that in all the examples the adaptive scheme is initialized with a time step that is far from the optimum time step of each SDE tolerance value. In general, the larger the distance between the two the more computational intensity is necessary.

In conclusion, we developed an adaptive scheme which numerically solves stiff systems of SDEs with multiple multiplicative noise by appropriately adjusting the time step, dynamically decreasing the time step of the numerical integration when stiffness exists, but dynamically increasing it when the system is no longer stiff. While this procedure adds stability in the integration algorithm it does not appear to have a significant negative impact on the speed up of the fixed Milstein method. This is important since for many applications the stability part of an algorithm is a crucial component of the modeling effort. The presented algorithm has been embedded successfully in a Hybrid multiscale algorithm we have developed, called Hy3S, used to propagate in time chemical kinetics models that are far from the thermodynamic limit [1]. Hy3S is available for download at `http://hysss.sourceforge.net/`.

Chapter 5

Model Reduction of Multiscale Chemical Langevin Equations: A Numerical Case Study

5.1 Introduction

Stochastic kinetic models are used to accurately represent the inherent probabilistic nature of systems of chemical or biochemical reaction networks that are far from the thermodynamic limit [90, 91, 74, 27, 29, 30, 2, 32, 37].

In this work we focus on networks of reactions that can be approximated as continuous Markov processes and are modeled through systems of chemical Langevin equations (CLEs). These are reactions that occur frequently in a given time interval and have participating species with relatively large concentrations [65]. For example, fast protein dimerization reactions where the monomer and dimer appear in concentrations of over 100 molecules each in a volume approximately the size of a small bacterial cell can be modeled by CLEs [64]. More precisely, the stochastic dynamics of such systems are governed by Fokker-Planck equations. Instead of solving the Fokker-Planck equations it is usually more convenient to sample the underlying probability distribution through trajectories obtained as solutions of the corresponding CLE or systems of CLEs [65].

CLEs are Itô stochastic differential equations (SDEs) with multiple multiplicative noise terms and like ordinary or partial differential equations their solution can be a stiff numerical problem whenever the underlying physical system exhibits multiple-time scales. One possible way to overcome stiffness in such systems is to efficiently adjust the integration time step using an adaptive time stepping algorithm (cf. Chapter 4) [3]. An alternative route for addressing multiple-time scales in a system of CLEs is through model reduction. While model reduction techniques are well studied for deterministic kinetics models, modeled by ordinary differential equations (ODEs) [92, 93, 94, 95, 96, 97], the same is not true for the stochastic regime and in particular for CLEs. In the stochastic framework, efforts have concentrated in the jump Markov process regime either by reduction of the chemical master equation [98, 99, 100, 101] or by using the quasi-steady state approximation to eliminate fast occurring reactions [61, 69, 67, 66]. Recently, Dong and coworkers proposed a reduction approach for CLEs where the reduction methodology is based on the corresponding system of ODEs [102].

In this chapter we describe and illustrate the application of a reduction framework for multi-scale systems of CLEs [103]. The proposed framework is semi-analytical and is based on a three-step systematic procedure that appropriately reduces the initial system of CLEs to subsystems with similar time scales. The first step entails the derivation of a linear transformation which decomposes the initial state vector of the CLE system into fast and slow varying variables [104]. This allows for a non-stiff description by treating each set differently. A set of sufficient and necessary conditions emerges, which must be met to ensure the existence of the appropriate transformation. The second step is to treat each of the two subsets independently by applying the method of adiabatic elimination to the systems under consideration [105]. Fast variables are assumed to relax to a pseudo-stationary density under the hypothesis that the slow variables remain constant. Slow variables are approximated through a Fokker-Planck equation which governs their probability density. Ultimately the distribution of the slow variables can be sampled through the solutions of a system of CLEs that correspond only to the slow subspace and can be integrated with large integration steps. The final step is to compute the approximated solution of the initial CLEs system by simply multiplying the two independent probability densities.

The chapter is organized as follows. First we briefly present background information on continuous Markov processes and their governing equation, the chemical Langecin equations. Then we present a motivating example that lies in the continuous Markov process regime. Then the reduction framework is formulated theoretically. We finally test the numerical accuracy and demonstrate the computational efficiency of the proposed scheme.

5.2 Continuous Markov Processes: Chemical Langevin Equations

Consider a system of N distinct chemical species, X_i $(i = 1, \ldots, N)$, participating in M chemical reactions in a well-mixed bacterial size volume V

$$\sum_{i=1}^{N} r_i^j X_i \xrightarrow{k_j} \sum_{i=1}^{N} p_i^j X_i, \qquad j = 1, \ldots, M \tag{5.1}$$

The corresponding system of chemical Langevin equations (CLEs) under the assumption that the system of reactions can be described as a continuous Markov process [3] is

$$d\underline{X} = \sum_{j=1}^{M} \underline{\nu}_j k_j c_j(\underline{X}(t))dt + \sum_{j=1}^{M} \underline{\nu}_j \sqrt{k_j c_j(\underline{X}(t))}dW_j, \tag{5.2}$$

where $\underline{X} = \left[\begin{array}{ccc} X_1 & \cdots & X_N \end{array}\right]^T$ is the state vector of the system containing the concentration of the N species, k_j corresponds to the mesoscopic reaction rate of the j^{th} reaction, $\underline{W} = \left[\begin{array}{ccc} W_1 & \cdots & W_M \end{array}\right]^T$ is an $M-$dimensional Wiener process, $\underline{\nu}_j$ and c_j denote the stoichiometric vector associated with the j^{th} reaction and the number of distinct combinations of the reacting species participating in the j^{th} reaction respectively and are defined as follows

$$\underline{\nu}_j = \begin{bmatrix} p_1^j - r_1^j \\ \vdots \\ p_N^j - r_N^j \end{bmatrix} \qquad c_j(\underline{X}) = \prod_{i=1}^{N} \frac{X_i!}{r_i^j\,!\,\left(X_i - r_i^j\right)!} \tag{5.3}$$

The product of $\alpha_j(\underline{X}) = k_j c_j(\underline{X})$ is usually referred to as the reaction propensity and corresponds to the probabilistic reaction rate of the j^{th} reaction.

For any reaction to be considered as a continuous Markov process the following two conditions must be met [64]

$$\alpha_j(\underline{X}) \geq \lambda \gg 1 \qquad X_i > \epsilon|\nu_{ji}| \tag{5.4}$$

The two parameters λ and ϵ define respectively the numbers of reactions occurring within time Δt and the lower limit in the number of molecules of the reactants and products for the effect of a reaction to be negligible. In practice, typical values for λ and ϵ can be empirically determined for computational efficiency and acceptable accuracy [64]. In the present work we consider $\lambda = 10$ and $\epsilon = 100$.

5.3 Motivating Example

Let us consider the network of reactions shown in Table 5.1 taking place in a bacterial-sized volume of 10^{-15} L. Species A, B can represent monomer proteins that fuse to form a dimer protein C that in turn fuses with D to form multimer E. The first reversible reaction is assumed to be much faster in both directions than the second one. Note that k_i's are the macroscopic reaction rates. Given both the initial conditions and the kinetic parameters (cf. Table I) one can infer that conditions in Eq. (5.4) are met for all reactions and therefore the system lies entirely in the continuous Markov process regime both initially and until the equilibrium state is reached as it will become evident below. The corresponding system of CLEs for all five species is

$$\begin{aligned}
dX_A &= (-k_1X_AX_B + k_2X_C)dt - \sqrt{k_1X_AX_B}dW_1 + \sqrt{k_2X_C}dW_2 \\
dX_B &= (-k_1X_AX_B + k_2X_C)dt - \sqrt{k_1X_AX_B}dW_1 + \sqrt{k_2X_C}dW_2 \\
dX_C &= (k_1X_AX_B - k_2X_C - k_3X_CX_D + k_4X_E)dt+ \\
&\quad + \sqrt{k_1X_AX_B}dW_1 - \sqrt{k_2X_C}dW_2 - \sqrt{k_3X_CX_D}dW_3 + \sqrt{k_4X_E}dW_4 \\
dX_D &= (-k_3X_CX_D + k_4X_E)dt - \sqrt{k_3X_CX_D}dW_3 + \sqrt{k_4X_E}dW_4 \\
dX_E &= (k_3X_CX_D - k_4X_E)dt + \sqrt{k_3X_CX_D}dW_3 - \sqrt{k_4X_E}dW_4
\end{aligned} \tag{5.5}$$

Using Hy3S [1], a suite of multiscale algorithms that use CLEs to propagate systems of reactions that belong in the continuous Markov process regime, or SynBioSS [4], a

Table 5.1: Reactions and Parameters for the Motivating Example

Set of Reactions	Mesoscopic Reaction Rates[†]	Initial Values[‡]
	$k_1 = 4.981$	$[A]_0 = 2800$
$A + B \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C$	$k_2 = 1.5 \times 10^2$	$[B]_0 = 2500$
$C + D \underset{k_4}{\overset{k_3}{\rightleftharpoons}} E$	$k_3 = 1.66 \times 10^{-5}$	$[C]_0 = 1600$
	$k_4 = 4 \times 10^{-3}$	$[D]_0 = 1900$
		$[E]_0 = 3000$

[†] for 1^{st} order reactions the units are s^{-1} and for 2^{nd} order reactions the units are $\mathrm{molecules}^{-1}\mathrm{s}^{-1}$.

[‡] initial values are in number of molecules.

cross-platform and user friendly version of Hy3S, we simulated Eq. (5.5) in the interval [0, 200.0] s and run $2,000$ independent trials for accurate statistical sampling. We used the fixed step Euler-Maruyama method [73] as our integration method and set the initial time step to 0.1 s. Every time the integration failed we decreased the time step using the following series of time steps 0.1 s, 0.05 s, 0.01 s, 5×10^{-3} s, 1×10^{-3} s,...etc; 5×10^{-5} s was the first time step for which the integration did not fail, i.e. species populations did not attain negative values. The selected integration time step provided relative accurate results compared to the SSA. Average normalized weak mean and variance errors [3] are relatively small for all species. Reducing the time step only impacts the variance error of the fast evolving species while the corresponding errors for the slow varying species seem to be invariant. Simulation results are shown in Fig. 5.1. In the remainder of the present chapter whenever we state that the integration fails for larger time steps it is assumed that we followed a similar approach to arrive at the selected time step.

Looking at the time axes, it is apparent in Fig. 5.1 that species evolve over time affected by the two different time scales. In particular species A and B are mainly affected by the fast dynamics and reach very quickly (less than 0.01 s) what appears to be a pseudo-steady state (cf. Fig. 5.1(a)) or more accurately a pseudo-stationary

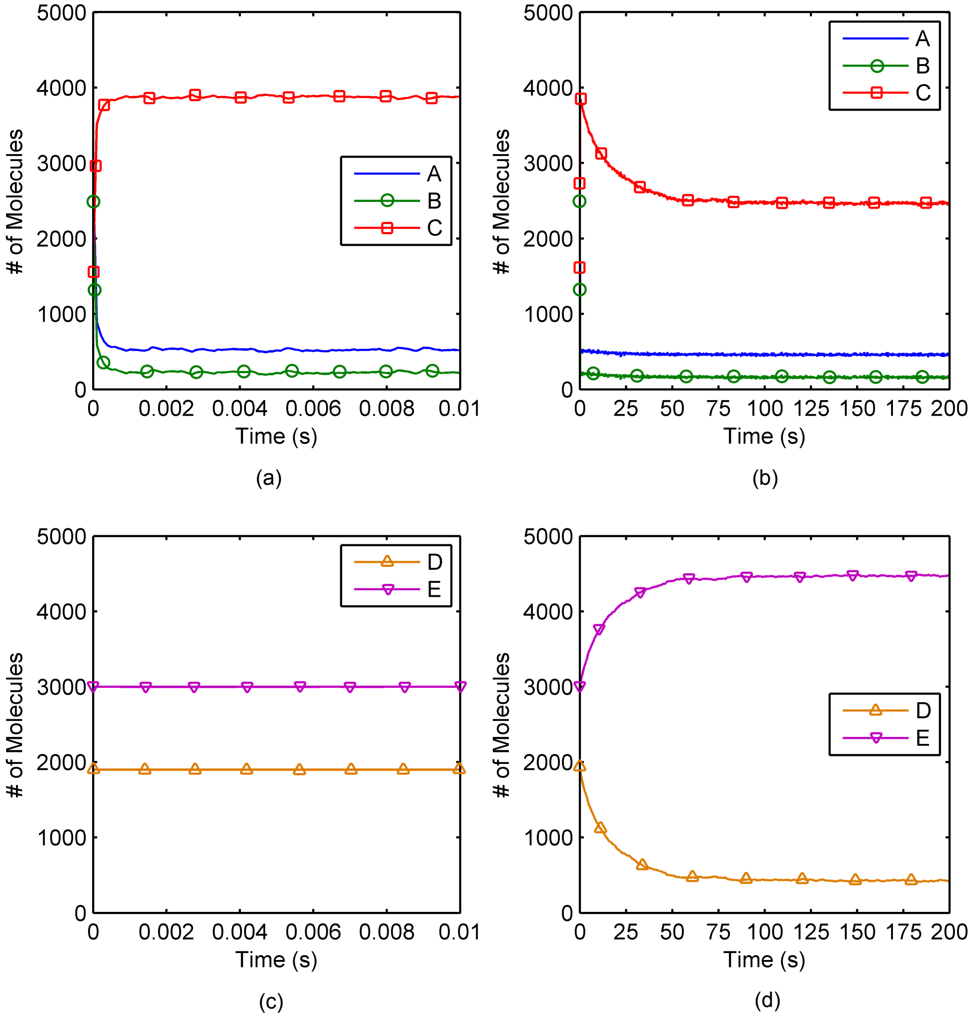


Figure 5.1: A simulated trajectory of system (5.5) using the fixed step Euler-Maruyama method with time step 5×10^{-5} s. (a) Time evolution of species A, B and C in the first 0.01 s. (b) Time evolution of species A, B and C in the time interval [0, 200] s. (c) Time evolution of species D and E in the first 0.01 s. (d) Time evolution of species D and E in the time interval [0, 200] s.

distribution. On the other hand, species D and E are mainly influenced by the slow reactions and in the time interval of 0.01 s their concentrations are practically unaltered (cf. Fig. 5.1(c)). Species C seems to be affected equally by both the fast and slow dynamics. Observing Fig. 5.1(b) and 5.1(d) we conclude that the system practically reaches its final equilibrium distribution after 150 s. Lastly, we want to point out that during the chosen time interval all reactions indeed satisfy Eq. (5.4).

Simulating the system of stochastic differential equations depicted in Eq. (5.5) requires a small integration time step since the existence of stiffness causes species concentration to become negative if a relatively large time steps is used. For the large time interval that the system requires reaching equilibrium the computational cost is significant. Consequently a reduction framework that will identify the minimal number of fast and slow variables and will propagate them separately in time will in principle improve the efficiency of integration.

5.4 Problem Formulation

If multiple time scales are present in the system of reactions (cf. Eq. (6.28)), then in Eq. (5.2) the state variables are affected by reaction rates of both fast and slow reactions resulting in a stiff numerical problem. In the majority of cases, there is no intuitive way to distinguish between slow and fast variables. Therefore we seek a coordinate change that will transform the original system in a new singularly perturbed SDE system where fast and slow variables are explicitly identified; in equation form we want to transform the initial system of CLEs (cf. Eq. (5.2)), to the following form

$$\begin{aligned} d\underline{Y}_s &= \underline{D}_s(\underline{Y}_s,\underline{Y}_f)dt + \underline{\underline{E}}_s(\underline{Y}_s,\underline{Y}_f)d\underline{W}_s \\ d\underline{Y}_f &= \frac{1}{\delta}\underline{D}_f(\underline{Y}_s,\underline{Y}_f)dt + \frac{1}{\sqrt{\delta}}\underline{\underline{E}}_f(\underline{Y}_s,\underline{Y}_f)d\underline{W}_s, \end{aligned} \tag{5.6}$$

where matrices $\underline{D}_s(\underline{Y}_s,\underline{Y}_f)$, $\underline{D}_f(\underline{Y}_s,\underline{Y}_f)$, $\underline{\underline{E}}_s(\underline{Y}_s,\underline{Y}_f)$ and $\underline{\underline{E}}_f(\underline{Y}_s,\underline{Y}_f)$ will be explicitly identified in the remainder of the paper. In Eq. (5.6), fast, $\underline{Y}_f$, and slow, $\underline{Y}_s$, variables are identified, but are still coupled through matrices $\underline{D}_s$, $\underline{D}_f$, $\underline{\underline{E}}_s$ and $\underline{\underline{E}}_f$. Decoupling them will allow for a non-stiff description and enable to treat each set differently.

5.5 Model Reduction Framework

The idea of the model reduction framework is to reduce the initial system of CLEs to subsystems with similar time scales and then treat each one independently [103]. We next highlight the steps of identifying an appropriate coordinate transformation, of decoupling the two subsets and of approximating the probability distribution of the initial system.

5.5.1 Identifying Fast and Slow Variables

Assume that there are p fast reactions in Eq. (6.28) and without loss of generality these are assumed to be the last p, which stated in equation form are

$$\begin{aligned} &\frac{k_i}{k_j} \simeq O(1), && i, j \geq M - p + 1 \\ &k_{M-p+1} \gg k_i, && i \leq M - p + 1, \end{aligned} \tag{5.7}$$

where all k_i are in consistent units. Then the $N \times M$ stoichiometric matrix can be written as $\underline{\underline{\nu}} = \left[\underline{\underline{\nu}}_s \;\; \underline{\underline{\nu}}_f \right]$ where $\underline{\underline{\nu}}_s$ is an $N \times (M - p)$ matrix corresponding to the stoichiometric vectors associated with the slow reactions and $\underline{\underline{\nu}}_f$ (an $N \times p$ matrix) with the fast reactions respectively. In equation form we have

$$\underline{\underline{\nu}}_s = \left[\underline{\nu}_1 \; \dots \; \underline{\nu}_{M-p} \right] \qquad \underline{\underline{\nu}}_f = \left[\underline{\nu}_{M-p+1} \; \cdots \; \underline{\nu}_M \right] \tag{5.8}$$

The identification of fast and slow variables and hence the transformation of Eq. (5.2) to the singularly perturbed form of Eq. (5.6) is possible if and only if the following condition between the images of the two stoichiometric submatrices is met [103, 104]

$$\mathrm{Im}\underline{\underline{\nu}}_s \bigcap \mathrm{Im}\underline{\underline{\nu}}_f = \emptyset \tag{5.9}$$

Approaching the relation from a purely mathematical point of view the condition requires the intersection of the images of the two stoichiometric matrices to be the null space. On a more practical note the condition requires that there are no fast and slow reactions that are stoichiometrically dependent. For example, a reversible reaction being fast only in one direction violates the condition.

Let us define the following three subspaces

$$\mathcal{S}_s = \ker \underline{\underline{\nu}}_f^T \setminus \ker \underline{\underline{\nu}}^T \quad \mathcal{S}_c = \ker \underline{\underline{\nu}}^T \quad \mathcal{S}_f = \ker \underline{\underline{\nu}}_s^T \setminus \ker \underline{\underline{\nu}}^T, \tag{5.10}$$

where *ker* denotes the null space of a matrix. Then if the condition in Eq. (5.9) is met the linear coordinate change $\underline{Y} = \underline{\underline{T}}\ \underline{X}$ where

$$\underline{\underline{T}} = \left[\ \underline{\underline{T}}_s^T\ \ \underline{\underline{T}}_f^T\ \right]^T = \left[\ \widehat{\underline{\underline{T}}}_s^T\ \ \widehat{\underline{\underline{T}}}_c^T\ \ \widehat{\underline{\underline{T}}}_f^T\ \right]^T, \tag{5.11}$$

and where $\widehat{\underline{\underline{T}}}_s$, $\widehat{\underline{\underline{T}}}_c$ and $\widehat{\underline{\underline{T}}}_f$ are full rank matrices satisfying $\widehat{\underline{\underline{T}}}_s^T \in \mathcal{S}_s$, $\widehat{\underline{\underline{T}}}_c^T \in \mathcal{S}_c$ and $\widehat{\underline{\underline{T}}}_f^T \in \mathcal{S}_f$, transforms the original system of CLEs to a system of the form of Eq. (5.6).

The transformed, singularly perturbed, SDE system is

$$\begin{aligned} d\underline{Y}_s &= \underline{\underline{T}}_s \underline{\underline{\nu}}_s \underline{A}_s(\underline{Y}_s, \underline{Y}_f)dt + \underline{\underline{T}}_s \underline{\underline{\nu}}_s \underline{\underline{\mathcal{D}}}\left(\sqrt{\underline{A}_s(\underline{Y}_s, \underline{Y}_f)}\right) d\underline{W}_s \\ d\underline{Y}_f &= \frac{1}{\delta} \underline{\underline{T}}_f \underline{\underline{\nu}}_f \underline{A}_f(\underline{Y}_s, \underline{Y}_f)dt + \frac{1}{\sqrt{\delta}} \underline{\underline{T}}_f \underline{\underline{\nu}}_f \underline{\underline{\mathcal{D}}}\left(\sqrt{\underline{A}_f(\underline{Y}_s, \underline{Y}_f)}\right) d\underline{W}_f, \end{aligned} \tag{5.12}$$

where the small parameter δ is defined as $\delta = 1/k_{M-p+1}$, the notation $\underline{\underline{\mathcal{D}}}(\sqrt{\underline{F}})$ denotes the diagonal matrix whose $(i,i)^{th}$ element is the square root of the i^{th} component of vector $\underline{F}$, $\underline{W}_s$ and $\underline{W}_f$ correspond to appropriate vectors of independent Wiener processes and the vectors $\underline{A}_s$ and $\underline{A}_f$ are

$$\underline{A}_s = \begin{bmatrix} k_1 c_1(\underline{Y}_s, \underline{Y}_f) \\ \vdots \\ k_{M-p} c_{M-p}(\underline{Y}_s, \underline{Y}_f) \end{bmatrix} \qquad \underline{A}_f = \frac{1}{k_{M-p+1}} \begin{bmatrix} k_{M-p+1} c_{M-p+1}(\underline{Y}_s, \underline{Y}_f) \\ \vdots \\ k_M c_M(\underline{Y}_s, \underline{Y}_f) \end{bmatrix} \tag{5.13}$$

The coordinate transformation leads to the following three subsets of variables,

$$\underline{Y}_s = \widehat{\underline{\underline{T}}}_s^T\ \underline{X} \qquad \underline{Y}_c = \widehat{\underline{\underline{T}}}_c^T\ \underline{X} \qquad \underline{Y}_f = \widehat{\underline{\underline{T}}}_f^T\ \underline{X} \tag{5.14}$$

classified as slow, constant and fast, respectively. Constant variables represent conservation relations and can also be designated as slow ones since they do not change over time. Thus instead of $\widehat{\underline{\underline{T}}}_s$ and $\widehat{\underline{\underline{T}}}_c$ we can mask both slow and constant variables under the matrix $\underline{\underline{T}}_s$. Equation (5.12) represents a transformed system of SDEs where slow and fast variables have been identified but still are coupled through the matrices $\underline{A}_s$ and $\underline{A}_f$.

5.5.2 Decoupling Fast and Slow Variables

Removing the coupling between fast and slow variables is the next important step of the proposed framework. This decoupling can be achieved by applying the method of adiabatic elimination to the system of the transformed SDEs [105]. Fast variables are assumed to relax to a pseudo-stationary density under the assumption that the slow variables remain constant. The pseudo-stationary distribution, $p_{\underline{Y}_s}(\underline{Y}_f)$, can be retrieved as the solution of the following homogeneous Fokker-Planck equation

$$\begin{aligned} & -\frac{\partial}{\partial \underline{Y}_f}\left[(\widehat{\underline{\underline{T}}}_f \underline{\underline{\nu}}_f \underline{A}_f(\underline{Y})) p_{\underline{Y}_s}(\underline{Y}_f)\right]+ \\ & +\frac{1}{2}\frac{\partial}{\partial \underline{Y}_f}\frac{\partial}{\partial \underline{Y}_f} : \left[\left(\widehat{\underline{\underline{T}}}_f \underline{\underline{\nu}}_f \underline{\underline{\mathcal{D}}}(\underline{A}_f(\underline{Y}))\underline{\underline{\nu}}_f^T \widehat{\underline{\underline{T}}}_f^T\right) p_{\underline{Y}_s}(\underline{Y}_f)\right] = 0, \end{aligned} \tag{5.15}$$

where the colon operator represents differentiation.

On the other hand the dynamics of the slow variables are approximated through a Fokker-Planck equation which governs the probability density of only the slow variables, $\widehat{p}(\underline{Y}_s;t)$

$$\begin{aligned} \frac{\partial \widehat{p}(\underline{Y}_s;t)}{\partial t} = & -\frac{\partial}{\partial \underline{Y}_s}\left[(\widehat{\underline{\underline{T}}}_s \; \underline{\underline{\nu}}_s \; \tilde{\underline{A}}_s(\underline{Y}))\widehat{p}(\underline{Y}_s;t)\right]+ \\ & +\frac{1}{2}\frac{\partial}{\partial \underline{Y}_s}\frac{\partial}{\partial \underline{Y}_s} : \left[\left(\widehat{\underline{\underline{T}}}_s \; \underline{\underline{\nu}}_s \; \underline{\underline{\mathcal{D}}}(\tilde{\underline{A}}_s(\underline{Y}))\underline{\underline{\nu}}_s^T \widehat{\underline{\underline{T}}}_s^T\right)\widehat{p}(\underline{Y}_s;t)\right], \end{aligned} \tag{5.16}$$

where the colon operator represents differentiation and $\tilde{\underline{A}}_s$ is a vector representing the mean reaction rates and its i^{th} component is defined as follows

$$\tilde{A}^i_s(\underline{Y}_s,\underline{Y}_c) = \int A^i_s(\underline{Y}_s,\underline{Y}_c,\underline{Y}^{'}_f) p_{\underline{Y}_s}(\underline{Y}^{'}_f) d\underline{Y}^{'}_f \tag{5.17}$$

Instead of solving Eq. (5.16) it is more convenient to sample the underlying probability distribution through the solutions of a system of CLEs that correspond only to the slow variables

$$d\underline{Y}_s = \underline{\underline{\widehat{T}}}_s \underline{\underline{\nu}}_s \ \underline{\tilde{A}}_s(\underline{Y}_s, \underline{Y}_c)dt + \underline{\underline{\widehat{T}}}_s \ \underline{\underline{\nu}}_s \ \underline{\underline{\mathcal{D}}}\left(\sqrt{\underline{\tilde{A}}_s(\underline{Y}_s, \underline{Y}_c)}\right)d\underline{W}_s \tag{5.18}$$

From Eq. (5.17) it is evident that in order to solve for the slow variables we have to know the pseudo-stationary distribution that the fast variables relax into. Additionally Eq. (5.16) describes the time evolution of the probability distribution of only the slow variables. Constant variables are not included since they remain constant over time. Finally, Eq. (5.16) and (5.17) imply that using the reduction framework we are able to initiate the time integration of the slow variables on the "equilibrium manifold". This is achieved because the fast variables have effectively reached a pseudo-stationary distribution.

5.5.3 Approximated Probability Distribution

The final step in the context of the framework is to compute the approximated probability distribution emerging from the initial system of CLEs by simply multiplying the two independent probability densities

$$p(\underline{Y}_s, \underline{Y}_f; t) = p_{\underline{Y}_s}(\underline{Y}_f)\widehat{p}(\underline{Y}_s; t) \tag{5.19}$$

Note that the constant variables are not included in the solution of the slow variables which is a direct result of the reduction approach method that allows the identification of the minimal number of variables for describing the dynamics of the system. First and second moments values, of all initial state variables can be retrieved from the joint probability distribution and given the definition of constant species.

The reduction framework is built upon the assumption that the initial system of reactions remains in the continuous Markov process regime for the entire simulated time interval. If for example the pseudo-stationary distribution turns out to have a mean that is below the value of ϵ or the value of the propensity of any given reaction falls below λ then not all reactions lie in the continuous Markov process regime and a different approach should be applied. Of course since both values for ϵ and λ are user defined there is a flexibility but still attention should be paid to ensure that Eq. (5.4) is satisfied during all times. On the other hand, while ϵ and λ can be varied at will their values should be handled with care to ensure that the CLE approximation is valid.

5.6 Examples

The examples used in this section aim to test the numerical accuracy of the proposed algorithm and demonstrate the benefits of the presented reduction formalism when integrating systems of stiff CLEs. All examples lie in the continuous Markov process regime, i.e. species and reaction rates satisfy conditions in Eq. (5.4). Paradigms of reacting systems that lie in the continuous Markov process regime may include but are not limited to fast occurring dimerization reactions, inducer-protein interactions and protein binding to non-specific DNA sites.

5.6.1 Example Revisited

First, we revisit the motivating example presented in the methods section (cf. Table 5.1). Recall that the reaction network dynamics are described through Eq. (5.5), requiring a relatively small time step for its accurate integration. Instead of using a small time step, we can apply the reduction framework under consideration. Even though the example looks simple, identifying the fast, slow or constant variables is not intuitive due to the coupling in the system of CLEs. Using the notation presented in the methods section the matrices $\underline{\underline{\nu}}_s$ and $\underline{\underline{\nu}}_f$ are

$$\underline{\underline{\nu}}_s = \begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix}^T \qquad \underline{\underline{\nu}}_f = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 \end{bmatrix}^T \tag{5.20}$$

The two matrices satisfy Eq. (5.9) and thus the appropriate diffeomorphism exists. Using the proposed framework the coordinate transformation is of the form

$$\begin{array}{l} \underline{\underline{\widehat{T}}}_s = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \\ \underline{\underline{\widehat{T}}}_f = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad \underline{\underline{\widehat{T}}}_c = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{5.21}$$

with $Y_1 = X_E$ being the slow species, $Y_5 = X_A$ the fast species and $Y_2 = X_B - X_A = q_1$, $Y_3 = X_D - X_A - X_C = q_2$ and $Y_2 = X_A + X_C + X_E = q_3$ the constant species (values for all $q_i$can be inferred from the initial conditions shown in Table I). The dimensionality of the problem is reduced from five variables to only two.

Taking into account the transformed variables, vectors and are defined through the following equations

$$\underline{A}_s = \begin{bmatrix} k_3\big(q_3 - Y_5 - Y_1\big)\big(q_2 + q_3 - Y_1\big) \\ k_4 Y_1 \end{bmatrix} \qquad \underline{A}_f = \frac{1}{k_1}\begin{bmatrix} k_1 Y_5\big(q_1 + Y_5\big) \\ k_2\big(q_3 - Y_5 - Y_1\big) \end{bmatrix} \tag{5.22}$$

After obtaining the coordinate transformation and identifying the transformed variables as fast, slow and constant the next step is to compute the pseudo-stationary distribution through Eq. (5.15). Since there is only one fast variable the homogeneous Fokker-Planck equation in probability (cf. Eq. (5.15)), reduces into a linear homogeneous ordinary differential equation (ODE).

$$-\frac{d}{d\underline{Y}_f}\Big[\underline{A}\, p_{\underline{Y}_s}(Y_f)\Big] + \frac{1}{2}\frac{d^2}{d\underline{Y}_f^2}\Big[\underline{\underline{B}}\, p_{\underline{Y}_s}(Y_f)\Big] = 0, \tag{5.23}$$

where after some calculations

$$\begin{aligned} \underline{A} &= \Big[\ \frac{k_2}{k_1}\big(q_3 - Y_5 - Y_1\big) - Y_5\big(q_1 + Y_5\big)\ \Big] \\ \underline{\underline{B}} &= \Big[\ \frac{k_2}{k_1}\big(q_3 - Y_5 - Y_1\big) + Y_5\big(q_1 + Y_5\big)\ \Big] \end{aligned} \tag{5.24}$$

For the boundary we can impose either Dirichlet or Neuman boundary conditions. Both approaches produce the same final result (data not shown). In this example we use reflecting boundary conditions, i.e. Neuman, since when a homogeneous Fokker-Planck equation assumes reflecting boundary conditions then under certain conditions there exists an analytical solution that depends only on matrices $\underline{A}$ and $\underline{\underline{B}}$ (cf. Eq. (5.24)). In the case of only one fast variable such an analytical solution exists always while in the case of more than one variable the solution exists under certain conditions [105]. For

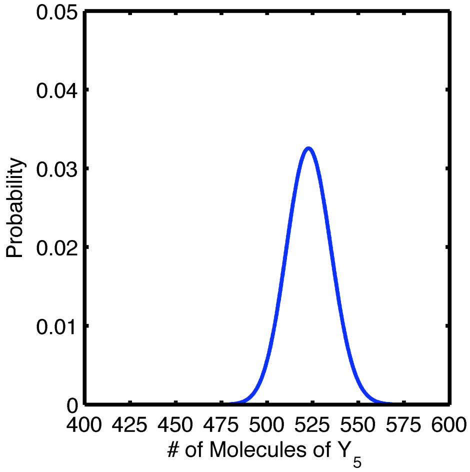


Figure 5.2: The pseudo-stationary distribution that the fast variable, Y_5, relaxes into obtained as the solution of Eq. (5.23).

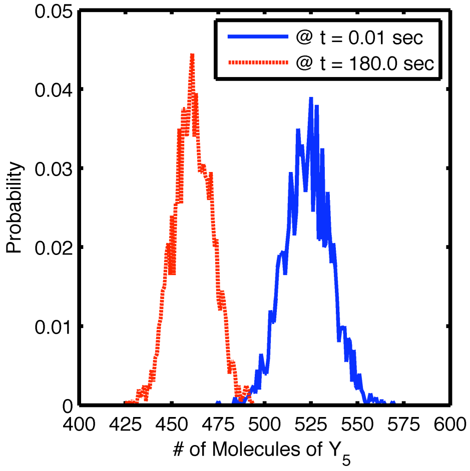


Figure 5.3: Probability distribution of the fast variable $Y_5 = X_A$ at two different time points, $t = 0.01$ s and $t = 180$ s, using 2,000 independent trials of the original set of CLEs (cf. Eq. (5.5)).

the case of only one variable the solution is of the form

$$p_{\underline{Y}_s}(Y_f) = exp\left[2\int_{\alpha}^{Y_f} A(x)/B(x)dx\right], \tag{5.25}$$

where α corresponds to the lower boundary and $A(x)$ and $B(x)$ are scalars for the one variable case (cf. Eq. (5.24)). We use Simpson's method with discretization step $h = 0.1$ to compute the numerical value of the integral. In this case the method becomes semi-analytical but still it is more easily programmable than the use of finite differences for integrating Eq. (5.23). The solution interval is $[10^2, 10^4]$, chosen large enough to ensure that the solution lies within the interval and also satisfies the assumption for the reflecting boundary conditions. Finally note that the solution of Eq. (5.23) depends on the slow species Y_1. Its value is set equal to its initial condition as depicted by the adiabatic elimination hypothesis.

We obtained the pseudo-stationary distribution depicted in Fig. 5.2. The mean and standard deviation are $E\big(Y_5|Y_1\big) = 523.43$ molecules and $\sigma = 12.27$ molecules respectively. For any such solution to make sense we have to compare the pseudo-stationary distribution with the probability distribution obtained from the initial system (cf. Eq. (5.5)). Results from the initial system for two different time points are shown in Fig. 5.3. Studying the results in Fig. 5.3 one notices that the distribution of the fast species A shifts to the left as time goes by. Indeed the mean value of the probability distribution at 0.01 s is 522.95 molecules and moves to 461.16 molecules at 180.0 s. Additionally, Fig. 5.1 also suggests that at time 180.0 s the system has practically attained its equilibrium distribution. At the same time the standard deviation of the distribution decreases over time from 12.50 molecules at 0.01 s to 11.15 molecules at 180.0 s. Note that the initial concentration of species A was 2800 molecules and in just 0.01 s it shifts to 522.95 molecules. This evidence suggests that species A, the fast species according to the reduction framework, relaxes particularly fast into a pseudo-stationary distribution, and then evolves slowly to its equilibrium distribution. These observations, while expected from Fig. 5.1 are now confirmed through Fig. 5.3. Based on the previous analysis, the distribution at time $t = 0.01$ s corresponds to a pseudo-stationary distribution, whereas the distribution at time $t = 180$ s corresponds to the equilibrium distribution. The important conclusion is that the solution of Eq. (5.23) indeed corresponds to a pseudo-stationary distribution close to 0.01 s. Further comparing Fig. 5.2 and 5.3 we observe that the two distributions have similar shapes, i.e. similar higher moments, and the error between the reduced and full system is less than 0.1 % in terms of the mean and the standard deviation.

The next step in the reduction framework is to compute the probability distribution of the slow species through Eq. (5.18). First we compute the mean reaction rates through Eq. (5.17) while taking into account the pseudo-stationary distribution and Eq. (5.22). After some calculations we have for the mean reaction rates, $\underline{\tilde{A}}_s(Y_s)$

$$\begin{aligned} \tilde{A}_s^1(Y_1) &= k_3\Big[Y_1^2 + \Big(E\big(Y_5|Y_1\big) - q_2 - 2q_3\Big)Y_1 + \big(q_2 + q_3\big)\Big(q_3 - E\big(Y_5|Y_1\big)\Big)\Big] \\ \tilde{A}_s^2(Y_1) &= k_4 Y_1, \end{aligned} \tag{5.26}$$

where $E\big(Y_5|Y_1\big)$ denotes the conditional expectation of species Y_5 and also corresponds to the dependence in the pseudo-stationary distribution. Since there is only one slow variable, Eq. (5.18) reduces to a single CLE. For an accurate sampling of the probability

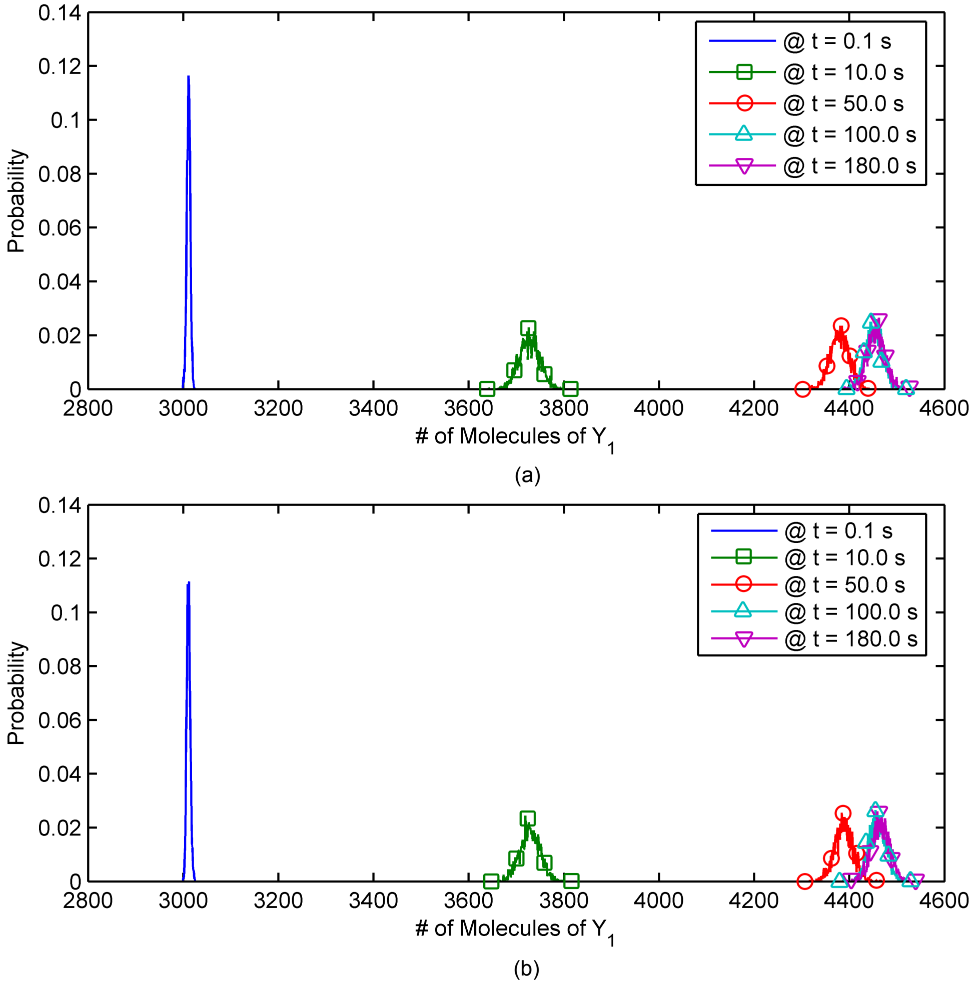


Figure 5.4: Probability distribution of the slow species, Y_1, at different time points. (a) Using the reduced model. (b) Using the full model.

distribution for the slow species we run 2,000 independent trials of Eq. (5.18). We used the Euler-Maruyama method to integrate the CLE with a time step 0.1 s in the time interval [0, 200.0] s. Note that the time step is relatively large compared to the one used to solve the initial stiff system (cf. Eq. (5.5)). Results for the probability distribution at different time points are shown in Fig. 5.4(a), whereas in Fig. 5.4(b) we depict the probability distributions at the same time points calculated from the full order model instead of the reduced one.

Through Fig. 5.4 we observe that as we move along the time interval the probability distribution shifts towards the right, i.e. its mean increases. This is evident for both the reduced and the full model. Additionally as the system approaches its equilibrium distribution the distributions converge. Comparing Fig. 5.4(a) and 5.4(b) we can conclude that the error introduced in the probability distributions of the slow species by the reduction framework is not significant, since distributions obtained by the full and reduced model do not differ appreciably. In order to quantify this observation we calculated the mean and standard deviation for the probability distributions at each time point and for the two methods. Results are depicted in Table 5.2 along with the corresponding error percentages. As the simulation time increases so does the error in the mean while that in the standard deviation decreases, but both remain relative small as the system approaches equilibrium. This error is the direct result of the discrepancy between the pseudo-stationary and the equilibrium distribution (cf. Fig. 5.2 and 5.3), which in this case is approximately 14 % for the mean and 13 % for the standard deviation. Evidently the reduction framework attenuates the error in the fast dynamics and accurately captures the dynamics of the slow species.

Examining the average normalized weak mean and variance errors [3] we observed that the errors introduced in the reduced system is invariant to a time step decrease and at least the variance error is similar to the full order system. On the other hand, the normalized error in the mean appears to be an order of magnitude larger. In the full order system there are two errors, the Langevin approximation and the integration error terms. For the reduced system we have an additional error term that comes from the reduction framework itself. This last term is the reason for the larger normalized mean error (cf. Table 5.2). Of course, for practical purposes the error is insignificant but it is still larger than the full order system.

Table 5.2: Mean and Standard Deviation Errors for the Slow Dynamics of the Motivating Example

	Mean (molecules)			Standard Deviation (molecules)		
Time (s)	Full System	Reduced System	Error (%)	Full System	Reduced System	Error (%)
0.1	3010.90	3011.00	0.003	3.66	3.62	1.093
10.0	3728.63	3727.30	0.036	21.20	21.86	3.113
50.0	4388.15	4379.50	0.197	19.74	19.73	0.051
100.0	4459.36	4451.42	0.178	18.65	18.50	0.804
180.0	4465.46	4456.38	0.203	18.88	18.77	0.583

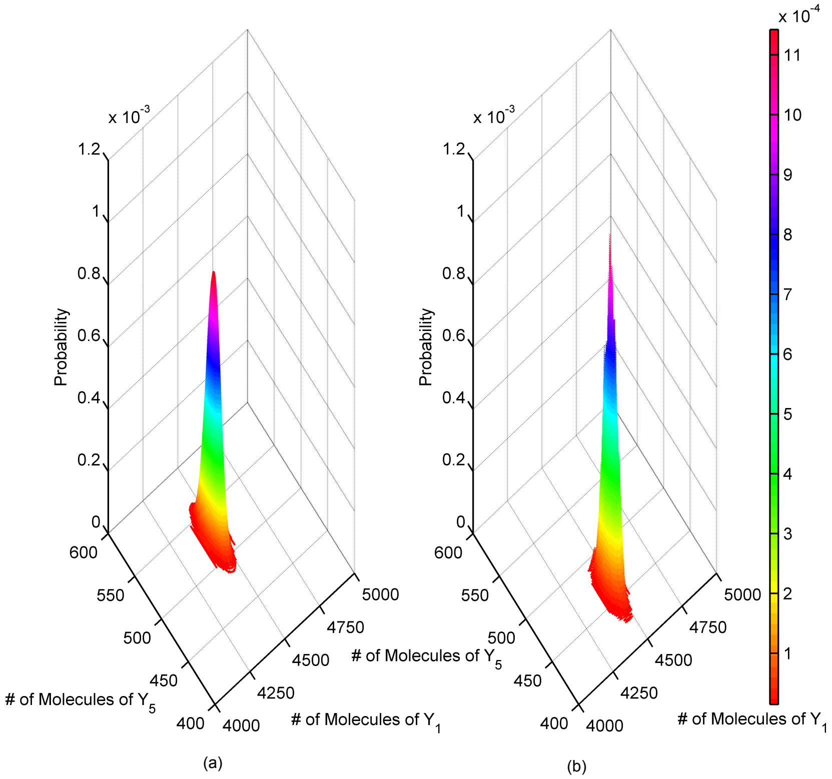


Figure 5.5: Joint probability distribution $t = 180$ s. (a) Using the reduced model. (b) Using the full model.

The final step in the reduction framework is to reproduce the joint probability distribution through Eq. (5.19). In Fig. 5.5 we plot the joint probability density at time $t = 180$ s as this is produced from the reduced model (Fig. 5.5(a)) and the full model (Fig. 5.5(b)). Fig. 5.5 essentially captures all the advantages and the limitations of the reduction framework. First comparing Fig. 5.5(a) 5.5(b) we notice that there is a shift in the joint probability distribution in the y-direction, i.e. the number of molecules of Y_5, which is a direct result of the error between the pseudo-stationary distribution calculated through the reduction approach and the actual equilibrium distribution. On the x-axis, i.e. the number of molecules of Y_1, the shift is negligible as result of the small error introduced by the reduction framework in the slow species. On the other hand the shape of the two distributions in terms of higher moments looks similar which is also an important aspect. The sharper look of the distribution in Fig. 5.5(b) is a result of the number of trials conducted. Using more than 2,000 trials would result in a smoother distribution. Note that the joint probability distribution depends only on the fast and slow species, while all information regarding the initial state variables can simply be reproduced through the joint probability distribution and the relations for the constant species.

A final point to make is the comparison of execution times between the full and reduced order models (details for each code have been mentioned throughout this section). For that, full and reduced order models where coded in Matlab (Intel Core 2 Duo 2.0 GHz processor with 3 GB RAM); no attempts to optimize the codes were made. Therefore results should only be considered as indicative. The full systems requires more than 10^5 s while the reduced model approximately 1000 s. Execution times correspond to the total computational cost since a reduction to per trial cost is not applicable for the reduced system. Results show that the reduction framework greatly decreases computational costs.

5.6.2 Second Example

As a second example we use an altered cycle test system. Consider the system of reactions detailed in Table 5.3 taking place in a bacterial-sized volume of 10^{-15} L. The reversible reaction is assumed to be much slower in both directions than the rest of the reactions. The corresponding system of CLEs is

$$\begin{aligned}
dX_A &= (-k_1X_A + k_3X_C - k_4X_AX_D + k_5XE)dt - \\
&\quad - \sqrt{k_1X_A}dW_1 + \sqrt{k_3X_C}dW_3 - \sqrt{k_4X_AX_D}dW_4 + \sqrt{k_5XE}dW_5 \\
dX_B &= (k_1X_A - k_2X_B)dt + \sqrt{k_1X_A}dW_1 - \sqrt{k_2X_B}dW_2 \\
dX_C &= (k_2X_B - k_3X_C)dt + \sqrt{k_2X_B}dW_2 - \sqrt{k_3X_C}dW_3 \\
dX_D &= (-k_4X_AX_D + k_5X_E)dt - \sqrt{k_4X_AX_D}dW_4 + \sqrt{k_5X_E}dW_5 \\
dX_E &= (k_4X_AX_D - k_5X_E)dt + \sqrt{k_4X_AX_D}dW_4 - \sqrt{k_5X_E}dW_5
\end{aligned} \tag{5.27}$$

Using Hy3S [1] or SynBioSS [4], we conducted $2,000$ independent trials of Eq. (5.27) in the time interval $[0, 50]$ s, long enough so that it reaches its equilibrium distribution,

Table 5.3: Reactions and Parameters for the Second Example

Set of Reactions	Mesoscopic Reaction Rates[†]	Initial Values[‡]
$A \xrightarrow{k_1} B$	$k_1 = 1.5 \times 10^3$	$\left[A\right]_0 = 1200$
$B \xrightarrow{k_2} C$	$k_2 = 5 \times 10^3$	$\left[B\right]_0 = 800$
$C \xrightarrow{k_3} A$	$k_3 = 10^3$	$\left[C\right]_0 = 1500$
$A + D \underset{k_5}{\overset{k_4}{\rightleftharpoons}} E$	$k_4 = 1.66 \times 10^{-4}$	$\left[D\right]_0 = 500$
	$k_5 = 8 \times 10^{-2}$	$\left[E\right]_0 = 200$

[†] for 1^{st} order reactions the units are s^{-1} and for 2^{nd} order reactions the units are $\mathrm{molecules}^{-1}\mathrm{s}^{-1}$.

[‡] initial values are in number of molecules.

approximately after 25 s. Results are not presented for brevity, but we want to point out that during the chosen time interval all reactions satisfy conditions Eq. (5.4) and thus can be described as continuous Markov processes. Examining the time trajectories of the system more closely we infer that species concentrations evolve over time affected by the two different time scales. In particular species A and B are mainly affected by the fast dynamics and reach very quickly, in approximately 0.01 s, what appears to be a pseudo-steady state or more accurately a pseudo-stationary distribution. On the other hand, species D and E are mainly influenced by the slow reactions and in the time interval of 0.01 s their concentrations are practically unaltered. Similar to the first example we used the fixed step Euler-Maruyama method [73] as our integration method with time step size set to 10^{-4} s. For larger time steps the integration fails; species populations attain negative values. This is a direct consequence of the disparate time scales, i.e. mesoscopic kinetic rates values differ by six orders of magnitude (cf. Table 5.3).

Therefore we can apply the reduction framework to the system under consideration. Using conditions in Eq. (5.7) and the data from Table 5.3 we obtain the corresponding stoichiometric submatrices defined through Eq. (5.8). More importantly, $\underline{\underline{\nu}}_s$ and $\underline{\underline{\nu}}_f$

satisfy condition in Eq. (5.9) thus the appropriate diffeomorphism exists and is of the form

$$\underline{\underline{\widehat{T}}}_f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \underline{\underline{\widehat{T}}}_c = \begin{bmatrix} -1 & -1 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \qquad (5.28)$$

$$\underline{\underline{\widehat{T}}}_s = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with X_E being the slow species, $X_D - X_A - X_B - X_C = q_1$ and $X_A + X_B + X_C + X_E = q_2$ the constant species and X_B and X_C the fast species. Importantly, the dimensionality of the problem is reduced from five variables to only three. The values for q_1 and q_2 can be calculated from the initial values (cf. Table 5.3). *A priori* identification of fast, slow or constant variables is not intuitive due to the coupling in the system of CLEs (cf. Eq. (5.27)). Given the transformed variables, vectors $\underline{A}_s$ and $\underline{A}_f$ are defined as

$$\underline{A}_s = \begin{bmatrix} k_4\big(q_2 - Y_1 - Y_4 - Y_5\big)\big(q_1 + q_2 - Y_1\big) \\ k_5 Y_1 \end{bmatrix}$$

$$\underline{A}_f = \frac{1}{k_1} \begin{bmatrix} k_1\big(q_2 - Y_1 - Y_4 - Y_5\big) \\ k_2 Y_2 \\ k_3 Y_5 \end{bmatrix}, \qquad (5.29)$$

where Y_i correspond to the new variables starting with the slow, continuing with the constant and ending with the fast ones.

After obtaining the coordinate transformation and identifying the transformed variables as fast, slow and constant the next step is to compute the pseudo-stationary distribution through Eq. (5.15). Since there are two fast variable the homogeneous Fokker-Planck equation in probability forms a homogeneous linear partial differential equation (PDE). An analytical solution does not exist as was the case in the first example [105]; we use central finite differences to approximate the first and second order derivatives in concentrations and then solve the corresponding system of linear algebraic equations. For the boundary we impose Dirichlet boundary conditions. Setting zero boundary values is realistic, but the system would then only attain the trivial solution. A more practical approach is to consider that at distances larger than five

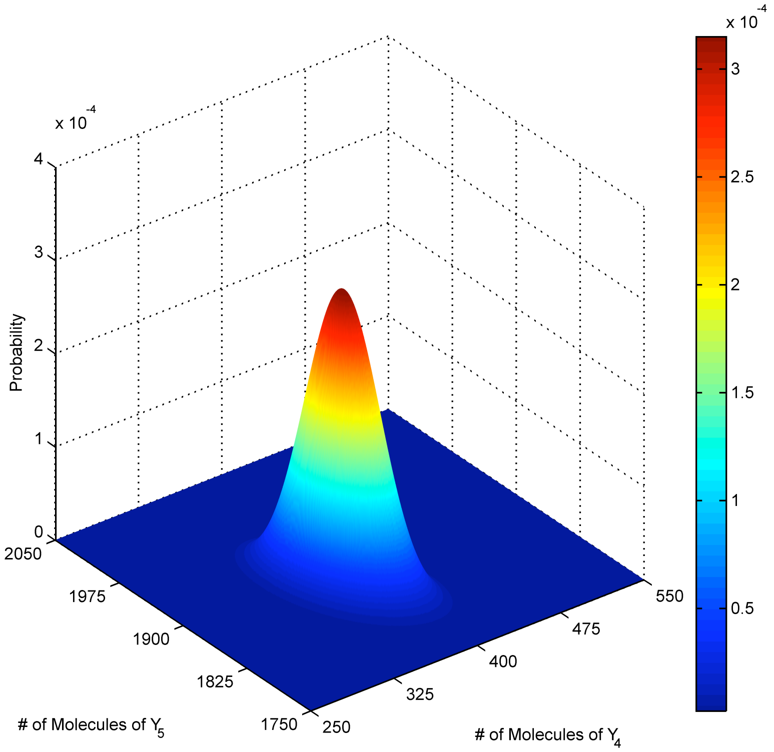


Figure 5.6: The pseudo-stationary distribution that the fast variables, Y_4 and Y_5, relax into obtained as the solution of Eq. (5.15).

standard deviations the tail of the probability distribution approaches zero but is not actually zero, i.e. on the boundaries we set the probability distribution value equal to 10^{-10}. Notice that through Eq. (5.29) the solution of the PDE depends on the slow species, Y_1. Its value is set equal to its initial condition as depicted by the adiabatic elimination hypothesis.

We solve Eq. (5.15) using an equally spaced mesh. The discretization step is 0.5, for which convergence in the desired accuracy is achieved (data not shown). The pseudo-stationary distribution is depicted in Fig. 5.6. The conditional expectation of species B with respect to C is $E\big(Y_4|Y_5, Y_1\big) = 374.99$ molecules and that of C with respect to B is $E\big(Y_5|Y_4, Y_1\big) = 1875.03$ molecules. By comparing (data not shown) the reduced and full system solutions we infer that indeed the result in Fig. 5.6 corresponds to a

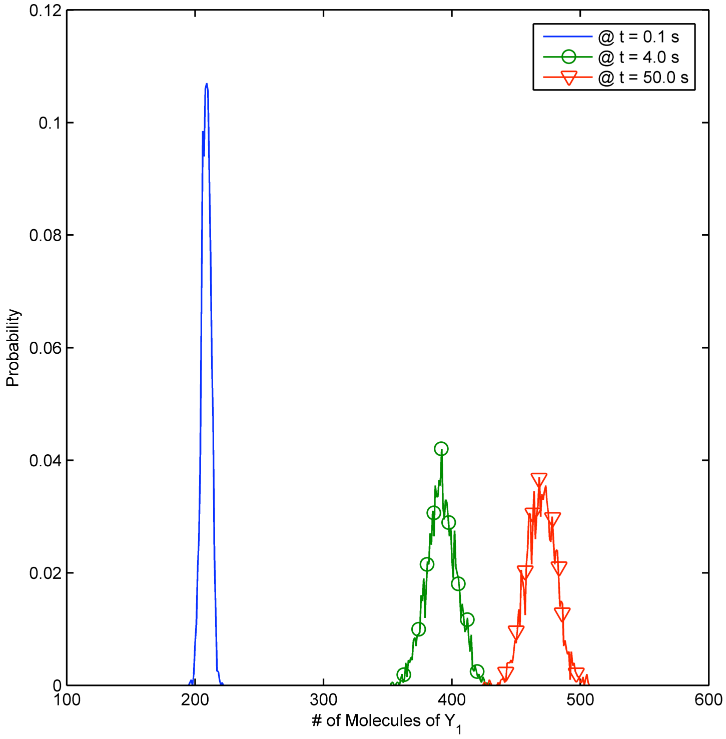


Figure 5.7: Probability distribution of the slow species, Y_1, at different time points.

pseudo-stationary distribution located at or close to 0.01 s. Importantly, the shapes of the two distributions in terms of higher moments look similar. The error between the reduced and the solution of the full system at time $t = 0.01$ s is less than 0.1 %.

Next we use Eq. (5.18) to compute the probability distribution of the slow species. First we compute the mean kinetic rates through Eq. (5.17) using Eq. (5.29). Skipping through some tedious calculations we have

$$\begin{aligned}\tilde{A}_s^1(Y_1) = k_4\Big[Y_1^2 + \Big(E\big(Y_4|Y_5,Y_1\big) + E\big(Y_5|Y_4,Y_1\big) - q_1 - 2q_2\Big)Y_1 \\ + \big(q_1+q_2\big)\big(q_2 - E\big(Y_4|Y_5,Y_1\big) - E\big(Y_5|Y_4,Y_1\big)\big)\Big] \qquad (5.30)\\ \tilde{A}_s^2(Y_1) = k_5 Y_1,\end{aligned}$$

where the dependence on the pseudo-stationary distribution is manifested through the two conditional expectations. Recall that there is only one slow variable thus we have a single CLE. For an accurate sampling of the probability distribution for the slow species we run $2,000$ independent trials using the Euler-Maruyama method to integrate Eq. (5.18) with a time step 0.1 s in the time interval $[0, 50]$ s. Note that the time step is relatively large compared to the one used to integrate the initial stiff system (cf. Eq. (5.27)). As in the first example the accuracy of the framework is invariant in a decrease in the integration time step since the reduction approach introduced the leading error term. The probability distributions at different time points are shown in Fig. 5.7. As we move along the time interval the probability distribution shifts towards the right, i.e. its mean increases.

In order to have an estimate of the error introduced by the reduction approach we calculated the mean and standard deviation for the probability distribution at each time point using both the reduced and full order models. Results are depicted in Table 5.4 along with the corresponding error percentages. As the simulation time increases so does the error in the mean while that in the standard deviation decreases, but both remain relative small as the system approaches equilibrium. The error in the slow species is a direct consequence of the discrepancy, if any, between the pseudo-stationary and the equilibrium distribution of the fast species. In the present example the conditional expectations of both fast species exhibit an error of approximately 10 % between their equilibrium and pseudo-stationary distributions. Thus the reduction framework is able to attenuate the error in the fast dynamics. Overall the reduction framework captures the dynamics of the slow species more accurately than for the fast ones.

Finally we compare the execution times between the full and reduced order models (details for each code have been mentioned during the present section). For this part both models, full and reduced, were coded in Matlab (Intel Core 2 Duo 2.0 GHz processor with 3GB RAM); no attempts to optimize the codes were made. Therefore results should only be considered as indicative. The full system requires more than 3×10^5 s while the reduced approximately 1350 s (includes the time needed to solve the PDE and the time it takes to obtain 2,000 trajectories for the CLE describing the slow species). Note that for the reduced model the finite difference mesh used is $\big[250, 550\big] \times \big[1750, 2050\big]$. Of course, use of a larger mesh in the finite difference scheme

Table 5.4: Mean and Standard Deviation Errors for the Slow Dynamics of the Second Example

	Mean (molecules)			Standard Deviation (molecules)		
Time (s)	Full System	Reduced System	Error (%)	Full System	Reduced System	Error (%)
0.1	208.68	208.71	0.01	3.35	3.49	4.18
4.0	401.84	391.99	2.45	12.24	11.84	3.27
50.0	492.95	469.35	4.79	11.75	11.67	0.68

would result in a considerable increase in CPU time. The choice of the mesh is based on an *a priori* knowledge. Such an insight can be obtained by running a single trial of the initial stiff system. In that case the total CPU time for the reduced model, in the context of the present example, would be approximately 1550 s. In summary, having an insight on the dynamics of the system results in the use of a reasonable mesh that further contributes in the significant decrease in computational cost observed with the reduction framework.

5.6.3 Sugar Cataract Development Model

As a final example we consider a previously studied biological model that describes the formation of sugar cataract. This sugar cataract development (SCD) model depicts how the enzyme sorbitol dehydrogenase (SDH) catalyzes the reversible oxidation of sorbitol and other polyalcohols to the corresponding ketosugars [106]. Accumulation of excess sorbitol to the lens is responsible for the sugar cataract development that distorts light passing through the lens. In particular, ratios of sorbitol over fructose larger than one have been shown to be an indication of early stages of cataract formation.

The original SCD model involves 7 reactions with 7 participating species. All but one reaction can be modeled as a continuous Markov processes, i.e. their dynamics are governed through a system of CLEs [107]. Only the conversion of SDH to its inactive form lies in the discrete Markov process regime. In other words, its reaction rate is many orders of magnitude smaller than the rest of the reactions and its impact on the system can be neglected for a short initial time frame. Therefore for the purposes of the present work we do not consider the inactivation of SDH. The details of the SCD model used in the present study are depicted in Table 5.5, where SDH represents the enzyme sorbitol dehydrogenase, S and F represent sorbitol and fructose, respectively, NADH represents the nicotinamide adenine dinucleotide and NAD+ is the oxidized form of NADH. Initial conditions correspond to the average concentrations according to Table II in Ref. [107]. All reactions take place in a bacterial-sized volume of 10^{-15} L.

Using Hy3S [1] or SynBioSS [4], we conducted $2,000$ independent trials of the corresponding system of CLEs [107] in the time interval $[0, 60]$ s, long enough so that it reaches its equilibrium distribution, but at the same time short enough so that the system dynamics are not affected by the reaction we neglected. Examining the time trajectories of the system more closely we infer that species concentrations evolve over time affected by the two different time scales. In particular species SDH, $E - NAD^+$ and NAD^+ are mainly affected by the fast dynamics and reach very quickly, in approximately 0.1 s, what appears to be a pseudo-steady state or more accurately a pseudo-stationary distribution. On the other hand, species S and F are mainly influenced by the slow reactions and in the time interval of 0.1 s their concentrations are practically unaltered. Similar to the previous two examples we used the fixed step Euler-Maruyama method [73] as our integration method with time step size set to 10^{-3} s. For larger time steps the integration fails; species populations attain negative values. This is a direct consequence of the different time scales.

From conditions in Eq. (5.7) and the data from Table 5.5 we obtain the corresponding stoichiometric submatrices defined through Eq. (5.8). More importantly, $\underline{\underline{\nu}}_s$ and $\underline{\underline{\nu}}_f$ satisfy condition Eq. (5.9) thus the appropriate diffeomorphism exists, i.e. the reduction framework is applicable. The transformed system has two fast variables and one slow variable, while there are also four constant species. X_F is the slow species, $X_{SDH} + X_{E-NADH} + X_{E-NAD^+} = q_1$, $X_{NADH} + X_{NAD^+} - X_{SDH} = q_2$, $X_{NADH} + X_{E-NADH} + X_S = q_3$ and $X_F - X_{NADH} - X_{E-NADH} = q_4$ the constant species and X_{SDH} and X_{NADH} the fast species. Importantly, the dimensionality of the problem is reduced from seven variables to only three. The values for all $q_i' s$ can be calculated from the initial values (cf. Table 5.5). The identification of NADH as fast species was unexpected given the data we had from the full order system. This goes to show that an *a priori* identification of fast, slow or constant variables is not straightforward.

Interestingly, both the SCD model and the second example reduce to a similar transformed system with two fast and one slow species even though their full order models differ significantly. Therefore the reduction methodology for the SCD model is similar to the second example, meaning that the pseudo-stationary distribution of the fast species arises as the solution of a homogeneous PDE and the dynamics of the slow species are described through a single CLE.

The solution of Eq. (5.15) for the sugar cataract development model is depicted in Fig. 5.8. Again we used an equally spaced mesh with the discretization step set to 0.5. From the pseudo-stationary distribution we infer that the conditional expectation of species SDH with respect to $NADH$ is $E\big(X_{SDH}|X_{NADH}, X_F\big) = 2826.7$ molecules

Table 5.5: Reactions and Parameters for the Sugar Cataract Development Model

Set of Reactions	Mesoscopic Reaction Rates[†]	Initial Values[‡]
$SDH + NADH \underset{k_2}{\overset{k_1}{\rightleftharpoons}} E - NADH$	$k_1 = 1.03 \times 10^{-2}, k_2 = 3.3 \times 10^1$	$\left[SDH\right]_0 = 301$
$E - NADH + F \underset{k_4}{\overset{k_3}{\rightleftharpoons}} E - NADH^+ + S$	$k_3 = 3.65 \times 10^{-6}$	$\left[NADH\right]_0 = \left[E - NADH\right]_0 = 3012$
$E - NADH^+ \underset{k_6}{\overset{k_5}{\rightleftharpoons}} SDH + NAD^+$	$k_4 = 1.31 \times 10^{-5}$	$\left[NAD^+\right]_0 = \left[E - NAD^+\right]_0 = 3012$
	$k_5 = 2.27 \times 10^2,\ k_6 = 10^{-2}$	$\left[F\right]_0 = \left[S\right]_0 = 1.507 \times 10^5$

[†] for 1st order reactions the units are s^{-1} and for 2nd order reactions the units are $\text{molecules}^{-1}s^{-1}$.

[‡] initial values are in number of molecules.

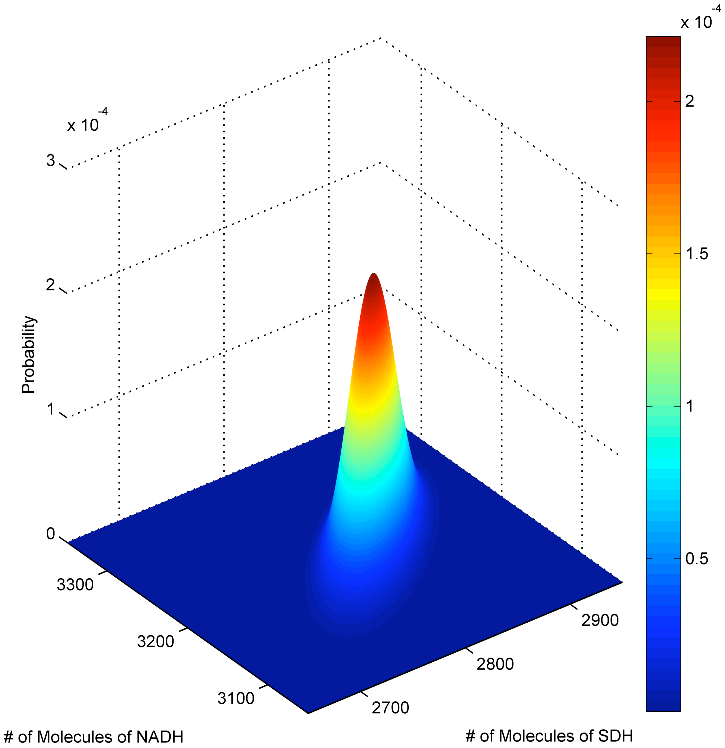


Figure 5.8: The pseudo-stationary distribution that the fast variables, X_{SDH} and X_{NADH}, relax into obtained as the solution of Eq. (5.15) as it applies to the SCD model.

and that of $NADH$ with respect to SDH is $E\big(X_{NADH}|X_{SDH}, X_F\big) = 3200.5$ molecules. Comparing the reduced and full system solutions we conclude that indeed the result in Fig. 5.8 corresponds to a pseudo-stationary distribution located at or close to 0.1 s. Importantly, the shapes of the two distributions in terms of higher moments look similar. The error between the reduced and the full system solution at time $t = 0.1$ s is less than 0.1 %. Note that the conclusions drawn for the SCD are qualitatively similar to the second example.

For the slow dynamics we use Eq. (5.18) to perform multiple trials in order to obtain an accurate sampling of the probability distribution of the slow species. Mean kinetic rates are calculated similarly to the second example. Recall that there is only

Table 5.6: Mean and Standard Deviation Errors for the Slow Dynamics of the Sugar Cataract Development Model

	Mean (molecules)			Standard Deviation (molecules)		
Time (s)	Full System	Reduced System	Error (%)	Full System	Reduced System	Error (%)
0.1	1.50694×10^5	1.50678×10^5	0.01	16.75	17.05	1.79
4.0	1.50341×10^5	1.50613×10^5	0.18	51.74	25.22	51.25
60.0	1.50290×10^5	1.50613×10^5	0.21	52.08	25.66	50.75

one slow variable in the transformed system, which translates to a single CLE. We run $2,000$ independent trials using the Euler-Maruyama method to integrate Eq. (5.18) with a time step 0.1 s in the time interval $[0, 60]$ s. Note that the time step is two orders of magnitude larger than the one used to integrate the initial stiff system. The error introduced in the slow dynamics through the reduction framework is estimated in Table 5.6 where we compare the first two moments of the probability distributions between the full and reduced order models at three different time points. Contrary to the behavior observed in the previous two examples, as the simulation time increases so do the errors in the mean and in the standard deviation. Focusing in the standard deviation error we notice a 50 % difference between the full and reduced order solution. That would be an unacceptable result if the concentration of the slow species was not in the order of 10^5. Factoring in the large concentration of the slow species we observe that if we normalize the error in the standard deviation with the observed mean, then the error is only 0.01 %. Therefore, we conclude that the reduction framework captures the dynamics of the slow species accurately.

Finally, execution times for the reduced model are very similar to the ones recorded in the second example, as both examples reduce to the same number of fast and slow variables. For the full order system we report execution times in the range of 2×10^4 s (Results should only be considered as indicative and all simulations were run in Matlab, Intel Core 2 Duo 2.0 GHz processor with 3 GB RAM). There is an approximate 10-fold decrease in the computational cost which is less than that observed in the previous two examples, as the present example is less stiff.

5.7 Summary

A new, semi-analytical reduction framework for multiscale systems of chemical Langevin equations was presented and its advantages and limitations were examined through illustrative examples. Whenever a necessary and sufficient condition is met the identification of fast, slow and constant variables becomes possible. Next the framework utilizes the already established method of adiabatic elimination in order to compute the pseudo-stationary distribution of the fast species under the assumption that slow variables remain constant. The probability distribution of the slow species is obtained as a solution of a Fokker-Planck equation or equivalently a system of CLEs governed only by the slow dynamics. The final step is to compute the joint probability distribution by multiplying each independent probability.

The reduction framework proves to be relatively easy to implement and at the same time accurate. It accurately captures the slow dynamics, which usually exhibit the most interesting phenomena, while reproducing the fast dynamics, which are usually less important, with an acceptable error. Implementation of the framework is straightforward, especially for the case of only one fast variable, where a semi-analytical solution renders its applicability attractive. Importantly, a notable advantage is the computational efficiency of the proposed algorithm that results in significant decrease in computational resource cost, ranging from one to two orders of magnitude depending on the stiffness of the system.

Chapter 6

Analytical Derivation of Moment Equations in Stochastic Chemical Kinetics

6.1 Introduction

Many stochastic processes in physics and chemistry can be considered to follow the Markov property; the movement of the system in the available state space depends only on the previous state and not on the path that the stochastic process has followed before.

Any stochastic process that obeys the Markov property is a Markov process. The underlying probability distribution, that is the probability of finding the system at any possible state at a certain time, is governed by a single partial differential equation (PDE) called the master equation (ME). If an initial condition is known and given the transitional probabilities, i.e the probabilities of the system transitioning from any state to any other state, ME uniquely determines the probability distribution at any later time [47, 105].

The solution of ME requires enumerating the states and finding the transition matrix, a task that is tractable only for the simplest of systems. For more complex systems, in lieu of a solution of ME, kinetic Monte Carlo techniques are often used to sample the underlying probability distribution. There is a growing community of researchers that develops computationally efficient and accurate stochastic simulation algorithms [64, 69, 53, 58, 56, 67, 70, 72, 3, 5, 1, 4].

Alternatively, instead of sampling the probability distribution all important information for the system's behavior can be had through the moments of the probability distribution [108]. The first moment relates to the mean of the probability distribution, the second to the variance, the third to skewness and the fourth to kurtosis. Higher order moments may also contain important information on systems dynamics especially when these systems are complex and highly non-linear.

In this work we start from the master equation and derive a system of ordinary differential equations (ODE) that describe the dynamics of the probability distribution moments. We express the transient moment dynamics in terms of the derivative or jump moments [109]. Jump moments quantify the effect of any transition between states in the probability distribution moment values. In other words, they are measures of the rate at which moment values change when the process moves from state to state.

What is new in the current work is a scheme to derive analytical expressions for the jump moments for any N-dimensional Markov process. These expressions can enable scientist to quickly construct the infinite linear system of ODEs that describes the moment dynamics of any such process.

We then apply this scheme to stochastic chemical kinetics. Stochastic chemical kinetics have emerged in recent years as an appropriate modeling formalism for biological systems that are away from the thermodynamic limit [74, 2, 90, 110, 30, 32, 21].

The idea of using moment equations to predict dynamics of systems in the stochastic chemical kinetics regime has also been considered early on [111]. In recent years, publications have proposed alternative ways to derive the moment equations [112, 113, 114, 115]. Our work complements this rich literature, providing useful analytical relations for the jump moments in stochastic chemical kinetics. These relations are then used to provide analytical equations for the probability distribution moments.

The paper is organized as follows. First we briefly present background information on Markov processes and their governing equations. We then define the jump moments of Markov processes and we use this definition to derive the general form of the moment equations. Then we concentrate on the stochastic chemical kinetics regime and we derive analytical relations for the jump moments. We use examples to discuss the applicability of the derived equations. Finally we discuss the drawbacks of any moment based method that mainly stem from the fact that the resulting ODE system is infinite dimensional. We discuss literature moment-closure schemes and speculate on how the analytical expressions we develop may assist in the development of accurate closing schemes. We also argue that higher order moments, at least up to order six, are necessary for biologically-relevant chemical kinetics systems. This is contrary to the widely held belief that only the mean and variance are important in stochastic chemical kinetic models.

6.2 Theory

6.2.1 Markov Processes

Consider a Markov process, $X(t)$, a stochastic process that for any n successive set of times, $t_1 < t_2 < \cdots < t_n$, obeys the following property, known as the *Markov property*

$$P_{1|n-1}\left(X_n, t_n \mid X_1, t_1; \ldots ; X_{n-1}, t_{n-1}\right) = P_{1|1}\left(X_n, t_n \mid X_{n-1}, t_{n-1}\right), \tag{6.1}$$

where $P_{1|n-1}$ and $P_{1|1}$ denote conditional probabilities and X_k is the state of the system at time t_k. The Markov property states that the transition of the system to state X_n at time t_n depends only on the previous state X_{n-1} at time t_{n-1} and therefore is independent of the history of the process, i.e. $X_1, t_1; \ldots ; X_{n-2}, t_{n-2}$.

Any stochastic process with state vector $\underline{X}(t) = (X_1(t), \ldots, X_N(t))$ and initial condition $\underline{X}(t = 0) = \underline{X}_0$ obeying the Markov property is governed by a differential difference equation in probability known as the *master equation* (ME) [47]

$$\frac{\partial P(\underline{X}, t)}{\partial t} = \int \Big[T\big(\underline{X}/\underline{X}^{'}\big)P\big(\underline{X}^{'}, t\big) - T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X}, t\big)\Big]d\underline{X}^{'}, \tag{6.2}$$

where $P(\underline{X}, t)$ is the probability of the system being at state $\underline{X}$ at time t and $T\big(\underline{X}/\underline{X}^{'}\big)$ is the transition probability per unit time for the system to jump from state $\underline{X}^{'}$ to state $\underline{X}$. In principle, the ME uniquely determines the probability $P(\underline{X}, t)$ of the system being at a state $\underline{X} = \underline{X}(t)$ at time $t > 0$. Note that using integrals in the ME we have made the assumption that variable $\underline{X}(t)$ is continuous.

An alternative description of the Markov process probability distribution is obtained through the *Kramers-Moyal* expansion [47].

$$\frac{\partial P(\underline{X},t)}{\partial t} = \sum_{m=1}^{\infty} \frac{(-1)^m}{m!} \sum_{j_1,\cdots,j_m=1}^{N} \frac{\partial^m}{\partial X_{j_1} \cdots \partial X_{j_m}} \left[\underline{a}_m(\underline{X}) P(\underline{X},t)\right], \tag{6.3}$$

where $\underline{a}_m$ are m-order tensors and were originally called *derivative* or *jump moments* by Moyal [109] and later referred to as propagator moment functions [108]. Jump moments are measures of the rate of change in the probability distribution moment values, i.e. what is the impact of any given transition between states on the moment values.

6.2.2 Moments and Jump Moments

Let us consider an N-dimensional Markov process $\underline{X}(t) = (X_1(t), \ldots, X_N(t))$ with probability distribution $P(\underline{X}(t), t)$.

The m-th moment of the probability distribution is defined as

$$\left\langle X_i^m \right\rangle = \int X_i^m P\big(\underline{X}, t\big) d\underline{X} \qquad i = 1, \ldots, N \tag{6.4}$$

The first order moments of any X_i is usually referred to as the mean while second order moments relate to the variance of the probability distribution through the relation, $var(X_i) = \left\langle X_i^2 \right\rangle - \left\langle X_i \right\rangle^2$.

The joint moments are defined through the following relations

$$\left\langle X_1^{m_1} X_2^{m_2} \cdots X_n^{m_n} \right\rangle = \int \left[X_1^{m_1} X_2^{m_2} \cdots X_n^{m_n}\right] P\big(\underline{X}, t\big) d\underline{X} \quad m_1, \cdots, m_n = 1, 2, \ldots \tag{6.5}$$

where the sum $m = m_1 + m_2 + \cdots m_n$ is the order of the joint moment. The set of the m-th order moments uniquely determines the underlying probability distribution. The existence of higher order moments depends on how fast $P(\underline{X}(t), t)$ approaches zero as $\|\underline{X}\| \to \infty$.

On the other hand, jump moments relate to the transition probability rather than the probability distribution. Based on Moyal's work jump moments are defined through the following relations [109]

$$a_m^i(\underline{X}) = \int (X_i^{'} - X_i)^m \; T(\underline{X}^{'}/\underline{X}) d\underline{X}^{'} \qquad i = 1, \dots, N \tag{6.6}$$

Index m refers to the order of the moment and index i to the element within the tensor.

Analogously the joint jump moments are

$$a_{m_i+m_j+\cdots}^{i,j\cdots}(\underline{X}) = \int \Big[(X_i^{'} - X_i)^{m_i} (X_j^{'} - X_j)^{m_j} \cdots \Big] T\big(\underline{X}^{'}/\underline{X}\big) d\underline{X}^{'} \tag{6.7}$$

Jump moments are tensors and the indexes $i, j, \dots$ denote the element of the $m = m_i + m_j + \cdots$ order tensor. Jump moments of order 2 and greater are symmetric tensors, i.e. $a_2^{1,2} = a_2^{2,1}$. The same is true for regular moments. In the simple case of an one-dimensional Markov process jump moments are simply scalars.

6.2.3 Derivation of Moment Equations

Given the definitions for both regular and jump moments we can now start deriving the moment equations, i.e. the system of ODEs that describes moment dynamics.

Starting with the first moment we have

$$\big\langle X_i \big\rangle = \int X_i P\big(\underline{X}, t\big) d\underline{X} \tag{6.8}$$

From the defining equation we have the exact identity for the time derivative of the first moment

$$\frac{d\big\langle X_i \big\rangle}{dt} = \int X_i \frac{\partial P\big(\underline{X}, t\big)}{\partial t} d\underline{X} \tag{6.9}$$

substituting Eq. (6.2) in the last equation we have

$$\begin{aligned} \frac{d\big\langle X_i \big\rangle}{dt} &= \int X_i \int \Big[T\big(\underline{X}/\underline{X}^{'}\big) P\big(\underline{X}^{'}, t\big) - T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) \Big] d\underline{X}^{'} d\underline{X} = \\ &= \int \int \Big[X_i T\big(\underline{X}/\underline{X}^{'}\big) P\big(\underline{X}^{'}, t\big) - X_i T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) \Big] d\underline{X}^{'} d\underline{X} \end{aligned} \tag{6.10}$$

Noticing that that the integration over $\underline{X}$ and $\underline{X}^{'}$ runs over the same domain we can interchange indexes in the last equation, i.e.

$$X_i T\big(\underline{X}/\underline{X}^{'}\big) P\big(\underline{X}^{'}, t\big) = X_i^{'} T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) \tag{6.11}$$

thus we have

$$\frac{d\langle X_i\rangle}{dt} = \int\int (X_i^{'} - X_i) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \tag{6.12}$$

Recalling the definition of the jump moment in section 6.2.2 and in particular that of the first moment

$$a_1^i(\underline{X}) = \int (X_i^{'} - X_i) T\big(\underline{X}^{'}/\underline{X}\big) d\underline{X}^{'} \tag{6.13}$$

the right hand side (RHS) of Eq. (6.12) can be recast in the following form

$$\frac{d\langle X_i\rangle}{dt} = \int a_1^i(\underline{X}) P\big(\underline{X}, t\big) d\underline{X} \tag{6.14}$$

where the RHS term denotes the average of $a_1^i(\underline{X})$.

The ODE equation describing the dynamics of the first moment of variable X_i is then simply

$$\frac{d\langle X_i\rangle}{dt} = \big\langle a_1^i(\underline{X})\big\rangle \qquad i = 1, \ldots, N \tag{6.15}$$

The set of N Eq. (6.15) describes the dynamics of all the first moments of any N-dimensional Markov process. In most physical processes of interest all or the majority of $a_1^i(\underline{X})$ are nonlinear functions and therefore the RHS side depends on higher order moment terms. Assuming that $a_1^i(\underline{X})$ are polynomial functions of second order then the RHS will depend on second order moments. If some $a_1^i(\underline{X})$ are third order polynomials then third order moments also appear in the RHS of Eq. (6.15). The dependency of lower order dynamics on higher order moments is an important characteristic of moment schemes when the underlying physical process is non-linear. Its significance will be the focus of a subsequent section. In the rare case where $a_1^i(\underline{X})$ are linear functions then

Eq. (6.15) simplifies to

$$\frac{d\langle X_i\rangle}{dt} = a_1^i(\langle \underline{X}\rangle) \qquad i = 1, \ldots, N \tag{6.16}$$

We can now derive the second order moment equations. The starting point is the defining Eq. (6.5) from which the following identity is derived for the time derivative

$$\frac{d\langle X_i X_j\rangle}{dt} = \int X_i X_j \frac{\partial P\big(\underline{X}, t\big)}{\partial t} d\underline{X} \qquad i, j = 1, \ldots, N \tag{6.17}$$

Analogously to the derivation of the first moment equations, we substitute Eq. (6.2) in the last equation and by interchanging notation using the same arguments as previously (cf. eq. (6.11)). Then

$$\frac{d\langle X_i X_j\rangle}{dt} = \int\int (X_i^{'} X_j^{'} - X_i X_j) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \tag{6.18}$$

Through algebraic manipulations we can rearrange the terms in the last equation. In particular we use the following identity

$$(X_i^{'} X_j^{'} - X_i X_j) = (X_i^{'} - X_i)(X_j^{'} - X_j) + X_j(X_i^{'} - X_i) + X_i(X_j^{'} - X_j) \tag{6.19}$$

Substituting the term in the RHS of Eq. (6.18) yields

$$\begin{aligned}\frac{d\langle X_i X_j\rangle}{dt} &= \int\int (X_i^{'} - X_i)(X_j^{'} - X_j) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ &+ \int\int X_j(X_i^{'} - X_i) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ &+ \int\int X_i(X_j^{'} - X_j) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X}\end{aligned} \tag{6.20}$$

Invoking the definition of the joint jump moments (cf. eq. (6.7)) and also that of averages the RHS becomes

$$\frac{d\langle X_i X_j\rangle}{dt} = \big\langle a_2^{ij}(\underline{X})\big\rangle + \big\langle X_i a_1^j(\underline{X})\big\rangle + \big\langle X_j a_1^i(\underline{X})\big\rangle \qquad i, j = 1, \ldots, N \tag{6.21}$$

where $\underline{\underline{a}}_2(\underline{X})$ is a second order tensor, whereas $\underline{a}_1(\underline{X})$ is a first order tensor. In the trivial case where $i = j$ the last equation simplifies to

$$\frac{d\langle X_i X_i\rangle}{dt} = \langle a_2^{ii}(\underline{X})\rangle + 2\langle X_i a_1^i(\underline{X})\rangle \qquad i = 1, \ldots, N \tag{6.22}$$

Following a similar way of thinking and carrying out slightly more complicated algebraic calculations, which we omit in the present section for brevity but we include in Appendix A for completeness, we can write the moment equations for third and fourth order moment dynamics.

$$\begin{aligned} \frac{d\langle X_i X_j X_l\rangle}{dt} &= \langle a_3^{ijl}(\underline{X})\rangle \\ &+ \langle X_i a_2^{jl}(\underline{X})\rangle + \langle X_j a_2^{il}(\underline{X})\rangle + \langle X_l a_2^{ij}(\underline{X})\rangle \\ &+ \langle X_i X_j a_1^{l}(\underline{X})\rangle + \langle X_i X_l a_1^{j}(\underline{X})\rangle + \langle X_j X_l a_1^{i}(\underline{X})\rangle \\ \\ & i, j, l = 1, \ldots, N \end{aligned} \tag{6.23}$$

$$\begin{aligned} \frac{d\langle X_i X_j X_l X_m\rangle}{dt} &= \langle a_4^{ijlm}(\underline{X})\rangle \\ &+ \langle X_i a_3^{jlm}(\underline{X})\rangle + \langle X_j a_3^{ilm}(\underline{X})\rangle \\ &+ \langle X_l a_3^{ijm}(\underline{X})\rangle + \langle X_m a_3^{ijl}(\underline{X})\rangle \\ &+ \langle X_l X_m a_2^{ij}(\underline{X})\rangle + \langle X_j X_m a_2^{il}(\underline{X})\rangle + \langle X_j X_l a_2^{im}(\underline{X})\rangle \\ &+ \langle X_i X_m a_2^{jl}(\underline{X})\rangle + \langle X_i X_l a_2^{jm}(\underline{X})\rangle + \langle X_i X_j a_2^{lm}(\underline{X})\rangle \\ &+ \langle X_i X_j X_l a_1^{m}(\underline{X})\rangle + \langle X_i X_j X_m a_1^{l}(\underline{X})\rangle \\ &+ \langle X_i X_l X_m a_1^{j}(\underline{X})\rangle + \langle X_j X_l X_m a_1^{i}(\underline{X})\rangle \\ \\ & i, j, l, m = 1, \ldots, N \end{aligned} \tag{6.24}$$

Using inductive reasoning and Eq. (6.15), (6.21), (6.23) and (6.24) we derive the general moment equation for the m^{th} order moment of an N-dimensional Markov process as a function of joint jump moments

$$\frac{d\big\langle X_1^{m_1}X_2^{m_2}\cdots X_N^{m_N}\big\rangle}{dt}=\sum_{j_1,j_2\ldots,j_N=0}^{m_1,m_2\ldots,m_N}\binom{m_1}{j_1}\binom{m_2}{j_2}\cdots\binom{m_N}{j_N}\times$$
$$\times\big\langle X_1^{m_1-j_1}X_2^{m_2-j_2}\cdots X_N^{m_N-j_N}a_{j_1+j_2+\cdots j_N}^{index}(\underline{X})\big\rangle \tag{6.25}$$

with initial conditions

$$\big\langle X_1^{m_1}X_2^{m_2}\cdots X_N^{m_N}\big\rangle_{t=0}=\big\langle X_1^{m_1}(t=0)X_2^{m_2}(t=0)\cdots X_{N0}^{m_N}(t=0)\big\rangle$$

where by identity $a_0(\underline{X})=0$ and *index* notation is used to symbolize the following operation

$$index=\{\underbrace{1,\ldots,1}_{m_1},\underbrace{2,\ldots,2}_{m_2},\ldots,\underbrace{N,\ldots,N}_{m_N}\}-\{\underbrace{1,\ldots,1}_{j_1},\underbrace{2,\ldots,2}_{j_2},\ldots,\underbrace{N,\ldots,N}_{j_N}\}$$
$$=\{\underbrace{1,\ldots,1}_{m_1-j_1},\underbrace{2,\ldots,2}_{m_2-j_2},\ldots,\underbrace{N,\ldots,N}_{m_N-j_N}\} \tag{6.26}$$

In other words, the *index* notation refers to the appropriate element of the j^{th} order tensor, $j=j_1+j_2+\cdots j_N$.

To our knowledge there have been no other attempts to deduce the general moment equation from jump moments in the case of a N-dimensional Markov process.

In the case of a 1-dimensional Markov process the above equation reduces to the following simpler relation [108]

$$\frac{d\big\langle X^m\big\rangle}{dt}=\sum_{j=1}^{m}\binom{m}{j}\big\langle X^{m-j}a_j(\underline{X})\big\rangle$$

$$\big\langle X^m\big\rangle_{t=0}=\big\langle X^m(t=0)\big\rangle \tag{6.27}$$

The system of Eq. (6.25) completely characterizes the moment dynamics for any given N-dimensional Markov process. The only prerequisite is the knowledge of analytical relations for the jump moments. These relations depend on the underlying physics of the problem as it will become evident in the following section. In particular, we derive analytical relations for the components of the jump moments tensors in the case where the underlying Markov process stems from a stochastic chemical kinetics model. The final equations may appear cumbersome but they are very simple to generate with the help of a computer program.

Even though we have made the silent assumption that variables $\underline{X}(t)$ are continuous, this does not need to be the case. The derivation of equation (6.25) could have been carried out similarly by starting from the discrete ME [47], alleviating the need for continuous variables. The main difference in the derivation is that instead of integrals, summation signs over all possible states would be used.

6.2.4 Jump Moments and Stochastic Chemical Kinetics

Consider a system of N distinct chemical species, $X_i \, (i = 1, \ldots, N)$, participating in M chemical reactions in a well-mixed bacterial size volume V

$$\sum_{i=1}^{N} r_i^j X_i \xrightarrow{k_j} \sum_{i=1}^{N} p_i^j X_i, \qquad j = 1, \ldots, M \tag{6.28}$$

and where $\underline{\nu}_j$ is the stoichiometric vector associated with the j^{th} reaction

$$\underline{\nu}_j = \begin{bmatrix} p_1^j - r_1^j \\ \vdots \\ p_N^j - r_N^j \end{bmatrix} \qquad j = 1, \ldots, M \tag{6.29}$$

Such systems of reactions are widely used to model biological interactions, such as transcription, translation, degradation, regulation and protein-protein interactions [27, 90, 116, 30, 2]. Dilute and sparse species populations render the traditional continuous-deterministic modeling approach false. Instead, stochastic chemical kinetics models are more appropriate to describe systems of reactions that are far from the thermodynamic limit [37]. Therefore system (6.28) is frequently considered as a Markov process where a chemical master equation (CME) governs the evolution of the probability distribution.

Under the stochastic chemical kinetics regime reaction rates become reaction propensities, $\alpha_j(\underline{X})$. These are the probabilistic equivalents and are defined as follows

$$\alpha_j(\underline{X}) = k_j c_j(\underline{X}) \qquad c_j(\underline{X}) = \prod_{i=1}^{N} \frac{X_i!}{r_i^j\,!\,\left(X_i - r_i^j\right)!} \tag{6.30}$$

Each constant k_j represents the mesoscopic reaction rate of the j^{th} reaction.

In what follows we derive analytical relations for the jump moments for any stochastic chemical kinetics model. Starting with the CME, we can determine the Kramers-Moyal expansion of the probability distribution (cf. Eq. (6.3)). If we truncate the expansion and retain only the first two terms then the resulting equation is the well-known Fokker-Plank equation (FPE) [47]

$$\frac{\partial P(\underline{X},t)}{\partial t} = -\frac{\partial}{\partial \underline{X}}\left[\underline{a}_1(\underline{X})P(\underline{X},t)\right] + \frac{1}{2}\frac{\partial}{\partial \underline{X}}\frac{\partial}{\partial \underline{X}} : \left[\underline{\underline{a}}_2(\underline{X})P(\underline{X},t)\right], \tag{6.31}$$

where $\underline{\underline{a}}_2$ and $\underline{a}_1$ are the first two jump moments tensors or using the terminology most often referred to them the drift vector and diffusion tensor respectively. Instead of solving the Fokker-Planck equations it is usually more convenient to sample the underlying probability distribution generating ensembles of trajectories obtained as solutions of the corresponding chemical Langevin equations (CLE) or systems of CLEs [65].

From the work of Gillespie we know that for systems of chemical reactions under the Markov process regime the corresponding chemical Langevin equation is [65]

$$d\underline{X} = \underline{\underline{\nu}}\;\underline{\alpha}(\underline{X})dt + \underline{\underline{\nu}}\;\underline{\underline{\mathcal{D}}}\big(\sqrt{\underline{\alpha}(\underline{X})}\big)\underline{dW}, \tag{6.32}$$

where $\underline{\underline{\nu}}$ corresponds to the $N \times M$ stoichiometric matrix, $\underline{\alpha}(\underline{X})$ corresponds to the $M \times 1$ propensities vector and the notation $\underline{\underline{\mathcal{D}}}\big(\sqrt{\underline{F}}\big)$ denotes the diagonal matrix whose $(i,i)^{th}$ element are the only nonzero elements and their value equals the square root of the i^{th} component of vector $\underline{F}$.

Given any FPE we can deduce the corresponding system of CLEs and vice versa. This direct relation between the two allows us to infer through inductive reasoning analytical relations for the jump moment tensors elements. From Eq. (6.32) we obtain the following representation for the FPE

$$\frac{\partial P(\underline{X},t)}{\partial t} = -\frac{\partial}{\partial \underline{X}}\Big[\underline{\underline{\nu}}\;\underline{\alpha}(\underline{X})P(\underline{X},t)\Big] + \frac{1}{2}\frac{\partial}{\partial \underline{X}}\frac{\partial}{\partial \underline{X}} : \Big[\underline{\underline{\nu}}\;\underline{\underline{\mathcal{D}}}\big(\underline{\alpha}(\underline{X})\big)\underline{\underline{\nu}}^T P(\underline{X},t)\Big] \tag{6.33}$$

Comparing Eq. (6.31) and (6.33) we retain the following relations for the first two jump moment tensors

$$\begin{aligned} \underline{a}_1(\underline{X}) &= \underline{\underline{\nu}}\;\underline{\alpha}(\underline{X}) \\ \underline{\underline{a}}_2(\underline{X}) &= \underline{\underline{\nu}}\;\underline{\underline{\mathcal{D}}}\big(\underline{\alpha}(\underline{X})\big)\;\underline{\underline{\nu}}^T \end{aligned} \tag{6.34}$$

or using summation notation each of the tensor elements is defined as follows

$$\begin{aligned} a_1^i(\underline{X}) &= \sum_{k=1}^{M} \nu_{ik}\alpha_k(\underline{X}) \\ a_2^{ij}(\underline{X}) &= \sum_{k=1}^{M} \nu_{ik}\nu_{jk}\alpha_k(\underline{X}) \end{aligned} \tag{6.35}$$

We notice that the difference between the two jump moments relations relies on an additional component of the stoichiometric vector. Following the discussion in the previous section we infer the relations for the third and fourth jump moments, where $a_3(\underline{X})$ and $a_4(\underline{X})$ are $N \times N \times N$ and $N \times N \times N \times N$ tensors respectively.

$$\begin{aligned} a_3^{ijl}(\underline{X}) &= \sum_{k=1}^{M} \nu_{ik}\nu_{jk}\nu_{lk}\alpha_k(\underline{X}) \\ a_4^{ijlm}(\underline{X}) &= \sum_{k=1}^{M} \nu_{ik}\nu_{lk}\nu_{jk}\nu_{mk}\alpha_k(\underline{X}) \end{aligned} \tag{6.36}$$

In general, for stochastic chemical kinetics models we infer that given the stoichiometric matrix and the reaction propensities vector all jump moments can be defined analytically through the following recursive formula

$$a_m^{\overbrace{ij\cdots l}^{m\ indeces}}(\underline{X}) = \sum_{k=1}^{M} \underbrace{\nu_{ik}\nu_{jk}\cdots\nu_{lk}}_{m\ \nu's}\alpha_k(\underline{X}) \tag{6.37}$$

The formula is relatively simple and intuitive. Substituting the last equation into Eq. (6.25) returns analytical relations for the moment equations. A first obvious comment is that the linear or non-linear character of the underlying reaction networks manifests itself through the reaction propensities which in turn render jump moments linear or non-linear functions of the chemical species concentrations. The impact of the non-linear jump moment relations will become evident in the examples following. In general, a similar approach may lead to analytical relations for jump moments for any given Markov process.

6.3 Examples

The aim of this section is to highlight the analytical form of the moment equations for stochastic chemical kinetic systems through illustrative examples. For this reason we use two toy examples, a linear and a non-linear reaction network. Both these reaction motifs can be thought of as components of a larger chemical kinetics model which may be used to capture the dynamic behavior of a biomolecular interactions network. Note that the corresponding systems of ODEs are identical with the ODE systems obtained through an alternative derivation of the moment equations [112]. Note that the moment scheme can be easily implemented using any numerical computing environment such as Matlab.

6.3.1 Linear Kinetics

In volume, V, consider the reversible reaction

$$A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B \tag{6.38}$$

The reaction may, for example, represent the transition between the active and non-active state of a protein molecule. The state vector of the system is $\underline{X}(t) = [X_A\ X_B]^T$ while the corresponding stoichiometric matrix and the reaction propensities vector are

$$\underline{\underline{\nu}} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \underline{\alpha}(\underline{X}) = \begin{bmatrix} k_1 X_A \\ k_2 X_B \end{bmatrix} \tag{6.39}$$

We observe that reaction propensities are linear functions of the state variable as expected from the linearity of the underlying reaction network. Through Eq. (6.39) and (6.37) we can compute all the jump moments tensors, up to any desired order.

$$\underline{a}_1(\underline{X}) = \begin{bmatrix} -k_1X_A + k_2X_B \\ k_1X_A - k_2X_B \end{bmatrix}$$
$$\underline{\underline{a}}_2(\underline{X}) = \begin{bmatrix} k_1X_A + k_2X_B & -k_1X_A - k_2X_B \\ -k_1X_A - k_2X_B & k_1X_A + k_2X_B \end{bmatrix} \tag{6.40}$$
$$\vdots$$
$$a_m^{\overbrace{ij\cdots l}^{m\ indeces}}(\underline{X}) = \sum_{k=1}^{M} \underbrace{\nu_{ik}\nu_{jk}\cdots\nu_{lk}}_{m\ \nu's}\alpha_k(\underline{X})$$

Higher order jump moments are hard to write down as they are m_{th} order tensors, but their elements can be easily computed through (6.37). Notice that jump moment elements are also linear functions of the state variables. This feature, the linearity of the jump moments functions, will prove to be very useful in the generation of the desired moment equations.

Combining (6.40) and (6.25) we obtain the corresponding system of moment equations up to any desired order. In the moment equations the state vector refers to moments of the state variables, i.e.

$$\underline{Y}(t) = \begin{bmatrix} \langle X_A\rangle & \langle X_B\rangle & \langle X_A^2\rangle & \langle X_AX_B\rangle & \langle X_B^2\rangle & \langle X_A^3\rangle & \langle X_A^2X_B\rangle & \cdots \end{bmatrix}^T \tag{6.41}$$

In principle the dimensionality of $\underline{Y}(t)$ is infinite but for practical purposes we only require a finite number of moments to obtain all necessary statistical information and approximately reconstruct the probability distribution with a relatively small error tolerance.

The moment equations for the first nine elements of $\underline{Y}(t)$, i.e. up to third order moments, using matrix notation are

$$\frac{d}{dt}\begin{bmatrix} \langle X_A\rangle \\ \langle X_B\rangle \\ \langle X_A^2\rangle \\ \langle X_AX_B\rangle \\ \langle X_B^2\rangle \\ \langle X_A^3\rangle \\ \langle X_A^2X_B\rangle \\ \langle X_AX_B^2\rangle \\ \langle X_B^3\rangle \end{bmatrix} = \begin{bmatrix} -k_1 & +k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_1 & -k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_1 & k_2 & -2k_1 & 2k_2 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & -k_2 & k_1 & -(k_1+k_2) & k_2 & 0 & 0 & 0 & 0 \\ k_1 & k_2 & 0 & 2k_1 & -2k_2 & 0 & 0 & 0 & 0 \\ -k_1 & k_2 & 3k_1 & 3k_2 & 0 & -3k_1 & 3k_2 & 0 & 0 \\ k_1 & -k_2 & -2k_1 & k_1-2k_2 & k_2 & k_1 & -(2k_1+k_2) & 2k_2 & 0 \\ -k_1 & k_2 & k_1 & k_2-2k_1 & -2k_2 & 0 & 2k_1 & -(k_1+2k_2) & k_2 \\ k_1 & -k_2 & 0 & 3k_1 & 3k_2 & 0 & 0 & 3k_1 & -3k_2 \end{bmatrix} \begin{bmatrix} \langle X_A\rangle \\ \langle X_B\rangle \\ \langle X_A^2\rangle \\ \langle X_AX_B\rangle \\ \langle X_B^2\rangle \\ \langle X_A^3\rangle \\ \langle X_A^2X_B\rangle \\ \langle X_AX_B^2\rangle \\ \langle X_B^3\rangle \end{bmatrix} \tag{6.42}$$

Note that the equations form a linear system of ODEs that can be easily integrated given any initial condition. Since the system is linear the equations describing the means correspond to the equations that we obtain applying mass action kinetics principles.

6.3.2 Non-Linear Kinetics

In volume V, consider the non-linear reversible reaction

$$A + B \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \tag{6.43}$$

The state vector of the system is $\underline{X}(t) = [X_A \; X_B \; X_C]^T$ while the corresponding stoichiometric matrix and the reaction propensities vector are

$$\underline{\underline{\nu}} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \underline{\alpha}(\underline{X}) = \begin{bmatrix} k_1 X_A X_B \\ k_2 X_C \end{bmatrix} \tag{6.44}$$

Notice that the reaction propensities vector contains nonlinear functions of the state vector. As a consequence jump moments will also be non-linear functions of the state vector. Using Eq. (6.44) and (6.37) we compute the first two jump moment tensors

$$\begin{aligned} \underline{a}_1(\underline{X}) &= \begin{bmatrix} -k_1 X_A X_B + k_2 X_C \\ -k_1 X_A X_B + k_2 X_C \\ k_1 X_A X_B - k_2 X_C \end{bmatrix} \\ \underline{\underline{a}}_2(\underline{X}) &= \begin{bmatrix} k_1 X_A X_B + k_2 X_C & k_1 X_A X_B + k_2 X_C & -k_1 X_A X_B - k_2 X_C \\ k_1 X_A X_B + k_2 X_C & k_1 X_A X_B + k_2 X_C & -k_1 X_A X_B - k_2 X_C \\ -k_1 X_A X_B - k_2 X_C & -k_1 X_A X_B - k_2 X_C & k_1 X_A X_B + k_2 X_C \end{bmatrix} \end{aligned} \tag{6.45}$$

while higher order moment tensor elements can be easily computed through relation (6.37).

Combining (6.45) and (6.25) the corresponding system of moment equations using matrix notation and up to second order moments is

$$\frac{d}{dt}\begin{bmatrix} \langle X_A \rangle \\ \langle X_B \rangle \\ \langle X_C \rangle \\ \langle X_A^2 \rangle \\ \langle X_A X_B \rangle \\ \langle X_A X_C \rangle \\ \langle X_B^2 \rangle \\ \langle X_B X_C \rangle \\ \langle X_C^2 \rangle \end{bmatrix} = \begin{bmatrix} 0 & 0 & k_2 & 0 & -k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_2 & 0 & -k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k_2 & 0 & k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_2 & 0 & k_1 & 2k_2 & 0 & 0 & 0 \\ 0 & 0 & k_2 & 0 & k_1 & k_2 & 0 & k_2 & 0 \\ 0 & 0 & -k_2 & 0 & -k_1 & -k_2 & 0 & 0 & k_2 \\ 0 & 0 & k_2 & 0 & k_1 & 0 & 0 & 2k_2 & 0 \\ 0 & 0 & -k_2 & 0 & -k_1 & 0 & 0 & -k_2 & k_2 \\ 0 & 0 & k_2 & 0 & k_1 & 0 & 0 & 0 & -2k_2 \end{bmatrix} \begin{bmatrix} \langle X_A \rangle \\ \langle X_B \rangle \\ \langle X_C \rangle \\ \langle X_A^2 \rangle \\ \langle X_A X_B \rangle \\ \langle X_A X_C \rangle \\ \langle X_B^2 \rangle \\ \langle X_B X_C \rangle \\ \langle X_C^2 \rangle \end{bmatrix} + $$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2k_1 & 0 & 0 \\ -k_1 & -k_1 & 0 \\ k_1 & 0 & -k_1 \\ 0 & -2k_1 & 0 \\ 0 & k_1 & -k_1 \\ 0 & 0 & 2k_1 \end{bmatrix} \begin{bmatrix} \langle X_A^2 X_B \rangle \\ \langle X_A X_B^2 \rangle \\ \langle X_A X_B X_C \rangle \end{bmatrix} \tag{6.46}$$

The last equation represents again a linear system of ODEs but in contrast to Eq. (6.42) the system is infinite dimensional. Note that lower order moments depend on higher order moments. More specifically, second order moments depend on third order moments. Apparently then the system of needed ODEs becomes infinite. Therefore an exact analytical or numerical solution is impossible. This is the direct consequence of non-linear jump moments functions, or, better, of the non-linear character of the underlying physical problem.

6.4 Discussion

6.4.1 Infinite Dimensional Moment Equations

For any given non-linear model, whether it is a chemical kinetics model or an electrical circuit model, jump moments will be non-linear functions of the state vector, hence the system of ODEs describing the moment evolution will be infinite dimensional.

The general form of the ODEs describing the moment dynamics of a stochastic chemical kinetics model up to a desired order can be summarized through the following matrix equation

$$\frac{d\underline{Y}}{dt} = \underline{\underline{A}}\,\underline{Y} + \underline{\underline{A}}_h\,\underline{Y}_h + \underline{B} \tag{6.47}$$

where $\underline{Y}$ is a vector containing the moment elements of interest and $\underline{\underline{A}}$ is a square matrix with appropriate dimensions. Its elements depend on the kinetic constants and the stoichiometric matrix of the reactions set. Matrix $\underline{\underline{A}}$ may be sparse depending on the connectivity of the system. Vector $\underline{B}$ contains the terms that depend on any zeroth order reactions and are always constant as the corresponding reaction rates are independent from the state vector. Finally, the term $\underline{\underline{A}}_h\,\underline{Y}_h$ denotes the dependence of lower order moments on higher order moments. Vector $\underline{Y}_h$ contains higher order moments and matrix $\underline{\underline{A}}_h$, which is not necessarily square as observed in section 6.3.2, depends on the reaction constants and stoichiometric matrix. $\underline{\underline{A}}_h\,\underline{Y}_h$ vanishes only when the underlying system is linear.

6.4.2 Moment Closure Schemes

As mentioned earlier Eq. (6.47) is not solvable when the system of reactions is non-linear, as the dependence on higher moments renders the system infinite. In this section we briefly discuss existing approaches, called moment closure techniques, that attempt to bypass the problem and obtain approximate solutions.

The main idea behind any moment closure scheme is to approximate the infinite dimensional system of ODEs depicted in Eq. (6.47) with a finite system of the form

$$\frac{d\underline{Y}_t}{dt} = \underline{\underline{A}}\,\underline{Y}_t + \underline{f}_h(\underline{Y}_t) + \underline{B} \tag{6.48}$$

where $\underline{Y}_t$ is a truncated version of the infinite dimensional vector $\underline{Y}$ (cf. eq. (6.47)). The last equation explicitly states that the term $\underline{\underline{A}}_h\, \underline{Y}_h$ in Eq. (6.47) should be substituted with a vector containing linear or non-linear functions of $\underline{Y}$, $\underline{f}_h(\underline{Y}_t)$, that would closely approximate the effect of the higher order moment terms. In other words, the effect of higher order moment terms is approximated through functions of lower order moment terms. This approximation renders the system finite but also non-linear as all the developed approaches consider $\underline{f}$ to be non-linear functions. At the same time the approximation introduces an error in the moment values with its significance depending on how well $\underline{f}$ follows $\underline{\underline{A}}_h\, \underline{Y}_h$.

Two major approaches exist on how the components of vector $\underline{f}$ are chosen. In the first, a shape for the underlying distribution is assumed, for example Gaussian or log-normal, which allows to establish relations between higher and lower order moments [117]. Advantages include easily determined relations between lower order and higher order terms that are not system specific. On the down side, if the assumption for the shape of the underlying distribution is not valid then the error would be significant. In practise it is difficult to have *a priori* knowledge on the state variables distributions, which may even change over the simulation time interval.

In a similar approach, it is usually convenient to assume that higher order central moments are negligible. This is indeed the case for normal distributions where central moments higher than two are always zero. For other distributions there is no intuitive way to distinguish which central moments can be neglected, as this would require to know the shape of the underlying distribution. The general approach followed is to consider central moments of order higher than the desired truncation order negligible. The truncation order refers to the highest order moment included in vector $\underline{Y}$. Setting them equal to zero provides non-linear relations between the higher order moments included in $\underline{Y}_h$ and those in $\underline{Y}_t$ [118, 114]. The advantage is that such relations are easily obtained but disadvantages include poor performance when the underlying distribution are anything but Gaussian. Usually there is a good estimation for the mean and the variance but higher order moments values deviate significantly. The significance of higher order moments is discussed in detail in section 6.4.3.

An alternative approach, developed recently, defines the functional form of vector $\underline{f}$ by imposing that the values and derivatives of vectors $\underline{Y}$ and $\underline{Y}_t$ differ by a small value

Table 6.1: Reactions and Parameters for the Schlögl Model

Set of Reactions[§]	Mesoscopic Reaction Rates[†]	Initial Values[‡]
$A + 2X \xrightarrow{k_1} 3X$	$k_1 = 3 \times 10^{-7}$	$\left[X\right]_0 = 247$
$3X \xrightarrow{k_2} 2X + A$	$k_2 = 10^{-4} \times 10^2$	
$B \xrightarrow{k_3} X$	$k_3 = 10^{-3}$	$\left[A\right] = 10^5$
$X \xrightarrow{k_4} B$	$k_4 = 3.5 \times 10^2$	$\left[B\right] = 2 \times 10^5$

[§] A and B are buffer species.
[†] for 1st order reactions the units are s^{-1}, for 2nd order reactions the units are $\text{molecules}^{-1}s^{-1}$ and for 3rd order reactions the units are $\text{molecules}^{-2}s^{-1}$.
[‡] initial values are in number of molecules.

ϵ [112]. The resulting functional form of $\underline{f}$ suggests that the underlying probability distributions are log-normal. In the case of stochastic chemical kinetics this is usually true when species concentrations are close to the origin. In all other cases there is deviation between actual and approximated moments, especially as the system departs from log-normal or gaussian behavior. Additionally higher order moments values exhibit large deviations and are mostly used to increase accuracy in the first two moments. When the system pertains complex non-linear dynamics, i.e. bimodality, the performance is suboptimal.

A summary of the existing moment closure techniques in stochastic chemical kinetics system can be found in reference [119]. To our knowledge there is no satisfactory solution to the moment closure scheme with broad applicability and the topic is still being actively researched.

6.4.3 Importance of Higher Moments in Non-linear Chemical Kinetics

In this section we determine the probability distribution of a simple non-linear system of chemical reaction to illustrate how higher than the first two order moments are necessary to capture the dynamic behavior. Consider the well studied Schlögl model that under specific values for the kinetic rates exhibits bistability, similar to the bistable switch [15]

Table 6.2: First Eight Moment Values for the Schlögl Model

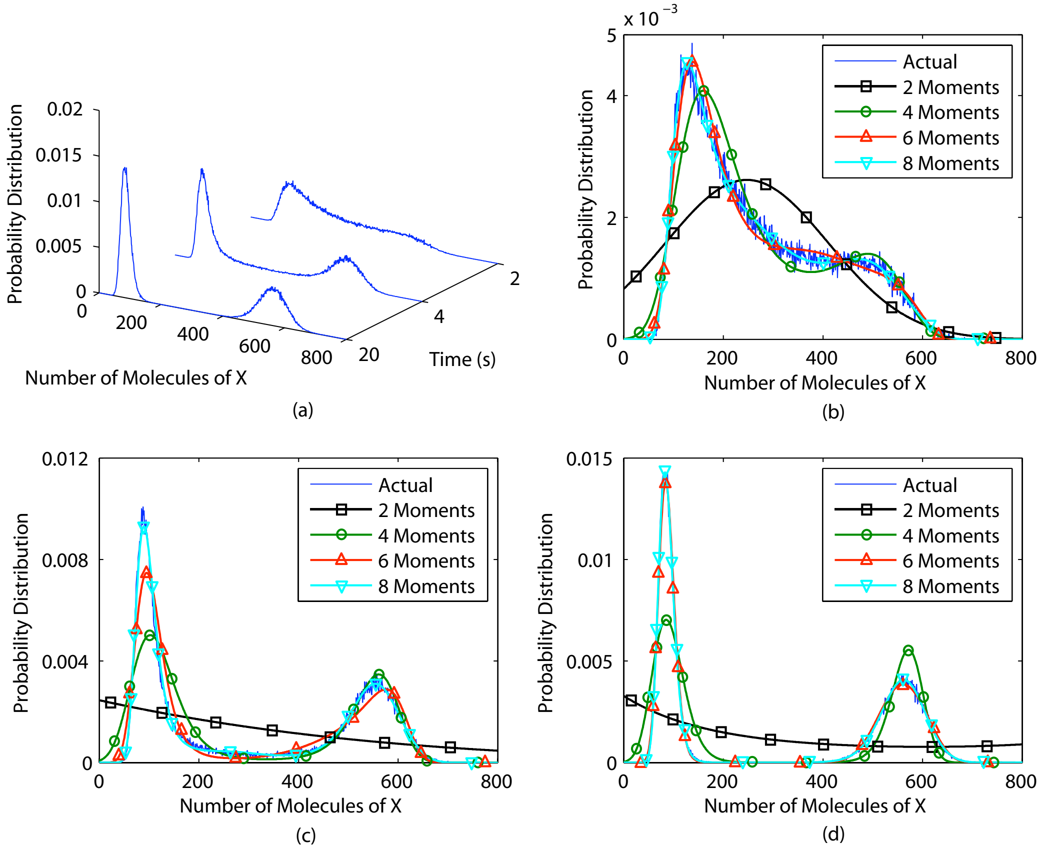
Moments	Value at t = 2 s	Value at t = 4 s	Value at t = 20 s
$\langle X \rangle$	268.9871	295.6759	301.6574
$\langle X^2 \rangle$	9.2903×10^4	1.3364×10^5	1.4733×10^5
$\langle X^3 \rangle$	3.7966×10^7	7.0001×10^7	8.1413×10^7
$\langle X^4 \rangle$	1.7179×10^{10}	3.8403×10^{10}	4.6203×10^{10}
$\langle X^5 \rangle$	8.2743×10^{12}	2.1488×10^{13}	2.6470×10^{13}
$\langle X^6 \rangle$	4.1528×10^{15}	1.2170×10^{16}	1.5262×10^{16}
$\langle X^7 \rangle$	2.1465×10^{18}	6.9578×10^{18}	8.8511×10^{18}
$\langle X^8 \rangle$	1.1347×10^{21}	4.0100×10^{21}	5.1617×10^{21}

and or the λ-phage infection [116]. Consider the network of reactions shown in Table 6.1. Kinetic data and initial conditions are taken from reference [120].

The kinetic values depicted in Table 6.1 render the system bistable. The use of the Schlögl model enables us to determine the minimum number of moments needed to reconstruct the underlying probability distribution within a reasonable error. Using Hy3S and SynBioSS [1, 4], a suite of multiscale algorithms, we generate 10^5 independent trajectory trials in the interval [0,20] s.

We use Matlab to compute the underlying probability distribution of species X at three different time points. The shapes and characteristics of the corresponding probability distributions are depicted in Fig. 6.1(a). Next we compute the first eight moments of X at the three different time points using the data obtained from the stochastic simulation. Their values are shown in Table 6.2. Using the numerical values of the moments we can reconstruct the corresponding probabilities using an algorithm based on the maximum entropy principle and developed by Mohammad-Djafari [121].

In order to determine the minimum number of moments we first reconstruct the distributions using only the first two moments and then we keep increasing the number of moments by two and up to order eight.



(a)

(b)

(c)

(d)

Figure 6.1: Actual and reconstructed probability distributions for the Schlögl model, depicted in Table 6.1, using the moment data in Table 6.2. (a) Probability distribution of species X at times t = 2 s, 4s and 20 s. (b) Comparison between the actual and reconstructed probability distributions using different moment sets at time t = 2 s. (c) Comparison between the actual and reconstructed probability distributions using different moment sets at time t = 4 s. (d) Comparison between the actual and reconstructed probability distributions using different moment sets at time t = 20 s.

In Fig. 6.1(b), 6.1(c) and 6.1(d) the comparison between the actual and reconstructed probabilities are depicted. Reconstructed probabilities using only the first two moments expectedly fail to capture the bimodality of the distribution especially in the cases of t = 4 s and 20 s. Using four moments seems to capture the essential futures of the distributions, i.e. the two modes, but it fails to weight in the relative peaks of the two modes. Six moments produce adequate results, especially when the separation between the two modes is distinct (cf. Figure 6.1(d)), but results are inferior when there is no distinct separation (cf. Fig. 6.1(b) and 6.1(c)). Still results are relatively accurate. Finally, using the first eight moments produces distributions that are almost identical to the actual distribution with minor, if any, deviations.

Overall, it is evident that using the first two moments is not adequate to reconstruct the probability distribution. Our analysis shows that use of at least the first six moments and in some case eight is needed.

6.5 Summary

A new derivation of the moment equations for any N-dimensional Markov process using the definition of jump moments was presented. The applicability of the scheme is general and leads to analytical relations given that the functional form of the jump moments is known.

Focusing on stochastic chemical kinetics models, we derived analytical relations for the elements of any jump moment tensor, demonstrating the ease of equation setup. The elements of the analytical relations are a function of the stoichiometric matrix and the reaction propensities, i.e the probabilistic reaction rates.

Using two toy examples, a linear and a non-linear set of reactions, we demonstrated the applicability of the jump-moments derivation. Through the two examples we established that when the underlying system contains non-linear dynamics the corresponding system of moment equations becomes an infinite dimensional system of linear ODEs. We briefly mentioned already developed moment closure schemes, highlighting the need for a closure scheme that has broad applicability and produces accurate moment values.

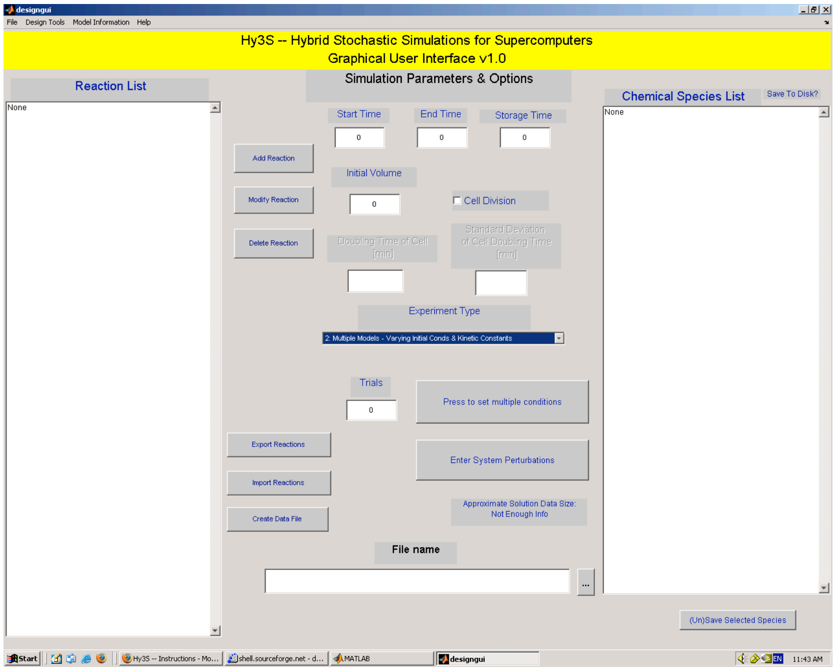
Using a simple example of non-linear kinetics we illustrated how higher than second order moments are important to accurately reconstruct the probability distribution for biologically-relevant chemical kinetics systems. Certainly, then, more effort is warranted on the development of moment closure schemes.

Chapter 7

Software Tools for Computer-Aided Design of Synthetic Gene Networks

7.1 Guided User Interfaces for Gene Network Engineering

In this dissertation the focus is on developing multiple time scale algorithms that can be used for computer-aided design (CAD) of synthetic gene networks. The prospect is to eventually put these algorithms to the test and attempt to use them for CAD of synthetic gene networks. This endeavor requires the algorithms to be publicly available and most importantly in a user friendly form, where even those with limited programming knowledge can use them. Our main target group is biologists and our aspiration is for them to use our algorithms actively. For that, our group has concentrated in producing software tools that intent to simplify computer-aided design of gene networks. Nowadays there is a large number of available software tools and a good source for many of the application is the systems biology markup language (SBML) website, `http://sbml.org/SBML_Software_Guide`.



designgui
File Design Tools Model Information Help
Hy3S -- Hybrid Stochastic Simulations for Supercomputers
Graphical User Interface v1.0
Simulation Parameters & Options
Add Reaction
Modify Reaction
Delete Reaction
Export Reactions
Import Reactions
Create Data File
Press to set multiple conditions
Enter System Perturbations
Save To Disk?
(Un)Save Selected Species
...
Start
Hy3S -- Instructions - Mo...
shell.sourceforge.net - d...
MATLAB
designgui
EN
11:43 AM

Reaction List: None

Start Time: 0
End Time: 0
Storage Time: 0
Initial Volume: 0
□ Cell Division
Doubling Time of Cell [min]:
Standard Deviation of Cell Doubling Time [min]:
Experiment Type: 2: Multiple Models - Varying Initial Conds & Kinetic Constants
Trials: 0
Approximate Solution Data Size: Not Enough Info
File name:

Chemical Species List: None

Figure 7.1: Screenshot of the main GUI window of Hy3S.

7.2 Hybrid Stochastic Simulation for Supercomputers

A user friendly program that implements the hybrid stochastic algorithm discussed in Sec. 3.2, called Hy3S short for hybrid stochastic simulation for supercomputers, is readily available [1]. Hy3S incorporates a Graphical User Interface (GUI) which makes easier for the user to enter the reactions and determine model parameters, such as the number of trials, time internal, save times etc. A screenshot of the main GUI window is presented in Fig. 7.1

Hy3S incorporates also some interesting features. It utilizes MPI, treats special events such as cell division sufficiently and encompasses options for introducing perturbation. For example the user can choose to perturb the reacting species concentrations, kinetic constants or both during the simulations. Additionally, sensitivity analysis can be performed through a combinatorial scheme allowing for running multiple simulations with varying initial concentrations and kinetic constants. Since large networks of reactions can be simulated with Hy3S, an optimized binary format (NetCDF) accounts for storing and handling large simulations data. All the above features make Hy3S a very attractive simulation platform for biological systems.

The algorithms upon which Hy3S is build are mostly described in Sec. 3.2. Hy3S uses fixed or adaptive time step schemes to numerically integrate the coupled system of CLEs and differential Jump equations. The adaptive scheme is part of the work conducted under the scope of the present dissertation and has been extensively discussed in Chapter 4.

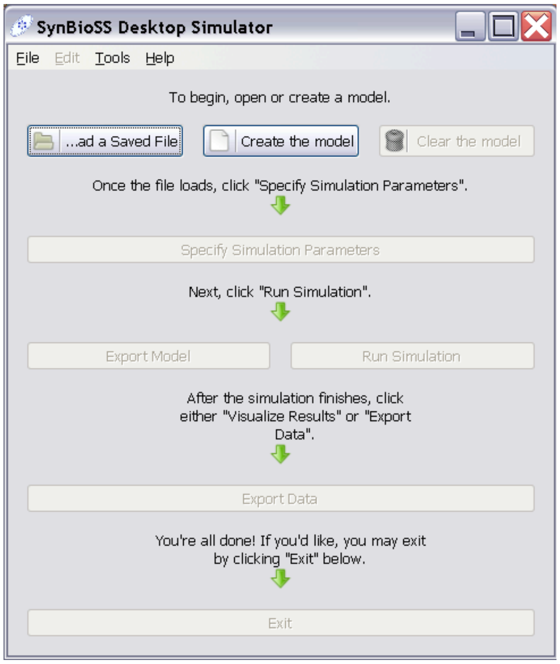
Hy3S is among the pioneering software tools in the field. Nonetheless, limitations are still present. For instance the GUI is built using the Matlab environment, which obliges any potential user to have access to a commercial software. Additionally, there is no embedded visualizing software thus a third party application is required for post processing.

7.3 Synthetic Biology Software Suite

In an attempt to overcome the limitations while also strengthening the computer-aided design capabilities of Hy3S, our group developed the synthetic biology modeling suite, known as SynBioSS [4]. SynBioSS is capable of creating models of arbitrary synthetic gene constructs. The modeling suite is built upon three distinct entities. Each of them will be briefly presented in the following sections. Detailed instructions along with useful examples regarding SynBioSS can be found on the project's website, `http://synbioss.sourceforge.net/`.

7.3.1 Synthetic Biology Software Suite: Desktop Simulator

The first entity is the SynBioSS Desktop Simulator (SynBioSS DS), a cross-platform (Windows, Mac OS and Linux are supported) desktop application. SynBioSS DS uses the same algorithms as Hy3S to efficiently and accurately simulate dynamics of synthetic gene constructs. Much effort has been spend in creating an even friendlier user interface compared to Hy3S. A typical screenshot of the GUI is depicte in Fig 7.2. The dependence on Matlab is no longer present while there is also a proprietary visualizing application for post-processing of simulation results. NetCDF file support is kept intact and SBML [122] file support has been added to accommodate the fact that SBML has become the standard markup language for representing chemical kinetics models.



To begin, open or create a model.

Once the file loads, click "Specify Simulation Parameters".

Next, click "Run Simulation".

After the simulation finishes, click either "Visualize Results" or "Export Data".

You're all done! If you'd like, you may exit by clicking "Exit" below.

Figure 7.2: Screenshot of the main GUI window of SynBioSS Desktop Simulator.

7.3.2 Synthetic Biology Software Suite: Wiki

The second entity is the SynBioSS Wiki. The Wiki is a searchable and curated database, similar to wikipedia, that stores reaction kinetic data necessary to build chemical kinetics models to represent biological interactions. The Wiki is a completely web-based application (`http://synbioss.sourceforge.net/`) that addresses the critical limitation of the chemical kinetics approach, namely the sparsity and lack of easily available kinetic data. A very attractive feature of the SynBioSS Wiki is the ability to collect and combine reactions along with their corresponding kinetic rates to create desired chemical kinetics models on the fly. Models can then be exported in SBML format and simulated using the SynBioSS Desktop Simulator.

7.3.3 Synthetic Biology Software Suite: Designer

The last entity in the SynBioSS modeling suite is the SynBioSS Designer (`http://synbioss.sourceforge.net/`). Similar to Wiki the Designer is also a web based application that combines available data from the Wiki, predefined design rules and BioBricks mentality to predict or suggest a cascade of reactions that can be used as a model for a given system of gene network. In other works, the Designer uses as inputs networks of standardized parts, i.e. BioBricks [123], that form the synthetic gene network under investigation. Then based on a set of design principles the software generates the appropriate chemical kinetics model and assigns the suitable kinetic rates using the Wiki database. The model is then exported in either NetCDF or SBML file format and imported in the Desktop Simulator.

The aspiration of the Designer project is synonymous to computer-aided design. The Designer encompasses and embraces our group's philosophy of how state of the art algorithms, wrapped around user friendly interfaces, can greatly benefit the advancement of synthetic biology and gene network engineering in particular.

Chapter 8

Synthetic Tetracycline-Inducible Regulatory Networks: Computer Aided Design of Dynamic Phenotypes

8.1 Introduction

In this chapter we model novel tetracycline-inducible regulatory gene networks using the principles of computer-aided design and the software tools presented in the previous chapters. This part of the dissertation conveys how sophisticated algorithms can reduce the vast amount of possible design alternatives by rapidly and rationally shifting through possible design combinations. In every step, we propose, test, and accept or reject design principles for each alternative, eventually developing design principles for novel tetracycline-inducible gene networks.

Tightly regulated gene networks, precisely controlling the expression of protein molecules, have received considerable interest [10] by the biomedical community due to their promising applications. Recently, regulatory gene networks have been used in exciting biomedical applications, such as delivery of therapeutic genes, treating cancer, diabetes, and other diseases [124, 125]. Proposed designs have been tested both *in vitro* and *in vivo* [124, 125], leading to encouraging results.

Desirable characteristics of a fine tuned system include silent expression in the absence of inducer (low expression leakiness), high induced expression, high specificity and sensitivity to inducers, quick response to inducers, regulation by an orally bioavailable inducer, minimal or no immune impact to the host and finally *in vivo* applicability. The most widely used inducible transcription systems that largely meet these criteria are the tetracycline regulatory expression systems based on the tetracycline resistance operon of *Escherichia coli* (*E. coli*) [126]. Tet-Off and Tet-On systems, also known as rTA and rtTA, respectively, are among the most well studied systems of this category [127, 128, 14, 129].

Tet-Off, first employed by Gossen and Bujard is a binary transgenic system in which expression of a target transgene is dependent on the activity of an inducible transcriptional activator [14]. The transcriptional activator is a tetracycline-controlled transactivator protein (tTA), which is a fusion between the Tet repressor DNA binding protein (TetR) and a transactivator, such as VP16 of the herpes simplex virus. The target gene is under transcriptional control of a tetracycline-responsive promoter element (TRE), a seven Tet operators (TetO) moiety placed upstream of a minimal promoter, typically derived from the human cytomegalovirus (hCMV). Expression of the transgene can be regulated both reversibly (expression is turned back on again when tetracycline has cleared out of the system) and quantitatively by exposing the system to varying concentrations of tetracycline (Tc), or Tc derivatives such as doxycycline (Dox) or minocycline. Transcription is silenced when tetracycline derivatives are administered, since TetR loses its affinity for TetO.

While Tet-Off requires the absence of Tc for expression of the transgene, in the Tet-On system the transgene is expressed when Tc or its analogues are present. Four amino acid substitutions on the TetR sequence led to reverse TetR, which binds TetO sequences only in the presence of Tc. Reverse TetR fused with a transactivator domain (rtTA) has the reverse phenotype of tTA, allowing transgene expression in the presence of Tc or its analogues. This last characteristic makes Tet-On systems more attractive than Tet-Off, since in general, organisms are more easily saturated with an inducer than depleted of it.

Despite their initial success, both systems (tTA and rtTA) still face limitations that need to be addressed before routinely using them in human gene therapies:

- High-level expression of Tet-OFF or Tet-ON transactivators might cause cellular toxicity, or selective pressure against the stable incorporation of vectors expressing the transactivators.

- Therapeutic gene expression leakage is still present despite the strength of biomolecular interactions comprising Tet-OFF or Tet-ON. For example, there is residual affinity of Tet-ON for TetO, even in the absence of Tc.

- Only traces of Tc or Dox appear to be sufficient for silencing expression in Tet-OFF, requiring days before the systems behavior is reversed.

- Fairly high levels of Dox are required for Tet-ON to be activated, a concentration that cannot be readily achieved in the brain of mice.

To address these issues, novel tetracycline regulated systems were engineered that display both low basal expression levels and higher affinity for Dox [130, 131]. Additionally acidic activation domains can replace VP16, creating a wide selection of possible transactivators. Another strategy employed to reduce basal activity was the fusion of TetR with a KRAB domain (tTS) [132], which led to repression of unwanted transgene when Tc was absent without affecting expression in the presence of Tc. Combined tTS and rtTA [133] systems demonstrate promising results. Moreover autoregulated expression vectors have been successfully used to control expression of Tet-OFF or Tet-ON [134]. Additional strategies working towards improving the original Tet designs include the use of adenovirus vector systems [135] and the use of histone deacetylases in mammalian cells [136].

Nonetheless, limitations persist. Instead of looking at these networks statically, and simply changing or mutating the promoter and operator regions with trial and error, a systematic investigation of the dynamic behavior of the network can result in rational design of regulatory gene expression systems. The observation that gene networks are inherently stochastic [37], allow to numerically simulate complex networks of regulated biological reactions. By combining fast supercomputers and greater knowledge of the molecular mechanisms of gene expression, we can numerically simulate the stochastic dynamics of gene networks and understand in depth how components of the network affect system level performance. Multiple time-scale algorithms [74] are able to accurate capture the dynamical behavior of complex gene networks, such as the bistable switch [29], the fim switch [27], the oscillator [30] and the lac operon [28].

Using computer simulations, we aim to facilitate rational synthesis of tetracycline-inducible regulatory networks and propose new designs that aim to address some of the limitations, for example enable tighter control of expression. We first generate four novel gene regulatory networks based on the tTA, rtTA and the wild type operon of *E. coli.* Then a chemical kinetics model based on the interactions present in the network is constructed and the dynamical behavior of the wild type network is simulated. The model consists of all distinct biomolecular interactions involved in transcription, translation, regulation and induction. The behavior is evaluated and design principles, such as mutations, are introduced, which aim in a fine tuned dynamical behavior. We propose, mutations in TetR sequence which affect both the relative binding affinity with TetO [137, 138, 139] and with tetracycline [140] allowing for a fine tuned design. Moreover, we suggest mutations in the TetO [141, 142, 143, 144] sequence that affect the relative binding affinity with TetR.

8.2 Four Novel Networks Based on the Tetracycline Resistance Operon of *E. coli*

Based on the components of Tet-Off, Tet-On and the tetracycline resistance operon of *E. coli* we introduce four novel model networks that address limitations present in current designs. By computationally identifying the important molecular interactions, the objective is to find ways to fine tune the dynamic response of the systems. We will mainly concentrate in controlling levels of repressor-transactivator proteins, response times and sensitivity to Tc or its analogues. A schematic representation of the proposed gene networks is shown in Fig. 8.1. The connectivity between network components in the absence or presence of Tc are shown in Fig. 8.2, where nodes represent genes and arrows represent repression, activation and blocking of activation.

Selection of network designs is a crucial subject in the present work. Based on the building blocks of the wild-type rTA and rtTA we constructed four networks with both different number of genes and components. The driving force behind the particular

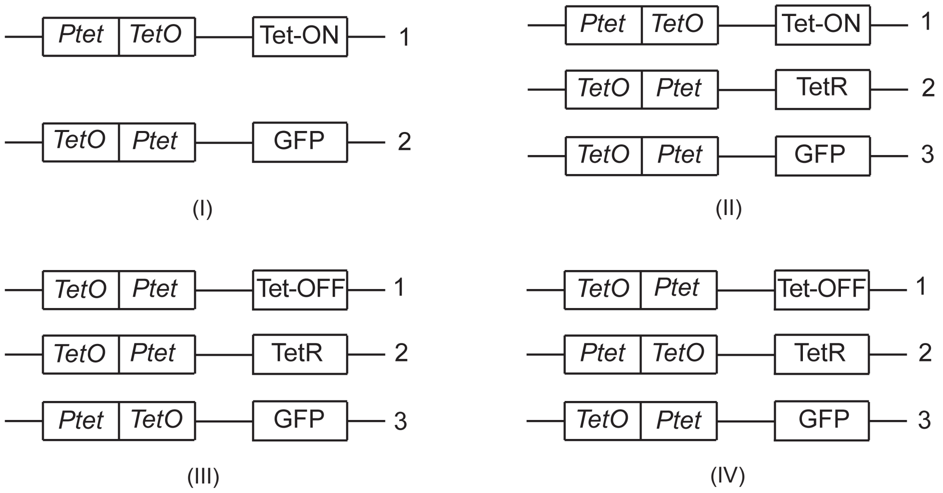


Figure 8.1: Schematic representation of the four regulatory gene networks showing the way components of the Tet-Off, Tet-On and the tetracycline resistance operon are combined. TetR: the wild type Tet repressor, TetO: the wild type Tet operator, Ptet: the wild type Tet Promoter, Tet-OFF: protein fusion of TetR with a transactivator domain, Tet-ON: protein fusion of reverse TetR with a transactivator domain and GFP: Green Fluorescent Protein.

design selection is mainly intuition. Intuition and the understanding of the existing design flaws lead to propose the present four networks. Of course there are millions other configurations we could have used, given the enormous amount of DNA sequences and proteins available, such as a combination of promoters from the lac and tetracycline resistance operon. But our objective was to keep the proposed networks as simple as possible for two reasons. First, at present it is hard to experimentally realize gene networks with more than three genes and secondly as nature has taught us, simple is always better.

The major components of the four networks are the wild type Tet repressor DNA binding protein (TetR), the wild type Tet operator (TetO), the wild type Tet promoter (Ptet), the Tet-OFF protein (fusion of TetR with an appropriate transactivator domain) and the Tet-ON protein (fusion of reverse TetR with an appropriate transactivator domain). Note that the proteins Tet-OFF and Tet-ON are written using capital letters for

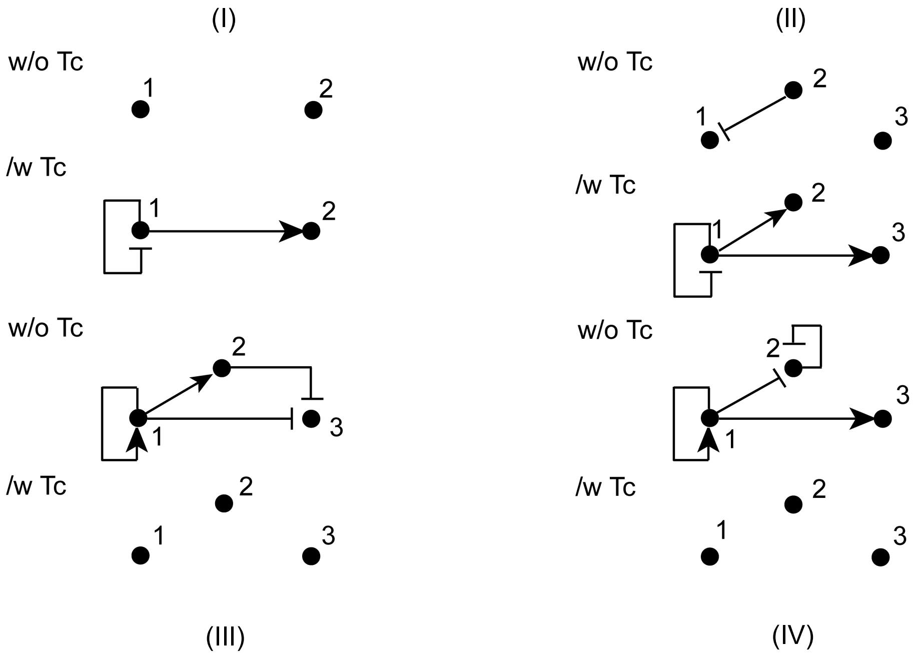


Figure 8.2: Schematic representation of the network connectivities in the presence or absence of Tc. Nodes represent genes and are numbered according to Fig. 8.1. Arrows represent repression or activation.

ON and OFF in contrast to the Tet-Off and Tet-On systems where lower cases are used. When TetO is located upstream of Ptet we assume that there is no overlapping of the two sequences, with the proximity of the two being appropriate for the transactivators to interact with the transcriptional machinery. The promoter is not silenced while TetR is bound to TetO; RNA polymerase can still be recruited. Green Fluorescent Protein (GFP) is used as a reporter gene in our networks. In practical gene therapy applications, a therapeutic gene can replace GFP.

The complexity and the degree of uncertainty we have for mammalian transcription and translation stages prohibit us currently from simulating the gene networks in mammalian cells. Therefore we study the time evolution of the networks within bacteria, in particular *E. coli* where molecular mechanisms are less complex and well studied. We use strings of *E. coli* that do not contain the natural tetracycline resistance operon. Our models incorporate all individual molecular species and interactions known to be involved in the transcription, translation, regulation, and induction steps in the tetracycline-regulated expression system. For example we end up with 93 reactions that model all the individual biomolecular interaction events in Network III. The detailed network of reactions, along with a short description can be found in the Methods section of the present chapter (cf. Sec. 8.3.1).

We start by looking into the wild type dynamical behavior of each network, using chemical kinetics models. This approach allows us to look into the molecular level and investigate how species concentrations vary over time and how they affect the actual phenotype. First we determine important interactions and secondly we propose ways to manipulate sequences and binding affinities to achieve design goals. Although intuition can help us to decide what new designs to construct based on qualitative arguments, it is the insight of the molecular level that guides us to propose changes that will attempt to address limitations and also lead to design rules for fine tuned systems. Experimentally realizable changes include the use of new TetR or reverse TetR [137, 138, 139] and TetO [141, 142, 143, 144] variants as well as TetR variants that do not bind Tc [140]. Effects are studied and the suggested changes are accepted, rejected, or combined.

8.3 Methods

8.3.1 Chemical Kinetics Models

Representing the physicochemical interactions between biomolecules, such as recruitment of RNA polymerase on promoter sites, as a set of chemical reactions enables us to study the time evolution of a gene network using stochastic algorithms. The knowledge for the molecular mechanism of transcription and translation provides us with enough insight to classify interactions in the molecular level as first order, second order, Michaelis Menten type, etc. reactions. Reversible phenomena, as binding and unbinding of tetracycline to TetR, are represented as two separate reactions (forward and reverse reactions). Special events as transcriptional elongation are modeled as gamma distributed events [53], whereas interactions between three or more species, where one of the species has a binary state, (one or zero number of molecules, for instance non occupied and occupied operators) are assumed to follow power law kinetics.

As an example, consider the synthetic Network III (cf. Fig. 8.1). The network consists of 63 species, participating in 93 distinct chemical reactions. In Table 8.1, we present all reactions with their kinetic parameters. These parameters are largely taken from the existing literature and others adjusted to give specific values, for example the rates of mRNA half-life and mRNA ribosome binding (initialization of translation) are adjusted to produce approximately 20 protein molecules per mRNA transcript. Table 8.1 represents the wild-type behavior of the genes. We will briefly describe how we assigned the appropriate kinetic data to the set of reactions. For brevity we will focus on the reactions depicted by design III (cf. Table 8.1), but the approach is similar for the rest of the networks.

Administration of Tc into the medium causes diffusion through the cytoplasmic membrane of *E. coli* [145]. The process has a half-equilibration time of approximately 35 ± 15 min and is modeled as first order chemical reaction (k = 3.3×10^4 s^{-1}).

Dimerization of TetR and Tet-OFF are reversible reactions and their equilibrium constants, in the absence of specific experimental information, are assumed to be similar to lac [146]. Binding of tetracycline to TetR is also a reversible phenomenon and equilibrium constants are readily available in the literature [126, 147]. In the case of Tet-OFF, we assume it has the same binding affinity for Tc as TetR, a reasonable assumption if one notes that the inducer binding domain of TetR is not affected when the transactivator is added. Each TetR or Tet-OFF dimer requires two molecules of Tc to be fully induced. Due to the stochastic nature of the algorithm it is in general difficult to model a reaction where more than two species are simultaneously involved. We break down the reaction of the two Tc molecules with either one TetR or Tet-OFF dimer into two steps. In the first step, one Tc molecule reacts with one TetR/Tet-OFF dimer molecule with rate constant 2.0×10^6 M^{-1}s^{-1} and in the next step the formed complex reacts very fast (1.0×10^{15} M^{-1}s^{-1})with another Tc molecule to form the fully induced complex. It is obvious that the first step is the rate limiting one and that the underlying assumption is that Tc induction depends linearly on the concentration of Tc. Finally, due to the short life of the intermediates we do not consider them degrading.

Binding constants for TetR and Tet-OFF dimers in the operator sequence (TetO) are available in the literature [147, 148]. As previously, we assume a similar behavior between the two dimers. Presence of Tc causes TetR or Tet-OFF dimmers bound to an operator to unbind faster [149], meaning that that the complexes have a smaller half-life

Table 8.1: A Chemical Kinetics Representation of Network III

#	Reaction	k	Ref.
1	TcEx $\longrightarrow$ Tc	3.3×10^{-4}	[145]
2	2 TetR $\longrightarrow$ TetR2	1.0×10^{9}	[146]
3	TetR2 $\longrightarrow$ 2 TetR	10.0	[146]
4	2 TetOFF $\longrightarrow$ TetOFF2	1.0×10^{9}	[146]
5	TetOFF2 $\longrightarrow$ 2 TetOFF	10.0	[146]
6	Tc + TetR2 $\longrightarrow$ Tc:2TetR	2.0×10^{6}	[126]
7	Tc + Tc:2TetR $\longrightarrow$ Tc:TetR2	1.0×10^{15}	§
8	Tc:TetR2 $\longrightarrow$ TetR2 + 2 Tc	1.0×10^{-3}	[126]
9	Tc + TetOFF2 $\longrightarrow$ Tc:2TetOFF	2.0×10^{6}	[126]
10	Tc + Tc:2TetOFF $\longrightarrow$ Tc:TetOFF2	1.0×10^{15}	§
11	Tc:TetOFF2 $\longrightarrow$ TetOFF2 + 2 Tc	1.0×10^{-3}	[126]
12	TetOFF $\longrightarrow$ ∅	1.2×10^{-3}	∗
13	TetOFF2 $\longrightarrow$ ∅	1.2×10^{-3}	∗
14	TetR $\longrightarrow$ ∅	1.2×10^{-3}	∗

Continued on Next Page...

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
15	TetR2 $\longrightarrow \varnothing$	1.2×10^{-3}	$*$
16	Tc:TetOFF2 $\longrightarrow$ 2 Tc	1.2×10^{-3}	$*$
17	Tc:TetR2 $\longrightarrow$ 2 Tc	1.2×10^{-3}	$*$
18	GFP-LAA $\longrightarrow \varnothing$	2.88×10^{-4}	[150]
19	Tc $\longrightarrow \varnothing$	2.67×10^{-6}	[151]
20	TcEx $\longrightarrow \varnothing$	2.67×10^{-6}	[151]
21	TetOFF2 + OP1 $\longrightarrow$ TetOFF2:OP1	2.86×10^{6}	[148]
22	TetOFF2:OP1 $\longrightarrow$ TetOFF2 + OP1	5.11×10^{-4}	[148]
23	TetR2 + OP1 $\longrightarrow$ TetR2:OP1	2.86×10^{6}	[148]
24	TetR2:OP1 $\longrightarrow$ TetR2 + OP1	5.11×10^{-4}	[148]
25	TetOFF2:OP1 + Tc $\longrightarrow$ Tc:2TetOFF:OP1	2.0×10^{6}	[126]
26	Tc:2TetOFF:OP1 + Tc $\longrightarrow$ Tc:TetOFF2:OP1	1.0×10^{15}	§
27	Tc:TetOFF2:OP1 $\longrightarrow$ Tc:TetOFF2 + OP1	5.83×10^{-3}	[149]
28	TetR2:OP1 + Tc $\longrightarrow$ Tc:2TetR:OP1	2.0×10^{6}	[126]

Continued on Next Page…

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
29	Tc:2TetR:OP1 + Tc $\longrightarrow$ Tc:TetR2:OP1	1.0×10^{15}	§
30	Tc:TetR2:OP1 $\longrightarrow$ Tc:TetR2 + OP1	5.83×10^{-3}	[149]
31	RNAp + P1 + OP1 $\longrightarrow$ RNAp:P1:OP1	8.6×10^{6}	[152]
32	RNAp:P1:OP1 $\longrightarrow$ RNAp + P1 + OP1	1.0×10^{-2}	[152]
33	RNAp:P1:OP1 $\longrightarrow$ RNAp*:P1:OP1	1.3×10^{-2}	[152]
34	RNAp*:P1:OP1 $\longrightarrow$ RNAp*:DNA11 + P1 + OP1	30	[153]
35	RNAp + P1 + OP1:TetR2 $\longrightarrow$ RNAp:P1:OP1:TetR2	8.6×10^{6}	[152]
36	RNAp:P1:OP1:TetR2 $\longrightarrow$ RNAp + P1 + OP1:TetR2	1.0×10^{-2}	[152]
37	RNAp:P1:OP1:TetR2 $\longrightarrow$ RNAp*:P1:OP1:TetR2	1.3×10^{-2}	[152]
38	RNAp*:P1:OP1:TetR2 $\longrightarrow$ RNAp*:DNA11 + P1 + OP1:TetR2	30	[153]
39	RNAp + P1 + OP1:TetOFF2 $\longrightarrow$ RNAp:P1:OP1:TetOFF2	8.6×10^{6}	[152]
40	RNAp:P1:OP1:TetOFF2 $\longrightarrow$ RNAp + P1 + OP1:TetOFF2	1.0×10^{-2}	[152]
41	RNAp:P1:OP1:TetOFF2 $\longrightarrow$ RNAp*:P1:OP1:TetOFF2	1.3×10^{-1}	¶
42	RNAp*:P1:OP1:TetOFF2 $\longrightarrow$ RNAp*:DNA11 + P1 + OP1:TetOFF2	30	[153]

Continued on Next Page…

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
43	$\text{RNAp*:DNA11} \longrightarrow \text{RNAp} + \text{mRNA1}$	$30,660$	[153]
44	$\text{mRNA1} \longrightarrow \varnothing$	2.0×10^{-3}	†
45	$\text{mRNA1} + \text{Ribosome} \longrightarrow \text{Rib:mRNA1}$	1.0×10^{5}	†
46	$\text{Rib:mRNA1} \longrightarrow \text{Rib:mRNA11} + \text{mRNA1}$	100	[154]
47	$\text{Rib:mRNA11} \longrightarrow \text{Ribosome} + \text{TetOFF}$	$100,220$	[154]
48	$\text{TetOFF2} + \text{OP2} \longrightarrow \text{TetOFF2:OP2}$	2.86×10^{6}	[148]
49	$\text{TetOFF2:OP2} \longrightarrow \text{TetOFF2} + \text{OP2}$	5.11×10^{-4}	[148]
50	$\text{TetR2} + \text{OP2} \longrightarrow \text{TetR2:OP2}$	2.86×10^{6}	[148]
51	$\text{TetR2:OP2} \longrightarrow \text{TetR2} + \text{OP2}$	5.11×10^{-4}	[148]
52	$\text{TetOFF2:OP2} + \text{Tc} \longrightarrow \text{Tc:2TetOFF:OP2}$	2.0×10^{6}	[126]
53	$\text{Tc:2TetOFF:OP2} + \text{Tc} \longrightarrow \text{Tc:TetOFF2:OP2}$	1.0×10^{15}	§
54	$\text{Tc:TetOFF2:OP2} \longrightarrow \text{Tc:TetOFF2} + \text{OP2}$	5.83×10^{-3}	[149]
55	$\text{TetR2:OP2} + \text{Tc} \longrightarrow \text{Tc:2TetR:OP2}$	2.0×10^{6}	[126]
56	$\text{Tc:2TetR:OP2} + \text{Tc} \longrightarrow \text{Tc:TetR2:OP2}$	1.0×10^{15}	§

Continued on Next Page...

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
57	Tc:TetR2:OP2 ⟶ Tc:TetR2 + OP2	5.83×10^{-3}	[149]
58	RNAp + P2 + OP2 ⟶ RNAp:P2:OP2	8.6×10^{6}	[152]
59	RNAp:P2:OP2 ⟶ RNAp + P2 + OP2	1.0×10^{-2}	[152]
60	RNAp:P2:OP2 ⟶ RNAp∗:P2:OP2	1.3×10^{-2}	[152]
61	RNAp∗:P2:OP2 ⟶ RNAp∗:DNA21 + P2 + OP2	30	[153]
62	RNAp + P2 + OP2:TetR2 ⟶ RNAp:P2:OP2:TetR2	8.6×10^{6}	[152]
63	RNAp:P2:OP2:TetR2 ⟶ RNAp + P2 + OP2:TetR2	1.0×10^{-2}	[152]
64	RNAp:P2:OP2:TetR2 ⟶ RNAp∗:P2:OP2:TetR2	1.3×10^{-2}	[152]
65	RNAp∗:P2:OP2:TetR2 ⟶ RNAp∗:DNA21 + P2 + OP2:TetR2	30	[153]
66	RNAp + P2 + OP2:TetOFF2 ⟶ RNAp:P2:OP2:TetOFF2	8.6×10^{6}	[152]
67	RNAp:P2:OP2:TetOFF2 ⟶ RNAp + P2 + OP2:TetOFF2	1.0×10^{-2}	[152]
68	RNAp:P2:OP2:TetOFF2 ⟶ RNAp∗:P2:OP2:TetOFF2	1.3×10^{-1}	¶
69	RNAp∗:P2:OP2:TetOFF2 ⟶ RNAp∗:DNA21 + P2 + OP2:TetOFF2	30	[153]
70	RNAp∗:DNA21 ⟶ RNAp + mRNA2	$30,621$	[153]

Continued on Next Page...

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
71	mRNA2 ⟶ ∅	2.0×10^{-3}	†
72	mRNA2 + Ribosome ⟶ Rib:mRNA2	1.0×10^{5}	†
73	Rib:mRNA2 ⟶ Rib:mRNA21 + mRNA2	100	[154]
74	Rib:mRNA21 ⟶ Ribosome + TetR	$100,207$	[154]
75	TetOFF2 + OP3 ⟶ TetOFF2:OP3	2.86×10^{6}	[148]
76	TetOFF2:OP3 ⟶ TetOFF2 + OP3	5.11×10^{-4}	[148]
77	TetR2 + OP3 ⟶ TetR2:OP3	2.86×10^{6}	[148]
78	TetR2:OP3 ⟶ TetR2 + OP3	5.11×10^{-4}	[148]
79	TetOFF2:OP3 + Tc ⟶ Tc:2TetOFF:OP3	2.0×10^{6}	[126]
80	Tc:2TetOFF:OP3 + Tc ⟶ Tc:TetOFF2:OP3	1.0×10^{15}	§
81	Tc:TetOFF2:OP3 ⟶ Tc:TetOFF2 + OP3	5.83×10^{-3}	[149]
82	TetR2:OP3 + Tc ⟶ Tc:2TetR:OP3	2.0×10^{6}	[126]
83	Tc:2TetR:OP3 + Tc ⟶ Tc:TetR2:OP3	1.0×10^{15}	§
84	Tc:TetR2:OP3 ⟶ Tc:TetR2 + OP3	5.83×10^{-3}	[149]

Continued on Next Page. . .

Table 8.1: A Chemical Kinetics Representation of Network III – Continued

#	Reaction	k	Ref.
85	$\text{RNAp} + \text{P3} + \text{OP3} \longrightarrow \text{RNAp:P3:OP3}$	8.6×10^{6}	[152]
86	$\text{RNAp:P3:OP3} \longrightarrow \text{RNAp} + \text{P3} + \text{OP3}$	1.0×10^{-2}	[152]
87	$\text{RNAp:P3:OP3} \longrightarrow \text{RNAp}{*}\text{:P3:OP3}$	1.3×10^{-2}	[152]
88	$\text{RNAp}{*}\text{:P3:OP3} \longrightarrow \text{RNAp}{*}\text{:DNA31} + \text{P3} + \text{OP3}$	30	[153]
89	$\text{RNAp}{*}\text{:DNA31} \longrightarrow \text{RNAp} + \text{mRNA3}$	$30, 723$	[153]
90	$\text{mRNA3} \longrightarrow \varnothing$	2.0×10^{-3}	†
91	$\text{mRNA3} + \text{Ribosome} \longrightarrow \text{Rib:mRNA3}$	1.0×10^{5}	†
92	$\text{Rib:mRNA3} \longrightarrow \text{Rib:mRNA31} + \text{mRNA3}$	100	[154]
93	$\text{Rib:mRNA31} \longrightarrow \text{Ribosome} + \text{GFP}$	$100, 248$	[154]

Units on k: 1st order reaction: s^{-1}, 2nd order: $M^{-1}s^{-1}$, power law kinetics: $M^{-2}s^{-1}$. Reactions with two kinetic constants are γ-distributed events, where the first number is the rate of each step and the second is the number of steps. References are noted next to each reaction. §: rate adjusted for fast reacting intermediates. ∗: rates adjusted for 10 min half-life. ¶: rate adjusted from the function of activator. †: rates adjusted to give 20 protein molecules per mRNA transcript. Numbering of the genes starts from top to bottom as they appear in Fig. 8.1. For example, P1: promoter of gene 1, OP1: operator of gene 1.

(2 min) as compared to the normal half-life of approximately 20 min. Similar to the binding of two Tc molecules to free TetR, the reaction of two Tc molecules with bound to an operator TetR is again broken down to two steps.

Protein degradation can be modelled as a first order reaction, with the kinetic constant calculated from half-life data. Protein half-lifes can vary by many orders of magnitudes and depend on the cell type and environmental conditions. Consequently it would be invalid to consider a typical value that would apply universally. The solution to this problem comes by adding a C-terminal tag. In the present study we assumed that all proteins, except GFP, have an initial half-life of approximately 10 min (0.0012 s^{-1}). Wild type GFP degradation is slow, has a half-life of approximately 26 hours. For distinct turn on and off times of the reporter gene smaller half-life times are desired. New unstable variants of GFP proteins have been introduced [150]. We choose GFP-L-A-A (the last three letters denote the amino acids of the C-terminal tag), which has a reported half-life of 40 min. Finally, since for *E. coli* there is no specific pathway for biodegrading Tc, we assume that the rate at which Tc is removed from the system is equal to the half-life (48 h) of Tc in distilled water [151].

E. coli RNA polymerase recruitment to the promoter region, interaction with the occupied or not operator region, formation of close complex and then formation of the open complex are modelled through a cascade of reactions. Literature data [152, 155] provide the desirable kinetic constants. Open complex formation is assumed to be irreversible, since cells try to minimize their energy use. Transactivators in general attract, position and modify RNA polymerase. Given that Ptet is a strong promoter we assume that the presence of either Tet-OFF or Tet-ON in close proximity to the promoter mainly affects the formation of the open complex. In the present study we assume that the kinetic constant of the irreversible formation of the open complex is increased by a factor of 10, since we were not able to obtain kinetic data for prokaryotic transactivators.

Recruitment of RNA polymerase is described using power law kinetics. Initiation of transcription is modeled as a first order reaction, whereas elongation is considered to be a gamma distributed event [53]. Movement of the RNA polymerase across the DNA (coding sequence), occurs at a rate of approximately 30 nucleotides/s [154]. The parameter N of the gamma distribution is equal to the number of nucleotides each coding sequence has. TetR is comprised of 207 Amino Acids (AA), whereas Tet-OFF has the extra AA from the transactivator domain. The GFP variant is comprised of 238 AA, plus three AA from the peptide chain.

As in the case of proteins, mRNA can be degraded. Similar to proteins there are complex pathways for biodegrading mRNA. Again we considered degradation to be a first order chemical reaction. Furthermore, mRNA is translated in the ribosomes where proteins are the final product. Initiation of translation is considered to be irreversible since the cell utilizes energy in the form of ATP. Kinetic constants for both stages are adjusted so that for each mRNA transcript approximately 20 protein molecules are being produced. Elongation of translation is treated similarly to transcription. Movement of the ribosomal subunits across the mRNA occurs at a rate of 100 AA/s [26]. Similar to transcription, we model translation as a gamma distributed event [53].

Assumptions that relate to our specific system and have not been mentioned in the previous paragraphs are the following. Monomer forms of TetR protein or fusion of TetR with tranactivators are not able to bind to operator regions. Furthermore we assume that monomer TetR and Tet-OFF are not able to react with Tc molecules, only the dimer forms do.

8.3.2 Model Assumptions and Initial Conditions

The main underlying assumptions on which the model is based are as follows. The reactant volume is considered to be well stirred and the species are diluted in a large number of water molecules (homogeneous). In all simulations we consider a cell which we follow over time as it divides to produce ancestors (cell division with the doubling time of cells being 30 ± 4 min). Each cell is considered to be of initial volume 10^{-15} L and then grows exponentially until it divides, with division times following a normal distribution with mean 30 min and standard deviation 4 min. Furthermore, the species velocities follow a Maxwell Boltzmann distribution leading to a large number of neutral collisions that add to the homogeneity and a small number of reacting collisions. The system is considered to be isolated, that is other genes or organelles are assumed not to interfere, while mass transfer through the boundaries is allowed (for example nucleotides bases are readily available). Also, the cell has all the nutrients it needs to fuel its metabolism, which keeps major components (for example, free and available RNA polymerase, proteolytic enzymes) concentrations constant. Temperature and pH are assumed to remain constant throughout the simulations.

Turning our attention to the initial conditions we briefly discuss how they were generated. When inserting a vector in a cell that expresses non natural occurring proteins (proteins that are not expressed naturally within the cell) you do not expect to have any previous accumulation of those proteins. In our case there is no previous accumulation of TetR, TetOFF or TetON and GFP. For TetR this is true because we use strings of *E. coli* that do not contain the natural tetracycline resistance operon. TetOFF and TetON are not naturally occurring proteins in any bacterial or mammalian cell and finally GFP is also not being naturally expressed in *E. coli.* Therefore all their concentrations are set initially to zero. This also means that all intermediates involved in their transcription, translation, regulation will also be absent, hence have zero initial concentrations. On the other hand we set the initial concentration of each promoter/operator sites equal to one since that is the amount the cell will recognize. Finally available and free RNA polymerase and ribosome numbers are set accordingly to represent average values. In our case, all simulations use 180 molecules of free and available RNA polymerase and 300 free and available ribosomes [29, 30, 26]. The initialization of the simulations with more free RNA polymerase or free ribosomes will only result in sifted up values of translated and transcribed proteins, such as GFP, but the qualitative characteristics remain unaltered.

8.3.3 Computer-aided Design Software Tools

For the time integration of the generated chemical kinetics models for each of the four Networks we use either Hy3S or SynBioSS (cf. Chapter 7). Both depend on the hybrid algorithm described in Sec 3.2 which has been enhanced with the implementation of the adaptive algorithm presented in Chapter 4.

We have made available the necessary files for simulating Network IV in the website of SynbioSS (`http://synbioss.sourceforge.net/`). Accessibility of the files is plausible through either of the GUIs. In these files initial conditions are set as discussed in Sec 8.3.2 and the values used for the kinetic parameters in the case of wild-type dynamics are depicted in Table 8.1. Averages are computed from 100 independent trajectories.

In our work, all realizations were obtained using Itanium2 1.5 GHz processors. Average simulation times for Network IV range from 4 to 6 hours per trial on supercomputers.

8.4 Network I

8.4.1 Dynamical Behavior Based on Wild Type Kinetics

Starting with Network I (cf. Fig. 8.1), we intend to control the concentration of Tet-ON with self-repression and decrease the sensitivity of the network to low Tc levels. This decreased sensitivity will result in shorter time intervals before gene expression of the reporter gene is turned back off again, after Tc administration. With Tc present, Tet-ON can bind on TetO, downstream of Ptet and self-repress while transcription of the reporter gene is on and activated by Tet-ON. The rate of the transcription depends on the amount of Tet-ON induced with Tc available and on the promoter strength. In the absence of Tc, Tet-ON does not bind to either TetO sequences and expression levels of Tet-ON and GFP will depend on promoter strength (basal activity). For small Tc concentrations, self-repression of Tet-ON will result in a decrease of Tet-ON concentration and lower expression of reporter gene. A schematic representation of these interactions can be seen in Fig. 8.2(a).

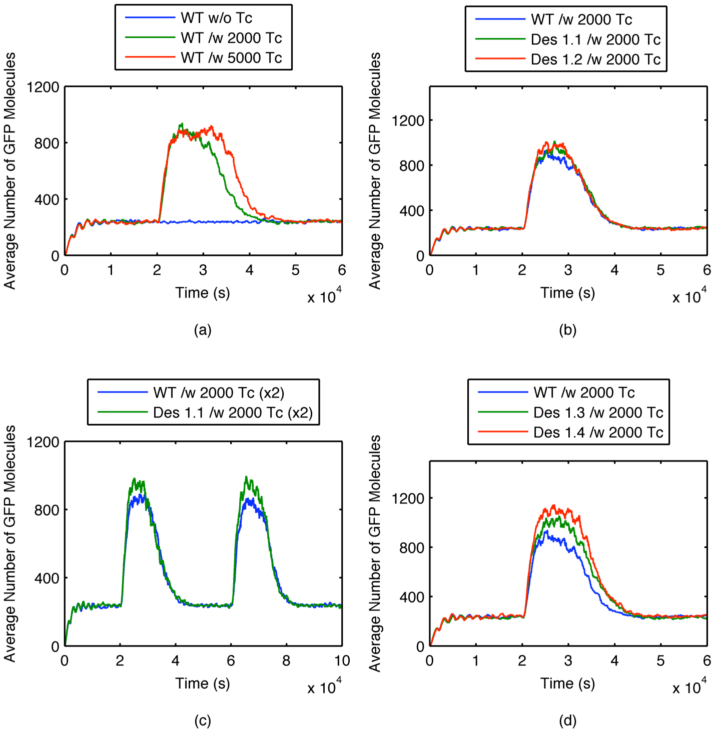
First we investigate the dynamical behavior, both transient and at equilibrium, of the network. The dynamical behavior of Network I in the presence or absence of Tc, over a time period of 6×10^4 s (16.7 h) is presented in Fig. 8.3(a). In the absence of Tc basal expression of GFP is approximately 250 molecules. The system reaches an equilibrium state after 10^4 s (2.8 h) with Tet-ON dimer concentrations values of approximately 40 molecules (data not shown). On the other hand, GFP production is increased when Tc is added, 2000 and 5000 molecules at time 2×10^4 s (5.6 h) (cf. Fig. 8.3(a)). Maximum GFP values reach levels of approximately 850 molecules, a 200 % increase from basal expression. The differences between the two cases are the time that the system sustains maximum levels of GFP and eventually the turning off time. This differentiation is a direct result of Tet-ON dimer concentration before the addition of Tc and the concentration of added Tc. In both cases, before adding Tc the concentration of Tet-ON monomers and dimers was the same and that led to similar levels of induced Tet-ON, meaning same GFP values. The lengthened duration of the pulse in case two is mainly due to the larger amount of free and available Tc molecules that sustained induced Tet-ON molecules longer. In both cases maximum free Tc amounts in the cell were below toxicity levels, approximately 600 (0.44 $\mu g/mL$) and 1600 (1.18 $\mu g/mL$) molecules respectively.

8.4.2 Fine Tuning Using Mutated TetR and TetO Variants

As we observe the system experiences high levels of basal expression. This has been anticipated since the Ptet promoter is naturally a strong promoter. Overcoming this limitation will require to change Ptet with a minimal one, something attempted successfully in mammalian cells with a promoter from hCMV. For this we will focus on other strategies to fine tune the system, such as mutating operator sequences and changing half-lifes.

The first proposed change is to mutate TetO of the gene encoding Tet-ON. Mutating TetO in general will result in less binding affinity for induced Tet-ON, since TetR or revTetR interaction with TetO is considered to be very strong. Therefore we will consider only cases where a decrease in the binding affinity is observed. Implementing the change in our model requires increasing the dissociation constant of the induced Tet-ON dimer from TetO (decrease half-life of the complex). Initially, the wild type kinetic constant was set to 0.01 s^{-1}. We changed the dissociation kinetic constant from 0.01 s^{-1} to 0.2 s^{-1} and then again to 0.5 s^{-1}.The results are shown in Fig. 8.3(b), where at time 2×10^4 s there is an addition of 2000 molecules of Tc in the medium. Although we observe a decrease in the occupancy of the mutated operator and a small increase in the number of induced Tet-ON dimers as well as the time period they are present, no significant change in the occupancy of the operator in the reporter gene is observed. Therefore as the kinetic parameter is increased, GFP levels are slightly altered whereas pulse duration practically remains the same. This becomes more evident in Fig. 8.3(c) where we compare the dynamical behavior of the wild type kinetics with the operator having a 20 fold (0.2 s^{-1}) decrease in the affinity for Tet-ON.

On the other hand mutating the operator adjacent to Ptet encoding GFP will only result in decreased production of GFP, a direct consequence of the reduced occupancy of the operator. The same outcome can be achieved by adding less Tc into the system.



(a)

(b)

(c)

(d)

Figure 8.3: Dynamical behavior of Network I: (a) Average number of GFP molecules in the absence of Tc (WT w/o Tc, blue line) and when 2000 molecules (WT /w 2000 Tc, green line) or 5000 molecules (WT /w 2000 Tc, red line) of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics. (b) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line), a 20 fold (Des 1.1 /w 2000 Tc, green line) and a 50 fold (Des 1.2 /w 2000 Tc, red line) increase in the dissociation constant of induced Tet-ON from TetO of the gene encoding Tet-ON. (c) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s and 6×10^4 s, using wild-type kinetics (WT /w 2000 Tc (x2), blue line), a 20 fold (Des 1.1 /w 2000 Tc (x2), green line) increase in the dissociation constant of induced Tet-ON from TetO of the gene encoding Tet-ON. (d) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line), a doubled (Des 1.3 /w 2000 Tc, green line) and a quadrupled (Des 1.4 /w 2000 Tc, red line) half-life of Tet-ON.

Another way to fine tune the systems behavior is by increasing the half-life of Tet-ON, leading to higher amounts of Tet-ON before administration of Tc. Applying such a change requires the addition of a C-terminal tag. The half-life of Tet-ON was initially set to 10 min; we looked into the cases of doubling it (5.7762×10^{-4} s^{-1}) and quadrupling it (2.8881×10^{-4} s^{-1}). The results are shown in Fig. 8.3(d), where again 2000 molecules of Tc are added at 2×10^4 s. In both cases we observe a change in the phenotype, GFP levels are increased over time and the duration is also increased. Looking into the levels of free and induced Tet-ON molecules we observe an increase of almost 75 % and a 150 % for doubled and quadrupled half-lifes, respectively. Comparable increases in GFP levels or duration are not monitored, due to non significant alternation in the occupancy of TetO in the reporter gene.

Concluding, controlling GFP levels and duration of the pulse (turning on and off times) cannot be accomplished separately. Both are related to the amount of Tet-ON dimers in the system prior to inducers administration and on the amount of the inducer added. The higher the amounts of Tet-ON dimers and of inducer, the higher GFP levels are going to be. On the other hand longer duration is achieved, by keeping the levels of induced Tet-ON constant over time.

8.5 Network II

8.5.1 Dynamical Behavior Based on Wild Type Kinetics

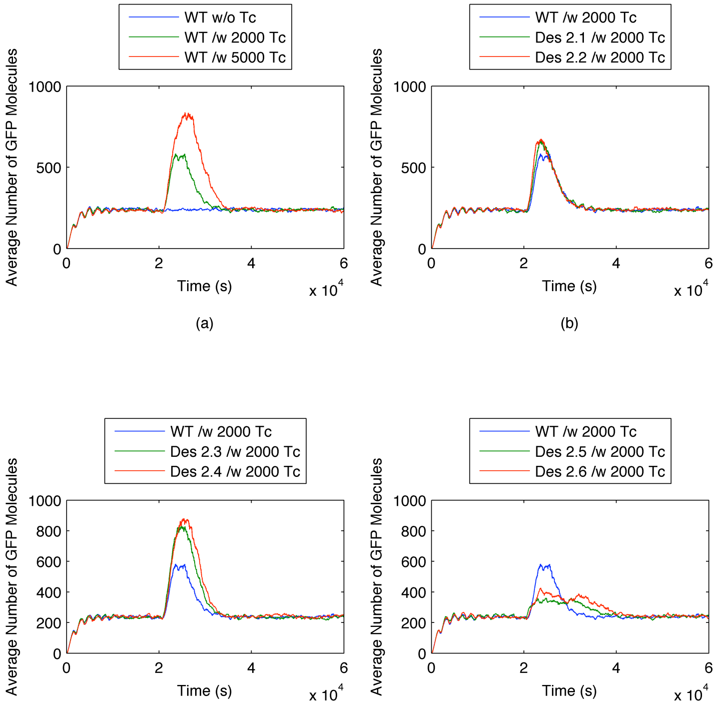
In Network II (cf. Fig. 8.1), we add a third gene encoding TetR to improve the regulation achieved with the first design. In the absence of Tc, TetR will minimize expression of all genes including itself. In the presence of Tc, TetR will no longer repress. Tet-ON levels will increase depending on the strength of self-repression, activating GFP expression. Tet-ON will also activate expression of TetR. At low levels of Tc, TetR binds to Tc and represses Tet-ON and reporter gene expression. Schematically these interactions are shown in Fig. 8.2(b).

Adding a third gene in the first network encoding TetR actually makes the system less sensitive to Tc concentrations. Fig. 8.4(a) compares the behavior of the system when there is no inducer present and when we add Tc, 2000 and 5000 molecules respectively at time 2×10^4 s. If we further compare these results with the ones presented in Fig. 8.4(a) we observe that the system exhibits the same phenotype when Tc is absent, while GFP levels are down and the duration of the pulse is smaller when the inducer is present. Network II has an extra source of Tc consumption, TetR dimers, so there is less free and available Tc concentration to induce Tet-ON. Also there is one extra TetO competing for the transactivator as well as Tet-ON has to compete with TetR for TetO. Concentrations of induced TetR reach maximum levels of approximately 140 and 180 molecules, for addition of 2000 and 5000 molecules of Tc respectively. On the contrary, induced Tet-ON molecules reach maximum levels of about 28 and 35 molecules respectively, lower compared to Network I. These lower values, together with the presence of TetR account for less GFP production. Comparing the different scenarios of inducer administered we notice that the more Tc present the more GFP is produced and for a longer period. At last, maximum free Tc amounts in the cell were below toxicity levels, approximately 400 (0.30 $\mu g/mL$) and 1200 (0.89 $\mu g/mL$) molecules, respectively.

8.5.2 Fine Tuning Using Mutated TetR and TetO Variants

It can be again observed that the system experiences high levels of basal expression. As mentioned before, a plausible solution for *E. coli* is the use of a minimal promoter, while in mammalian cells rTS could substitute TetR. In the remaining section the focus is to improve the design by proposing strategies, for instance mutated operator sequences or coding sequences that alter the dynamical behavior.

First we start by introducing a mutation in the operator controlling the expression of Tet-ON. This change will affect both TetR and induced Tet-ON binding. For simplicity we assume that the change is analogues for both cases. The approach is indeed simplified, but this assumption is made due to the similarity of the two proteins that differ only by a small number of mutations. The idea behind this mutation is to decrease the turning on response time but also increase GFP levels. TetR dimers will bind weaker, resulting in higher Tet-ON dimer levels at equilibrium. At the same time self repression is limited in the presence of the inducer. We changed the dissociation kinetic constants from 0.01 s^{-1} (wild-type) to 0.1 s^{-1} and then again to 0.5 s^{-1} in the case of induced Tet-ON and from 5.11×10^{-4} s^{-1} (wild-type) to 5.11×10^{-3} s^{-1} to 2.555×10^{-4} s^{-1} in the case of TetR. The results are depicted in Fig. 8.4(b), where at time 2×10^4 s there is an addition of 2000 molecules of Tc. As the affinity decreases, we observe higher levels



(a)

(b)

(c)

(d)

Figure 8.4: Dynamical behavior of Network II: (a) Average number of GFP molecules in the absence of Tc (WT w/o Tc, blue line) and when 2000 molecules (WT /w 2000 Tc, green line) or 5000 molecules (WT /w 5000 Tc, red line) of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics. (b) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line), a 10 fold (Des 2.1 /w 2000 Tc, green line) and a 50 fold (Des 2.2 /w 2000 Tc, red line) increase in the dissociation constant of both TetR and induced Tet-ON from TetO of the gene encoding Tet-ON. (c) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line). All other plots have a 10 fold increase in the dissociation constant of both TetR and induced Tet-ON from TetO of the gene encoding Tet-ON, but differ in a 5 fold (Des 2.3 /w 2000 Tc, green line) and 20 fold (Des 2.4 /w 2000 Tc, red line) increase in the dissociation constant of both TetR and induced Tet-ON from TetO of the gene encoding TetR. (d) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line). All other plots have mutated TetR variants that do not bind Tc and show a decreased binding affinity for TetO (20 fold decrease in the dissociation constant), but differ in the half-life of Tet-ON, 40 min (Des 2.5 /w 2000 Tc, green line) and 24 h (Des 2.6 /w 2000 Tc, red line).

of GFP but also a small but visible decrease in the turning on time. Increased Tet-ON levels at equilibrium help the system to respond faster when Tc is added. Indeed, levels of Tet-ON dimer prior to Tc administration show a 500 % and 1250 % increase for a 10 fold and 50 fold increases in the dissociation constants, respectively. In contrast, the actual phenotype is only increased by 15 % (from approximately 570 to 660 molecules of GFP), since the actual increase of induced Tet-ON dimers is only 25 % for both cases.

We also propose to mutate TetO in the gene encoding TetR. The objective is to decrease production of TetR when Tc is added. The expectations are to observe higher expression of GFP and less Tc bound TetR. The latter is a consequence of the fact that less TetR is being produced when the inducer is present, while the former is a result of higher transactivator concentrations. Results are not shown for brevity but the objectives are largely met. Furthermore, the system appears to have a small increase in the turning off time.

Since both above mentioned mutations were in the same direction, the next logical step is to combine them. In Fig. 8.4(c), the wild type kinetics dynamical behavior is compared to the behavior of the mutated TetOs. In all cases, the dissociation constants concerning TetO of the Tet-ON gene where increased by a factor of 10, while those for the other TetO have a 5 or a 20 fold increase. From the figure it is obvious that levels of GFP are up, the turning off time is also increased while the turning on time is shortened. Apparently, the two mutations acted additively in the case of GFP production. The mutated TetO of TetR increased the turning off time whereas the other mutated TetO contributed to the decreased turning on time. Obviously, one can adjust the parameters accordingly in order to achieve the tergeted phenotype.

Finally, a more radical approach is attempted. The wild type behavior is compared with the ones resulting from a series of mutations (cf. Fig. 8.4(d)). First TetR is mutated so that it does not bind Tc and at the same time mutated to bind weaker to all TetO, a 20 fold increase in the dissociation constant. The two mutations do not overlap in the coding sequence, since different amino acids are responsible for the DNA binding and for Tc binding. For simplicity we assume there is no direct or indirect (allostery) interference between the two mutations. Second, the half-life of Tet-ON is increased from 10 min to 40 min and then again to 24 h. Briefly the idea is first to increase Tet-ON concentration before the addition of Tc and to reduce the need for Tc. The results are inferior if the interest is in GFP production but on the other hand the duration of the pulse is increased.

Concluding we observed that by adding TetR in the equation we are able to adjust and better control the expression of GFP. We are able to regulate both turning on and off times and at the same time manipulate levels of GFP. The downside is that for a given addition of Tc concentration Network I will reach higher GFP levels compared to Network II, since the latter has an extra source of Tc consumption, namely TetR.

8.6 Network III

8.6.1 Dynamical Behavior Based on Wild Type Kinetics

With Network III we anticipate to increase sensitivity to Tc. Without Tc, Tet-OFF production is on and self-activating. Tet-OFF also activates TetR expression. Furthermore, TetR production is also on but self-repressing and at the same time TetR represses Tet-OFF and GFP expression. In this network, Tet-OFF represses expression of the reporter gene instead of activating it. TetR stimulates the amount of both TetR and Tet-OFF dimers in the cell by competing with Tet-OFF for TetO. In the presence of Tc, GFP levels will mainly depend on the basal expression of the promoter and the ratio of Tc over Tet-OFF and TetR concentrations. Fig. 8.2(c) summarizes the interactions betweens genes in Network III.

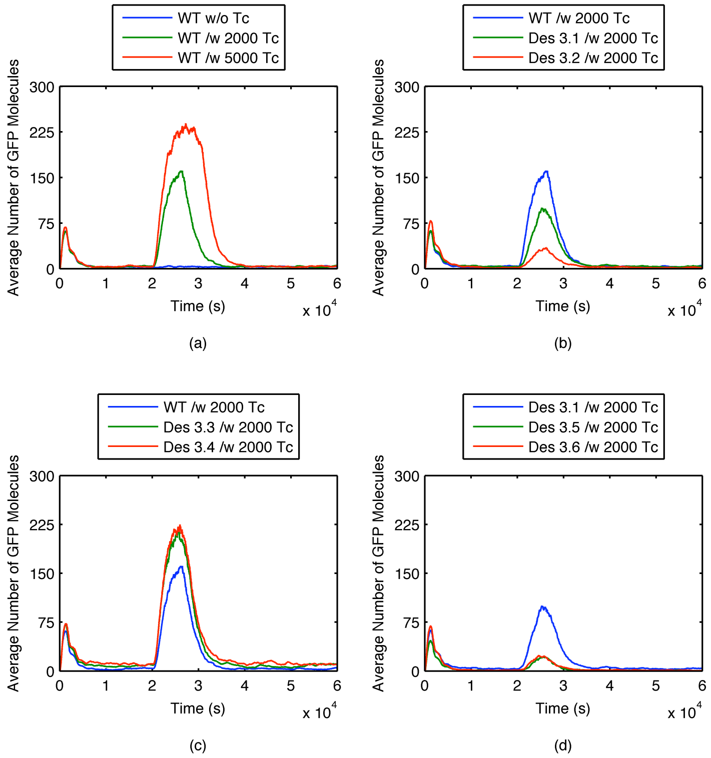
Simulating the time evolution of Network III (cf. Fig. 8.5(a)) using wild-type kinetics, results in a substantially different observed phenotype. In the absence of Tc, equilibrium state values of GPF approach zero, approximately 4 molecules of GFP. A sharp pick in the concentration of GFP in the transient period is a result of small initial TetR and Tet-OFF concentrations, which at equilibrium sum up to a total average number of approximately 300 molecules. Next we add 2000 and 5000 molecules of Tc at 2×10^4 s (cf. Fig. 8.5(a)). The higher the inducer concentration, the more time the operator, located downstream in the reporter gene, will be unoccupied leading to more GFP for longer time periods. GPF maximum concentrations values approach levels of approximately 250 molecules (basal expression).This network takes into account and effectively uses the high basal expression of Ptet. Again Tc levels remained below toxicity levels.

8.6.2 Fine Tuning Using Mutated TetR and TetO Variants

The challenges that this network poses are first to eliminate expression leaking when Tc is absent and second, to increase the sensitivity of the network to Tc. Beginning with the first challenge an obvious step is to increase repressors levels, meaning the total amount of both TetR and Tet-OFF dimmer molecules, when Tc is absent. This can be accomplished if we allow Tet-OFF to occupy operator sites for longer times compared to TetR. One alternative is to mutate TetR so that it shows weaker binding to TetO. Increasing the dissociation of TetR from TetO by a factor of 10 or 50, we managed indeed to achieve a decrease in the levels of GFP, but we did not manage to make them zero (cf. Fig. 8.5(b)). Levels of repressors indeed raised 50 % and 100 % for a 10 and 50 fold increase in the kinetic parameter, respectively. To generalize, as TetR binds progressively more weakly to TetO, GFP levels in the absence of Tc decrease by two molecules (50 %) when the dissociation constant is increased 50 times. Conversely, GFP levels decrease drastically for a given (2000 molecules of Tc added at 2×10^4 s) concentration of inducer, less sensitivity to Tc.

On the other hand, trying to increase the sensitivity of the system to Tc concentrations requires the opposite, a decrease in the repressors concentration. Similarly, mutating Tet-OFF instead of TetR, leads to smaller production of both proteins. In Fig. 8.5(c) we compare the wild type phenotype with the one observed by increasing the dissociation constant 5 (2.555×10^{-3} s^{-1}) and 20 (1.022×10^{-2} s^{-1}) times. It is obvious that GFP levels increase towards basal expression levels. Repressors levels in the cell decrease approaching total values of 80 molecules, with leaking becoming more evident in the absence of inducer.

Since the above two mutations do not point on the same direction, it would not be fruitful to try to combine them. For this we tried something more extreme in order to eliminate leaking. We mutated TetR for smaller binding affinity to TetO sequences and also we increased the half-life of both TetR and Tet-OFF with the purpose of increasing the overall concentration of repressors. We keep the same TetR mutant in all simulations presented, 10 fold increase in the kinetic parameter, while we triple, (30 min) and quadruple (40 min) the half-lifes. The results are shown in Fig. 8.5(d). The increase in the repressors total concentration is approximately 100 molecules (22 %) for both half-life cases whereas the actual decrease in GFP is 3 molecules (75 %). In



(a)

(b)

(c)

(d)

Figure 8.5: Dynamical behavior of Network III: (a) Average number of GFP molecules in the absence of Tc (WT w/o Tc, blue line) and when 2000 molecules (WT /w 2000 Tc, green line) or 5000 molecules (WT /w 5000 Tc, red line) of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics. (b) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line), a 10 fold (Des 3.1 /w 2000 Tc, green line) and a 50 fold (Des 3.2 /w 2000 Tc, red line) increase in the dissociation constant of TetR for all TetO (mutated TetR variant). (c) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using wild-type kinetics (WT /w 2000 Tc, blue line), a 5 fold (Des 3.3 /w 2000 Tc, green line) and a 20 fold (Des 3.4 /w 2000 Tc, red line) increase in the dissociation constant of Tet-OFF for all TetO (mutated Tet-OFF variant). (d) Average number of GFP molecules when 2000 molecules of Tc are added into the medium at time 2×10^4 s, using for all simulations a mutated TetR variant with 10 fold (Des 3.1 /w 2000 Tc, blue line) increase in the dissociation constant plus a tripled (Des 3.5 /w 2000 Tc, green line) and a quadrupled (Des 3.6 /w 2000 Tc, red line) half-life for both Tet-ON and TetR.

Conclusion, increasing the half-life eventually will lead to zero GFP levels but with large amounts of proteins molecules in the cell that may be toxic. At the same time large inducer amounts are required to transfer the system from the Off to the On state.

In conclusion we explored possible mutations that would allow us to eliminate expression leakage. We observed that even if we increased repressor molecules levels by 100 % leaking is still present but in limited amounts. For complete silencing large amounts of repressor molecules are required leading to toxicity concerns. On the other hand increasing sensitivity to Tc requires less repressor molecules being present. Therefore depending on application requirements we can adjust the system parameters in order to achieve either very low GFP expression or higher sensitivity.

8.7 Network IV

8.7.1 Dynamical Behavior Based on Wild Type Kinetics

Finally for Network IV in the absence of Tc, Tet-OFF production is on and self-activating and TetR production is also on, but self-repressing. GFP expression will also be on, but how strongly depends on the ratio of TetR and Tet-OFF amounts available. With Tc present, TetR production is on, Tet-OFF and reporter gene production depend on the promoter strength. A schematic representation of these interactions can be seen in Fig. 8.2(d). Note that constant Tc administration will be required for the expression to be silenced, a limitation following Tet-OFF. Adjusting turning on response times is the objective in the present network.

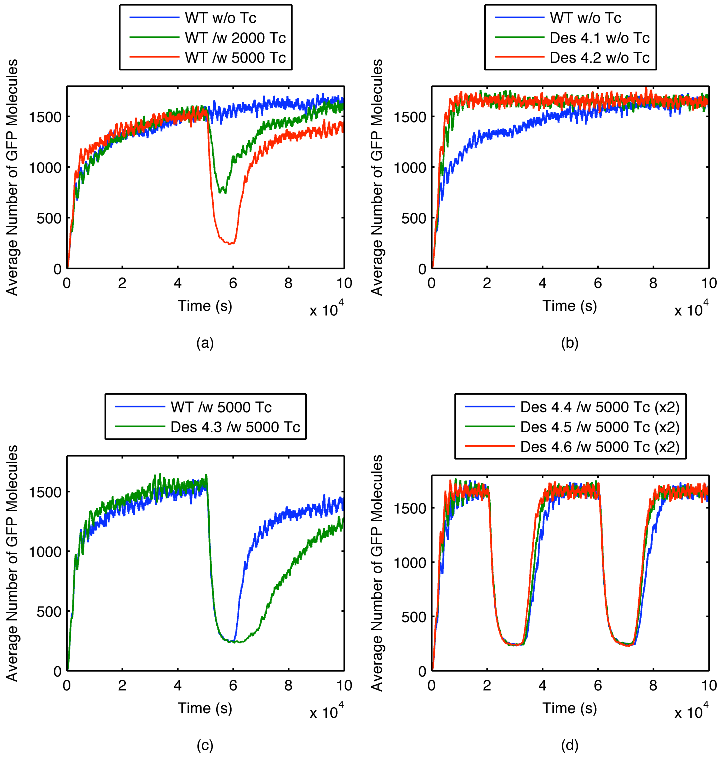
In Fig. 8.6(a) the time evolution of Network IV is shown. When Tc is absent the network produces higher concentration of GFP than the other networks, equilibrium values are approximately 1650 molecules. Obviously the system has not reached an equilibrium state, even after 28 hours. Addition of Tc causes an evident decrease in GFP production, with the transition from the Off to the On state having a large response time. This phenotype is a direct consequence of the competition between TetR and Tet-OFF dimers to occupy TetO sequences. Initially, or after Tc administration, concentrations of dimer TetR increase rapidly reaching a maximum concentration, only to fall rapidly short thereafter, approaching zero levels. Tet-OFF dimer concentration goes rapidly to 200 molecules and then requires 10 times more time to reach equilibrium values (approximately 300 molecules). These high concentration values are eventually responsible for the increased expression of GFP. High levels of Tc are required in order to drop production of GFP down to basal expression levels. Free maximum Tc concentrations reach levels below toxicity, approximately 100 (0.07 $\mu g/mL$) and 1200 (0.89 $\mu g/mL$) molecules for addition of 2000 and 5000 molecules of Tc, respectively.

8.7.2 Fine Tuning Using Mutated TetR and TetO Variants

By investigating the time evolution of the system, we can pinpoint limitations in the design and propose changes. First, the high basal expression is a common drawback among the proposed networks. Secondly, it is apparent that the response of the system is slow, both initially and after administration of the inducer. Finally, one would like to make the system more sensitive to Tc concentrations for two reasons; easier transition between the On and Off states and better control over the duration of the Off state.

In the previous section, we noted that the slow response is due to competitive binding between Tet-OFF and TetR with TetO. Improving the response time will require altering the relative binding affinity of the two dimers. We mutate TetR, since it is the one that does not add to GFP production. The appropriate kinetic parameters were altered from 5.11×10^{-4} s^{-1} to 5.11×10^{-3} s^{-1} to 1.022×10^{-2} s^{-1}, a 10 and 20 fold increases respectively. Simulating the new network we find that the initial lag is reduced significantly (cf. Fig. 8.6(b)). The system reaches equilibrium values in only 2.8 h. Note that the difference in the responses between the two mutations is small, which means that a little alternation is capable of producing the targeted behavior. Additionally, Tet-OFF levels reach their equilibrium values much faster than before. The difference is also noted by looking at the percent of Tet-OFF occupying TetOs over time, increasing as TetR affinity for TetOs decreases.

In this particular system, increased sensitivity to Tc can be achieved through a decrease in the equilibrium values of TetR and Tet-OFF. However, this will also cause a decrease in GFP levels. Another way to go about this problem is to use TetR variants that do not bind to Tc. The downside is that TetR will always be able to bind to TetO sequences. Using the last alternative, we simulate the system and the results are presented in Fig. 8.6(c). Obviously the new system appears to have longer pulse duration. Still the response time for transition between On and Off states remains large.



(a)

(b)

(c)

(d)

Figure 8.6: Dynamical behavior of Network IV: (a) Average number of GFP molecules in the absence of Tc (WT w/o Tc, blue line) and when 2000 molecules (WT /w 2000 Tc, green line) or 5000 molecules (WT /w 5000 Tc, red line) of Tc are added into the medium at time 5×10^4 s, using wild-type kinetics. (b) Average number of GFP molecules in the absence of Tc, using wild-type kinetics (WT w/o Tc, blue line), a 10 fold (Des 4.1 w/o Tc, green line) and a 20 fold (Des 4.2 w/o Tc, red line) increase in the dissociation constant of TetR from all TetO in the network(mutated TetR variant, less affinity for TetO). (c) Average number of GFP molecules when 5000 molecules of Tc are added into the medium at time 5×10^4 s, using wild-type kinetics (WT /w 5000 Tc, blue line) and mutated TetR variant that does not bind Tc (Des 4.3 /w 2000 Tc, green line). (d) Average number of GFP molecules when 5000 molecules of Tc are added into the medium at times 2×10^4 s and 6×10^4 s, using a mutated TetR variant that does not bind Tc and also shows different levels of binding affinity for TetO, 10 fold (Des 4.4 /w 2000 Tc (x2), blue line), 20 fold (Des 4.5 /w 2000 Tc (x2), green line) and 50 fold (Des 4.6 /w 2000 Tc (x2), red line) decrease in the dissociation constant.

Since both mutations in the coding sequence of TetR improved the design we decided to combine them, assuming the two mutations do not interfere with each other. In Fig. 8.6(d) the time evolution of the system is presented. Tc is added into the system at two time points, 2×10^4 s and 6×10^4 s. Though all TetR variants have no affinity for Tc, they have different levels of binding affinity for TetO, namely 10, 20 and 50 fold increase in the dissociation constant. Indeed the behavior of the system looks superior compared to the wild type. Furthermore, the network exhibit shorter turn off times as the binding affinity of TetR for TetO weakens. Constant Tc administration for low GFP production as well as the high GFP production in the absence of inducer (560 % above basal expression) render this system difficult but at the same time attractive for applications.

In summary, we achieved to decrease the response times of the network in both the transient period and also after Tc administration. Adjusting the corresponding kinetic parameter gives the required edge to Tet-OFF over TetR and hence improves the response. Additionally we explored ways to decrease the necessity for Tc in order to silence the system. We observed that by mutating TetR appropriately the required amounts of Tc are indeed reduced.

8.8 Summary

Using all the molecular components of transcription, translation, regulation and induction, the dynamic behavior of the proposed synthetic gene networks can be simulated and screened for possible improvements. It should not go unnoticed that the simulation of a system that spans many orders of magnitude in kinetic constant values is indeed realizable. To achieve this, we use a hybrid dynamic stochastic-discrete and stochastic-continuous algorithm equipped with an adaptive time stepping method for numerically integrating the set of stochastic differential equations in the model. The simulations allow the quick and inexpensive investigation and comparison of multiple alternative designs. They provide a clear insight at the molecular level, while experiments focus on the phenotype. The key is to identify the important interactions and based on them propose design rules. Important interactions can be obvious but non-apparent in terms of their impact on the phenotype of the system. Ideally, the computational approach will be able to investigate thoroughly all possible alternatives and adjust the dynamical behavior of a gene network to fit certain demands.

Based on the tetracycline-regulated systems, we propose four novel regulatory gene networks in order to alleviate limitations faced in widely used systems. We improved the design of all networks using mutations in the coding and operator sequences. Though there is still plenty of room for improvement, especially if one considers the amount of available operator, promoter and coding sequences that exist in nature. Our model-driven designs can become the first step in improved gene regulatory networks.

Chapter 9

Concluding Remarks

9.1 Summary of Contributions

The goal of this dissertation is the computer-aided design of synthetic gene networks. In the preceding chapters of this dissertation the focus has been on methodologies, algorithms, software tools and their applicability for computer-aided design of novel gene regulatory networks. The key contributions are summarized below.

- **Efficient and accurate integration of stiff chemical Langevin equations**

  An important challenge in any computer-aided design process is the accurate and efficient numerical integration of the underlying models. In gene network engineering and in synthetic biology in general, chemical Langevin equations (CLEs) appear when in a chemical kinetics model there are species and reactants in the same reaction that are minimally affected with each reaction occurrence and when that specific reaction channel fires frequently. The mathematical representation, i.e. the system of chemical Langevin equations, becomes stiff when there are different and deviant firing rates, meaning different timescales.

  In Chapter 4 we address the existence of stiffness by proposing a new adaptive time step scheme. The scheme is general and applies to any Itô SDEs with multiple multiplicative noise terms, a subclass of which are CLEs. The time step is adjusted, increased, decreased or kept the same, depending on two error controls. Contrary to the fixed step counterpart, the adaptive scheme is more stable in all the examples tested. The integration proceeds even when the initial time step is relatively large where the fixed step scheme fails to produce meaningful trajectories. This feature is extremely useful, especially when the dynamical behavior of the CLE system is unknown, as the solution may become numerically unstable during the course of the simulation. In terms of the required execution time, the method does not outperform the fixed step method as expected by comparing them with their deterministic counterparts. In fact, it slightly underperforms. Overall, the adaptive methodology adds much needed stability to the integration algorithm without crippling execution times. This led us to successfully incorporate the adaptive scheme into the hybrid stochastic algorithm, described in Sec. 3.2, which is now the "engine" for both Hy3S and SynBioSS (cf. Chapter 7).

In Chapter 5 we approached the same problem from a completely different perspective. Instead of treating the complete system of CLEs we examined the effectiveness of a semi-analytical reduction framework. Variables are identified and separated into fast and slow subsets under a necessary and sufficient condition. Then each subset is treated independently through the use of the adiabatic elimination methodology. Fast variables relax relatively fast into a pseudo-stationary distribution under the assumption of unchanged slow variables. Next the probability distribution of the transformed slow species is obtained as the solution of the corresponding system of CLEs for only the slow variables. Finally the overall probability distribution is obtained as the product of the two probabilities. On all the examples we have tested the method, results indicate a significant reduction in the computational cost, reaching up to two orders of magnitude. More importantly, the algorithm accurately reproduces the slow dynamics, while, depending on the system, deviations are more significant but within acceptable limits in the fast dynamics. Compared to the adaptive scheme, the main advantage is the resulting decrease in computational cost by sacrificing the accuracy in the fast species.

- **Deterministic description of a Markov process**

  The mathematical description of a chemical kinetics model as it applies to gene networks maybe challenging to propagate in time. The stochastic nature and the different time scales complicate any such approach. Therefore, a transformation of the problem into any mathematical equivalent form that involves only ordinary differential equations is welcome. In Chapter 6 we present a derivation of the moment equations starting from the master equation using the definition of jump moments. Analytical expressions are derived that apply to any process described as a Markov process. Similar to Chapter 4 the approach is generic and is not restrictive to chemical kinetics systems. As again our interest is in chemical kinetics models we derived analytical relations for the elements of any jump moment tensor, demonstrating the ease of equation setup. The elements of the analytical relations are a function of the stoichiometric matrix and the reaction propensities, i.e the probabilistic reaction rates. The drawback of the method, which is currently and actively researched, is that moment equations form an infinite dimensional system that does not accept a solution unless it is ”cleverly” truncated. Applicability is briefly illustrated through toy examples. Finally, using a simple non-linear example, known as the Schlögl model, we prove the need for higher than second order moments to accurately reconstruct the probability distribution for biologically-relevant chemical kinetics systems.

- **Software tools for computer-aided design**

  To cope with the ever increasing knowledge of the biological world and the exponential increase in available DNA sequences, a set of modeling algorithms would be useful in synthetic biology. They should be sophisticated enough to capture the complexity of biomolecular systems, yet easy enough to use by biologists, who are not experts in high performance computing. For this reason, in Chapter 7 we present two software tools, Hybrid Stochastic Simulation for Supercomputers (Hy3S) and Synthetic Biology Software Suit (SynBioSS), that make the use of our algorithms easy for all.

- **A computer-aided design example**

  In Chapter 8 we use two promising synthetic gene constructs as the seeds to generate novel gene networks to address current known performance limitations. In particular, we use the tetracycline regulatory expression systems, i.e. Tet-Off and Tet-On systems, based on the tetracycline resistance operon of *E. coli*, to propose new synthetic networks with improved characteristics. A computational approach is followed, where four alternative designs are proposed, tested and screened for any possible improvements. Results suggest that by setting the desired objectives for any given network, important interactions are identified, which may be obvious but non-apparent in terms of their impact on the phenotype of the system, and the strength of the interactions engineered to satisfy design criteria. A set of clear design rules is developed and appropriate mutants of regulatory proteins and operator sites are proposed.

Instead of looking at these networks statically, and simply changing or mutating the promoter and operator regions with trial and error, we use simulations to guide the design process based on desired objectives. Even thought the original networks were constructed in mammalian cells we created our respective models to correspond to bacteria cells physiology as the higher complexity in mammalian cells hinders any accurate model representation (cf. Sec 2.4.1). Detailed models, including all the molecular components of transcription, translation, regulation and induction provide the necessary detail level to propose actual changes in the DNA sequence. For instance mutations in the operator and or protein sequences. The simulation packages Hy3S and SynBioSS DS, discussed in Chapter 7, were used to profile the dynamic behavior of each of the networks. All in all, Chapter 8 conveys the usefulness of model-driven design in synthetic biology while at the same time the findings of the study propose actual improvements in the design blueprints, in a miniscule amount of time, leading to novel tetracycline-inducible regulatory gene networks.

9.2 Future Research Directions

As the filed of synthetic biology grows so will the need for efficient and accurate algorithms that will guide design efforts. We are far from any software tool or algorithm that resembles the use and efficiency of computer-aided design packages used to design the modern marvels of technology, for instance the software packages used by Airbus or Boing to design their fleets. Certainly, much more effort is warranted on the development of multiple time scale algorithms for gene networks.

Several open questions remain in order for the endeavor to be succesfull. Of great importance is the development of multiscale algorithms that can efficiently and accurately handle the different time scales present in any given chemical kinetics model of a real biological system. Looking back at Fig. 3.1 the question is how can we capture all the important and interesting dynamics present in any of the four regions of the problem space without sacrificing accuracy and speed. The answer, in our opinion, lies within the concepts developed in Chapter 6. Chapter 6 includes the foundation for describing Markov processes, such as stochastic chemical kinetics models, using a deterministic description. The advantages for such an approach will be significant since ODEs are well studied and many of the needed tools are already available. The bottleneck, as mentioned earlier, is that the system of moment equations is infinite and all the approximate solutions proposed to this date are not satisfactory. They have limited applicability or require *a priori* knowledge on the shape of the underlying distribution. Questions such as how many moments are indeed important, or at what order should the system be truncated in order to retain an accurate resolution of the lower order moments, are of importance, and to date they are only partially answered. But the single most important question is whether there exists a universal truncation formula. Answering this question will benefit numerous fields where Markov processes do play a crucial role for modeling purposes, including the design of novel gene networks.

Moreover, as the capabilities of CAD software for gene networks increase, there will also be a constant strive to prove their rigorousness. So far there have been a few studies combining detailed chemical kinetics models with actual experiments. By comparing wet lab and simulation experiments we can determine qualitative and quantitative performance of our algorithms. This will allow us to improve our CAD algorithms as well fine tune our modeling approaches. Under this scope, our group works to build *in vivo* equivalent networks to those discussed in Chapter 8. Experimental data will confirm or reject modeling predictions. In any case, this step is important and will only benefit the development of models. Arising questions range from, why do or don't we have qualitative or quantitative agreement, to are there interactions that are important but are either unknown or have been silently neglected? Comparing and contrasting simulation and wet lab experiments results is not only necessary but an important process in developing accurate modeling strategies and algorithms.

A final direction which promises to have a direct impact is the implementation of the reduction framework in the hybrid stochastic algoritm. While not a scientific undertaking, but rather an implementation challenge, the integration may reduce significantly execution times. At this point, the main disadvantage is that while the reduction algorithm is fairly simple on its own, incorporation into Hy3S is anything but straightforward. Hy3S is already a 10,000 plus line code with limited flexibility and a predefined architecture. So, in this case the open questions relate more on how do we create a better algorithm rather than how we improve the methods implemented in the algorithm. Improving algorithms and software tools should be a constant objective for all of those involved in computational synthetic biology.

References

[1] H. Salis, V. Sotiropoulos, and Y. N. Kaznessis. Multiscale hy3s: Hybrid stochastic simulation for supercomputers. *BMC Bioinformatics*, 7(1):93, 2006.

[2] V. Sotiropoulos and Y. N. Kaznessis. Synthetic tetracycline-inducible regulatory networks: computer-aided design of dynamic phenotypes. *BMC Systems Biology*, 1(1):7, 2007.

[3] V. Sotiropoulos and Y. N. Kaznessis. An adaptive time step scheme for a system of stochastic differential equations with multiple multiplicative noise: Chemical langevin equation, a proof of concept. *The Journal of Chemical Physics*, 128(1):014103, 2008.

[4] A. D. Hill, J. R. Tomshine, E. M. B. Weeding, V. Sotiropoulos, and Y. N. Kaznessis. Synbioss: the synthetic biology modeling suite. *Bioinformatics*, 24(21):2551–2553, 2008.

[5] V. Sotiropoulos, M.N. Contou-Carrere, P. Daoutidis, and Y. N. Kaznessis. Model reduction of multiscale chemical langevin equations: A numerical case study. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 6(3):470–482, July-Sept. 2009.

[6] J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[7] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of The Cell.* Garland Science, New York, 4th edition, 2002.

[8] S. A. Benner and A. M. Sismour. Synthetic biology. *Nat Rev Genet*, 6(7):533–43, 2005.

[9] A. P. Arkin. Synthetic cell biology. *Curr Opin Biotechnol*, 12(6):638–44, 2001.

[10] C. Toniatti, H. Bujard, R. Cortese, and G. Ciliberto. Gene therapy progress and prospects:transcription regulatory systems. *Gene Ther*, 11(8):649–57, 2004.

[11] C. Zimmer. Genomics. tinker, tailor: can venter stitch together a genome from scratch? *Science*, 299(5609):1006–7, 2003.

[12] Priscilla E. M. Purnick and Ron Weiss. The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol*, 10(6):410, 2009.

[13] D. Ferber. Synthetic biology. microbes made to order. *Science*, 303(5655):158–61, 2004.

[14] M. Gossen and H. Bujard. Tight control of gene expression in mammalian cells by tetracycline-responsive promoters. *Proc Natl Acad Sci U S A*, 89(12):5547–51, 1992.

[15] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–42, 2000.

[16] A. Becskei and L. Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590–3, 2000.

[17] A. Becskei, B. Seraphin, and L. Serrano. Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion. *Embo J*, 20(10):2528–35, 2001.

[18] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–8, 2000.

[19] M. R. Atkinson, M. A. Savageau, J. T. Myers, and A. J. Ninfa. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in escherichia coli. *Cell*, 113(5):597–607, 2003.

[20] R.S. Cox, M.G. Surette, and M.B. Elowitz. Programming gene expression with combinatorial promoters. *Molecular Systems Biology*, 3(1), 2007.

[21] K. I. Ramalingam, J. R. Tomshine, J. A. Maynard, and Y. N. Kaznessis. Forward engineering of synthetic bio-logical and gates. *Biochemical Engineering Journal*, In Press, Accepted Manuscript, 2009.

[22] C. C. Guet, M. B. Elowitz, W. Hsing, and S. Leibler. Combinatorial synthesis of genetic networks. *Science*, 296(5572):1466–70, 2002.

[23] H. Kobayashi, M. Kaern, M. Araki, K. Chung, T. S. Gardner, C. R. Cantor, and J. J. Collins. Programmable cells: interfacing natural and engineered gene networks. *Proc Natl Acad Sci U S A*, 101(22):8414–9, 2004.

[24] J. Aleksic, F. Bizzari, Y. Cai, B. Davidson, K. De Mora, S. Ivakhno, S.L. Seshasayee, J. Nicholson, J. Wilson, A. Elfick, C. French, L. Kozma-Bognar, H. Ma, and A. Millar. Development of a novel biosensor for the detection of arsenic in drinking water. *Synthetic Biology, IET*, 1(1.2):87–90, June 2007.

[25] J. Lohmueller, N. Neretti, B. Hickey, A. Kaka, A. Gao, J. Lemon, V. Lattanzi, P. Goldstein, L.K. Tam, M. Schmidt, A.S. Brodsky, K. Haberstroh, J. Morgan, T. Palmore, G. Wessel, A. Jaklenec, H. Urabe, J. Gagnon, and J. Cumbers. Progress toward construction and modelling of a tri-stable toggle switch in e. coli. *Synthetic Biology, IET*, 1(1.2):25–28, June 2007.

[26] A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, 149(4):1633–48, 1998.

[27] D. M. Wolf and A. P. Arkin. Fifteen minutes of fim: control of type 1 pili expression in e. coli. *Omics*, 6(1):91–114, 2002.

[28] J. M. Vilar, C. C. Guet, and S. Leibler. Modeling network dynamics: the lac operon, a case study. *J Cell Biol*, 161(3):471–6, 2003.

[29] H. Salis and Y. Kaznessis. Numerical simulation of stochastic gene circuits. *Computers & Chemical Engineering*, 29(3):577, 2005.

[30] L.M. Tuttle, H. Salis, J. Tomshine, and Y.N. Kaznessis. Model-driven designs of an oscillating gene network. *Biophysical journal*, 89(6):3873–3883, 2005.

[31] N. J. Guido, X. Wang, D. Adalsteinsson, D. McMillen, J. Hasty, C. R. Cantor, T. C. Elston, and J. J. Collins. A bottom-up approach to gene regulation. *Nature*, 439(7078):856–60, 2006.

[32] H. Salis and Y. N. Kaznessis. Computer-aided design of modular protein devices: Boolean and gene activation. *Phys Biol*, 3(4):295–310, 2006.

[33] F. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[34] Mark Ptashne and Alexander Gann. *Genes & Signals*. Cold Spring Harbor, New York, 2002.

[35] W. Hillen and C. Berens. Mechanisms underlying expression of tn10 encoded tetracycline resistance. *Annu Rev Microbiol*, 48:345–69, 1994.

[36] H. H. McAdams and A. Arkin. It's a noisy business! genetic regulation at the nanomolar scale. *Trends Genet*, 15(2):65–9, 1999.

[37] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–6, 2002.

[38] D.A. McQuarrie. *Statistical Mechanics*. University Science Books, 2000.

[39] H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proc Natl Acad Sci U S A*, 94(3):814–9, 1997.

[40] T. B. Kepler and T. C. Elston. Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations. *Biophys J*, 81(6):3116–36, 2001.

[41] M. Thattai and A. van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proc Natl Acad Sci U S A*, 98(15):8614–9, 2001.

[42] R. Bundschuh, F. Hayot, and C. Jayaprakash. The role of dimerization in noise reduction of simple genetic networks. *J Theor Biol*, 220(2):261–9, 2003.

[43] R. Bundschuh, F. Hayot, and C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophys J*, 84(3):1606–15, 2003.

[44] C. D. Cox, G. D. Peterson, M. S. Allen, J. M. Lancaster, J. M. McCollum, D. Austin, L. Yan, G. S. Sayler, and M. L. Simpson. Analysis of noise in quorum sensing. *Omics*, 7(3):317–34, 2003.

[45] J. Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–418, 2004.

[46] A. Bar-Even, J. Paulsson, N. Maheshri, M. Carmi, E. O'Shea, Y. Pilpel, and N. Barkai. Noise in protein expression scales with natural protein abundance. *Nature genetics*, 38(6):636–643, 2006.

[47] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier, Amsterdam, revised and enlarged edition, 2004.

[48] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403 – 434, 1976.

[49] C. Gadgil, C.H. Lee, and H.G. Othmer. A stochastic analysis of first-order reaction networks. *Bulletin of mathematical biology*, 67(5):901–946, 2005.

[50] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem*, 81(25):2340 – 2361, 1977.

[51] Hiroyuki Kuwahara and Ivan Mura. An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. *The Journal of Chemical Physics*, 129(16):165101, 2008.

[52] Dan T. Gillespie, Min Roh, and Linda R. Petzold. Refining the weighted stochastic simulation algorithm. *The Journal of Chemical Physics*, 130(17):174103, 2009.

[53] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, 104(9):1876, 2000.

[54] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J Chem Phys*, 121(9):4059–67, 2004.

[55] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716, 2001.

[56] M. Rathinam, L. R. Petzold, Cao Yang, and D. T. Gillespie. Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. *Journal of Chemical Physics*, 119(24):12784, 2003.

[57] Y. Cao, D. T. Gillespie, and L. R. Petzold. Avoiding negative populations in explicit poisson tau-leaping. *J Chem Phys*, 123(5):054104, 2005. 0021-9606 (Print) Journal Article.

[58] Y. Cao, D. T. Gillespie, and L. R. Petzold. Efficient step size selection for the tau-leaping simulation method. *J Chem Phys*, 124(4):44109, 2006.

[59] D. T. Gillespie and L. R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *Journal of Chemical Physics*, 119(16):8229, 2003.

[60] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of chemical physics*, 121(21):10356, 2004.

[61] C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: application to the gillespie algorithm. *Journal of Chemical Physics*, 118(11):4999–5010, 2003.

[62] J. Puchalka and A. M. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophys J*, 86(3):1357–72, 2004.

[63] E. L. Haseltine and J. B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *Journal of Chemical Physics*, 117(15):6959, 2002.

[64] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J Chem Phys*, 122(5):54103, 2005.

[65] D. T. Gillespie. The chemical langevin equation. *Journal of Chemical Physics*, 113(1):297, 2000.

[66] J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J Chem Phys*, 122(18):184102, 2005.

[67] Y. Cao, D. T. Gillespie, and L. R. Petzold. The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, 122(1):014116, 2005.

[68] W. E, Liu Di, and E. Vanden-Eijnden. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *Journal of Chemical Physics*, 123(19):194107, 2005.

[69] H. Salis and Y. N. Kaznessis. An equation-free probabilistic steady-state approximation: dynamic application to the stochastic simulation of biochemical reaction networks. *J Chem Phys*, 123(21):214106, 2005.

[70] A. Samant and D. G. Vlachos. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: a multiscale monte carlo algorithm. *J Chem Phys*, 123(14):144114, 2005.

[71] Radek Erban, Ioannis G. Kevrekidis, David Adalsteinsson, and Timothy C. Elston. Gene regulatory networks: A coarse-grained, equation-free approach to multiscale computation. *The Journal of Chemical Physics*, 124(8):084106, 2006.

[72] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J Chem Phys*, 124(4):44104, 2006.

[73] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer-Verlag, Berlin, 1992.

[74] Y. N. Kaznessis. Multi-scale models for gene network engineering. *Chemical Engineering Science*, 61(3):940, 2006.

[75] Stochsim [computer program]. http://stochsim.sourceforge.net, 2007.

[76] Stochkit [computer program]. http://www.engineering.ucsb.edu/ cse/stochkit/, 2007.

[77] S. Ramsey, D. Orrell, and H. Bolouri. Dizzy: stochastic simulation of large-scale genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, 3(2):415–36, 2005.

[78] K. Burrage, P. M. Burrage, and T. Tian. Numerical methods for strong solutions of stochastic differential equations: an overview. *Proceedings of the Royal Society of London, Series A (Mathematical, Physical and Engineering Sciences)*, 460(2041):373, 2004.

[79] G. N. Milstein, E. Platen, and H. Schurz. Balanced implicit methods for stiff stochastic systems. *SIAM Journal on Numerical Analysis*, 35(3):1010, 1998.

[80] T. Tian and K. Burrage. Implicit taylor methods for stiff stochastic differential equations. *Applied Numerical Mathematics*, 38:167–185, 2001.

[81] P. M. Burrage and K. Burrage. A variable stepsize implementation for stochastic differential equations. *SIAM Journal of Scientific Computation*, 24(3):848–864, 2002.

[82] P. M. Burrage, R. Herdiana, and K. Burrage. Adaptive stepsize based on control theory for stochastic differential equations. *Journal of Computational and Applied Mathematics*, 171(1-2):317, 2004.

[83] J. G. Gaines and T. J. Lyons. Variable step size control in the numerical solution to stochastic differential equations. *SIAM Journal on Applied Mathematics*, 57(5):1455, 1997.

[84] H. Lamba. An adaptive timestepping algorithm for stochastic differential equations. *Journal of Computational and Applied Mathematics*, 161(2):417, 2003.

[85] A. Szepessy, R. Tempone, and G. E. Zouraris. Adaptive weak approximation of stochastic differential equations. *Communications on Pure and Applied Mathematics*, 54(10):1169–1214, 2001.

[86] P. Levy. Processus stochastiques et mouvement brownien. *Monographies des Probabilites*, 1948.

[87] H. Lamba, J. C. Mattingly, and A. M. Stuart. An adaptive euler-maruyama scheme for sdes: convergence and stability. *IMA J Numer Anal*, 27(3):479–506, 2007.

[88] R. Srivastava, M. S. Peterson, and W. E. Bentley. Stochastic kinetic analysis of the escherichia coli stress circuit using sigma(32)-targeted antisense. *Biotechnol Bioeng*, 75(1):120–9, 2001.

[89] K. Takahashi, K. Kaizu, B. Hu, and M. Tomita. A multi-algorithm, multi-timescale method for cell simulation. *Bioinformatics*, 20(4):538, 2004.

[90] Y. N. Kaznessis. Models for synthetic biology. *BMC Systems Biology*, 1(1):47, 2007.

[91] M. Kaern, T. C. Elston, W. J. Blake, and J. J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nat Rev Genet*, 6(6):451, 2005.

[92] M. S. Okino and M. L. Mavrovouniotis. Simplification of mathematical models of chemical reaction systems. *Chem. Rev.*, 98(2):391–408, 1998.

[93] N. Vora and P Daoutidis. Nonlinear model reduction of chemical reaction systems. *AIChE Journal*, 47(10):2320–2332, 2001.

[94] Ziomara P. Gerdtzen, Prodromos Daoutidis, and W. S. Hu. Non-linear reduction for kinetic models of metabolic reaction networks. *Metabolic Engineering*, 6(2):140, 2004.

[95] M. R. Maurya, S. J. Bornheimer, V. Venkatasubramanian, and S. Subramaniam. Reduced-order modelling of biochemical networks: application to the gtpase-cycle signalling module. *Systems Biology, IEE Proceedings*, 152(4):229, 2005.

[96] Iman Famili and Bernhard O. Palsson. The convex basis of the left null space of the stoichiometric matrix leads to the definition of metabolically meaningful pools. *Biophys. J.*, 85(1):16–26, 2003.

[97] E. V. Nikolaev, A. P. Burgard, and C. D. Maranas. Elucidation and structural analysis of conserved pools for genome-scale metabolic reconstructions. *Biophys. J.*, 88(1):37–49, 2005.

[98] J. A. M. Janssen. The elimination of fast variables in complex chemical reactions. ii. mesoscopic level (reducible case). *Journal of Statistical Physics*, 57(1):171, 1989.

[99] J. A. M. Janssen. The elimination of fast variables in complex chemical reactions. iii. mesoscopic level (irreducible case). *Journal of Statistical Physics*, 57(1):187, 1989.

[100] T. Shibata. Reducing the master equations for noisy chemical reactions. *The Journal of Chemical Physics*, 119(13):6629, 2003.

[101] S. Peles, B. Munsky, and M. Khammash. Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of Chemical Physics*, 125(20):204104, 2006.

[102] G. Dong, L. Jakobowski, M. Iafolla, and D. McMillen. Simplification of stochastic chemical reaction models with fast and slow dynamics. *Journal of Biological Physics*, 33(1):67, 2007.

[103] M.N. Contou-Carrere. *Model Reduction and Control of Multi Scale Processes: Reaction-Convection Processes and Chemical Langevin Models*. PhD thesis, University of Minnesota, 2005.

[104] M. N. Contou-Carrere and P. Daoutidis. Decoupling of fast and slow variables in chemical langevin equations with fast and slow reactions. In *American Control Conference, 2006*, page 6, 2006.

[105] C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and Natural Sciences*. Synergetics. Springer, Berlin, 3rd edition, 2004.

[106] I. Marini, L. Bucchioni, P. Borella, A. Del Corso, and U. Mura. Sorbitol dehydrogenase from bovine lens: Purification and properties. *Archives of Biochemistry and Biophysics*, 340(2):383, 1997.

[107] D. Riley, X. Koutsoukos, and K. Riley. Verification of biochemical processes using stochastic hybrid systems. In *IEEE 22nd International Symposium on Intelligent Control (ISIC)*, page 100, 2007.

[108] D.T. Gillespie. *Markov processes: An introduction for physical scientists.* Academic Press, 1992.

[109] J.E. Moyal. Stochastic processes and statistical physics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(2):150–210, 1949.

[110] J. Tomshine and Y. N. Kaznessis. Optimization of a stochastically simulated gene network model via simulated annealing. *Biophysical journal*, 91(9):3196–205, 2006.

[111] D. A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413, 1967.

[112] J. P. Hespanha and A. Singh. Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems. *International Journal of Robust and Nonlinear Control*, 15(15):669–689, 2005.

[113] John Goutsias. Classical versus stochastic kinetics modeling of biochemical reaction systems. *Biophys. J.*, 92(7):2350–2365, 2007.

[114] C. S. Gillespie. Moment-closure approximations for mass-action models. *Systems Biology, IET*, 3(1):52, 2009.

[115] C. H. Lee, K. Kim, and P. Kim. A moment closure method for stochastic reaction networks. *The Journal of Chemical Physics*, 130(13):134107, 2009.

[116] F. St-Pierre and D. Endy. Determination of cell fate selection during phage lambda infection. *Proceedings of the National Academy of Sciences*, 105(52):20705, 2008.

[117] I. Krishnarajah, A. Cook, G. Marion, and G. Gibson. Novel moment closure approximations in stochastic epidemics. *Bulletin of mathematical biology*, 67(4):855–873, 2005.

[118] A. K. Thakur, A. Rescigno, and C. DeLisi. Stochastic theory of second-order chemical reactions. *The Journal of Physical Chemistry*, 82(5):552–558, 1978.

[119] A. Singh and J.P. Hespanha. Moment closure techniques for stochastic models in population biology. In *American Control Conference*, page 6, June 2006.

[120] R. Gunawan, Y. Cao, L. Petzold, and 3rd Doyle, F. J. Sensitivity analysis of discrete stochastic systems. *Biophys J*, 88(4):2530–40, 2005.

[121] Ali Mohammad-Djafari. A matlab program to calculate the maximum entropy distributions, 2001.

[122] M. Hucka, A. Finney, HM Sauro, H. Bolouri, JC Doyle, H. Kitano, AP Arkin, BJ Bornstein, D. Bray, A. Cornish-Bowden, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[123] B. Canton, A. Labno, and D. Endy. Refinement and standardization of synthetic biological parts and devices. *Nat Biotech*, 26(7):787 – 793, 2008.

[124] S. Agha-Mohammadi and M. T. Lotze. Regulatable systems: applications in gene therapy and replicating viruses. *J Clin Invest*, 105(9):1177–83, 2000.

[125] S. Goverdhana, M. Puntel, W. Xiong, J. M. Zirger, C. Barcia, J. F. Curtin, E. B. Soffer, S. Mondkar, G. D. King, J. Hu, S. A. Sciascia, M. Candolfi, D. S. Greengold, P. R. Lowenstein, and M. G. Castro. Regulatable gene expression systems for gene therapy applications: progress and future challenges. *Mol Ther*, 12(2):189–211, 2005.

[126] C. Berens and W. Hillen. Gene regulation by tetracyclines. constraints of resistance regulation in bacteria shape tetr for application in eukaryotes. *Eur J Biochem*, 270(15):3109–21, 2003.

[127] U. Baron and H. Bujard. Tet repressor-based system for regulated gene expression in eukaryotic cells: principles and advances. *Methods Enzymol*, 327:401–21, 2000.

[128] U. Baron, M. Gossen, and H. Bujard. Tetracycline-controlled transcription in eukaryotes: novel transactivators with graded transactivation potential. *Nucleic Acids Res*, 25(14):2723–9, 1997.

[129] M. Gossen, S. Freundlieb, G. Bender, G. Muller, W. Hillen, and H. Bujard. Transcriptional activation by tetracyclines in mammalian cells. *Science*, 268(5218):1766–9, 1995.

[130] S. Urlinger, U. Baron, M. Thellmann, M. T. Hasan, H. Bujard, and W. Hillen. Exploring the sequence space for tetracycline-dependent transcriptional activators: novel mutations yield expanded range and sensitivity. *Proc Natl Acad Sci U S A*, 97(14):7963–8, 2000.

[131] J. K. Koponen, H. Kankkonen, J. Kannasto, T. Wirth, W. Hillen, H. Bujard, and S. Yla-Herttuala. Doxycycline-regulated lentiviral vector system with a novel reverse transactivator rtta2s-m2 shows a tight control of gene expression in vitro and in vivo. *Gene Ther*, 10(6):459–66, 2003.

[132] U. Deuschle, W. K. Meyer, and H. J. Thiesen. Tetracycline-reversible silencing of eukaryotic promoters. *Mol Cell Biol*, 15(4):1907–14, 1995.

[133] S. Freundlieb, C. Schirra-Muller, and H. Bujard. A tetracycline controlled activation/repression system with increased potential for gene transfer into mammalian cells. *J Gene Med*, 1(1):4–12, 1999.

[134] C. A. Strathdee, M. R. McLeod, and J. R. Hall. Efficient control of tetracycline-responsive gene expression from an autoregulated bi-directional expression vector. *Gene*, 229(1-2):21–9, 1999.

[135] M. Molin, M. C. Shoshan, K. Ohman-Forslund, S. Linder, and G. Akusjarvi. Two novel adenovirus vector systems permitting regulated protein expression in gene transfer experiments. *J Virol*, 72(10):8358–61, 1998.

[136] W. Jiang, L. Zhou, B. Breyer, T. Feng, H. Cheng, R. Haydon, A. Ishikawa, and T. C. He. Tetracycline-regulated gene expression mediated by a novel chimeric repressor that recruits histone deacetylases in mammalian cells. *J Biol Chem*, 276(48):45168–74, 2001.

[137] C. Berens, L. Altschmied, and W. Hillen. The role of the n terminus in tet repressor for tet operator binding determined by a mutational analysis. *J Biol Chem*, 267(3):1945–52, 1992.

[138] A. Wissmann, R. Baumeister, G. Muller, B. Hecht, V. Helbl, K. Pfleiderer, and W. Hillen. Amino acids determining operator binding specificity in the helix-turn-helix motif of tn10 tet repressor. *Embo J*, 10(13):4145–52, 1991.

[139] A. Wissmann, Jr. Wray, L. V., U. Somaggio, R. Baumeister, M. Geissendorfer, and W. Hillen. Selection for tn10 tet repressor binding to tet operator in escherichia coli: isolation of temperature-sensitive mutants and combinatorial mutagenesis in the dna binding motif. *Genetics*, 128(2):225–32, 1991.

[140] O. Scholz, P. Schubert, M. Kintrup, and W. Hillen. Tet repressor induction without mg2+. *Biochemistry*, 39(35):10914–20, 2000.

[141] V. Helbl and W. Hillen. Stepwise selection of tetr variants recognizing tet operator 4c with high affinity and specificity. *J Mol Biol*, 276(2):313–8, 1998.

[142] V. Helbl, B. Tiebel, and W. Hillen. Stepwise selection of tetr variants recognizing tet operator 6c with high affinity and specificity. *J Mol Biol*, 276(2):319–24, 1998.

[143] C. Sizemore, A. Wissmann, U. Gulland, and W. Hillen. Quantitative analysis of tn10 tet repressor binding to a complete set of tet operator mutants. *Nucleic Acids Res*, 18(10):2875–80, 1990.

[144] A. Wissmann, I. Meier, and W. Hillen. Saturation mutagenesis of the tn10-encoded tet operator o1. identification of base-pairs involved in tet repressor recognition. *J Mol Biol*, 202(3):397–406, 1988.

[145] A. Sigler, P. Schubert, W. Hillen, and M. Niederweis. Permeation of tetracyclines through membranes of liposomes and escherichia coli. *Eur J Biochem*, 267(2):527–34, 2000.

[146] M. M. Levandoski, O. V. Tsodikov, D. E. Frank, S. E. Melcher, R. M. Saecker, and Jr. Record, M. T. Cooperative and anticooperative effects in binding of the first and second plasmid osym operators to a laci tetramer: evidence for contributions of non-operator dna binding by wrapping and looping. *J Mol Biol*, 260(5):697–717, 1996.

[147] A. Kamionka, J. Bogdanska-Urbaniak, O. Scholz, and W. Hillen. Two mutations in the tetracycline repressor change the inducer anhydrotetracycline to a corepressor. *Nucleic Acids Res*, 32(2):842–7, 2004.

[148] S. Kedracka-Krok and Z. Wasylewski. Kinetics and equilibrium studies of tet repressor-operator interaction. *J Protein Chem*, 18(1):117–25, 1999.

[149] W. Hillen, C. Gatz, L. Altschmied, K. Schollmeier, and I. Meier. Control of expression of the tn10-encoded tetracycline resistance genes. equilibrium and kinetic investigation of the regulatory reactions. *J Mol Biol*, 169(3):707–21, 1983.

[150] J. B. Andersen, C. Sternberg, L. K. Poulsen, S. P. Bjorn, M. Givskov, and S. Molin. New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria. *Appl Environ Microbiol*, 64(6):2240–6, 1998.

[151] A. R. English, S. Y. P'an, T. J. McBride, J. F. Gardocki, G. Van Halsema, and A. W. Wright. Tetracycline-microbiologic, pharmacologic, and clinical evaluation. *Antibiotics Annual*, pages 70–80, 1953-1954.

[152] E. Bertrand-Burggraf, J. F. Lefevre, and M. Daune. A new experimental approach for studying the association between rna polymerase and the tet promoter of pbr322. *Nucleic Acids Res*, 12(3):1697–706, 1984.

[153] U. Vogel and K. F. Jensen. The rna chain elongation rate in escherichia coli depends on the growth rate. *J Bacteriol*, 176(10):2807–13, 1994.

[154] M. A. Sorensen and S. Pedersen. Absolute in vivo translation rates of individual codons in escherichia coli. the two glutamic acid codons gaa and gag are translated with a threefold difference in rate. *J Mol Biol*, 222(2):265–80, 1991.

[155] C. Kleinschmidt, K. Tovar, W. Hillen, and D. Porschke. Dynamics of repressor-operator recognition: the tn10-encoded tetracycline resistance control. *Biochemistry*, 27(4):1094–104, 1988.

Appendix A

Results for Chapter 6

A.1 Derivation of Third Order Moment Equations

In the derivation of the third moment equations we start again with the defining moment equation (cf. section 6.2.2)

$$\langle X_i X_j X_l \rangle = \int X_i X_j X_l P(\underline{X}, t) d\underline{X} \tag{A.1}$$

and the time derivative is defined as follows

$$\frac{d\langle X_j X_i X_l \rangle}{dt} = \int X_i X_j X_l \frac{\partial P(\underline{X}, t)}{\partial t} d\underline{X} \tag{A.2}$$

substituting equation (6.2) in the last equation we have

$$\begin{aligned} \frac{d\langle X_j X_i X_l \rangle}{dt} &= \int X_j X_i X_l \int \Big[T(\underline{X}/\underline{X}^{'}) P(\underline{X}^{'}, t) \Big] - T(\underline{X}^{'}/\underline{X}) P(\underline{X}, t) \Big] d\underline{X}^{'} d\underline{X} = \\ &= \int \int \Big[X_j X_i X_l T(\underline{X}/\underline{X}^{'}) P(\underline{X}^{'}, t) - X_j X_i X_l T(\underline{X}^{'}/\underline{X}) P(\underline{X}, t) \Big] d\underline{X}^{'} d\underline{X} \end{aligned} \tag{A.3}$$

Noticing that that the integration over $\underline{X}$ and $\underline{X}^{'}$ runs over the same domain we can interchange indexes in the last equation, i.e.

$$X_j X_i X_l T(\underline{X}/\underline{X}^{'}) P(\underline{X}^{'}, t) = X_j^{'} X_i^{'} X_l^{'} T(\underline{X}^{'}/\underline{X}) P(\underline{X}, t) \tag{A.4}$$

thus we have

$$\frac{d\big\langle X_j X_i X_l \big\rangle}{dt} = \int\int (X_j^{'} X_i^{'} X_l^{'} - X_j X_i X_l) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \tag{A.5}$$

Skipping through some tedious calculations we use the following relation to transform the RHS of equation (A.5)

$$\begin{aligned}(X_j^{'} X_i^{'} X_l^{'} - X_i X_j X_l) = {} & (X_i^{'} - X_i)(X_j^{'} - X_j)(X_l^{'} - X_l) \\ & + X_i(X_j^{'} - X_j)(X_l^{'} - X_l) + X_j(X_i^{'} - X_i)(X_l^{'} - X_l) \\ & + X_l(X_i^{'} - X_i)(X_j^{'} - X_j) + X_i X_j(X_l^{'} - X_l) \\ & + X_i X_l(X_j^{'} - X_j) + X_j X_l(X_i^{'} - X_i)\end{aligned} \tag{A.6}$$

substituting yields

$$\begin{aligned}\frac{d\big\langle X_i X_j X_l \big\rangle}{dt} = {} & \int\int (X_i^{'} - X_i)(X_j^{'} - X_j)(X_l^{'} - X_l) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_i(X_j^{'} - X_j)(X_l^{'} - X_l) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_j(X_i^{'} - X_i)(X_l^{'} - X_l) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_l(X_i^{'} - X_i)(X_j^{'} - X_j) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_i X_j(X_l^{'} - X_l) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_i X_l(X_j^{'} - X_j) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \\ & + \int\int X_j X_l(X_i^{'} - X_i) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X}\end{aligned} \tag{A.7}$$

Using the definitions of the joint jump moments (cf. eq. (6.7)) and that of the averages the RHS eventually becomes

$$\begin{aligned}\frac{d\big\langle X_iX_jX_l\big\rangle}{dt} &= \big\langle a_3^{ijl}(\underline{X})\big\rangle \\ &+ \big\langle X_i a_2^{jl}(\underline{X})\big\rangle + \big\langle X_j a_2^{il}(\underline{X})\big\rangle + \big\langle X_l a_2^{ij}(\underline{X})\big\rangle \\ &+ \big\langle X_iX_j a_1^{l}(\underline{X})\big\rangle + \big\langle X_iX_l a_1^{j}(\underline{X})\big\rangle + \big\langle X_jX_l a_1^{i}(\underline{X})\big\rangle \\ \\ & i,j,l = 1,\ldots,N\end{aligned} \tag{A.8}$$

where $\underline{\underline{a}}_3(\underline{X})$ is a third order tensor. In the trivial case where $i = j = l$ the last equation simplifies to

$$\frac{d\big\langle X_i^3\big\rangle}{dt} = \big\langle a_3^{iii}(\underline{X})\big\rangle + 3\big\langle X_i a_2^{ii}(\underline{X})\big\rangle + 3\big\langle X_i^2 a_1^{i}(\underline{X})\big\rangle \qquad i = 1,\ldots,N \tag{A.9}$$

A.2 Derivation of Fourth Order Moment Equations

For the fourth order moments the starting point is again the moment defining equation (cf. section 6.2.2)

$$\big\langle X_iX_jX_lX_m\big\rangle = \int X_iX_jX_lX_m P\big(\underline{X},t\big)d\underline{X} \tag{A.10}$$

and the time derivative is defined as follows

$$\frac{d\big\langle X_jX_iX_lX_m\big\rangle}{dt} = \int X_iX_jX_lX_m \frac{\partial P\big(\underline{X},t\big)}{\partial t}d\underline{X} \tag{A.11}$$

substituting equation (6.2) in the last equation we have

$$\begin{aligned}&\frac{d\big\langle X_jX_iX_lX_m\big\rangle}{dt} = \int X_jX_iX_lX_m \int \Big[T\big(\underline{X}/\underline{X}^{'}\big)P\big(\underline{X}^{'},t\big) - T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)\Big]d\underline{X}^{'}d\underline{X} = \\ &= \int\int \Big[X_jX_iX_lX_m T\big(\underline{X}/\underline{X}^{'}\big)P\big(\underline{X}^{'},t\big) - X_jX_iX_lX_m T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)\Big]d\underline{X}^{'}d\underline{X}\end{aligned} \tag{A.12}$$

Similarly to the third order moments case the integration in equation (A.12) over

$\underline{X}$ and $\underline{X}^{'}$ runs over the same domain thus we can interchange indexes, i.e.

$$X_j X_i X_l X_m T\big(\underline{X}/\underline{X}^{'}\big) P\big(\underline{X}^{'}, t\big) = X_j^{'} X_i^{'} X_l^{'} X_m^{'} T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) \tag{A.13}$$

thus we have

$$\frac{d\big\langle X_j X_i X_l\big\rangle}{dt} = \int\int (X_j^{'} X_i^{'} X_l^{'} X_m^{'} - X_j X_i X_l X_m) T\big(\underline{X}^{'}/\underline{X}\big) P\big(\underline{X}, t\big) d\underline{X}^{'} d\underline{X} \tag{A.14}$$

Skipping through some tedious calculations we use the following relation to transform the RHS of equation (A.5)

$$\begin{aligned}
(X_j^{'} X_i^{'} X_l^{'} X_m^{'} - X_i X_j X_l X_m) = {} & (X_i^{'} - X_i)(X_j^{'} - X_j)(X_l^{'} - X_l)(X_m^{'} - X_m) \\
& + X_i(X_j^{'} - X_j)(X_l^{'} - X_l)(X_m^{'} - X_m) \\
& + X_j(X_i^{'} - X_i)(X_l^{'} - X_l)(X_m^{'} - X_m) \\
& + X_l(X_i^{'} - X_i)(X_j^{'} - X_j)(X_m^{'} - X_m) \\
& + X_m(X_i^{'} - X_i)(X_j^{'} - X_j)(X_l^{'} - X_l) \\
& + X_i X_j(X_l^{'} - X_l)(X_m^{'} - X_m) + X_i X_l(X_j^{'} - X_j)(X_m^{'} - X_m) \\
& + X_i X_m(X_j^{'} - X_j)(X_l^{'} - X_l) + X_j X_l(X_i^{'} - X_i)(X_m^{'} - X_m) \\
& + X_j X_m(X_i^{'} - X_i)(X_l^{'} - X_l) + X_l X_m(X_i^{'} - X_i)(X_j^{'} - X_j) \\
& + X_i X_j X_l(X_m^{'} - X_m) + X_i X_j X_m(X_l^{'} - X_l) \\
& + X_i X_l X_m(X_j^{'} - X_j) + X_j X_l X_m(X_i^{'} - X_i)
\end{aligned} \tag{A.15}$$

substituting yields

$$
\begin{aligned}
\frac{d\langle X_i X_j X_l\rangle}{dt} &= \int\int (X_i^{'}-X_i)(X_j^{'}-X_j)(X_l^{'}-X_l)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_i(X_j^{'}-X_j)(X_l^{'}-X_l)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_j(X_i^{'}-X_i)(X_l^{'}-X_l)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_l(X_i^{'}-X_i)(X_j^{'}-X_j)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_m(X_i^{'}-X_i)(X_j^{'}-X_j)(X_l^{'}-X_l)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_j(X_l^{'}-X_l)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_l(X_j^{'}-X_j)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_m(X_j^{'}-X_j)(X_l^{'}-X_l)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_jX_l(X_i^{'}-X_i)(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_jX_m(X_i^{'}-X_i)(X_l^{'}-X_l)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_lX_m(X_i^{'}-X_i)(X_j^{'}-X_j)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_jX_l(X_m^{'}-X_m)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_jX_m(X_l^{'}-X_l)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_iX_lX_m(X_j^{'}-X_j)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \\
&+\int\int X_jX_lX_m(X_i^{'}-X_i)T\big(\underline{X}^{'}/\underline{X}\big)P\big(\underline{X},t\big)d\underline{X}^{'}d\underline{X} \qquad \text{(A.16)}
\end{aligned}
$$

Using the definitions of the joint jump moments (cf. eq. (6.7)) and that of the averages the RHS eventually becomes

$$
\begin{aligned}
\frac{d\big\langle X_i X_j X_l X_m \big\rangle}{dt} &= \big\langle a_4^{ijlm}(\underline{X}) \big\rangle \\
&+ \big\langle X_i a_3^{jlm}(\underline{X}) \big\rangle + \big\langle X_j a_3^{ilm}(\underline{X}) \big\rangle \\
&+ \big\langle X_l a_3^{ijm}(\underline{X}) \big\rangle + \big\langle X_m a_3^{ijl}(\underline{X}) \big\rangle \\
&+ \big\langle X_l X_m a_2^{ij}(\underline{X}) \big\rangle + \big\langle X_j X_m a_2^{il}(\underline{X}) \big\rangle + \big\langle X_j X_l a_2^{im}(\underline{X}) \big\rangle \\
&+ \big\langle X_i X_m a_2^{jl}(\underline{X}) \big\rangle + \big\langle X_i X_l a_2^{jm}(\underline{X}) \big\rangle + \big\langle X_i X_j a_2^{lm}(\underline{X}) \big\rangle \\
&+ \big\langle X_i X_j X_l a_1^{m}(\underline{X}) \big\rangle + \big\langle X_i X_j X_m a_1^{l}(\underline{X}) \big\rangle \qquad \text{(A.17)} \\
&+ \big\langle X_i X_l X_m a_1^{j}(\underline{X}) \big\rangle + \big\langle X_j X_l X_m a_1^{i}(\underline{X}) \big\rangle \\
\\
& i, j, l, m = 1, \ldots, N
\end{aligned}
$$

where $\underline{\underline{a}}_4(\underline{X})$ is a fourth order tensor. In the trivial case where $i = j = l = m$ the last equation simplifies to

$$
\frac{d\big\langle X_i^4 \big\rangle}{dt} = \big\langle a_4^{iiii}(\underline{X}) \big\rangle + 4\big\langle X_i a_3^{iii}(\underline{X}) \big\rangle + 6\big\langle X_i^2 a_2^{ii}(\underline{X}) \big\rangle + 4\big\langle X_i^3 a_2^{i}(\underline{X}) \big\rangle \qquad i = 1, \ldots, N \qquad \text{(A.18)}
$$