Diss. ETH No. 19598

Numerical Methods for Stochastic Processes with Applications in Chemical Kinetics and Biology

A dissertation submitted to
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by
BASIL BAYATI
B.S., University of California,rvine
M.S., Chalmers University of Technology

Date of birth
July, 27th 1982

citizen of
U.S.A.

accepted on the recommendation of
Prof. Dr. P. Koumoutsakos, examiner
Prof. Dr. L. Petzold, co-examiner
Prof. Dr. J. Stelling, co-examiner

2011
Abstract

In this thesis continuous-time stochastic processes will be considered, which are increasingly being used as models for physical phenomena ranging from economics to biochemistry. The sine qua non of these models is that there exist variables that are assumed to be governed by the laws of probability theory. Stochastic chemical kinetics, which is an important example of a stochastic system, describes the time evolution of chemically reacting systems, in which molecules are inherently discrete and exhibit some degree of randomness in their dynamical behavior.

Numerical methods for continuous-time Markov processes are presented for problems that have a particular feature or characteristic. Specifically, methods are expounded for the simulation of stochastic reaction-diffusion problems using both uniform and adaptive, multiresolution meshes. In the latter case, an algorithm for the application of adaptive mesh refinement to mesoscopic stochastic simulations of spatially evolving reaction-diffusion processes is developed. Another method is presented that is capable of handling, in a time-adaptive fashion, stochastic systems that have events with delays. Lastly, a method is described for the simulation of stochastic systems that are characterized by stiffness, the error of which is shown to exhibit a cutoff phenomenon as a function of its speed-up. Examples from the field of chemical kinetics are provided for all of the proposed methods, as well as analyses of the numerical errors and optimizations.

Two problems are presented that are modeled using continuous-time stochastic processes, the first of which is a biologically-oriented study of the cytoplasmic transport of adenoviruses, and the second is the dissemination and proliferation of brain tumors.
Zusammenfassung


Zwei Probleme werden präsentiert, die mittels zeitkontinuierlicher Prozesse modelliert werden. Das Erstere ist eine biologische Untersuchung des zytoplasmatischen Transports von Adenoviren. Das zweite Problem befasst sich mit der Dissemination und dem Wachstum von Hirntumoren.
Acknowledgements

My gratitude goes first to my supervisor, Petros Koumoutsakos, for giving me this opportunity. He was a continuous source of ideas and encouragement, without whom this thesis would not be what it is. Furthermore, Petros initiated and fostered collaborations - both in the lab and abroad - that significantly contributed to this thesis.

I am also indebted to Philippe Chatelain for his constant help, inquisitiveness, and sense of humor, all of which I appreciated throughout his time in the lab.

Houman Owhadi at Caltech and Eric Mjolsness at UC Irvine deserve a special mention, not only for being precipitants of rewarding collaborations, but for being mentors and teachers as well.

I am also grateful for all of the collaborations I had within the lab, particularly with Mattia Gazzola and Diego Rossinelli. Urs Greber, Christoph Burckhardt, and Martin Engelke at the University of Zürich are people to whom I am indebted for their eagerness to share their biological knowledge.

The doctoral committee, in particular Linda Petzold and Jörg Stelling, deserve recognition as well for their willingness to be critical reviewers of this thesis.

This thesis is dedicated to my parents and my sister, who endearingly support me in any endeavor I choose to undertake.
Chaos umpire sits,
And by decision more embroils the fray
By which he reigns: next him high arbiter,
Chance governs all.

—Milton—
Contents

1 Introduction 1

I Theory 7

2 Theory of Stochastic Processes 9
  2.1 Introduction ......................................................... 9
  2.2 Elements of Probability Theory ...................................... 10
    2.2.1 Elementary Probability Theory .............................. 10
    2.2.2 Discrete Random Variables ................................. 12
    2.2.3 Continuous Random Variables ............................... 13
  2.3 Markov Processes .............................................. 14
    2.3.1 Example of Brownian Motion ............................... 14
    2.3.2 Definition .................................................. 15
    2.3.3 Chapman-Kolmogorov Equation ............................. 16
  2.4 Master Equation ............................................... 17
    2.4.1 Derivation from the Chapman-Kolmogorov Equation ......... 17
    2.4.2 $\Psi$—Matrices & General Solution ......................... 18
    2.4.3 Kramers-Moyal Expansion .................................. 19
    2.4.4 $\Omega$-Expansion ......................................... 20
  2.5 Stochastic Differential Equations ............................. 27
    2.5.1 Introduction ............................................... 27
    2.5.2 Definition ............................................... 27
2.5.3 Itô's Lemma ........................................ 28
2.6 Chemical Reactions ................................. 29
  2.6.1 Chemical Reactions as Stochastic Phenomena 29
  2.6.2 Master Equation Formulation ................. 29

II Methods .............................................. 33

3 Monte Carlo Methods ............................... 35
  3.1 Introduction ....................................... 35
  3.2 Convergence of $O(N^{-1/2})$ .................. 36
  3.3 Numerical Integration ............................ 37
  3.4 Random Walk ...................................... 38
  3.5 Poisson's Equation ............................... 39
    3.5.1 Periodic Boundaries ......................... 40
    3.5.2 Broken Disk Problem ....................... 41
  3.6 Stochastic Simulation Algorithm ............... 43
    3.6.1 Gillespie Algorithm ......................... 43
    3.6.2 Convergence .................................. 44
    3.6.3 Discrepancy Between Continuum & Discrete Simulations 45
  3.7 Approximate Stochastic Simulation Algorithms 48
    3.7.1 $\tau$-Leaping Algorithm .................. 48
    3.7.2 $R$-Leaping Algorithm ...................... 50
    3.7.3 Connection between $\tau$- and $R$-Leaping 52
  3.8 Euler-Maruyama Method ........................... 53

4 Accelerated Stochastic Simulations in Space .... 57
  4.1 Uniform Discretizations of the Diffusion Process 57
    4.1.1 Introduction .................................. 57
    4.1.2 Reaction-Diffusion: Stochastic and Deterministic Models 57
    4.1.3 Computational Methods ...................... 60
4.1.4 Numerical Results ........................................ 63
4.1.5 Conclusion ................................................. 65

4.2 Multiresolution Stochastic Reaction-Diffusion Systems in 1-D ............ 66
4.2.1 Introduction ................................................. 66
4.2.2 The Method ................................................. 67
4.2.3 Temporal Scale Disparity: Diffusion Propensities ................. 68
4.2.4 Temporal Scale Disparity, Reaction and Diffusion Propensities ...... 68
4.2.5 Numerical Results ........................................... 71
4.2.6 Conclusion ................................................. 73

4.3 Multiresolution Stochastic Reaction-Diffusion Systems using AMR ....... 74
4.3.1 Introduction ................................................. 74
4.3.2 Reaction-Diffusion Master Equation ................................ 75
4.3.3 General Formulation of Stochastic Simulation Algorithms .......... 76
4.3.4 Diffusion .................................................... 77
4.3.5 Reactions and the Homogeneity Condition .......................... 81
4.3.6 Adaptivity .................................................... 82
4.3.7 Numerical Results ........................................... 86
4.3.8 Conclusion ................................................. 95

5 Accelerated Stochastic Simulations in Time .................................... 97
5.1 Processes with Delays ........................................ 97
5.1.1 Introduction ................................................. 97
5.1.2 Continuum Chemical Kinetics and Delay Differential Equations .... 98
5.1.3 SSA and Accelerated, Approximate Algorithms .................... 98
5.1.4 Accelerated, Approximate SSA for delayed reactions .............. 100
5.1.5 D-Leaping .................................................. 101
5.1.6 Numerical Results ........................................... 105
5.1.7 Conclusion .................................................. 109

5.2 Flow-Averaging Algorithms ....................................... 111
5.2.1 Introduction ................................................. 111
5.2.2 Flow Averaging Integration for Ordinary Differential Equations ... 112
5.2.3 Flow Averaging Integration for Markov Processes ................. 113

III Applications 125

6 A Stochastic Model of Microtubule Motors 127
6.1 Introduction .......................................................... 127
6.2 A Stochastic Model for Cargo Transport .......................... 128
6.3 Results ................................................................. 129
6.4 Discussion ............................................................... 135

7 Pattern Formation 139
7.1 Introduction .......................................................... 139
7.2 Uniform-Mesh Simulations ......................................... 140
7.3 Adaptive Simulations ............................................... 140

8 Simulations of Brain Tumors 145
8.1 Introduction .......................................................... 145
8.2 Numerical Method ................................................... 146
  8.2.1 Inhomogeneous Diffusion ...................................... 147
  8.2.2 Boundary Conditions ........................................... 149
  8.2.3 Numerical Validation ........................................... 150
8.3 3-D Simulations using MRAG ..................................... 150
  8.3.1 Biological Data .................................................. 151
  8.3.2 Stochastic Model ............................................... 151

IV Conclusion & Outlook 155

9 Conclusion 157
10 Outlook 161
V Appendix & Curriculum Vitae
Chapter 1

Introduction

The subject of stochastic processes is one of the two children of probability theory - the other being statistics - that has its roots in games of chance. It was in the 20th century that scientists began to realize that certain physical and chemical processes could be likened to these games. This was, coincidentally enough, not serendipitous; for scientists were coming to the conclusion that their favorite tools for modeling the world, differential equations, were either themselves becoming unpredictable (chaos theory) or were losing their predictive power (quantum theory).

One of the earliest accounts of a stochastic phenomenon in nature was given by the botanist Robert Brown in an unpublished article entitled A Brief Account of Microscopical Observations Made in the Months of June, July, and August, 1827, on the Particles Contained in the Pollen of Plants; and on the General Existence of Active Molecules in Organic and Inorganic Bodies, in which he described his experimental research [1]:

“The plant was Clarkia pulchella, of which the grains of pollen, taken from antherae full grown of unusually large size, varying from nearly \( \frac{1}{4000} \) th to about \( \frac{1}{5000} \) th of an inch in length, and of a figure between cylindrical and oblong, perhaps slightly flattened, and having rounded and equal extremities. While examining the form of these particles immersed in water, I observed many of them very evidently in motion; their motion consisting [...] of a change of place in the fluid, manifested by alterations in their relative positions, [...]”

Brown continued his enquiry with an experiment involving inanimate objects:

“In plants, either dried or immersed in spirit for a few days only, the particles of pollen of both kinds were found in motion equally evident with that observed in the living plant; specimens of several plants, some of which had been dried and preserved in an...
"berbarium for upwards of twenty years, and others not less than a century, still exhibited the molecules or smaller spherical particles in considerable numbers, and in evident motion, [...]"

Figure 1.1: Brownian motion: the trajectory of a pollen granule in suspension [1].

This peculiar motion, as Brown later called it, became an active area of research, particularly by those who were interested in the internal motion of fluids; and it later became clear that the erratic motion of the particles of pollen was due to collisions with the fluid molecules. An example of a trajectory of a granule suspended in a fluid is shown in Figure 1.1. This example of Brownian motion is important not only because it was one of the first accounts of stochastic processes in nature, but also because it is one of the most fundamental processes in nature.

Brownian motion has ties to kinetic theory - the theory of particles that are constantly exhibiting random motion - which has had consequences in the theory of chemical reactions. Since particles in a chemical system are discrete, the evolution cannot simply be regarded as a continuous process. Moreover, if the particles are suspended in a fluid, it is impossible to know the exact position of any individual particle due to Brownian motion.

It is interesting in its own right, but Brownian motion is often part of a more complex physical process such as chemical systems, an example of which is the Belousov-Zhabotinsky reaction-diffusion system. This phenomenon was first discovered experimentally in the 1950s and is an example of an excitable chemical system. If a stimulus is applied in an otherwise homogeneous system, patterns and wave propagation arise as shown in Figure 1.2.
Figure 1.2: Images from experiments with the Belousov-Zhabotinsky reactions [2, 3].

This behavior was also predicted theoretically by Turing and it was hypothesized that rather simple reaction-diffusion equations can give rise to self-organization in biology, examples of which range from the trivial, e.g. the pattern of an animal's coat, to the essential, e.g. morphogenesis. Indeed Turing opined in The Chemical Basis of Morphogenesis [4] that reaction-diffusion systems are sufficient for describing morphogenesis:

“It is suggested that a system of chemical substances, called morphogens, reacting together and diffusing through a tissue, is adequate to account for the main phenomena of morphogenesis. Such a system, although it may originally be quite homogeneous, may later develop a pattern or structure due to an instability of the homogeneous equilibrium, which is triggered off by random disturbances.”

Figure 1.3 shows pattern formation in numerical simulations as described by Turing. These patterns are the result of the Grey-Scott equations, which will be discussed further in this thesis.

The underlying theme throughout his thesis will be that of simulating stochastic processes via Monte Carlo methods. Much of the thesis deals with chemical kinetics but it should be noted this is an example of a process that depends on random events and we need not limit ourselves to chemistry; indeed most of the methods presented in this thesis can be applied to any discrete-space, continuous-time Markov process.

Although there exists a large corpus for deterministic numerical methods, the same cannot necessarily be said of stochastic methods. Monte Carlo methods have become exten-
Chapter 1. Introduction

Figure 1.3: Pattern formation simulations of the Grey-Scott equations (Image from Pearson [5]).

sively studied, but not to the same extent of numerical methods for ordinary and partial differential equations. Research in this field is warranted since there is a keen interest in Monte Carlo methods not only because they are well suited for stochastic problems, but also because intractable or insoluble deterministic problems can be recast into equivalent stochastic processes that may be efficiently solved [6].

This thesis is organized as follows:

Chapter 2 introduces the notation and elementary concepts from probability theory and the theory of stochastic processes that will be used throughout the thesis. Essential topics such as discrete and continuous random variables, Markov processes, the master equation, and stochastic differential equations will be expounded here. A special emphasis in this chapter has been placed on the expansions of the master equation owing to my own partiality. Nothing in this section is my own work, but was collected predominantly from the texts of A.N. Shiryaev [7], W. Feller [8], N.G. van Kampen [9], C. Gardiner [10], and T.C. Gard [11].

Chapter 3 is essentially a survey of currently used Monte Carlo methods, some of which are classical and well-known, e.g. methods for numerical integration, and some of which are
more esoteric, e.g. for solving Poisson's equation. This section also introduces the Stochastic Simulation Algorithm [12,13], which is the primary numerical method used in this thesis, and its descendants [14–16]. The theory in this section, unlike the numerical examples, is not my own work. The texts of I.T. Dimov [6] and D.T. Gillespie [12,13] are heavily cited in this chapter.

Chapter 4 contains methods for accelerated stochastic simulations in space and time. Specifically, reaction-diffusion processes using both uniform and multiresolution meshes are presented. Furthermore, temporal scale disparities are analyzed and methods are discussed for simulations using adaptive mesh refinement. Numerical examples of propagating fronts and pattern forming systems [4] are used to validate the methods. Section 4.1 was done in collaboration with Diego Rossinelli and Petros Koumoutsakos, and Philippe Chatelain and Petros Koumoutsakos were involved in the work for Sections 4.2 and 4.3.

In Chapter 5, accelerated stochastic simulations in time are discussed. Section 5.1 is an exposition of a numerical method for the simulation of stochastic chemical reactions, in which the products are not instantaneously produced when the reactants collide but rather are produced some time later. These so-called delayed processes are more realistic models of chemical reactions since time is often needed to convert the reactants to products. The research carried out in this chapter was done with Philippe Chatelain and Petros Koumoutsakos. Section 5.2 expounds a numerical method for the simulation of stiff stochastic processes. The method is based on averaging the flow of a dynamical system that has hidden slow and fast variables in the spirit of flow averaging integrators for differential equations [17]. Additionally, a cutoff phenomenon is shown to exist with respect to a parameter that dictates the error and speed-up of the numerical simulation. Two examples are provided to validate the method. Houman Owhadi and Petros Koumoutsakos contributed to this chapter.

Chapter 6 presents biologically-oriented research on the transport of viruses inside of cells. Molecular motors, due to their transportation function, are essential to the cell, but they are often exploited by viruses to reach their replication site. In this work, in vivo imaging is used to collect intracellular virus trajectories, which are in turn used to optimize the parameters of a stochastic model of motor proteins. The model is able to explain the in vivo trajectories by virtue of a stochastic interaction between motors. Furthermore it enables prediction to be made on the number of motors and binding sites on pathogens, the values of which are difficult to obtain experimentally. Mattia Gazzola, Christoph Burckhardt, Martin Engelke, Urs Greber, and Petros Koumoutsakos all contributed to this research.

Chapter 7 consists of results from the simulations of pattern-formation equations. The Grey-Scott equations are dealt with using the methods presented in Chapter 4. The differences between continuum and discrete simulations are presented here.
Chapter 8 presents 3-D simulations of brain tumor growth. A stochastic differential equation is used to model the spread of tumorous cells inside of a brain modeled by white and grey matter. Diego Rossinelli and Petros Koumoutsakos contributed to the research carried out in this chapter.

Chapter 9 concludes this thesis and Chapter 10 consists of a discussion of possible future work emanating from this thesis.
Part I

Theory
Chapter 2

Theory of Stochastic Processes

2.1 Introduction

The epigraph of this thesis was purposely chosen to be an excerpt from a poem since it was the analysis of the form and structure of verse that inspired Andrei Markov in his work on probability theory [18]. Specifically, the question that Markov dealt with was the following: given an arbitrary consonant from the poem *Eugeny Onegin* by Alexander Pushkin, what is the probability that the next letter would be a vowel? Moreover, given just the current state, i.e. a letter, is it possible to determine the probability of the next state? Table 2.1 shows the results that Markov obtained. This analysis led to Markov’s eponymous stochastic processes, which are arguably the most important stochastic processes because they can be related to fundamental physical phenomena such as Brownian motion and chemical kinetics. These processes form the theoretical basis of this thesis.

<table>
<thead>
<tr>
<th>letter(s)</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>0.568</td>
</tr>
<tr>
<td>$V$</td>
<td>0.432</td>
</tr>
<tr>
<td>$C</td>
<td>C$</td>
</tr>
<tr>
<td>$V</td>
<td>C$</td>
</tr>
<tr>
<td>$C</td>
<td>V$</td>
</tr>
<tr>
<td>$V</td>
<td>V$</td>
</tr>
</tbody>
</table>

Table 2.1: Markov’s results after analyzing 20,000 letters from *Eugeny Onegin*: $C$ denotes a consonant, $V$ a vowel, $V|C$ denotes a consonant followed by a vowel [18].
2.2 Elements of Probability Theory

Notation, definitions, and elementary relations of probability theory will be provided in this section. The rest of the thesis, and in particular Markov processes and the master equation, will rely heavily upon the rudimentary ideas developed here. Following Shiryaev [7], the subject matter is that which has:

- *deterministic irregularity*, meaning that observations yield different outcomes,
- *statistical regularity*, in the sense that observations have regularity in their frequencies.

2.2.1 Elementary Probability Theory

Following Shiryaev [7] and Feller [8], let \( s_1, \ldots, s_n \) be elementary events, and denote the space of elementary events, or *sample space*, by

\[
S = \{s_1, \ldots, s_n\}.
\]

For a single letter from a text, the sample space \( S \) consists of two elementary points, namely \( S = \{C, V\} \), where \( C \) and \( V \) represent consonant and vowel, respectively. The sample space for \( k \) letters results in \( S = \{s : s = (a_1, \ldots, a_k), a_i = C \text{ or } V\} \), where the number \( N(S) \) is \( 2^k \).

An *event* is defined as a subset of the sample space, namely \( A \subset S \), where \( s \in A \) or \( s \notin A \). Given a collection of events, new events can be formed by virtue of logical connectives. The *union* of two sets \( A \) and \( B \) is

\[
A \cup B = \{s \in S : s \in A \text{ or } s \in B\},
\]

and likewise the *intersection* is

\[
A \cap B = \{s \in S : s \in A \text{ and } s \in B\}
\]

Consider an example involving two letters. Let \( A = \{CC, CV, VC\} \) and \( B = \{VV, VC, CV\} \), then \( A \cup B = S = \{CC, CV, VC, VV\} \) and \( A \cap B = \{CV, VC\} \). The *complement* of \( A \), \( \bar{A} \), is defined such that \( A \cap \bar{A} = \emptyset \). The *difference* between two sets \( A \) and \( B \), which is defined as the set of points in \( A \) but not in \( B \), is denoted by \( A \setminus B \). If \( A \) and \( B \) are disjoint, \( A \cap B = \emptyset \), then \( A \cup B = A + B \), which is called the *sum* of \( A \) and \( B \).

Let \( S_0 \) denote the collection of subsets of \( S \), namely \( S_0 \subseteq S \), the operators introduced above can be used to construct a new collection of sets, which are themselves events. Adjoining the
2.2. Elements of Probability Theory

certain, event $S$ and the impossible event $\emptyset$ to $\mathcal{S}_0$ yields a so-called algebra, which is denoted by $\mathcal{S}$. This algebra has the following properties:

1. $S \in \mathcal{S}$

2. if $A \in \mathcal{S}$ and $B \in \mathcal{S}$, then $A \cup B \in \mathcal{S}$, $A \cap B \in \mathcal{S}$, and $A \setminus B \in \mathcal{S}$.

Typically, $\mathcal{S}$ is taken to be the algebra of all subsets of $S$.

**Definition of Probability** Given a discrete sample space $S$ with elementary points $s_1, \ldots, s_n$, we shall assume that with each point $s_j$ there is an associated number, called the probability of $s_j$ and denoted by $P(s_j)$. It is to be non-negative and such that

$$P(s_1) + \ldots + P(s_n) = 1. \quad (2.4)$$

**Definition of the Probability of an Event** The probability $P(A)$ of any event $A$ is the sum of the probabilities of all elementary points in it, namely

$$P(A) = \sum_{\{i: s_i \in A\}} P(s_i). \quad (2.5)$$

**Definition of the Probability Space** Given a sample space $S$, an algebra of subsets of $S$ denoted by $\mathcal{S}$, and $P = \{P(A) : A \in \mathcal{S}\}$, then

$$(S, \mathcal{S}, P) \quad (2.6)$$

constitutes the probability space.

From (2.5), the following properties can be stated:

$$P(S) = 1, \quad (2.7)$$
$$P(\emptyset) = 0, \quad (2.8)$$
$$P(A \cup B) = P(A) + P(B) - P(A \cap B), \quad (2.9)$$

if $P(A \cap B) = \emptyset$, then

$$P(A + B) = P(A) + P(B), \quad (2.10)$$
$$P(\bar{A}) = 1 - P(A). \quad (2.11)$$
Table 2.2: Example of a random variable: the number of consonants in a probabilistic model of two letters.

<table>
<thead>
<tr>
<th>$s$</th>
<th>$CC$</th>
<th>$CV$</th>
<th>$VC$</th>
<th>$VV$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi(s)$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Definition of the Conditional Probability** The *conditional probability* of event $A$, under the assumption of event $B$ and $P(B) > 0$ is

$$P(A|B) = \frac{P(A \cap B)}{P(B)} ,$$  \hspace{1cm} (2.12)

where $P(A \cap B)$ will be written as $P(AB)$ if the context is clear. Note that if $A$ and $B$ are independent, $P(A|B) = P(A)P(B)/P(B) = P(A)$.

**2.2.2 Discrete Random Variables**

**Definition of a Random Variable** A *random variable* is any function $\xi = \xi(s)$ defined on a sample space $S$.

**Example 1** As an example, let $\xi(s)$ be the number of consonants in a model with two letters. The possible values of $s$ are $CC, CV, VC$, and $VV$, and the corresponding values of the random variable $\xi(s)$ will be 2, 1, 1 and 0, as shown in Table 2.2.

Let $X = \{x_1, \ldots, x_m\}$, where $\xi \in X$. In other words, $X$ is a set that represents all possible values of $\xi$.

**Definition of the Probability Distribution of a Discrete Random Variable** The *probability distribution* or *probability density of the random variable* $\xi$ is set of numbers $\{P_\xi(x_1), \ldots, P_\xi(x_m)\}$, where

$$P_\xi(x_i) = P\{s : \xi(s) = x_i\}. \hspace{1cm} (2.13)$$

In example 1, note that $P(VC) = P(V|C)P(C) = P(CV) = 0.377$, and therefore, $P_\xi(x = 2) = 0.191$, $P_\xi(x = 1) = 0.754$, and $P_\xi(x = 0) = 0.055$.

**Definition of the Distribution Function of a Discrete Random Variable** The *distribution* or *cumulative density function* of the random variable $\xi$ is

$$F_\xi(x) = P\{s : \xi(s) \leq x\} = \sum_{\{i : x_i < x\}} P_\xi(x_i), \hspace{1cm} (2.14)$$
2.2. Elements of Probability Theory

which has the property that $F_\xi(-\infty) = 0$ and $F_\xi(+\infty) = 1$.

**Definition of the Expectation of a Discrete Random Variable** The *expectation, mathematical expectation, expected value, or first moment* of the random variable $\xi$ is

$$E[\xi] = \sum_i x_i P_\xi(x_i),$$

(2.15)

In example 1, $E[\xi] = 0 \cdot 0.055 + 1 \cdot 0.754 + 2 \cdot 0.191 = 1.136$.

**Definition of the Moments of a Discrete Random Variable** The (centered) *moments* of the random variable $\xi$ are

$$M_r^\xi = \sum_i (x_i - E[\xi])^r P_\xi(x_i) = E[(\xi - E[\xi])^r].$$

(2.16)

If $r = 2$, the moment is called the *variance* and is denoted by $\mathbb{V}[\xi]$.

### 2.2.3 Continuous Random Variables

Analogously to discrete random variables, let $\xi = \xi(s)$ be a continuous random variable.

**Definition of the Distribution Function of a Continuous Random Variable** The *distribution or cumulative density function* of the random variable $\xi$ is

$$F_\xi(x) = P\{s : \xi(s) \leq x\}, \quad x \in \mathbb{R}.$$  

(2.17)

**Definition of the Probability Distribution of a Continuous Random Variable** The *probability distribution or probability density of the random variable* $\xi$ is

$$P_\xi(x) = \frac{d}{dx} F_\xi(x).$$

(2.18)

Moreover, the relation between the distribution function and the density function is

$$F_\xi(x) = \int_{-\infty}^x P_\xi(x') \, dx'.$$

(2.19)

The expectation and moments are integral versions of the discrete case:
Definition of the Expectation of a Continuous Random Variable  The expectation, mathematical expectation, expected value, or first moment, of the random variable $\xi$ is

$$E[\xi] = \int x P_\xi(x) \, dx. \quad (2.20)$$

Definition of the Moments of a Continuous Random Variable  The (centered) moments of the random variable $\xi$ are

$$M_\xi = \int (x - E[\xi])^r P_\xi(x) \, dx = E[(\xi - E[\xi])^r], \quad (2.21)$$

where $\forall[\xi] \triangleq M_\xi^2$.

## 2.3  Markov Processes

### 2.3.1  Example of Brownian Motion

There is perhaps no more famous Markov process than that of Brownian motion, which describes the random movement of particles suspended in a fluid. The following derivation by Einstein is indicative of the methods and assumptions that are pervasive in the modelling of random phenomena.

Following Gardiner [10] and Einstein [19], assume that there are a total of $n$ particles suspended in a fluid. It is assumed that each individual particle executes a motion that is independent of all of the other particles. Furthermore, the movement of a particle at a different time interval is an independent process relative to the current time interval. In a time interval $\tau$, the $x$-coordinates of each particle will increase by an amount $\Delta$, where $\Delta$ has a different value, both positive and negative, for each particle. A certain frequency law will dictate the values of $\Delta$. The following can be written for the evolution of the $n$ molecules:

$$\frac{dn}{d\Delta} = n\phi(\Delta), \quad (2.22)$$

where

$$\int_{-\infty}^{\infty} \phi(\Delta) \, d\Delta = 1, \quad (2.23)$$

and where $\phi(\Delta)$ is symmetric, namely $\phi(\Delta) = \phi(-\Delta)$.

Let $u(x, t)$ be the number of particles per unit volume. Then,

$$u(x, t + \tau) = \int_{-\infty}^{\infty} u(x + \Delta, t) \phi(\Delta) \, d\Delta. \quad (2.24)$$
Since $\tau$ is small, Taylor’s series may be used to expand $u(x, t + \tau)$.

$$u(x, t + \tau) = u(x, t) + \tau \frac{\partial u(x, t)}{\partial t}. \quad (2.25)$$

Furthermore,

$$u(x + \Delta, t) = u(x, t) + \Delta \frac{\partial u(x, t)}{\partial x} + \frac{\Delta^2}{2} \frac{\partial^2 u(x, t)}{\partial x^2}, \quad (2.26)$$

Inserting this into the integral equation (2.24) yields

$$u + \tau \frac{\partial u}{\partial t} = \int_{-\infty}^{\infty} u \phi(\Delta) d\Delta + \int_{-\infty}^{\infty} \Delta \frac{\partial u}{\partial x} \phi(\Delta) d\Delta + \int_{-\infty}^{\infty} \frac{\Delta^2}{2} \frac{\partial^2 u(x, t)}{\partial x^2} \phi(\Delta) d\Delta, \quad (2.27)$$

$$\tau \frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \int_{-\infty}^{\infty} \Delta \phi(\Delta) d\Delta + \frac{\partial^2 u}{\partial x^2} \int_{-\infty}^{\infty} \frac{\Delta^2}{2} \phi(\Delta) d\Delta, \quad (2.28)$$

$$\frac{\partial u}{\partial t} = \Delta \frac{\partial^2 u}{\partial x^2}, \quad (2.29)$$

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}. \quad (2.30)$$

Equation (2.30) is the diffusion equation. In the derivation we have used the fact that

$$\int_{-\infty}^{\infty} \Delta \phi(\Delta) d\Delta = 0, \quad (2.31)$$

because of the symmetry of $\phi(\Delta)$, and we have defined the macroscopic diffusion coefficient $D$ as

$$D \triangleq \tau^{-1} \int_{-\infty}^{\infty} \frac{\Delta^2}{2} \phi(\Delta) d\Delta. \quad (2.32)$$

Einstein was able to show the connection between a stochastic description of Brownian motion, using the density function $\phi(\Delta)$, and the continuum description, i.e. the diffusion equation. Furthermore, the derivation contains many of the concepts in the theory of stochastic processes such as the Chapman-Kolmogorov equation (similar to equation (2.24)), the Kramers-Moyal expansion (which can be likened to equations (2.25) and (2.26)), and Fokker-Planck Equation (equation (2.30) being a special case of the Fokker-Planck equation). Each of these essential steps will be treated in detail below.

### 2.3.2 Definition

Following van Kampen [9], let $Y(t) = f(\xi, t)$ be a time-dependent random variable, where $f$ is an arbitrary mapping and $\xi$ is a random variable. The function $Y = Y(t)$ is a **stochastic process** since it depends on a random variable and on time. In the simplest case, and the
one considered in this thesis, \( f \) and \( \xi \) are identity functions, and thus the stochastic process takes on an elementary point in the sample space at each point in time. \( Y \) can take on certain states \( y_1, \ldots, y_n \) successively, where \( y_k \in S \).

The probability that \( Y = y_1 \) at time \( t_1 \), \( Y = y_2 \) at time \( t_2 \), ..., and \( Y = y_n \) at time \( t_n \) is denoted by the joint probability distribution

\[
P(y_1, t_1; y_2, t_2; \ldots; y_n, t_n) \triangleq P(y_1 \cap y_2 \cap \ldots \cap y_n, t_n). \tag{2.33}
\]

The Markov property is

\[
P(y_n, t_n|y_1, t_1; \ldots; y_{n-1}, t_{n-1}) = P(y_n, t_n|y_{n-1}, t_{n-1}). \tag{2.34}
\]

where \( t_1 < t_2 < \ldots < t_n \) and \( P(y_n, t_n|y_{n-1}, t_{n-1}) \) is called the transition probability.

In words, the states and times from 1 to \( n - 2 \) are not taken into account. The process is memoryless, in the sense that the future state depends only on the current state.

A Markov process is fully determined by combinations of two functions, namely \( P(y_k, t_k) \) and \( P(y_{k+1}, t_{k+1}|y_k, t_k) \). Three steps of a Markov chain yield

\[
P(y_1, t_1; y_2, t_2; y_3, t_3) = P(y_3, t_3|y_1, t_1; y_2, t_2)P(y_1, t_1; y_2, t_2) \tag{2.35}
\]

\[
= P(y_3, t_3|y_2, t_2)P(y_1, t_1; y_2, t_2) \tag{2.36}
\]

\[
= P(y_3, t_3|y_2, t_2)P(y_2, t_2|y_1, t_1)P(y_1, t_1) \tag{2.37}
\]

\[
= P(y_1, t_1)P(y_2, t_2|y_1, t_1)P(y_3, t_3|y_2, t_2) \tag{2.38}
\]

This procedure can be done successively to find all time points.

### 2.3.3 Chapman-Kolmogorov Equation

Integrating the identity (2.38) over \( y_2 \) and dividing by \( P(y_1, t_1) \), one obtains

\[
P(y_1, t_1; y_2, t_2; y_3, t_3) = P(y_1, t_1)P(y_2, t_2|y_1, t_1)P(y_3, t_3|y_2, t_2) \tag{2.39}
\]

\[
\int P(y_1, t_1; y_2, t_2; y_3, t_3) \, dy_2 = \int P(y_1, t_1)P(y_2, t_2|y_1, t_1)P(y_3, t_3|y_2, t_2) \, dy_2 \tag{2.40}
\]

\[
P(y_1, t_1; y_3, t_3) = P(y_1, t_1)\int P(y_2, t_2|y_1, t_1)P(y_3, t_3|y_2, t_2) \, dy_2 \tag{2.41}
\]

\[
P(y_3, t_3|y_1, t_1)P(y_1, t_1) = P(y_1, t_1)\int P(y_2, t_2|y_1, t_1)P(y_3, t_3|y_2, t_2) \, dy_2 \tag{2.42}
\]

\[
P(y_3, t_3|y_1, t_1) = \int P(y_3, t_3|y_2, t_2)P(y_2, t_2|y_1, t_1) \, dy_2 \tag{2.43}
\]
2.4 Master Equation

Equation (2.43) is the Chapman-Kolmogorov equation. The transition probabilities of any Markov process must obey the Chapman-Kolmogorov equation. The time ordering is imperative: $t_2$ lies between $t_1$ and $t_3$. This equation expresses the fact that a process starting at $t_1$ with value $y_1$ reaches $y_3$ at $t_3$ via any one of the possible values $y_2$ at the intermediate time $t_2$. If $y$ takes discrete values, the integral becomes a summation.

2.4 Master Equation

2.4.1 Derivation from the Chapman-Kolmogorov Equation

To simplify the notation, we introduce the following variables:

\begin{align*}
P(y_3, t_3|y_2, t_2) &= T_\tau(y_3|y_2) \text{ where } \tau = t_3 - t_2, \quad (2.44) \\
P(y_2, t_2|y_1, t_1) &= T_\tau(y_2|y_1) \text{ where } \tau = t_2 - t_1. \quad (2.45)
\end{align*}

The Chapman-Kolmogorov equation (2.43) becomes

\begin{equation}
T_{\tau+\hat{\tau}}(y_3|y_1) = \int T_{\hat{\tau}}(y_3|y_2) T_{\tau}(y_2|y_1) \, dy_2. \quad (2.46)
\end{equation}

Under the assumption that $\hat{\tau}$ is small, Taylor’s series can be used to expand the transition probability $T_{\hat{\tau}}(y_3|y_2)$:

\begin{equation}
T_{\hat{\tau}}(y_3|y_2) = \delta(y_3 - y_2) + \hat{\tau}W(y_3|y_2) + O(\hat{\tau}^2), \quad (2.47)
\end{equation}

where the delta function expresses the probability of remaining in the same state. $W(y_3|y_2)$ is the derivative of the transition probability and is called the transition probability per unit time. This expression must integrate to unity, therefore

\begin{equation}
T_{\hat{\tau}}(y_3|y_2) = (1 - \hat{\tau}\alpha_0(y_2))\delta(y_3 - y_2) + \hat{\tau}W(y_3|y_2) + O(\hat{\tau}^2), \quad (2.48)
\end{equation}

where the coefficient represents the probability for no transition to have taken place and where

\begin{equation}
\alpha_0(y_2) = \int W(y_3|y_2) \, dy_3. \quad (2.49)
\end{equation}
Inserting equation (2.48) into equation (2.46) and disregarding terms of $O(\hat{\tau}^2)$ yields

$$T_{\tau+\hat{\tau}}(y_3|y_1) = \int [(1 - \hat{\tau}\alpha_0(y_2))\delta(y_3 - y_2) + \hat{\tau}W(y_3|y_2)] T_{\tau}(y_2|y_1) \, dy_2, \quad (2.50)$$

$$= \int (1 - \hat{\tau}\alpha_0(y_2))\delta(y_3 - y_2)T_{\tau}(y_2|y_1) \, dy_2 +$$

$$\hat{\tau} \int W(y_3|y_2)T_{\tau}(y_2|y_1) \, dy_2, \quad (2.51)$$

$$= (1 - \hat{\tau}\alpha_0(y_3))T_{\tau}(y_3|y_1) + \hat{\tau} \int W(y_3|y_2)T_{\tau}(y_2|y_1) \, dy_2, \quad (2.52)$$

$$= \left(1 - \hat{\tau}\int W(y_3|y_2) \, dy_2\right) T_{\tau}(y_3|y_1) + \hat{\tau} \int W(y_3|y_2)T_{\tau}(y_2|y_1) \, dy_2. \quad (2.53)$$

Rearranging yields

$$\frac{T_{\tau+\hat{\tau}}(y_3|y_1) - T_{\tau}(y_3|y_1)}{\hat{\tau}} = \hat{\tau} \int \left\{W(y_3|y_2)T_{\tau}(y_2|y_1) - W(y_2|y_3)T_{\tau}(y_3|y_1)\right\} \, dy_2. \quad (2.55)$$

Sending $\hat{\tau} \to 0$ yields

$$\frac{\partial T_{\tau}(y_3|y_1)}{\partial \tau} = \int \left\{W(y_3|y_2)T_{\tau}(y_2|y_1) - W(y_2|y_3)T_{\tau}(y_3|y_1)\right\} \, dy_2. \quad (2.57)$$

The equation can be rewritten into a simplified and more intuitive form as

$$\frac{\partial P(y,t)}{\partial t} = \int \left\{W(y'y)P(y',t) - W(y'|y)P(y,t)\right\} \, dy'. \quad (2.58)$$

If $y$ is discrete, i.e. the sample space $\Omega$ is discrete, then denoting the indices of $y$ by the labels $n$ (e.g. $n = y_2, n' = \{y_1, y_3, y_4, \ldots\}$) results in

$$\frac{dp_n(t)}{dt} = \sum_{n'} \left\{W_{n,n'}p_{n'}(t) - W_{n',n}p_n(t)\right\}. \quad (2.59)$$

### 2.4.2 W-Matrices & General Solution

The master equation, (2.59), is a system of ordinary differential equations that can be written in the following matrix/vector form

$$\frac{dp}{dt} = W \cdot p. \quad (2.60)$$
where the matrix $\mathbb{W}$ has elements defined as

$$\mathbb{W}_{n,n'} = a_{n,n'} - \delta_{n,n'} \left( \sum_{n''} a_{n,n''} \right).$$  \hfill (2.61)

where $a_{n,n'} \, \text{dt}$ represents the probability of transitioning from state $n'$ to $n$ within some infinitesimal time-step $\text{dt}$. The rates $a_{n,n'}$ are called propensities.

**Example of Propensities from a Text**  If the sample space is $\Omega = \{A, B, C, \ldots, Z\}$, then $a_{n-E,n'-R} \, \text{dt}$ would represent the probability of transitioning from state $R$ to state $E$, where the states could represent the characters in a text.

**Example of Propensities from Chemical Kinetics**  Consider the process of a chemical decay. The sample space is $\Omega = \mathbb{Z}$, where e.g. $a_{99,99'-100} \, \text{dt}$ would represent the probability of transitioning from a state in which there are 100 molecules to a state in which there are 99 molecules, i.e. one molecule decayed within the time-step $\text{dt}$.

The formal solution of equation (2.60) is

$$p(t) = e^{t\mathbb{W}} \cdot p(0).$$  \hfill (2.62)

The solution however is unsatisfactory since $\mathbb{W}$ is usually large or countably infinite. Since $\mathbb{W}$ is in general not symmetric, eigenvalue/eigenvector decomposition methods cannot be used.

The properties of the $\mathbb{W}$-matrices are

$$\mathbb{W}_{n,n'} \geq 0 \text{ for } n \neq n',$$  \hfill (2.63)

$$\sum_n \mathbb{W}_{n,n'} = 0 \text{ for each } n.$$  \hfill (2.64)

### 2.4.3 Kramer-Moyal Expansion

Following Gardiner [10], Kramers [20], and Moyal [21], we will make a change of variables in the master equation and then expand the equation using a power series to obtain the so-called Fokker-Planck or Kolmogorov Forward equation. Substituting $x = y - y'$ into the first term and $x = y' - y$ into the second term, and defining $t(x, y) \triangleq W(y + x|y)$, yields

$$\frac{\partial P}{\partial t} = \int \left\{ W(y|y')P(y') - W(y'|y)P(y) \right\} \, dy',$$  \hfill (2.65)

$$= \int \left\{ t(x, y - x)P(y - x) - t(x, y)P(y) \right\} \, dx.$$  \hfill (2.66)
Now we expand in a power series,
\[
\frac{\partial P}{\partial t} = \int \left\{ t(x, y - x)P(y - x) - t(x, y)P(y) \right\} \, dx, \tag{2.67}
\]
\[
= \int \left\{ \sum_{n=1}^{\infty} \frac{(-x)^n}{n!} \frac{\partial^n}{\partial y^n} [t(x, y)P(y)] \right\} \, dx. \tag{2.68}
\]
Defining the moments as
\[
\alpha_n(y) = \int (y' - y)^n W(y'|y) \, dy' = \int x^n t(x, y) \, dx,
\]
yields
\[
\frac{\partial P}{\partial t} = \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\partial^n}{\partial y^n} [\alpha_n(y)P(y)]. \tag{2.70}
\]
Disregarding terms \( n > 2 \) gives the Fokker-Planck equation:
\[
\frac{\partial P}{\partial t} = -\frac{\partial}{\partial y} [\alpha_1(y)P(y)] + \frac{1}{2} \frac{\partial^2}{\partial y^2} [\alpha_2(y)P(y)]. \tag{2.71}
\]
Although elegant, the Kramers-Moyal expansion is not a systematic expansion of a small parameter, meaning that we cannot safely disregard terms \( n > 2 \) in equation (2.70). Van Kampen formulated the so-called \( \Omega \)-Expansion as a perturbation problem [22] by identifying a meaningful parameter, the system size \( \Omega \), in the master equation. The idea of a perturbation theory is to identify a small parameter, usually denoted by \( \epsilon \), and then to stipulate the solution of a problem as a series (the ansatz), e.g.: \( u(t) = u_0(t) + \epsilon u_1(t) + \epsilon^2 u_2(t) \), such that the higher order terms become negligible and the \( u_i \)'s are unknown.

### 2.4.4 \( \Omega \)-Expansion

Two examples will be given, the first of which elucidates the expansion for a simple problem, and the second of which attempts to justify the ansatz used in the expansion.

**\( \Omega \)-Expansion for a Single Species Chemical System.** Let \( S = \{s_1 = 0, s_2 = 1, s_2 = 3, \ldots \} = \mathbb{Z} \), i.e. the sample space is the number of molecules of the species \( X \). The \( \Omega \)-expansion is essentially a mapping from the sample space \( S \) to \( S' \), where \( S' = \mathbb{R} \). In other words, the objective is to approximate the discrete sample space (master equation (2.59)) by a continuous sample space (Fokker-Planck equation (2.71)).

We begin with a model chemical system:
\[
A \xrightarrow{k} X, \tag{2.72}
\]
\[
2X \xrightarrow{k'} B. \tag{2.73}
\]
2.4. Master Equation

where $A$ is a constant and $B$ is not of interest. We therefore have a single variable, namely $X$. The following differential (macroscopic) equation can be written for the evolution of the concentration of $X$

$$\frac{d\phi}{dt} = k\phi_A - 2k'\phi^2,$$  \hspace{1cm} (2.74)

where the concentration of $k\phi_A = 1$, $k = 1$, and $k' = 1/2$.

![Diagram of states of the stochastic process](image)

(a) States of the stochastic process

![Diagram of evolution of the density function](image)

(b) Evolution of the density function

Figure 2.1: Shown in 2.1(a) is the jump process with discrete states labelled by $n$. Figure 2.1(b) depicts the ansatz and the typical evolution of the density function (image from [9]).

The master equation can be written for the stochastic process (see Figure 2.1):

$$\frac{dp_n}{dt} = \Omega \left( \mathbb{L}^{-1} - 1 \right) p_n + \frac{1}{2\Omega} \left( \mathbb{L}^{+2} - 1 \right) n(n-1)p_n,$$  \hspace{1cm} (2.75)
Chapter 2. Theory of Stochastic Processes

where \( n \) denotes the number of molecules of \( X \), \( p_n(t) \) is the probability of having \( n \) molecules at time \( t \) and the shift operators are defined as

\[
L^+[p_n] = p(n + m, t),
\]
\[
L^-[p_n] = p(n - m, t).
\]  

(2.76)

(2.77)

Now we define the following ansatz (see Figure 2.1) for the number of molecules in the system:

\[
n = \Omega \phi(t) + \Omega^{1/2} \xi,
\]

(2.78)

which leads to the change of variables

\[
p_n(t) = p(n, t) = p(\Omega \phi(t) + \Omega^{1/2} \xi, t) = \Pi(\xi, t),
\]

(2.79)

where \( \Pi(\xi, t) \) is now the function of interest. We notice that if \( \xi \) is replaced by \( \xi + \Omega^{-1/2} \), this is equivalent to an increment of \( n \), namely: \( \xi \to \xi + \Omega^{-1/2} \Rightarrow n \to n + 1 \). The shift operators can then be expanded in the form of a power series (note that using the notation of perturbation theory [22], \( \epsilon = \Omega^{-1/2} \))

\[
\Pi(\xi + \Omega^{-1/2}, t) = \mathcal{L}^{+1}[\Pi(\xi, t)] = \left(1 + \Omega^{-1/2} \frac{\partial}{\partial \xi} + \frac{1}{2!} \Omega^{-1} \frac{\partial^2}{\partial \xi^2} + \ldots \right) \Pi(\xi, t),
\]

(2.80)

and in general

\[
L^+ = \left(1 + m \Omega^{-1/2} \frac{\partial}{\partial \xi} + \frac{m^2}{2!} \Omega^{-1} \frac{\partial^2}{\partial \xi^2} + \ldots \right),
\]

\[
L^- = \left(1 - m \Omega^{-1/2} \frac{\partial}{\partial \xi} + \frac{m^2}{2!} \Omega^{-1} \frac{\partial^2}{\partial \xi^2} - \ldots \right).
\]

(2.81)

(2.82)

The time-derivative of \( p_n \) becomes

\[
\frac{dp_n}{dt} = \frac{\partial \Pi}{\partial t} - \Omega^{1/2} \frac{d\phi}{dt} \frac{\partial \Pi}{\partial \xi}
\]

(2.83)

We can now write the Master-Equation in terms of the new variables:

\[
\frac{d\Pi}{dt} - \Omega^{1/2} \frac{d\phi}{dt} \frac{\partial \Pi}{\partial \xi} = \Omega \left(-\Omega^{-1/2} \frac{\partial}{\partial \xi} + \frac{1}{2!} \Omega^{-1} \frac{\partial^2}{\partial \xi^2} \right) \Pi + \frac{1}{2} \Omega^{-1} \left[2\Omega^{-1/2} \frac{\partial}{\partial \xi} + 2\Omega^{-1/2} \frac{\partial^2}{\partial \xi^2} \right] \Pi \left(\Omega \phi(t) + \Omega^{1/2} \xi \right) \left(\Omega \phi(t) + \Omega^{1/2} \xi - 1 \right).
\]

(2.84)

Expanding the Master-Equation and collecting terms in decreasing powers of \( \Omega \) yields

\[
\frac{d\Pi}{dt} - \Omega^{1/2} \frac{d\phi}{dt} \frac{\partial \Pi}{\partial \xi} = \Omega^{1/2} \left(-\frac{\partial \Pi}{\partial \xi} + \phi^2 \frac{\partial^2 \Pi}{\partial \xi^2} \right) + \Omega^0 \left(2\phi \frac{\partial \Pi}{\partial \xi} (\xi^2 \Pi) + \frac{1}{2} \frac{\partial^2 \Pi}{\partial \xi^2} + \phi^2 \frac{\partial^3 \Pi}{\partial \xi^3} \right) + \Omega^{-1/2} \left(\frac{\partial}{\partial \xi} \left(\xi^2 \Pi \right) - \phi \frac{\partial \Pi}{\partial \xi} + 2\phi \frac{\partial^2 \Pi}{\partial \xi^2} (\xi \Pi) \right) + \Omega^{-1} \left(-\frac{\partial}{\partial \xi} \left(\xi^2 \Pi \right) + \frac{\partial^2 \Pi}{\partial \xi^2} + 2\phi \frac{\partial^3 \Pi}{\partial \xi^3} \right) - \Omega^{-3/2} \left(\frac{\partial^2 \Pi}{\partial \xi^2} (\xi \Pi) \right).
\]

(2.85)
2.4. Master Equation

![Comparison of the result of the Ω-Expansion (top) with the solution of the master equation (bottom). Note the change of the state space from discrete, \( S \), to continuous, \( S' \).](image)

At first sight this is not a proper expansion since there are terms that diverge when \( \Omega \to \infty \). However, the \( \Omega^{1/2} \) terms vanish if we insert \( \frac{d\phi}{d\pi} \) leaving a proper expansion and justifying our initial ansatz

\[
\frac{\partial \Pi}{\partial t} = \Omega^0 \left( 2\phi \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{1}{2} \frac{\partial^2 \Pi}{\partial \xi^2} + \phi^2 \frac{\partial^2 \Pi}{\partial \xi^2} \right) + \Omega^{-1/2} \left( \frac{\partial}{\partial \xi} (\xi^2 \Pi) - \phi \frac{\partial \Pi}{\partial \xi} + 2\phi \frac{\partial^2 \Pi}{\partial \xi^2} \right) + \Omega^{-1} \left( -\frac{\partial}{\partial \xi} (\xi \Pi) + \frac{\partial^2}{\partial \xi^2} (\xi^2 \Pi) - \phi \frac{\partial^2 \Pi}{\partial \xi^2} \right) - \Omega^{-3/2} \left( \frac{\partial^2}{\partial \xi^2} (\xi \Pi) \right).
\]  

(2.86)

Collecting the lowest order terms yields

\[
\frac{\partial \Pi}{\partial t} = 2\phi \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{1}{2} \left( 1 + 2\phi^2 \right) \frac{\partial^2 \Pi}{\partial \xi^2},
\]  

(2.87)

which is the Fokker-Planck equation (vide equation (2.71)), the solution of which is a Gaussian distribution [9]. It therefore suffices to determine the first and second moments (vide equations (2.20)-(2.21)). Multiplying both sides of the Fokker-Planck equation by \( \xi \) and \( \xi^2 \) and integrating yields, respectively

\[
\int_{-\infty}^{\infty} \xi \frac{\partial \Pi}{\partial t} \, d\xi = \int_{-\infty}^{\infty} \xi \left[ 2\phi \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{1}{2} \left( 1 + 2\phi^2 \right) \frac{\partial^2 \Pi}{\partial \xi^2} \right] \, d\xi
\]  

(2.88)

\[
\int_{-\infty}^{\infty} \xi^2 \frac{\partial \Pi}{\partial t} \, d\xi = \int_{-\infty}^{\infty} \xi^2 \left[ 2\phi \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{1}{2} \left( 1 + 2\phi^2 \right) \frac{\partial^2 \Pi}{\partial \xi^2} \right] \, d\xi
\]  

(2.89)

The following integrals reappear in all Fokker-Planck equations, and thus in all expansions of the Master-Equation, and it is useful to solve, with appropriate boundary conditions,
Chapter 2. Theory of Stochastic Processes

beforehand:

$$\int_{-\infty}^{\infty} \xi \left[ \frac{\partial}{\partial \xi} (\xi \Pi) \right] \, d\xi = -\langle \xi \rangle, \quad (2.90)$$

$$\int_{-\infty}^{\infty} \xi \left[ \frac{\partial^2 \Pi}{\partial \xi^2} \right] \, d\xi = 0, \quad (2.91)$$

$$\int_{-\infty}^{\infty} \xi^2 \left[ \frac{\partial}{\partial \xi} (\xi \Pi) \right] \, d\xi = -2 \langle \xi^2 \rangle, \quad (2.92)$$

$$\int_{-\infty}^{\infty} \xi^2 \left[ \frac{\partial^2 \Pi}{\partial \xi^2} \right] \, d\xi = 2, \quad (2.93)$$

where we have used the fact that the function $\Pi$ vanishes at $\pm\infty$:

$$\Pi(\pm\infty, t) = 0, \quad (2.94)$$

$$\frac{\partial \Pi}{\partial \xi}(\pm\infty, t) = 0. \quad (2.95)$$

After integration, the following ordinary differential equations are obtained for the evolution of the moments:

$$\frac{d \langle \xi \rangle}{dt} = -2\phi \langle \xi \rangle, \quad (2.96)$$

$$\frac{d \langle \xi^2 \rangle}{dt} = -4\phi \langle \xi^2 \rangle + 1 + 2\phi^2. \quad (2.97)$$

The initial condition is a Delta function, therefore $\langle \xi \rangle_0 = \langle \xi^2 \rangle_0 = 0$ and the only solution for the first moment is the trivial solution $\langle \xi \rangle_t = 0$. If $\phi(t = 0) = 0$ then

$$\phi(t) = \frac{-1 + e^{2t}}{1 + e^{2t}} \quad (2.98)$$

$$\langle \xi^2 \rangle_t = \frac{-3 - 8e^{2t} + 8e^{6t} + 3e^{8t} + 8e^{4t}}{4(1 + e^{2t})^4} \quad (2.99)$$

and $\langle \xi^2 \rangle_{t \to \infty} = \frac{3}{3}$. We are now in a position to use our initial ansatz:

$$\langle n \rangle_t = \Omega \phi(t) + \Omega^{1/2} \langle \xi \rangle_t, \quad (2.100)$$

$$\langle n^2 \rangle_t = \Omega \langle \xi^2 \rangle_t. \quad (2.101)$$

Additionally, we notice the following convergence to the differential (macroscopic) equation:

$$\langle n \rangle_t = \Omega \phi(t) + \Omega^{1/2} \langle \xi \rangle_t, \quad (2.102)$$

which has an error associated with $\langle \xi \rangle_t$

$$\langle n \rangle_t = \Omega \phi(t) + \Omega^{1/2} \left( \langle \xi \rangle_t + C_1 \Omega^{-1/2} + C_2 \Omega^{-1} + C_3 \Omega^{-3/2} \right), \quad (2.103)$$
2.4. Master Equation

The distinction lies in the fact that microscopically we must keep track of example consider a trilmolecular reaction such as $y q \nu m$ manner gsee [v] for more detailshk but first we must make a distinction between the macroscopic reaction propensities $gFigure pqhm equation that governs the change in the probability distribution of the fluctuationsk call it $\Omega^{1/2}$ is directly proportional to the molecule numbersk allowing the molecule numbers to be written

$$\frac{\langle n \rangle_t}{\Omega} = \phi(t) + \mathcal{O}(\Omega^{-1}).$$

(2.104)

The steady-state solution resulting from the expansion is shown in Figure 2.2 where the solution of the master equation is also shown for reference. A schematic is shown in Figure 2.3 that illustrates the expansion.

**Justifying the Ansatz** Consider the reversible isomerization reaction:

$$X_1 \xrightarrow{k} X_2; \quad (2.105)$$
$$X_2 \xrightarrow{k} X_1, \quad (2.106)$$

where $X_1 + X_2 = \Omega$ (conservation). $S = \{s_1 = [0, U], s_2 = [1, U - 1], s_2 = [2, U - 2], \ldots, s_n = [U, 0]\} = [0, U]^2$. Let $n$ denote the number of molecules of $X_1$ and $\phi$ the deterministic solution of the concentration of $X_1$. We therefore have (macroscopic model)

$$\frac{d\phi}{dt} = -2k\phi + k, \quad (2.107)$$

and the Master-Equation is

$$\frac{\partial p(n, t)}{\partial t} = (L^{+1} - 1)knp + (L^{-1} - 1)k(\Omega - n)p. \quad (2.108)$$
Chapter 2. Theory of Stochastic Processes

Figure 2.4: Reversible isomerization reaction: [20, 0] denotes the state with 20 molecules of species $A$ and 0 molecules of species $B$. Shown is a comparison of the result of the $\Omega$-Expansion (top) with the solution of the master equation (bottom).

Again, we want to rewrite the discrete probability density function $p(n, t)$ as a new function $\Pi(\xi, t)$ such that

$$ n = \Omega \phi + \Omega^\gamma \xi. \quad (2.109) $$

It shall be shown that in order to get the Fokker-Planck equation (vide equation (2.71)), $\gamma$ must be $1/2$. Under this change of variables, the shift operators can be expanded in a power series:

$$ \mathbb{L}^+ = \left( 1 + r \Omega^{-\gamma} \frac{\partial}{\partial \xi} + \frac{r^2}{2!} \Omega^{-2\gamma} \frac{\partial^2}{\partial \xi^2} + \mathcal{O} \left( \Omega^{-3\gamma} \right) \right), \quad (2.110) $$

$$ \mathbb{L}^- = \left( 1 - r \Omega^{-\gamma} \frac{\partial}{\partial \xi} + \frac{r^2}{2!} \Omega^{-2\gamma} \frac{\partial^2}{\partial \xi^2} - \mathcal{O} \left( \Omega^{-3\gamma} \right) \right). \quad (2.111) $$

We obtain

$$ \frac{\partial \Pi}{\partial t} - \Omega^{1-\gamma} \frac{d\phi}{dt} \frac{\partial \Pi}{\partial \xi} = (2k\phi - k) \Omega^{1-\gamma} \frac{\partial \Pi}{\partial \xi} + 2k \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{\partial}{\partial \xi} \left( \frac{\partial^2 \Pi}{\partial \xi^2} \right) + \mathcal{O} \left( \Omega^{1-3\gamma} \right). \quad (2.112) $$

Therefore, $\gamma$ must be 1/2 in order to obtain the Fokker-Planck equation:

$$ \frac{\partial \Pi}{\partial t} = 2k \frac{\partial}{\partial \xi} (\xi \Pi) + \frac{c}{2} \frac{\partial^2 \Pi}{\partial \xi^2} + \mathcal{O} \left( \Omega^{-1/2} \right), \quad (2.113) $$

which has the following moment equations:

$$ \frac{d \langle \xi \rangle}{dt} = -2k \langle \xi \rangle, \quad (2.114) $$

$$ \frac{d \langle \xi^2 \rangle}{dt} = -4k \langle \xi^2 \rangle + k, \quad (2.115) $$
subject to the initial conditions: $\langle \xi \rangle_0 = \langle \xi^2 \rangle_0 = 0$. The steady-state solution is shown in Figure 2.4 using the state $[20, 0]$ for the initial condition.

## 2.5 Stochastic Differential Equations

### 2.5.1 Introduction

Stochastic differential equations substitute extrinsic noise for intrinsic noise in the master equation. This procedure can be likened to the $\Omega$-Expansion where the solution follows a deterministic trajectory and the fluctuations are proportional to $\Omega^{1/2}$. Indeed the two can be related via the Fokker-Planck equation [9,10,23].

### 2.5.2 Definition

Following Gard [11], let $X_t$ is a realization of a stochastic process $X_t = f(\xi, t)$.

\[
\text{d}X_t = \mu(X_t, t) \, \text{d}t + \sigma(X_t, t) \, \text{d}B_t, \tag{2.116}
\]

where $\mu(X_t, t)$ is the so-called drift term, $\sigma(X_t, t)$ is the expected standard deviation of the process, and $B_t$ is a Wiener process defined such that

1. $B_0 = 0$
2. $\mathbb{E}[B_b - B_a] = 0$
3. $B_b - B_a \sim \mathcal{N}(0, b - a)$, so-called independent increments where

\[
\mathbb{E}[|B_b - B_a|^2] = b - a. \tag{2.119}
\]

Equation (2.116) is short for

\[
X_t = X_0 + \int_0^t \mu(X_s, s) \, \text{d}s + \sigma(X_s, s) \, \text{d}B_s, \tag{2.121}
\]

where the second integral needs to be defined. The definition is not unique since there exist two different ways which yield different values. These are called the Itô and Stratonovich integrals. Let time be partitioned into uniform intervals denoted by $0 = t_0, t_1 = t/n, t_2 = 2t/n, \ldots, t_n = nt/n = t$. The Itô integral is defined as

\[
\int_0^t \sigma(X_s, s) \, \text{d}B_s \triangleq \lim_{n \to \infty} \sum_{i=1}^n \sigma(X_{t_i^*}, t_{i}^*) \Delta B_i \tag{2.122}
\]
where $\Delta B_t \triangleq B_{t_i} - B_{t_{i-1}}$ and $t_i^* = t_{i-1}$. The Stratonovich integral is

$$
\int_0^t \sigma(X_s, s) \circ dB_s \triangleq \lim_{n \to \infty} \sum_{i=1}^{n} \sigma(X_{t_i^*}, t_i^*) \Delta B_t
$$

(2.123)

where $t_i^* = (t_{i-1} + t_i)/2$. Note that the Itô integral preserves the Markov character of the stochastic process and, in general, $\int_0^t \sigma(X_s, s) \, dB_s \neq \int_0^t \sigma(X_s, s) \circ dB_s$ [11]. If $\mu = \alpha_1$ and $\sigma^2 = \alpha_2$ then $X_t$ using the Itô integral is a random variate or sample from the solution of the corresponding Fokker-Planck equation. Specifically, if $X_t^{(j)}$ denotes the $j$th realization, then $\{X_t^{(j)}\}$, as $j \to \infty$ will converge to $P(y, t)$ from equation (2.71) [10].

### 2.5.3 Itô’s Lemma

Itô’s lemma provides a way for changing the dependent variable of a stochastic process. It is useful for finding equivalent stochastic processes and can be used to solve stochastic differential equations if the transformation yields a simpler equation.

Let

$$
dx = \mu \, dt + \sigma \, dB_t
$$

(2.124)

and let $f = f(x, t) \in C^2$, then [11]

$$
df = \left( \frac{\partial f}{\partial t} + \mu \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} \right) \, dt + \sigma \frac{\partial f}{\partial x} \, dB_t.
$$

(2.125)

Following [10], Itô’s lemma can be informally proved by seeing that

$$
df = f(x + dx, t + dt) - f(x, t),
$$

(2.126)

$$
= f(x, t) + \frac{\partial f}{\partial x} \, dx + \frac{\partial f}{\partial t} \, dt + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} \, dx^2 + \ldots - f(x, t),
$$

(2.127)

$$
= \frac{\partial f}{\partial x} \, dx + \frac{\partial f}{\partial t} \, dt + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} \, dx^2,
$$

(2.128)

$$
= \frac{\partial f}{\partial x} (\mu \, dt + \sigma \, dB_t) + \frac{\partial f}{\partial t} \, dt + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} (\mu \, dt + \sigma \, dB_t)^2,
$$

(2.129)

$$
= \frac{\partial f}{\partial x} (\mu \, dt + \sigma \, dB_t) + \frac{\partial f}{\partial t} \, dt + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} (\mu^2 \, dt^2 + 2 \mu \sigma \, dt \, dB_t + \sigma^2 \, dB_t^2).
$$

(2.130)

Next, note that as $dt$ becomes small, $dt^2 \to 0$ and $dt \, dB_t \to 0$ [10], yielding

$$
df = \frac{\partial f}{\partial x} (\mu \, dt + \sigma \, dB_t) + \frac{\partial f}{\partial t} \, dt + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} \, dB_t^2.
$$

(2.131)
2.6. Chemical Reactions

We use the fact that \( dB_t^2 = dt \) \([10, 11]\), which can be informally seen by noticing that
\[
E[dB_t^2] = E[|B_{t+dt} - B_t|^2] = dt,
\]
and therefore
\[
df = \frac{\partial f}{\partial x} (\mu dt + \sigma dB_t) + \frac{\partial f}{\partial t} dt + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} dt,
\]
(2.132)
\[
= \left( \frac{\partial f}{\partial t} + \mu \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} \right) dt + \sigma \frac{\partial f}{\partial x} dB_t.
\]
(2.133)

2.6 Chemical Reactions

2.6.1 Chemical Reactions as Stochastic Phenomena

In this chapter we are concerned with formulating the entries of the \( \mathcal{W} \)-matrix for the master
equation. Historically the temporal evolution of a spatially homogeneous chemical system
was modelled using continuous, deterministic differential equations. However these
equations, called reaction-rate equations, assume that the number of molecules is very large, i.e.
\( \Omega \to \infty \) as described in Section 2.4.4. Indeed the differential equations are only valid if \( \Omega \) is
large and the variance of the process is neglected.

As will be numerically shown in the next chapter, discrete, stochastic and continuous, deter-
ministic models of chemical processes can yield vastly different results. Moreover, chemical
reactions are an inherently random process since, even after neglecting quantum physics,
the equations of classical mechanics - i.e. Newton’s equations of motion - would not allow
us to know with absolute certainty the exact position of molecules and therefore would not
be able to precisely evolve the system [13].

2.6.2 Master Equation Formulation

Chemical reactions can be written in the form
\[
r_1X_1 + r_2X_2 + \ldots \rightarrow g_1X_1 + g_2X_2 + \ldots \quad (2.134)
\]
where \( k \) is the reaction rate, \( r_k \) represents the number of \( X_k \) molecules that participate
in the reaction, and \( g_k \) the number of \( X_k \) products. Let \( \mathbf{X}(t) = (X_1(t), X_2(t), \ldots)^T \) be a
realization of the stochastic process, where \( \mathbf{X}(t) \in S \). The stoichiometric vector for reaction
(2.134) is \( \mathbf{\nu}_i = (g_1 - r_1, g_2 - r_2, \ldots)^T \) so that if the current state is \( \mathbf{X}(t) \) and reaction \( i \) occurred
within \( dt \), then \( \mathbf{X}(t+dt) = \mathbf{X}(t) + \mathbf{\nu}_i \).

The sample space \( S \) of this stochastic process can be visualized as an integer lattice \( \mathbb{Z}^d \), where
the dimension \( d \) is the number of species in the system. Usually the sample space is smaller
than the whole of $\mathbb{Z}^d$, thus it is $[0, \Omega]^d$ (see Figure 2.5). However, if there are reactions that are unbounded, e.g. $\emptyset \rightarrow X_1$, then formally the sample space is countably infinite. Although this may seem academic, the definition of the $\mathcal{W}$-matrix will need to be a subspace; and if this subspace does not encapsulate the space in which the chemical reaction resides, then the solution of the master equation will not integrate to unity.

A propensity is defined for a reaction indexed by $i$ as

$$a_i(\mathbf{X}(t)) \triangleq k\Omega \prod_{j=1}^{N} \left\{ \frac{\mathbf{X}_j(t) \left( \mathbf{X}_j(t) - 1 \right) \cdots \left( \mathbf{X}_j(t) - r_j + 1 \right)}{\Omega^{r_j}} \right\},$$

where $a_i(\mathbf{X}(t)) \, dt$ is the probability of a reaction. Intuitively, the product appears in (2.135) since the collisions of molecules are assumed to be independent, and the factor of $\Omega^{-r_j}$ is needed so that $a_i(\mathbf{X}(t))$ has units of $[\text{molecules}/\text{time}]$.

**Reversible Isomerization Example** The $\mathcal{W}$-matrix for the reaction set defined in equations (2.105)-(2.106), where $X_1 + X_2 = \Omega$ is

$$\mathcal{W} = \begin{pmatrix}
-k\Omega & k & 0 & \ldots & 0 \\
k\Omega & -k\Omega & \ddots & 0 & : \\
0 & k(\Omega - 1) & \ddots & k(\Omega - 1) & 0 \\
: & 0 & \ddots & -k\Omega & k\Omega \\
0 & \ldots & 0 & k & -k\Omega
\end{pmatrix},$$

(2.136)
which has the properties defined in (2.63) and (2.64). The time evolution of the probability density function is shown in Figure 2.6 after symbolically exponentiating the $\mathcal{W}$-matrix according to equation (2.62). The corresponding reaction rate equation for this system was given in equation (2.107).

Unfortunately the size of the $\mathcal{W}$-matrix is prohibitively large and in practice can only be of any practical use in very small systems. Therefore, although formally we are finished, practically we have not yet begun. The rest of the thesis will deal with numerical methods for practical problems arising from chemistry and biology.
Part II

Methods
Chapter 3

Monte Carlo Methods

3.1 Introduction

According to Dimov [6], there are two principle directions in the research and development of Monte Carlo algorithms.

The first direction is the so-called Monte Carlo Simulation, in which the algorithms follow the corresponding physical, chemical, or biological process under consideration. In these simulations, the result is a trajectory or realization of a stochastic process and is one possible outcome of the many different possible outcomes of the stochastic process under consideration. These methods are used for solving probabilistic problems by means of random variables.

Monte Carlo Numerical Algorithms represent the second principle direction of Monte Carlo algorithms. These numerical algorithms are used for solving deterministic problems by means of random variables. The fundamental idea is to construct an artificial stochastic process and show that the some property of it, usually the expectation, coincides with the solution of the deterministic problem.

The two principle directions need not however be mutually exclusive. The master equation presented in Section 2.4, which is completely deterministic, is the result of a probabilistic model, i.e. a Markov process. The straightforward way of solving the master equation, discussed in Section 2.4.2, is typically inefficient and/or infeasible. As will be shown below, an efficient way of solving the equation is to produce a realization of the master equation that represents a sample from the solution of the master equation - the solution being a discrete probability density, videlicet equation (2.62).
3.2 Convergence of $O(N^{-1/2})$

Let $\xi$ be a random variable for which the expectation exists, namely $E[\xi] = J$. Let $\{\xi_i\}$ be a sequence of independent and identically distributed random variables, which will be used to approximate the solution $J$. Also let the sample mean be defined as $\bar{\xi}_N = N^{-1} \sum_{i}^{N} \xi_i$, where $N$ is the number of samples. The convergence rate of Monte Carlo methods is:

$$
\lim_{N \to \infty} P\left\{ |\bar{\xi}_N - J| < \alpha \cdot \beta(\xi) \cdot N^{\gamma} \right\} = \Phi(\alpha),
$$

where

$$
\alpha = x_{1/2} \approx 0.67449, \quad \beta(\xi) = \sqrt{V[\xi]}, \quad \Phi(\alpha) = \frac{2}{\sqrt{2\pi}} \int_0^\infty e^{-t^2/2} dt, \quad \gamma = \frac{1}{2}.
$$

This can be shown by using the Central Limit Theorem [6]:

$$
\lim_{N \to \infty} P\left\{ x_1 < \frac{\sum_{i=1}^{N} (\xi_i - J)}{\sqrt{NV[\xi]}} < x_2 \right\} = \frac{1}{\sqrt{2\pi}} \int_{x_1}^{x_2} e^{-t^2/2} dt,
$$

which can be manipulated:

$$
\lim_{N \to \infty} P\left\{ \left| \frac{\sum_{i=1}^{N} (\xi_i - J)}{\sqrt{N\sqrt{\xi}}} \right| < x \right\} = \Phi(x),
$$

$$
\lim_{N \to \infty} P\left\{ \sum_{i=1}^{N} (\xi_i - J) < x \sqrt{N\sqrt{\xi}} \right\} = \Phi(x),
$$

$$
\lim_{N \to \infty} P\left\{ N^{-1} \sum_{i=1}^{N} (\xi_i - J) < N^{-1} x \sqrt{N\sqrt{\xi}} \right\} = \Phi(x),
$$

$$
\lim_{N \to \infty} P\left\{ N^{-1} \sum_{i=1}^{N} (\xi_i - \sum_{i=1}^{N} J) < N^{-1} x \sqrt{\sqrt{\xi}N^{-1/2}} \right\} = \Phi(x),
$$

$$
\lim_{N \to \infty} P\left\{ |\bar{\xi}_N - J| < x \sqrt{\sqrt{\xi}N^{-1/2}} \right\} = \Phi(x).
$$

We want the probability to be equal to 1/2, therefore,

$$
\lim_{N \to \infty} P\left\{ |\bar{\xi}_N - J| < x \sqrt{\sqrt{\xi}N^{-1/2}} \right\} = \Phi(x) = \frac{1}{2},
$$

$$
\lim_{N \to \infty} P\left\{ |\bar{\xi}_N - J| < x_{1/2} \sqrt{\sqrt{\xi}N^{-1/2}} \right\} = \Phi(x_{1/2}),
$$

$$
\lim_{N \to \infty} P\left\{ |\bar{\xi}_N - J| < x_{1/2} \sqrt{\sqrt{\xi}N^{-1/2}} \right\} = \Phi(x_{1/2}) \approx 0.67449 = \frac{1}{2}.
$$
3.3 Numerical Integration

The result states that, apart from $N^{-1/2}$ rate of convergence, the variance of the stochastic process must exist, i.e. be finite. A random variable for which the variance exists is said to be an $L_2$ random variable.

3.3 Numerical Integration

![Figure 3.1: Convergence of Monte Carlo methods](image)

(a) $f_1(x), \gamma \approx -0.49$

(b) $f_2(x), \gamma \approx -0.37$

Let \{\eta_i\} be a sequence of independent and identically distributed random variables that are uniformly distributed over the interval [0, 1], and let \{\xi_i = f_k(\eta_i)\} where the functions $f_k$ are defined as $f_k : [0, 1] \to \mathbb{R}$, for $k = 1, 2$:

\[
\begin{align*}
    f_1(x) &= x^{-2/5}, \\
    f_2(x) &= x^{-2/3}.
\end{align*}
\]  

(3.15)  

(3.16)

Monte Carlo simulations were used to calculate the expected value of the variable $\xi$, namely $\mathbb{E}[\xi] = J$. The number of samples, $N$, was varied from $N = 10$ to $N = 10^5$, using $\Delta N = 10$, and algorithm was run $L = 10^3$ times. The error was defined as $e_{N,l} \triangleq |J - \bar{\xi}_{N,l}|$, and the
mean error was plotted, namely \( e_N \triangleq L^{-1} \sum_{l=1}^{L} e_{N,l} \), versus \( N \) in log-log scale. in Figure 3.1. Equation (3.6) does indeed hold for \( \xi = f_1(\eta) \), yet does not hold for \( \xi = f_2(\eta) \). The analytical mean and variance for the equation \( f_1 \) are

\[
E[\xi_{f_1}] = \int_0^1 x^{-2/3} \, dx = \frac{5}{3}, \tag{3.17}
\]

\[
\text{Var}[\xi_{f_1}] = \int_0^1 (x^{-2/5} - E[\xi_{f_1}])^2 \, dx = \frac{20}{9}, \tag{3.18}
\]

whereas for function \( f_2 \), the variance does not exist:

\[
E[\xi_{f_2}] = \int_0^1 x^{-2/3} \, dx = 3, \tag{3.19}
\]

\[
\text{Var}[\xi_{f_2}] = \int_0^1 (x^{-2/3} - E[\xi_{f_2}])^2 \, dx = -\frac{3}{x^{1/3}} - 18x^{1/3} + 9x + C, \tag{3.20}
\]

and since the variance does not exist, nothing can be said about the rate of convergence.

### 3.4 Random Walk

In Section 2.3.1, it was shown that the random motion of particles suspended in a fluid is equivalent to the diffusion equation:

\[
\frac{\partial \omega}{\partial t} = \nu \Delta \omega, \quad x \in \mathbb{R}, \tag{3.21}
\]

where \( \omega = \omega(x, t) \) denotes the concentration of particles in a volume element denoted by \( x \). The solution of this equation is [24]

\[
\omega(x, t) = \frac{1}{(1 + 4\nu t)^{3/2}} x e^{-x^2/(1+4\nu t)}. \tag{3.22}
\]

The random walk process is defined as

\[
r_i(t + \Delta t) = r_i(t) + \sqrt{2\nu \Delta t} \xi_i, \tag{3.23}
\]

where \( r_i(t) \) is the position of the particle \( i \) at time \( t \), and \( \xi_i \) is a normally distributed random variable. After two time-steps, the method is

\[
r_i(t + \Delta t) = r_i(t) + \sqrt{2\nu \Delta t} \xi_i, \tag{3.24}
\]

\[
r_i(t + 2\Delta t) = \left( r_i(t) + \sqrt{2\nu \Delta t} \xi_i^{(1)} \right) + \sqrt{2\nu \Delta t} \xi_i^{(2)}, \tag{3.25}
\]

\[
= r_i(t) + K_1 + K_2, \tag{3.26}
\]
3.5. Poisson’s Equation

Figure 3.2: Comparison of the Random Walk method with the Diffusion Equation.

where $K_1 \sim \mathcal{N}(0, 2\nu \Delta t)$ and $K_2 \sim \mathcal{N}(0, 2\nu \Delta t)$, and using the property of additivity of independent and identically distributed normal random variables results in

$$r_i(t + 2\Delta t) = r_i(t) + \sqrt{2\nu(2\Delta t)}\xi_i.$$  \hspace{1cm} (3.27)

Therefore

$$r_i(t + n\Delta t) = r_i(t) + \sqrt{2\nu(n\Delta t)}\xi_i,$$  \hspace{1cm} (3.28)

$$r_i(T) = r_i(0) + \sqrt{2\nu T}\xi_i.$$  \hspace{1cm} (3.29)

In other words there is no restriction on the time-step so long as there is no boundary. Equation (3.22) was used to initialize $N$ randomly placed particles. The positions of the particles were then displaced according to (3.29) and the function value was computed by binning the particles and averaging the particle weights within each bin. Figure 3.2 shows the function values and the rate of convergence with respect to $N$.

3.5 Poisson’s Equation

In Sections 2.3.1 and 3.4 the connection between the diffusion operator and random processes were shown. This is because there is an inherent relation between the Laplace operator and random walks. In this section we apply random walks to another problem involving the Laplace operator.
3.5.1 Periodic Boundaries

We consider the Poisson boundary value problem for a function \( u = u(x, t) \):

\[
\begin{align*}
- \Delta u & = f, \quad x \in \mathcal{D}, \quad (3.30) \\
u & = \phi, \quad x \in \partial \mathcal{D}. \quad (3.31)
\end{align*}
\]

For a regular uniform discretization of the Poisson’s equation, namely

\[
\Delta_h u = -f_h, \quad (3.32)
\]

one can solve for the \( i^{th} \) point, \( i \triangleq (i_1, i_2) \)

\[
u_i = \frac{1}{4} \Lambda + \frac{h^2}{4} f_i, \quad (3.33)
\]

Intuitively, \( \Lambda \) represents a random walk since it is the average of the nearest neighbors. Defining the transition probability \( p_{j,i} \) as the probability of transitioning from node \( i \) to node \( j \), \( j \) being a neighboring node, the mathematical expectation of the following random variable coincides with the solution of the problem [25-27]:

\[
\begin{align*}
\xi & = \frac{h^2}{4} \sum_{i=1}^{i^* - 1} f_i + \phi_{i^*}, \quad (3.34) \\
p_{j,i} & = \frac{1}{4}, \quad \text{if } i \in \Omega, \quad (3.35) \\
p_{j,i} & = 0, \quad \text{if } i \in \partial \Omega, \quad (3.36)
\end{align*}
\]

where the sum is over all of the indices of the Markov chain (i.e. the sum of the path taken by the random walk) and \( i^* \) is the point where the chain reaches the boundary. This method for solving Poisson’s equation can be useful if the solution is desired at only a few points in the domain, since each point is solved for independently.

Monte Carlo simulations were performed using a periodic grid, for each of the two following functions:

\[
\begin{align*}
f_1(x, y) & = \sin(2\pi x) \sin(2\pi y), \quad (3.37) \\
f_2(x, y) & = \frac{1}{2\pi C^2} \exp\left\{ -\frac{(x - 1/2)^2 + (y - 1/2)^2}{2C^2} \right\}, \quad C = 10^{-2}. \quad (3.38)
\end{align*}
\]

where \( \Omega = [0, 1] \times [0, 1] \), and \( N = 10^3 \), the results of which are shown in Figure 3.3.
3.5. Poisson’s Equation

![Surface plots of Poisson’s equation, deterministic (left) and stochastic (right) for the functions $f_1$ and $f_2$. $x$- and $y$-axes show the index of the grid node.](image)

3.5.2 Broken Disk Problem

We consider the following broken disk problem [26]:

$$
- \Delta u = (2 - r^2)e^{-r^2/2}, \quad \text{in } D \\
\quad u(r, 0) = e^{-r^2/2} \\
\quad u(r, -3\pi/2) = -r^{1/3} + e^{-r^2/2} \\
\quad u(1, \theta) = \sin \left( \frac{\theta}{3} \right) + e^{-1/2}
$$

where $D$ is the unit disk minus the first quadrant. The analytical solution is [26]:

...
\begin{equation}
  u(r, \theta) = r^{1/3} \sin \left( \frac{\theta}{3} \right) + e^{-r^2/2},
\end{equation}

where $r \in [0, 1]$ and $\theta \in [0, -3\pi/2]$. The probabilities for the Monte Carlo method are modified near the boundary. Let $d$ denote the distance to the interface where $i$ is the index of a point inside $\mathcal{D}$ and is closest to the interface. Define the unscaled probability [28]

\begin{align}
  \hat{p}_{j,i} &= \frac{1}{hd}, \quad i \in \mathcal{D}, \quad j \notin \mathcal{D}, \\
  \hat{p}_{k,i} &= \frac{1}{h^2}, \quad i \in \mathcal{D}, \quad k \in \mathcal{D},
\end{align}
where $h$ is the mesh spacing. The probabilities of transitions are

$$ p_{j,i} = \frac{\hat{p}_{j,i}}{\sum_{l \in N(i)} \hat{p}_{l,i}}, \quad (3.46) $$

where $N(i)$ denotes all of the neighbors of $i$. Figure 3.4 shows the solution contours and Figure 3.5 shows the errors in space with respect to the analytical solution.

## 3.6 Stochastic Simulation Algorithm

### 3.6.1 Gillespie Algorithm

The Stochastic Simulation Algorithm (SSA) is a Monte Carlo method for the simulation of discrete-space, continuous-time Markov processes. Let the discrete-state be denoted by $s \in \mathbb{N}^N$, where $N$ is the number of species in the system. Additionally, let the complete enumeration of the state space be denoted by $S = \{s_1, s_2, \ldots\}$. SSAs deal with a realization of the time-dependent stochastic processes, namely a trajectory $X(t) \in S$. The process is simulated over time by the following update scheme:

$$ X(t + \tau) = X(t) + \nu_k, \quad (3.47) $$

where $\nu_k$ is the stoichiometric vector that denotes the change induced by reaction $k$. The random variable $X(t_f)$ is ultimately a sample from the unknown discrete probability density function at time $t_f$. The convergence rate of the numerical method is the well-known rate of Monte Carlo methods as shown in Section 3.2. In the classical formulation of the Stochastic Simulation Algorithm (SSA) [12, 13], the probability of a reaction $k$ with a time-step $\tau$ is chosen from the joint probability density function

$$ p(\tau, k) = a_k(t)e^{-a_0(t)\tau}, \quad (3.48) $$

where the propensities are defined by equation (2.13) and

$$ a_0(t) \triangleq \sum_k a_k(X(t)). \quad (3.49) $$

Equation (3.48) can be decomposed as $p(\tau, k) = p(\tau)p(k)$, where

$$ p(\tau) = a_0(t)e^{-a_0(t)\tau}, \quad (3.50) $$

$$ p(k) = \frac{a_k(t)}{a_0(t)}, \quad (3.51) $$
which amount to calculating the time-step in which a reaction occurred and finding the index of the reaction that occurred within said time-step. The inverse transform sampling method [29] is used to sample $\tau$ and $k$ from equations (3.50) and (3.51). For instance,

$$r_1 = \int_0^\tau p(\tau') \, d\tau',$$

where $r_1$ is sampled from a uniform distribution in the range $[0, 1)$. Solving for $\tau$ yields

$$\tau = -\frac{1}{a_0(t)} \ln(r_1).$$

The value for $j$ is the integer for which

$$\sum_{\beta=1}^{j-1} a_\beta(t) < r_2 a_0(t) \leq \sum_{\beta=1}^j a_\beta(t).$$

where $r_2$ is another sample from a uniform distribution in the range $[0, 1)$.

The algorithm is: 0. Initialize the time $t = t_0$ and the system’s state $X = X_0$.

1. With the system in state $X$ at time $t$, evaluate all of the propensities $a_j(t)$ (equation (2.135)) and their sum $a_0(t)$ (equation (3.49)).

2. Generate values for $\tau$ and $j$ where $\tau$ is an exponential random variable with parameter $a_0(t)$ (equation (3.53)) and $j$ is a discrete random variable with $P(j = k) = \frac{a_k(t)}{a_0(t)}$ (equation (3.54)).

3. Execute the next reaction by replacing $t \rightarrow t + \tau$ and $X \rightarrow X + \nu_j$ where $\nu_j$ is the stoichiometric vector that denotes the change induced by reaction $j$.

4. Record $(X, t)$ as desired. Return to step 1, or else end the simulation.

### 3.6.2 Convergence

In this section, we will analyze the reversible isomerization reactions given equations (2.105)-(2.106). As shown in Figure 3.6, as the number of samples increases, the solution obtained from the Stochastic Simulation Algorithm coincides with the solution obtained by explicitly solving the master equation.
3.6. Stochastic Simulation Algorithm

Figure 3.6: Convergence of the Stochastic Simulation Algorithm with respect to $N$, the number of samples. The chemical example is the reversible isomerization (vide equations (2.105)-(2.106)).

3.6.3 Discrepancy Between Continuum & Discrete Simulations

Given the following chemical reactions$^\dagger$:

\[
\begin{align*}
S_1 & \xrightarrow{\mathcal{C}_1} \emptyset \\
S_2 & \xrightarrow{\mathcal{C}_2} \emptyset \\
S_1 + S_2 & \xrightarrow{\mathcal{C}_3} 2S_1 \\
\emptyset & \xrightarrow{\mathcal{C}_4} S_2
\end{align*}
\]

$^\dagger$This example is from lecture notes from a course given by Prof. Dr. Tobias Jahnke, Karlsruhe Institute of Technology.
The following ordinary differential equations for the concentrations of \( S_1 \) and \( S_2 \) can be written:

\[
\begin{align*}
\frac{ds_1}{dt} &= -k_1 s_1 + k_3 s_1 s_2, \\
\frac{ds_2}{dt} &= -k_2 s_2 - k_3 s_1 s_2 + k_4,
\end{align*}
\]

where \( s_1(t = 0) = 4 \) and \( s_2(t = 0) = 0 \), \( t \in [0, 5] \) using \((k_1, k_2, k_3, k_4) = (1, 1, 1/5, 20)\). The system of equations can be written in matrix/vector form:

\[
\dot{s} = A \cdot s.
\]

We look for steady state solutions, namely

\[
\dot{s} = 0,
\]

therefore

\[
\dot{s} = f(s) = \begin{pmatrix} -k_1 s_1 + k_3 s_1 s_2 \\ -k_2 s_2 - k_3 s_1 s_2 + k_4 \end{pmatrix} = 0.
\]

There are two steady state solutions,

\[
s_1^* = \begin{pmatrix} 0 \\ \frac{k_4}{k_2} \end{pmatrix},
\]

and

\[
s_2^* = \begin{pmatrix} \frac{k_1}{k_4} - k_2 \\ \frac{k_1}{k_3} \end{pmatrix}.
\]

Performing an expansion around the steady states yields

\[
\begin{align*}
\dot{s} &= f(s) = f(s^*) + \mathcal{J}f(s^*)(s - s^*), \\
\dot{s} &= \mathcal{J}f(s^*)(s - s^*),
\end{align*}
\]

where the Jacobian is

\[
\mathcal{J}f(s) = \begin{pmatrix} -k_1 + k_3 s_2 & k_3 s_1 \\ -k_3 s_1 & -k_2 - k_3 s_1 \end{pmatrix}.
\]

Using the parameter values above, the eigenvalues for the two steady states are

\[
\det(\mathcal{J}f(s^*) - \lambda I) = 0 \implies \lambda = \{-1, 3\},
\]
3.6. *Stochastic Simulation Algorithm*

which is unstable, and

$$\text{det } (\mathcal{J}f(s^*_2) - \lambda \mathbf{I}) = 0 \implies \lambda = \{-3, -1\}, \tag{3.70}$$

which is stable. Therefore, practically speaking, the solution of the ODEs always tends to the point $s^*_2$. See Figure 3.7(a).

The rate constants for the stochastic simulation are defined as

\begin{align*}
c_1 &= k_1, \tag{3.71} \\
c_2 &= k_2, \tag{3.72} \\
c_3 &= \frac{k_3}{\Omega}, \tag{3.73} \\
c_4 &= \Omega k_4, \tag{3.74} \\
\end{align*}

and the initial number of particles is $(S_1, S_2) = (4\Omega, 0)$ and the concentrations of the stochastic simulation are defined as $s_i = S_i/\Omega$, where $\Omega$ is a scaling parameter that dictates the number of particles in the system. Stochastic simulations were performed which unexpectedly yielded different results. See Figures 3.7(b) and 3.8.

![Diagram](image)

(a) The two steady state solutions of the differential equations. (b) Concentrations over time. Black, differential equations. Blue ($S_1$) and red ($S_2$), stochastic simulations for different values of $\Omega$.

**Figure 3.7:** Shown in 3.7(a) are the stream lines of the differential equations and the two steady-state solutions. In 3.7(b) convergence with respect to $\Omega$ is clear, however, the discrepancy when $\Omega$ is small is *not* due to an inadequate number of samples.
Figure 3.8: When $\Omega = 10$, the probability density is Gaussian-like and centered around the solution of the ordinary differential equations, whereas when $\Omega = 1$ results in a nontrivial bimodal distribution with peaks around the stable and unstable states of the differential equations.

## 3.7 Approximate Stochastic Simulation Algorithms

Apart from the error of $O(N^{-1/2})$, the stochastic simulation algorithm is an exact procedure for generating random variates from the corresponding master equation. However, the exactitude of the simulation comes at the expense of computational efficiency because every reaction is explicitly simulated.

The objective of *leaping* algorithms is to advance the state of the simulation, either by a time-step $\tau$ - where $\tau \gg E[1/a_0]$ (cf. (3.50)) - or by $L$ reactions - where $L \gg 1$ (cf. (3.51)). The former method is the so-called $\tau$-Leaping algorithm, whilst the latter is the $R$-Leaping.

### 3.7.1 $\tau$-Leaping Algorithm

The definition of the propensities (2.135) have units of $[\text{molecules}] / [\text{time}]$, i.e. they are rates. Moreover, the rates time-dependent and if the propensities are known beforehand, then the simulation would simply be a Poisson process. The propensities, however, are not known *a priori* which is why SSA simulates each reaction: after each reaction, all of the propensities could have changed. Indeed the waiting time between Poisson events is an exponential, meaning that we need to update the propensities since a Poisson event has occurred. If however, we assume that the propensities have not changed significantly, the we get the
so-called \( \tau \)-Leaping method [14]:

\[
X(t + \tau) = X(t) + \sum_{j=1}^{M} k_j^{(\tau)} \nu_j,
\]

where \( \kappa \triangleq X(t) + \sum_{j=1}^{M} k_j^{(\tau)} \nu_j \) and

\[
k_j^{(\tau)} \sim \mathcal{P}(a_j \tau),
\]

\( \mathcal{P}(\lambda) \) being the Poisson distribution with parameter \( \lambda \). The parameter \( \tau \) is chosen such that the propensities, \( a_j \), do not change significantly during a time-step. Specifically,

\[
|a_j(x + \kappa) - a_j(x)| \approx \epsilon a_0,
\]

where the term on the right denotes the fraction \( (0 < \epsilon \ll 1) \) of the total propensity. A number of control mechanisms for \( \tau \) have been proposed in [30]. Note that if \( \sum_{j=1}^{M} k_j^{(\tau)} = 1 \), then the method is exact and equivalent to SSA.

Consider a decay problem, namely \( X_1 \to 0 \). Let \( \lambda_j(X, t, \tau) \) denote the value of the \( a_j \). This value dictates how accurate the simulation will be. The exact, \( \lambda_j(X, t, \tau)^{(e)} \), and the approximate, \( \lambda_j(X, t, \tau)^{(\tau)} \), values are

\[
\lambda_j(X, t, \tau)^{(e)} = \int_{t}^{t+\tau} a_j(X(t')) dt',
\]

\[
\lambda_j(X, t, \tau)^{(\tau)} = a_j(X(t)) \tau + \mathcal{O}(\epsilon),
\]

which, if the number of molecules is decaying, looks like Figure 3.9(a). The moments of the Poisson distribution depend on this value; and in fact, the first three moments are all equal to \( \lambda_j(X, t, \tau) \). Therefore, if an error is made, it is made across all of the moments of the Poisson distribution. This is important to note because the first and second moments in stochastic differential equations are correct as \( \Delta t \to 0 \), whereas in the approximate leaping algorithms, \( \tau \to 0 \) converges to SSA or if \( \tau \gg \mathbb{E}[p(\tau)] = 1/a_0 \) then an error is made across all of the moments.

When the number of molecules in the system becomes large, the Poisson distribution approaches the Gaussian distribution [31], viz.

\[
\mathcal{P}(a_j \tau) \approx \mathcal{N}(a_j \tau, a_j \tau),
\]

\[
= a_j \tau + \sqrt{a_j \tau} \mathcal{N}(0, 1),
\]

where \( \mathcal{N}(\mu, \sigma^2) \) denotes the Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \). The update scheme becomes

\[
X(t + \tau) = X(t) + \sum_{j=1}^{M} \nu_j a_j \tau + \sum_{j=1}^{M} \nu_j a_j \tau \xi_j
\]

(3.82)
Figure 3.9: Leaping Algorithms. For 3.9(a), light blue is the integral computed by using leaping algorithms, dark blue is the exact integral. The shift in the distributions in 3.9(b) signifies the incorrect computation of the first moment.

where \( \xi_j \sim \mathcal{N}(0, 1) \). Following Gillespie [31], let \( \tilde{a}_j(Z) \triangleq \Omega^{-1}a_j(X) \), where \( Z \triangleq \Omega^{-1}X \). This change of variables yields

\[
Z(t + \tau) = Z(t) + \sum_{j=1}^{M} \nu_j \tilde{a}_j \tau + \mathcal{O}(\tau^{1/2}),
\]  

which - after neglecting \( \mathcal{O}(\tau^{1/2}) \) terms - is the explicit Euler method applied to the deterministic reaction rate equations.

### 3.7.2 R-Leaping Algorithm

In R-Leaping [15], the number of reactions that are executed per iteration, \( L \), is prescribed instead of the time-step. These executions may occur across all the reaction channels instead of one channel as proposed in the \( k_{\alpha} \)-Leaping [14].

A straightforward way to accelerate the stochastic simulation algorithm consists of performing \( L \) iterations without updating the propensities \( \{a_j\}_{1 \leq j \leq M} \). The time-step \( \tau_L \) of these \( L \) reactions follows a Gamma distribution with parameters \((L, 1/a_0)\), viz.:

\[
\tau_L \sim \mathcal{G}(L, 1/a_0). 
\]  

The indices of the \( L \) reactions are determined by the random variables \( \{j_1, \ldots, j_L\} \), each of which obeys the same point-wise distribution of SSA (vide (3.51)). The number of executions, \( k^{(R)}_m \), for reaction channel \( R_m \) is:

\[
k^{(R)}_m = \sum_{k=1}^{L} \mathbf{1}_{\{j_k = m\}} \text{ for } m = 1, \ldots, M
\]  

(a) The error induced in leaping algorithms     
(b) R-Leaping (blue) vs \( \tau \)-Leaping (orange) vs SSA (black)
3.7. Approximate Stochastic Simulation Algorithms

where \( i_{\{ j_k = m \}} \) is the indicator function:

\[
i_{\{ j_k = m \}} = \begin{cases} 
1 & \text{if } j_k = m \\
0 & \text{otherwise}.
\end{cases}
\]  

(3.86)

The update scheme is

\[
X(t + \tau_L) = X(t) + \sum_{j=1}^{M} k_j^{(R)} \nu_j.
\]  

(3.87)

Note that (3.87) is identical to (3.75) save for the \( k_j^{(\tau)} \)'s and \( k_j^{(R)} \)'s which follow different distributions. The benefit of using this sampling procedure is that propensities are not computed as frequently as in SSA.

Assume that the indices \( \{ j_1, \ldots, j_L \} \) are drawn from point-wise distributions and let \( m \in [1, M] \). For each index \( j_k \), there are two possible outcomes: either \( j_k = m \) (with probability \( a_m/a_0 \)), or \( j_k \neq m \) (with probability \( 1 - a_m/a_0 \)). Each event is therefore a Bernoulli random variable with probability \( a_m/a_0 \), where \( k_m^{(R)} \) is the number of successes in \( L \) trials. The distribution of \( k_1^{(R)} \) is

\[
k_1^{(R)} \sim \mathcal{B} \left( L, \frac{a_m}{a_0} \right).
\]  

(3.88)

However, \( k_2^{(R)} \) is dependent on the outcome of \( k_1^{(R)} \). In \textit{R-Leaping}, the sampling procedure is given by the \textit{R-Leaping Theorem}:

\textbf{R-Leaping Theorem.} The distribution of \( k_1^{(R)} \) is \( \mathcal{B}(L, a_1/a_0) \) and for every \( m \in \{2, \ldots, M\} \), the distribution of \( k_m^{(R)} \) is conditioned on the set of events \( \{k_1^{(R)}, \ldots, k_{m-1}^{(R)}\} \) is

\[
\mathcal{B} \left( L - \sum_{i=1}^{m-1} k_i, \frac{a_m}{a_0 - \sum_{i=1}^{m-1} a_i} \right)
\]  

(3.89)

The result is invariant under any permutation of the indices.

This invariance can be exploited by sorting the propensities during the simulation. By sorting the propensities in descending order, reactions with large propensities can be dealt with first, the result of which may be an early exit from the sampling procedure. Specifically, if \( L - \sum_{i=1}^{j'} k_i^{(R)} = 0 \), then the rest of the values \( \{k_{j'+1}^{(R)}, \ldots, k_{m-1}^{(R)}\} \) need not be sampled.

Using this theorem, the following algorithm is proposed:

\textbf{Algorithm 1.} Sample the \( K_m \) variables

\textbf{Require:} \( L \geq 0, a_m \geq 0 \)

\textbf{Ensure:} \( \sum_{m=1}^{M} k_m^{(R)} = L \)
for $m = 0$ to $M$ do
  $k_m^{(R)} \leftarrow 0$
end for

$S \leftarrow 0$
$A \leftarrow 0$
$m \leftarrow 1$

while $S < L$ do
  $k_m^{(R)} \leftarrow \text{Sample} \left( B \left( L - S, \frac{a_m}{a_0 - A} \right) \right)$
  $S \leftarrow S + k_m^{(R)}$
  $A \leftarrow A + a_m$
  $m \leftarrow m + 1$
end while

In Figure 3.9(b), an example of the following decaying-dimerization [14] is shown:

\begin{align}
X_1 & \xrightarrow{c_1} \emptyset \quad (3.90) \\
2X_1 & \xrightarrow{c_2} X_2, \quad (3.91) \\
X_2 & \xrightarrow{c_3} 2X_1, \quad (3.92) \\
X_2 & \xrightarrow{c_4} X_3. \quad (3.93)
\end{align}

where $(c_1, c_2, c_3, c_4) = (1, 0.002, 0.5, 0.04)$. The initial conditions $X_1(0) = 4, 150$, $X_2(0) = 39, 565$, $X_3(0) = 3, 445$ were imposed. Note that a shift in the distributions of the leaping algorithms indicates that the computation of the first moment is incorrect.

### 3.7.3 Connection between $\tau -$ and $R-$Leaping

The procedures of executing more than one reaction per time-step and taking a larger time-step (larger with respect to the classical stochastic simulation algorithm) are related. Note that if $\tau$ is fixed according to the $\tau$-Leaping algorithm, the number of reactions within that time-step follows

\begin{equation}
L \sim \mathcal{P} \left( a_0 \tau \right). \quad (3.94)
\end{equation}

In other words, the $R$-Leaping algorithm can be performed by fixing the time-step as well, and then sampling the number of reactions that will occur within said time-step. Equivalently, the number of reactions can be fixed as in the $R$-Leaping algorithm, and the time-step follows

\begin{equation}
\tau \sim \mathcal{G} \left( L, 1/a_0 \right). \quad (3.95)
\end{equation}

In summary, either the time-step or the number of reactions can be fixed per iteration; this choice need not dictate the algorithm that is ultimately used, as the two can be interchanged.
3.8 Euler-Maruyama Method

Given a stochastic differential equation, an efficient numerical integration method is the so-called Euler-Maruyama method, which is essentially an extension of the explicit Euler method. The method is useful because of its simplicity and - more importantly - because unlike other integration methods, namely the Milstein method, analytical derivatives of $\sigma(X_t, t)$ are unnecessary.

The Euler-Maruyama numerical method is a direct discretization of the Itô integral (2.122):

$$X_{t+\Delta t}^{(i)} = X_t^{(i)} + \mu \left( X_t^{(i)}, t \right) \Delta t + \sigma \left( X_t^{(i)}, t \right) \sqrt{\Delta t} \xi,$$

(3.96)

where $X^{(i)}$ denotes the $i^{th}$ sample, $\Delta t$ is the time-step, and $\xi$ is a random variate from a standard normal distribution. The Euler-Maruyama method is known to have a weak error of $O(\Delta t)$ and a strong error of $O(\Delta t^{1/2})$ [11], where the weak error (error of the mean) is defined as:

$$E^{(W)} \triangleq \left| E \left[ X_T^{(exact)} \right] - \frac{1}{N} \sum_{i=1}^{N} X_T^{(i)} \right|,$$

(3.97)

and the strong error (mean of the error) is:

$$E^{(S)} \triangleq \frac{1}{N} \sum_{i=1}^{N} \left| X_T^{(i)} - X_T^{(exact)} \right|.$$

(3.98)

Consider the Geometric Brownian Motion problem

![Graphs](image)

(a) Weak error $\gamma \approx -1$

(b) Strong error $\gamma \approx -\frac{1}{2}$

Figure 3.10: Weak and strong errors: the slope is $\gamma$, simulation (red), and theoretical (black)

$$dX_t = rX_t \, dt + \alpha X_t \, dB_t,$$

(3.99)
Figure 3.11: The probability density function 3.11(a) produced from equation (3.104). In 3.11(b) |e| denotes the difference of the probability density function compared to the solution of the explicit solution of the Master Equation (see equations (2.62) and (2.136)). Langevin (red) and Ω-Expansion (blue) are shown. Cf. Figure 2.4.

where \( r \) and \( \alpha \) are constants. We can use Ito's lemma (2.125) to find the analytical solution by setting \( f(X_t, t) = \ln(X_t) \), in which case we get

\[
d(\ln(X_t)) = \left( 0 + \mu \frac{1}{X_t} - \frac{\sigma^2}{2} \frac{1}{X_t^2} \right) dt + \sigma \frac{1}{X_t} dB_t.
\]  

(3.100)

Using \( \mu = rX_t, \sigma = \alpha X_t \), and simplifying yields the solution

\[
d(\ln(X_t)) = \left( r + \frac{\alpha^2}{2} \right) dt + \alpha dB_t,
\]  

(3.101)

\[
\ln \left( \frac{X_t}{X_0} \right) = \left( r + \frac{\alpha^2}{2} \right) dt + \alpha dB_t.
\]  

(3.102)

Therefore at time \( t \) the exact solution is

\[
X_t^{(\text{exact})} = X_0 \exp \left\{ \left( r - \frac{1}{2} \alpha^2 \right) t + \alpha B_t \right\},
\]  

(3.103)

where \( B_t \) is a Brownian motion sample. Convergence of the Euler-Maruyama method for the geometric Brownian motion problem defined by equation (3.99) is shown in Figure 3.10.

It has been shown that the result of the Ω-Expansion (Section 2.4.4) is a Fokker-Planck equation (2.71), and in turn, a stochastic differential equation can be constructed such that it is a random variate from the Fokker-Planck equation. Moreover, it has been shown in [23] that
a stochastic differential equation called the \textit{Langevin} equation can indeed be constructed such that it numerically produces the analytical approximation of the $\Omega$-Expansion:

\begin{equation}
X^{(i)}_{t+\Delta t} = X^{(i)}_t + \Delta t \sum_{j=1}^{M} \nu_j a_j \left( X^{(i)}_t \right) + \sqrt{\Delta t} \sum_{j=1}^{M} \nu_j \sqrt{a_j} \left( X^{(i)}_t \right) \xi_j,
\end{equation}

which is simply equation (3.96) with $\mu$ and $\sigma$ equal to the propensities and square-root of the propensities, respectively. Note that this is because the mean and variance of the Poisson process in Section 3.7.1 were shown to be the propensities.

Chemical reactions described by equations (2.105)-(2.106) were simulated using the chemical Langevin equation (3.104), the results of which are shown in Figure 3.11. Note that the errors are due to the stochastic differential equation’s inability to capture higher order moments (moments $> 2$) as the errors are primarily in the tails of the distribution.
Chapter 4

Accelerated Stochastic Simulations in Space

4.1 Uniform Discretizations of the Diffusion Process

4.1.1 Introduction

Spatial distributions characterize the evolution of reaction-diffusion models of several physical, chemical, and biological systems. We present two novel algorithms for the efficient simulation of these models: Spatial $\tau$-Leaping ($S\tau$-Leaping), employing a unified acceleration of the stochastic simulation of reaction and diffusion, and Hybrid $\tau$-Leaping ($H\tau$-Leaping), combining a deterministic diffusion approximation with a $\tau$-Leaping acceleration of the stochastic reactions. The algorithms are validated by solving Fisher’s equation and used to explore the role of the number of particles in pattern formation. The results indicate that the present algorithms have a nearly constant time complexity with respect to the number of events (reaction and diffusion), unlike the exact stochastic simulation algorithm which scales linearly.

4.1.2 Reaction-Diffusion: Stochastic and Deterministic Models

Reaction-diffusion models are used to describe processes ranging from pattern formation in nature [32] and epidemics [33] to cancer induced angiogenesis [34]. These models are usually formulated either in terms of deterministic rate equations or by using stochastic descriptions of the underlying molecular processes. The stochastic description provides detailed information about the dynamics of the reaction-diffusion process, albeit at a significant computational cost over deterministic simulations.
The Stochastic Simulation Algorithm, Section 3.6, (SSA) \([12, 13]\) has been used extensively in biochemical modeling (\([35, 36]\) and references therein) of reactions that assume a homogeneous spatial distribution of the species involved. A number of algorithms \([14, 15, 37]\) have been presented for the acceleration of the SSA for homogeneous systems. In recent years the SSA has been extended to simulations involving spatially inhomogeneous molecular distributions undergoing diffusion and reaction processes \([38-40]\). The algorithm presented in \([38, 39]\) scales almost linearly with the number of events, but requires them to be scheduled thus prohibiting parallel execution. In \([40]\) the computational time is reduced by splitting the reaction-diffusion phenomena into two distinct diffusion and reaction phases. This splitting may introduce numerical artifacts for systems close to a microscopic level as the reaction and diffusion processes happen concurrently, in particular for systems that involve too few particles to be insensitive to this kind of splitting. Recent works have examined the qualitative behavior of stochastic systems and have provided extensions for the deterministic systems to include leading order corrections for molecular noise \([41, 42]\), hence losing some of the descriptive benefits of a completely stochastic simulation but with the advantage of a relative reduction in computational cost.

A number of issues remain open in spatial SSA, such as the modeling of the diffusion rates in complex geometries, algorithms of increased computational efficiency and accuracy, and the enforcement of the homogeneity assumption \([36]\).

Here we present two algorithms for the accelerated simulation of spatial reaction-diffusion processes: an accelerated spatial stochastic algorithm (\(S\tau\)-Leaping) employing a unified \(\tau\)-Leaping procedure for the stochastic simulation of both processes, and a hybrid method (\(H\tau\)-Leaping), combining a deterministic description for diffusion with a \(\tau\)-Leaping acceleration of the stochastic reactions. Both of the algorithms are validated in simulations of Fisher’s equation \([33]\). In addition, we explore the role of the number of particles in the pattern forming Gray-Scott equations \([5]\).

### Stochastic Modeling of Reaction-Diffusion Processes

Reaction-diffusion phenomena can be represented by stochastic models, where particles in a domain move via Brownian motion and are subject to molecular collisions. In the present spatial simulations, the domain is decomposed into independent cells such that a reactant molecule can only react with other reactants in its cell while diffusion events are modeled as unimolecular transitions to neighboring cells.

We consider a total of \(N\) species and a domain that is discretized into a set of uniform cells, \(C\), subject to the same set of reactions, \(R\). We denote by \(a_r(u)\), \(r \in R\), \(c \in C\), the propensity of the reaction \(r\) in the cell \(c\) and \(\nu^c_r = (\nu_{r1}, \ldots, \nu_{rN})\), the corresponding stoichiometric...
4.1. Uniform Discretizations of the Diffusion Process

vector. The set of diffusion transitions is $\mathbf{D}$, and $\nu_d^{(i,c)}$ is the stoichiometric vector of the diffusion transition $d \in \mathbf{D}$ for the species $i$ in the cell $c$. We can write the reaction-diffusion process in a unified framework in terms of generic transitions:

$$\sum_{i=1}^{N} \alpha_i^j A_i^j \rightarrow \sum_{i=1}^{N} \beta_i^j B_i^j, \quad j = 1, \ldots, M$$  \hspace{1cm} (4.1)

where $j$ is the index of the transition, $M$ is the number transitions, $A_i$ is the species undergoing a transition, $B_i$ is the species in the resulting transition, and $\alpha_i$ and $\beta_i$ are the stoichiometric values. As an example, the reaction transitions for the pattern forming Gray-Scott \cite{5} model are expressed as:

$$U_{x,y,z}^{i-1,y,z} + 2U_{x,y,z}^{i,y,z} \rightarrow 3U_{x,y,z}^{i,y,z},$$  \hspace{1cm} (4.2)

$$U_{x,y,z}^{i,y,z} \rightarrow U_{x,y,z}^{i+1,y,z},$$  \hspace{1cm} (4.3)

The diffusion process can be represented by transitions to neighboring cells:

$$U_{x,y,z}^{i,y,z} \frac{d_i}{\Delta x} U_{x,y,z}^{i-1,y,z} \quad U_{x,y,z}^{i,y,z} \frac{d_i}{\Delta y} U_{x,y,z}^{i,y-1,z} \quad U_{x,y,z}^{i,y,z} \frac{d_i}{\Delta z} U_{x,y,z}^{i,y,z+1},$$  \hspace{1cm} (4.4)

$$U_{x,y,z}^{i,y,z} \frac{d_j}{\Delta y} U_{x,y,z}^{i,y-1,z} \quad U_{x,y,z}^{i,y,z} \frac{d_j}{\Delta z} U_{x,y,z}^{i,y,z+1},$$  \hspace{1cm} (4.5)

$$U_{x,y,z}^{i,y,z} \frac{d_k}{\Delta z} U_{x,y,z}^{i,y,z-1} \quad U_{x,y,z}^{i,y,z} \frac{d_k}{\Delta y} U_{x,y,z}^{i,y,z+1},$$  \hspace{1cm} (4.6)

where $U_{x,y,z}^{i,y,z}$ denotes the molecular species $i$ inside the cell indexed as $(x, y, z)$, $d_i$ is the diffusion coefficient for species $i$ and $dl$ represents the cell sizes in all of the dimensions.

**Deterministic Modeling of Reaction-Diffusion Processes**

The stochastic model, presented in Section 4.1.2, can be represented by a deterministic reaction-diffusion model under the assumption of an infinite number of particles in the system. In the deterministic model, we evolve the concentration of substances, $u_i = u_i(x, t)$, $i \in \{1, \ldots, N\}$, according to partial differential equations of the form

$$\frac{\partial u_i}{\partial t} = d_i \Delta u_i + f^{(i)}(u),$$  \hspace{1cm} (4.7)

where $f^{(i)}$ denotes the rate of change in concentrations due to the reactions, and $d_i$ is the diffusion coefficient of substance $i$. This equation can be solved numerically, using techniques such as finite difference or particle strength exchange methods \cite{43}. 

4.1.3 Computational Methods

Spatial $\tau$-Leaping

Cao et al. in [16] have provided a computationally efficient method for calculating the time-step for the $\tau$-Leaping method without the need for evaluating derivatives. We follow [16] by creating a bound for the molecular population in each cell:

$$\tau_{\text{reaction}} = \min_{c \in C} \{ \tau_{c, \text{reaction}} \},$$  \hspace{1cm} (4.8)

and for each cell we have

$$\tau_{c, \text{reaction}} = \min_{i \in I} \left\{ \frac{\max \{ e_{i,c} u_{i,c} / g_{i} \}}{|\bar{\mu}_{i,c}^{\text{reaction}}(u)|}, \frac{\max \{ e_{i,c} u_{i,c} / g_{i} \}}{\left(\bar{\sigma}_{i,c}^{\text{reaction}}(u)\right)^{2}} \right\},$$  \hspace{1cm} (4.9)

where we have let $\epsilon$ be a control parameter such that $0 < \epsilon \ll 1$, $g_{i}$ is the highest order of reaction, $I$ is the set of different species and $\bar{\mu}_{i,c}^{\text{reaction}}(u)$ and $\left(\bar{\sigma}_{i,c}^{\text{reaction}}(u)\right)^{2}$ are given as follows:

$$\bar{\mu}_{i,c}^{\text{reaction}}(u) = \sum_{r \in R} \nu_{i,c}^{r} a_{r}(u),$$ \hspace{1cm} (4.10)

$$\left(\bar{\sigma}_{i,c}^{\text{reaction}}(u)\right)^{2} = \sum_{r \in R} \left(\nu_{i,c}^{r}\right)^{2} a_{r}(u),$$ \hspace{1cm} (4.11)

We can use the simple structure of the diffusion transitions in order to accelerate the computation of $\tau_{\text{diffusion}}$

$$\tau_{\text{diffusion}} = \min_{c \in C} \{ \tau_{c, \text{diffusion}} \},$$ \hspace{1cm} (4.12)

$$\tau_{c, \text{diffusion}} = \min_{i \in I} \left\{ \frac{\max \{ e_{i,c} u_{i,c} / g_{i} \}}{|\bar{\mu}_{i,c}^{\text{diffusion}}(u)|}, \frac{\max \{ e_{i,c} u_{i,c} / g_{i} \}}{\left(\bar{\sigma}_{i,c}^{\text{diffusion}}(u)\right)^{2}} \right\}. \hspace{1cm} (4.13)

The denominators can be computed as

$$\bar{\mu}_{i,c}^{\text{diffusion}}(u) = \frac{1}{d I^{2}} \sum_{c' \in N(c)} u_{i,c'} - u_{i},$$ \hspace{1cm} (4.14)

$$\left(\bar{\sigma}_{i,c}^{\text{diffusion}}(u)\right)^{2} = \frac{1}{d I^{2}} \sum_{c' \in N(c)} u_{i,c'} + u_{i},$$ \hspace{1cm} (4.15)

where $N(c)$ denotes the set of neighboring cells of $c$. Since equation (4.15) will always be greater than equation (4.14), the formula for $\tau_{c, \text{diffusion}}$ is simplified to:

$$\tau_{c, \text{diffusion}} = \min_{i \in I} \left\{ \frac{\max \{ e_{i,c} u_{i,c} / g_{i} \}}{\left(\bar{\sigma}_{i,c}^{\text{diffusion}}(u)\right)^{2}} \right\}. \hspace{1cm} (4.16)$$
4.1. Uniform Discretizations of the Diffusion Process

The time-step, $\tau$, is chosen as the minimum of the two time-steps,

$$\tau = \min\{\tau_{\text{reaction}}, \tau_{\text{diffusion}}\}. \tag{4.17}$$

We perform the transitions on the entire solution, $u = \{u^c\}_{c \in C}$, according to the following formula:

$$u(t + \tau) = u(t) + \sum_{c \in C} \sum_{r \in R} \nu_r^c \mathcal{P}(u_r(u^c), \tau) + \sum_{c \in C} \sum_{i \in I} \sum_{d \in D} \nu_d^{i,c} \mathcal{P}(d_i u_i^c, \tau), \tag{4.18}$$

where $\mathcal{P}(.)$ is a sample from a Poisson distribution.

**Hybrid $\tau$-Leaping**

In order to further accelerate the spatial modeling of reaction-diffusion systems, we propose a hybrid scheme where the reactions are simulated stochastically while diffusion is simulated deterministically. This approximation is suitable since the diffusion process is typically two orders of magnitude faster than the reaction process [28]. We consider a system where the particles, $u_i = u_i(x, t)$, evolve according to the following equation:

$$u_i(x, t + \tau) = u_i(x, t) + \mathcal{M}_1(d_i \Delta_d \mathcal{M}_2(u_i(x, t))) + f_s^{(i)}(u(x, t)), \tag{4.19}$$

where $f_s^{(i)}$ represents the stochastically simulated reactions, $\Delta_d$ represents a deterministic diffusion operator, and $\mathcal{M}_1$ and $\mathcal{M}_2$ are mapping functions such that $\mathcal{M}_1 : \mathbb{R}_+^N \to \mathbb{N}^N$ and $\mathcal{M}_2 : \mathbb{N}^N \to \mathbb{R}_+^N$.

$\mathcal{M}_1$ and $\mathcal{M}_2$ convert from between discrete and continuum representations of the field. $\mathcal{M}_2$ is trivial since in this mapping we have all of the information than we need, i.e. converting from a discrete to a continuum model. This can be done by dividing the number of particles by the value $P$, the number of particles per unit of concentration. Care, however, needs to be taken with $\mathcal{M}_1$ since we need to ensure both a fair mapping and also a conservation of mass within our system.

The procedure for $\mathcal{M}_1$ is as follows: suppose we have a single species on a 1-dimensional spatial domain where we denote $x_i$ as the cell discretization of the domain, for $i = 1, \ldots, N$, $\Gamma(x_i) := \Delta_d \mathcal{M}_2(u(x_i, t)))$, i.e. $\Gamma(x_i)$ is a concentration, and $P$ the number of particles per unit of concentration. First, we lift the value of $\Gamma(x_i)$,

$$\hat{\Gamma}(x_i) = \Gamma(x_i)P. \tag{4.20}$$
\( \hat{\Gamma}(x_i) \) can now be decomposed into a natural number part and a real part
\[
\hat{\Gamma}(x_i) = \hat{\Gamma}_N(x_i) + \hat{\Gamma}_R(x_i),
\]
where \( \hat{\Gamma}_N(x_i) \in \mathbb{N} \), \( \hat{\Gamma}_R(x_i) \in \mathbb{R}_+ \), and more specifically \( \hat{\Gamma}_R(x_i) \in [0, 1) \). If we crop the values of \( \hat{\Gamma}(x_i) \) such that
\[
\hat{\Gamma}(x_i) = \hat{\Gamma}_N(x_i),
\]
then we can distribute the “extra molecules” \( L \), where
\[
L = \sum_{i=1}^{N} \hat{\Gamma}_R(x_i),
\]
where \( L \in \mathbb{N} \). The objective now is to distribute these extra molecules by sampling from a probability density function where the probability of each cell is its fractional value \( \hat{\Gamma}_R(x_i) \). Therefore, we normalize all of the fractional values such that \( p(x_i) = \frac{\hat{\Gamma}_R(x_i)}{L} \). We denote the number of molecules gained for each cell \( i \) as \( k_i \) which is a realization of a random variable \( K_i \), for \( i = 1, \ldots, N \). We recall that a Binomial distribution, \( B(R, P) \), is a discrete probability density distribution giving the number of successes in a sequence of \( R \) independent Bernoulli trials having a success probability of \( P \). We consider the fractional values as Bernoulli trials where the probability of success is \( p(x_i) \), the probability of failure is \( 1 - p(x_i) \), and the number of trials is \( L \). Therefore, the distribution of \( K_i \) is \( k_i = B(L, p(x_i)) \), and all of the following variables \( m \in \{2, \ldots, N\} \), denoted as \( K_m \), are conditionally distributed on the previous events, i.e. on \( \{k_1, \ldots, k_{m-1}\} = \{K_1, \ldots, K_{m-1}\} \). Therefore, for these variables we need to scale their probabilities of success based on the previous events, and decrease the amount of trials based on the previous events. Hence, we can sample from the following distribution:
\[
k_m = B \left( L - \sum_{j=1}^{m-1} k_j, \frac{p(x_m)}{1 - \sum_{j=1}^{m-1} p(x_j)} \right),
\]
\[
\hat{\Gamma}(x_i) = \hat{\Gamma}_N(x_i) + k_i, \quad \text{for } i = 1 \ldots N.
\]
We note that at most \( N - 1 \) random numbers are needed and that the distribution of the molecules may terminate early if all \( L \) molecules have been distributed. It is also possible to distribute the \( L \) molecules in a point-wise manner instead of sampling from a Binomial distribution, but we have found that both \( L \) and the number of cells are large so that the method shown above is computationally more efficient.

The above equations trivially generalize to \( n \) dimensions where one has an \( n \)-dimensional space to distribute molecules instead of the 1-dimensional example given above. For example, in 3-dimensions where \( x_{i,j,k} \) is the discretization of the domain, for \( i = 1 \ldots N_i \),
4.1. Uniform Discretizations of the Diffusion Process

\[ j = 1 \ldots N_j, \text{ and } k = 1 \ldots N_k, \text{ then equation (4.24) becomes} \]
\[ k_{a,b,c} = B \left( L - \sum_{\alpha=1}^{a} \sum_{\beta=1}^{b} \sum_{\gamma=1}^{c-1} k_{\alpha,\beta,\gamma}, \frac{p(x_{a,b,c})}{1 - \sum_{\alpha=1}^{a} \sum_{\beta=1}^{b} \sum_{\gamma=1}^{c-1} p(x_{\alpha,\beta,\gamma})} \right), \tag{4.26} \]

and equation (4.25) becomes
\[ \hat{\Gamma}(x_{i,j,k}) = \hat{\Gamma}_n(x_{i,j,k}) + k_{i,j,k}, \text{ for } i = 1 \ldots N_i, j = 1 \ldots N_j, \text{ and } k = 1 \ldots N_k. \tag{4.27} \]

The function \( f \), performs an independent \( \tau \)-leaping procedure at the points \( x \) at time \( t \) with a time step of \( \tau \). Prescribing \( \tau \) is performed by binding the changes in molecular populations, as described in the previous section, at each cell. The final \( \tau \) is chosen as the minimum of all of these independent evaluations, and this \( \tau \) is used as the time step for all of the \( \tau \)-leaping procedures at each discretized volume.

The algorithm for the hybrid method is straightforward at this point. We choose a value for \( \tau \) and simulate the reactions in our volume. Then, using this \( \tau \), we may simulate the diffusion process. This procedure is performed iteratively until the final integration time is reached.

Note that the speed-up of this hybrid approximation lies not only in that deterministic diffusion is more efficient than sampling random numbers (either by a random walk or \( \tau \)-Leaping), but also because we only need to diffuse such that our numerical stability criterion for our deterministic diffusion scheme is satisfied. In other words, we do not necessarily need to diffuse after every reaction process.

4.1.4 Numerical Results

Validation: Fisher’s Equation

The proposed methods were validated by simulating Fisher’s equation [33], a model for the spreading of an advantageous gene in a population inhabiting a 1-dimensional space:
\[ \frac{\partial p}{\partial t} = k \Delta p + mp(1 - p). \tag{4.28} \]

Fisher’s equation admits traveling wave solutions [44] with a speed of \( c \), which is known analytically for certain initial conditions. We used a smoothed Heaviside step function (44), for the initial condition with \( k = 1 \) and \( m = 1 \), which has an analytical solution with a wavespeed of \( c = \frac{2}{\sqrt{m}} \) per unit of time. In addition, we used zero-flux Neumann boundary conditions. This initial condition was distributed across the entire \( y \) and \( z \)-axes of our
3-dimensional domain so that we should be able to obtain the same wavespeed in our simulation as in the 1-dimensional case. We solved Fisher’s equation with various resolutions in a domain of $[-14, 14] \times [-14, 14] \times [-14, 14]$ using the $S_\tau$-Leaping and $H_\tau$-Leaping methods with a varying number of particles per unit of concentration which we denote by $P$.

Figure 4.1 presents the error of the propagating front with respect to the analytical solution of Fisher’s equation. The plot on the left shows the error at a single time point ($t = 2$), and the plot on the right shows the error of the position of the front with respect to all time points. The results indicate that using $S_\tau$-Leaping with a high number of particles is the more accurate method. We observe that this fact can be justified by the absence of any decoupling of reaction and diffusion processes. However, when using $H_\tau$-Leaping, we can obtain a good approximation when the number of particles per concentration unit is above 1000.

In Figure 4.2 we show the relative performance of both methods in simulating Fisher’s equation by varying the number of cells and the number of particles. We observe that in both methods the number of cells is a more critical factor for CPU time as opposed to the number...
4.1. Uniform Discretizations of the Diffusion Process

Figure 4.2: CPU time in seconds for both methods, $S_\tau$-Leaping and $H_\tau$-Leaping, with different values for the number of particles per unit of concentration ($P$) and the number of cells per dimension ($N$). $S_\tau$-Leaping is denoted by ‘- - - - ‘, ‘- - -’ , and $H_\tau$-Leaping ‘x-x-x’, ‘- - -’. Shown on the left, the CPU time with varying values of $N$, where $P = 10^2, 10^6, 10^2, 10^6$ for the four methods, respectively. On the right, the CPU time with respect to $P$, where $N = 4, 64, 4, 64$, respectively.

of particles. We note that the CPU time does not grow linearly with respect to the number of particles (and thus the number of events), which is the case with the exact spatial SSA algorithm.

4.1.5 Conclusion

We presented two novel numerical methods for the efficient simulation of reaction-diffusion processes as described by stochastic and hybrid models. In $S_\tau$-Leaping, a unified $\tau$-Leaping procedure was used for both the reaction and the diffusion processes whereas in $H_\tau$-Leaping, diffusion was handled deterministically and the reactions stochastically. We validated the methods using the analytical solution of Fisher’s equation and we investigated the role of the number of molecules in pattern forming Gray-Scott equations (show in Chapter 7). The algorithms were shown to exhibit significant computational improvements over the exact spatial SSA.
4.2 Multiresolution Stochastic Reaction-Diffusion Systems in 1-D

4.2.1 Introduction

Spatially distributed stochastic simulations of reaction-diffusion processes are frequently used for the modeling of physical phenomena ranging from biology and social sciences to ecosystems and materials processing. Indeed spatial dynamics, such as wavefront propagation and pattern formation, are intrinsic to physical phenomena ranging from morphogenesis [4] and pedestrian traffic [45] to epitaxial growth [46] and epidemics [47]. Reaction-diffusion models of these phenomena often involve microscopic simulations using many-particle systems. The evolution of these systems can be modeled stochastically using algorithms known as the BKL [48] or the Stochastic Simulation Algorithm (SSA) [12] (Section 3.6). These methods were originally developed for homogeneous systems and their extension to spatially inhomogeneous systems is associated with a high computational cost. Spatially inhomogeneous, stochastic simulation methods divide the volume into uniform cells with reactions occurring within cells and diffusion events modeled as unimolecular transitions to neighboring cells. A number of recent works have employed such algorithms to simulate reaction-diffusion processes of biological systems [28, 39, 40, 49] using a uniform discretization of the computational domain. In these simulations, the finest spatial scales dictate the size of the cells, thus making the method highly inefficient in areas where coarser scales are operating. In order to overcome this difficulty, novel multiscale methods have been proposed [50–52], which combine stochastic, microscopic, deterministic, and coarse grained descriptions.

In this section, we present a novel multiresolution method for the efficient stochastic simulation of reaction-diffusion processes for spatially developing systems. The method entails discretizing the computational domain into cells of different sizes in the spirit of Adaptive Mesh Refinement (AMR) [53, 54], which was developed for the discretization of partial differential equations. The proposed multiresolution algorithm enables the stochastic handling of phenomena with disparate spatial scales, but at the same time it leads to a temporal disparity that increases the complexity of the simulations. We solve this problem by combining approximate, accelerated stochastic simulation algorithms [14, 15] with the AMR technique. In this work we quantify the scale disparity and the proposed algorithm is validated in simulations of one-dimensional wavefront propagation in a model reaction-diffusion system described by the Fisher-Kolmogorov equation [33]. The results demonstrate the need and the effectiveness of multiresolution simulations for inhomogeneous reaction-diffusion processes.
4.2.2 The Method

The governing reaction-diffusion processes are simulated using a stochastic particle description where particles in a computational domain, discretized by a series of meshes, move via Brownian motion and are subject to molecular collisions. In the present spatial simulations, the domain is decomposed into independent cells such that a reactant molecule can only react with other reactants in its cell, while diffusion events are modeled as unimolecular transitions to neighboring cells.

We consider a set of one-dimensional meshes indexed by an integer $\mathcal{L}$, with $\mathcal{L} = 0$ denoting the coarsest mesh, and the finer meshes denoted by increasing positive integers, such that the cell spacing for mesh level $\mathcal{L} + 1$ is half of that for level $\mathcal{L}$. Reaction-diffusion processes can be expressed in a unified framework in terms of generic transitions:

$$
\sum_{j=1}^{N} \alpha_{z,j} A_{i,j}^{\mathcal{L}_i} \rightarrow \sum_{j=1}^{N} \beta_{z,j} A_{k,j}^{\mathcal{L}_k},
$$

where $N$ is the total number of species, $\alpha_{z,j}$ and $\beta_{z,j}$ are the stoichiometric values for transition index $z$ for species $j$, and $A_{i,j}^{\mathcal{L}_i}$ represents the species $j$ at cell index $i$ at mesh level $\mathcal{L}_i$.

In the context of a multiresolution representation, the computational elements are mapped onto different levels of discretization corresponding to different mesh resolutions. This enables the efficient use of computational elements, since we can place larger numbers of computational elements in areas of the domain associated with fine spatial scales (e.g. around a propagating front), while other areas are discretized using fewer computational elements.

This representation requires communication between different discretization levels, a process that is facilitated by the discrete nature of the particles.

We let $U_i^{\mathcal{L}_i}$ denote the number of particles at cell index $i$ on mesh level $\mathcal{L}_i$. The refinement of the computational elements for a species $U_i^{\mathcal{L}_i}$ from level $\mathcal{L}_i$ to level $\mathcal{L}_i + 1$ is performed as:

$$
U_{2i+1}^{\mathcal{L}_i+1} \sim \mathcal{B} \left( U_i^{\mathcal{L}_i}, \frac{1}{2} \right),
$$

$$
U_{2i}^{\mathcal{L}_i+1} \sim \mathcal{B} \left( U_i^{\mathcal{L}_i} - U_{2i}^{\mathcal{L}_i+1}, \frac{1/2}{1 - 1/2} \right) = U_i^{\mathcal{L}_i} - U_{2i}^{\mathcal{L}_i+1},
$$

where $\mathcal{B}(N, p)$ represents a Binomial distribution of $N$ independent trials with a success rate of $p$, and we note that equation (4.31) represents a conditional distribution.

The coarsening of computational elements from level $\mathcal{L}_i + 1$ to $\mathcal{L}_i$ is performed by

$$
U_i^{\mathcal{L}_i} = U_{2i+1}^{\mathcal{L}_i+1} + U_{2i+1}^{\mathcal{L}_i+1}.
$$
4.2.3 Temporal Scale Disparity: Diffusion Propensities

We define $a_{D,i,j}^{L_i,L_j}$ as the diffusion propensity from cell $i$ on level $L_i$ to cell $j$ on level $L_j$, where $j$ is a neighboring cell to $i$:

$$a_{D,i,j}^{L_i,L_j} = U_i^{L_i} \kappa(L_i, L_j).$$  \hspace{1cm} (4.33)

The diffusion rate, $\kappa(L_i, L_j)$, can be derived by virtue of a Finite Volume approximation as shown in [28], and is given as:

$$\kappa(L_i, L_j) = \frac{2\nu}{h(L_i)(h(L_i) + h(L_j))},$$ \hspace{1cm} (4.34)

where $h(L)$ is the cell spacing at level $L$ and $\nu$ is the diffusion coefficient. Using the partial sum for a geometric series and equations (4.33)-(4.34), the mean change in propensities with respect to the coarsest level is:

$$a_{D,i,j'}^{L_i',L_j'} = a_{D,i,j}^{0,0} \zeta(L_i', L_j'),$$ \hspace{1cm} (4.35)

where

$$\zeta(L_i, L_j) = \left\{ \begin{array}{ll}
2 & \text{if } L_i = L_j \\
\frac{\alpha^{\max(L_i, L_j)}}{2^{1 + \max(L_i, L_j) - 1}} & \text{otherwise.}
\end{array} \right.$$

Equations (4.35)-(4.36) show that non-uniform cell sizes introduce disparities in the diffusion propensities since finer cells exhibit faster diffusion rates compared with coarser cells.

4.2.4 Temporal Scale Disparity, Reaction and Diffusion Propensities

*Stiffness*, which is a disparity in time-scales, is present in most stochastic, homogeneous chemical systems [55, 56]. Here we show that, by decreasing the cell size in a uniform discretization for inhomogeneous systems, the reaction and diffusion propensities become progressively disparate. Consequently, this forces exact stochastic simulation algorithms [12, 13] to spend more time sampling diffusion events than reaction events. This resolution-dependent stiffness warrants the efficient allocation of computational resources, such as adaptive meshes since the finest spatial scales are often localized in the domain.

We denote the dimension of the problem by $d$, and define a characteristic length scale $h_\lambda$ for each level of discretization such that:

$$h_\lambda = \frac{L_0}{2^\lambda},$$ \hspace{1cm} (4.37)

where $\lambda \geq 0$ and $L_0$ is the length of the domain. Additionally, we define the number of particles of species $s$ when $\lambda = 0$ to be $X_s$ and the corresponding concentration of species
4.2. Multiresolution Stochastic Reaction-Diffusion Systems in 1-D

Figure 4.3: Scale-disparity of Fisher-Kolmogorov equation: ratio of the maximum diffusion propensity to the maximum reaction propensity plotted against the cell size, $h_k$. ‘- - - -’ denotes the estimated value, $\xi(\lambda)$ (equation (4.44)), ‘- - -’ denotes the numerical value, $\xi(\lambda)$ (equation (4.46)) for $\nu = 1/160^2$, $k = 1$ and $L_0 = 1$.

$s$ to be $\chi_s = X_s/V_\lambda \leq 1$, where $V_\lambda$ is a normalization factor that depends on $\lambda$. Employing equations (4.33)-(4.34) and noting that the number of particles in a cell is proportional to the cell size, the maximum diffusion propensity for a species $X_1$ is given as:

$$\hat{a}_D = \left( X_1 h_\lambda^2 \right) \left( \frac{\nu}{h_\lambda^2} \right) = X_1 \nu L_0^{d-2} 2^{2\lambda-d}. \quad (4.38)$$

Without loss of generality, we consider a representative set of bimolecular reactions (frequently encountered in chemical kinetics and phase transition problems) such as the Fisher-Kolmogorov equation in [33], originally proposed as a model for the propagation of a gene in a population. This equation models reaction-diffusion processes admitting traveling wave solutions. The continuum form of this equation for the two species involved, $\chi_1$ and $\chi_2$, reads:

$$\frac{\partial \chi_1}{\partial t} - \nu \Delta \chi_1 = k \chi_1 \chi_2 = k(\chi_1^2 - \chi_1^4), \quad (4.39)$$

where $k$ is the deterministic reaction rate and the conservation relation, $\chi_1 + \chi_2 = 1$, has been used. If the initial condition of equation (4.39) satisfies $0 \leq \chi_1(x,0) \leq 1$, $\chi_1(x,0) = 1$ for $x < a$, $\chi_1(x,0) = 0$ for $x > b$, where $a < b$, then the solution is a traveling wave with a constant wavespeed [57]. In cases of low particle concentrations, the continuum equation can be replaced by its equivalent discrete form:

$$X_1 + X_2 \rightarrow 2X_1, \quad (4.40)$$
where $X_1$ and $X_2$ are both diffusing species. The propensity for any such biomolecular reaction can be written as

$$\dot{a}_R = (X_1 h^d_1)(X_2 h^d_2) \left( \frac{k}{V_\lambda} \right).$$  

(4.41)

The concentration of $X_2$ is obtained from equation (4.41),

$$\chi_2 = \frac{X_2 h^d_2}{V_\lambda} \leq 1,$$

(4.42)

thus, the maximum reaction propensity is (cf. equation (4.38))

$$\dot{a}_R = X_1 k L^d_0 2^{-\lambda d}.$$  

(4.43)

To estimate the relative disparity between reaction and diffusion propensities, we define a dimensionless scaling parameter $\hat{\xi}(\lambda)$ where

$$\hat{\xi}(\lambda) = \frac{\hat{a}_D}{\hat{a}_R} = \frac{2^{2\lambda}}{L_0^2} \left( \frac{\nu}{k} \right), \quad \lambda \geq 0.$$  

(4.44)

Furthermore, this can be recast in terms of mean waiting time for reaction and diffusion events as:

$$\hat{\xi}(h) = \frac{\hat{\tau}_R}{\hat{\tau}_D} = \frac{\hat{a}_R^{-1}}{\hat{a}_D^{-1}} = \frac{\nu}{h^2 k},$$

(4.45)

where $\hat{\tau}_R$ is the mean free time with respect to reactive collisions in a volume element and $\hat{\tau}_D$ is the mean time during which a molecule will remain in a volume element.

We observe that $\hat{\xi}(\lambda)$ is independent of the dimensionality of the problem, $d$. The Finite Volume approximation of the propensities in equations (4.33)-(4.34) scales with $O(h^2_\lambda)$ [28] thus, accurate simulations of the diffusion process engenders temporal scale disparities. The numerical value quantifying this scale-disparity is

$$\hat{\xi}(\lambda) = \frac{\max_i(a_{D,j,i}^{L_i})}{\max_i(a_{R,i}^{L_i})},$$

(4.46)

where $a_{D,j,i}^{L_i}$ is the diffusion propensity defined in equations (4.33)-(4.34), and $a_{R,i}^{L_i}$ is the reaction propensity for cell $i$ on level $L_i$ [12,13]. In Figure 4.3, we show $\hat{\xi}(\lambda)$ and $\hat{\xi}(\lambda)$ plotted against $h_\lambda$, which represents the temporal scale disparity of Fisher-Kolmogorov equation for $\nu = 1/160^2$, $k = 1$ and $L_0 = 1$. It can be seen that as $h_\lambda$ decreases, the ratio of the diffusion to reaction propensities increases, thus leading to a stiffer system.
4.2. Multiresolution Stochastic Reaction-Diffusion Systems in 1-D

Figure 4.4: Multiresolution mesh for the Fisher-Kolmogorov simulation: cell size \( h_\lambda \) against position, where \( \cdot \quad \bullet \cdot \cdot \cdot \cdot \) is the resolution at \( t = 0 \) and \( \cdot \quad \cdot \quad \bullet \cdot \) at \( t = t_{end} = 19.6 \). Wavefront center at \( t_{end} \) is \( x = 1/4 \).

4.2.5 Numerical Results

The Fisher-Kolmogorov equation exhibits a localization of fine spatial scales in the form of a traveling wave. To demonstrate the validity of the present method, we simulated the Fisher-Kolmogorov equation (see equations (4.39)-(4.40)) with \( \nu = 1/160^2 \), \( k = 1 \), on the domain \( x \in [-1/4, 3/4] \), using approximate, accelerated stochastic simulation algorithms [14, 15]. The analytical solution for the continuum form of the Fisher-Kolmogorov (equation (4.39)) is [44, 57]:

\[
\chi_1(x, t) = \frac{1}{(1 + ae^{b(x-ct)})^2},
\]

where \( a = \sqrt{2} - 1 \), \( b = 80\sqrt{2/3} \), and the wavespeed \( c = 1/(32\sqrt{6}) \). Equation (4.47) was used to generate an initial condition with a total of \( 8 \times 10^6 \) particles in the domain. Consequently, the shape of the wave, save for fluctuations, remains the same so that the error with respect to the velocity could be determined. We used four types of discretizations for comparison: three uniform meshes and one multiresolution mesh. The cell sizes were \( h_\lambda = 2.5 \times 10^{-2} \), \( 1.25 \times 10^{-2} \), and \( 6.25 \times 10^{-3} \) for the uniform meshes, and \( \min(h_\lambda) = 6.25 \times 10^{-3} \) and \( \max(h_\lambda) = 2.5 \times 10^{-2} \) for the multiresolution mesh. We note that these values are also used to show the scale-disparity in Figure 4.3. Simulations were performed until \( t = t_{end} = 19.6 \). The multiresolution mesh was refined and coarsened according to equations (4.30)-(4.32) using a priori knowledge of the wavespeed. In Figure 4.4, the initial and final multiresolution meshes are shown, where the initial mesh was centered around \( x = 0 \) and the final mesh around \( x = 1/4 \). The figure also indicates that a total of three
Figure 4.5: Error of Fisher-Kolmogorov simulation: pointwise error with respect to the analytical solution against the position. ‘- ▲ -’, ‘- x -’, ‘- - - - -’ uniform methods with $h_\lambda = 2.5 \times 10^{-2}, 1.25 \times 10^{-2}, 6.25 \times 10^{-3}$ respectively. ‘- - -’ multiresolution method with $\min(h_\lambda) = 6.25 \times 10^{-3}$ and $\max(h_\lambda) = 2.5 \times 10^{-2}$. Wavefront center at $t = t_{\text{end}} = 19.6$ is $x = 1/4$ for the analytical solution.

Figure 4.5 shows the pointwise error of the four simulations with respect to the analytical solution of the Fisher-Kolmogorov equation (4.47) at $t = t_{\text{end}} = 19.6$. The error indicates the affect of the discretizations with respect to the wavespeed. The coarsest uniform discretization clearly has an inaccurate wavespeed, while the multiresolution method displays an accuracy comparable to the uniform method with $h_\lambda = 6.25 \times 10^{-3}$, and yet it requires approximately 67% less computational time. The Gaussian-like shape of the error reveals that the center of the wave is a critical part of the chemical system. The center of the wavefront for the analytical solution is $x = 1/4$. 

levels were used for the simulation.
4.2.6 Conclusion

We presented a novel framework for multiresolution stochastic simulations of reaction-diffusion processes exhibiting disparate scales. The framework relies on the efficient combination of multiresolution discretizations to capture the disparate spatial scales of reaction-diffusion processes, and novel accelerated stochastic simulation algorithms capable of resolving the resulting scale disparities. The proposed methodology is general and applicable in a wide range of spatial stochastic many-particle models of physical processes ranging from social systems to biology.
4.3 Multiresolution Stochastic Reaction-Diffusion Systems using AMR

4.3.1 Introduction

Simulations of systems that exhibit multiple spatial scales often employ Adaptive Mesh Refinement methods (AMR) [53, 54] for the discretization of the governing Partial Differential Equations (PDEs). However, when the mesh spacing approaches the molecular scale and thermal fluctuations affect the dynamics of the system, the use of partial differential equations may become unjustifiable for variables that are inherently random and discrete [58–60]. At the same time simulations of microscopic phenomena may not be efficiently computed using atomistic level modeling as in Molecular Dynamics (MD). There is an increased interest [28, 49, 61–67] in formulating methods for mesoscopic simulations in which the computational elements contain neither so many particles as to be considered in a continuum, nor so few as to warrant MD simulations. Such mesoscopic simulations employ variables that are assumed to be governed by the laws of probability theory. The transition probabilities of any discrete-state, continuous-time Markov process must obey the Chapman-Kolmogorov equation (see Section 2.3.3), which in turn is equivalent to the so-called Master Equation (2.59) (M-equation) [9]. Since the number of states is large - possibly infinite - for all but the simplest systems, analytical or direct numerical integration methods for the M-equation are generally impractical. Being at a loss for a direct solution, numerical realizations of the stochastic process can alternatively be generated. These numerical realizations, performed via Stochastic Simulation Algorithms [12, 48] (see Section 3.6), amount to generating random variates from the unknown discrete probability density function.

The M-equation frequently appears in chemical kinetics since the time-evolution of chemically-reacting species in a system is inherently stochastic owing to Brownian motion [68, 69]. The applicability of the M-equation to chemical kinetics rests on the assumption that the system is in thermal equilibrium, such that any molecule has enough time, on average, to move throughout the volume before it participates in a reaction. If Brownian motion does not suffice, then the system is inhomogeneous and the problem has intrinsically local phenomena. The variables in the system are recast to represent the number of molecules of a species in a subvolume - or volume element - of space. This description has been previously studied theoretically [61–63] and numerically [64] using a uniform discretization in 1-D. Moreover, 2- and 3-D uniform discretizations have been used for simulating nonlinear reaction-diffusion processes [49]. Since inhomogeneities may arise at various scales throughout the domain, the placement of smaller or larger volume elements in certain regions of the domain readily lends itself to finite volume methods [70]. Non-uniform 1-D
discretizations, which have borrowed techniques from finite volume methods, have been employed for propagating wavefront systems [28, 65]. Recently, Engblom et al. [66] and Ferm et al. [71] have used adaptive and unstructured meshes in simulations of stochastic processes. Additionally, Drawert et al. [72] have used the Finite State Projection algorithm presented in [73] to simulate reaction-diffusion processes.

This work is concerned with formulating the transition rates, or propensities, for reaction-diffusion processes on adaptive locally refined structured meshes in the spirit of Adaptive Mesh Refinement (AMR) [53, 54, 74], and simulating mesoscopic reaction-diffusion processes using these rates. The convergence of the diffusion process is presented as well as a refinement criterion for the adaptive mesh. The method is applied to relatively high spatial-resolution simulations of the Fisher-Kolmogorov [33] and Gray-Scott [5] equations. The results indicate that the method is especially suited for, but not limited to, wavefront propagation and pattern formation problems.

The section is organized as follows: Subsections 4.3.2 and 4.3.3 introduce the Master Equation and Stochastic Simulation Algorithms, respectively. Subsection 4.3.4 presents a derivation for diffusion propensities on locally refined meshes, and Subsection 4.3.5 recapitulates reaction propensities and their validity with respect to the spatial discretization. Subsection 4.3.6 presents the mesh refinement criterion and stochastic interpolation method used in this study. Subsection 4.3.7 shows both the validity of the method and numerical examples, and Subsection 4.3.8 concludes this work.

### 4.3.2 Reaction-Diffusion Master Equation

Let \( \mathcal{G} \subseteq \mathbb{R}^d \) denote the domain in which stochastic reaction and diffusion processes reside. Following the notation presented in [75], let \( \mathcal{G} \) be subdivided into a collection of voxels, or volume elements, labelled by vectors \( \mathbf{i} \) that belong to an index set, namely \( \mathbf{i} \in \mathcal{I} \). Given \( S \) species in \( \mathcal{G} \), let the time-dependent random variable \( U^{(s)}_i(t) \) denote the number of molecules (or particles) of species \( s \) in volume element \( i \) at time \( t \). Furthermore, define \( \mathbf{U}_i \triangleq (U^{(1)}_i, \ldots, U^{(S)}_i) \), \( \mathbf{U}^{(s)} \triangleq \{ U^{(s)}_i \}_{i \in \mathcal{I}} \), and \( \mathbf{U}(t) \triangleq \{ \mathbf{U}_i \}_{i \in \mathcal{I}} \).

The Reaction-Diffusion Master Equation (RDME) describes the time evolution of a prob-
ability density function:

\[
\frac{dp(U, t)}{dt} = 
\sum_{i \in I} \sum_{j \in I} \sum_{s=1}^S \left\{ a_{ij}^{(s)} \left( U^{(s)} + 1_i^{(s)} \right) p \left( U + 1_i^{(s)} - 1_i^{(s)}, t \right) - a_{ij}^{(s)} \left( U^{(s)} \right) p \left( U, t \right) \right\} + 
\sum_{i \in I} \sum_{m=1}^M \left\{ a_i^{(m)} \left( U_i - \nu^{(m)} \right) p \left( U - \nu^{(m)} 1_i, t \right) - a_i^{(m)} \left( U_i \right) p \left( U, t \right) \right\},
\]

(4.48)

where the diffusion process has been modeled as the movement of a molecule to a neighboring volume element \([28, 61]\); \(a_{ij}^{(s)}(.)\), the so-called propensity function, represents the probability per unit time of the diffusion of a molecule from volume element \(j\) to \(i\); \(a_i^{(m)}(.)\) denotes the propensity of reaction \(m\), where \(m = 1, \ldots, M\); \(\nu^{(m)}\) represents the stoichiometric vector of reaction \(m\); and \(1_i^{(s)}\) denotes one molecule of species \(s\) at index \(i\) (i.e. \(U(t) + 1_i^{(s)} = U_i^{(s)}(t) + 1\)).

### 4.3.3 General Formulation of Stochastic Simulation Algorithms

Equation (4.48) constitutes a set of ordinary differential equations. Note that the evolution of the probability density function is determined by the time-dependent propensity functions \(a_{ij}^{(s)}(.)\) and \(a_i^{(m)}(.)\), which represent the unscaled probabilities per unit time of their respective reactions. Since the diffusion events are modeled as unimolecular transitions to neighboring cells, equation (4.48) can be completely characterized by a set of propensities that vary over time. Indeed, stochastic simulations work by generating random variates from the probability density function \(p(U, t)\) at every time-step. Stochastic simulations therefore simulate the underlying Markov process, which is defined by the time-dependent propensities.

Given an initial vector of the state space \(U_{sim}\), stochastic simulation algorithms proceed as follows:

\[
\Theta(t) \triangleq (\hat{a}_1(t), \ldots, \hat{a}_W(t)),
\]

(4.49)

\[
\tau \sim \zeta(\Theta(t), \epsilon),
\]

(4.50)

\[
k^{(w)} \sim \Psi(\Theta(t), \tau),
\]

(4.51)

\[
U_{sim}(t + \tau) = U_{sim}(t) + \sum_{w=1}^W k^{(w)} \nu^{(w)},
\]

(4.52)

where \(\Theta(t)\) represents propensities \(\hat{a}_w(t)\) only from the current state to all other of the \(W\) reachable states, where \(W\) is the sum of the diffusion and reaction events; the vector \(\nu^{(w)}\) denotes the change induced by transformation \(w\); and the distributions \(\zeta\) and \(\Psi\) vary depending
on the algorithm that is used (see [68] for a review). These algorithms proceed by iterating through equations (4.49)-(5.7) until the predefined final time \( t_f \) is reached. Ultimately, \( \mathbf{U}_{\text{sim}}(t_f) \) is an approximate random variate from \( p(\mathbf{U}, t_f) \). Specifically, the relationship between the approximate and exact solution is:

\[
p(\mathbf{U}_{\text{sim}}(t_f)) + C_1 \epsilon + C_2 N^{-1/2} + C_3 h^2 = p(\mathbf{U}, t_f)
\]

where \( C_1 \epsilon \) is the error from the integration over time of the stochastic simulation (\( \epsilon = 0 \) in the case of exact simulation algorithms [12,76]), \( C_2 N^{-1/2} \) is the well-known sampling error of Monte Carlo methods, and \( C_3 h^2 \) is the error from the spatial-discretization of the diffusion process.

### 4.3.4 Diffusion

The objective of this derivation is to determine the rates of transitions that represent diffusion for non-uniform volume elements. Following the work presented in [28], the rates will be derived by virtue of a finite volume method. The well-known transition rate for uniformly sized volume elements, \( D/h^2 \), will be derived again for the sake of completeness. It should be noted that the jump rates will be chosen to reproduce the statistics of random walks (see Section 4.3.7 and [28]) and to conform to a discretization of the Laplace operator (see Section 4.3.7 and [75]). Specifically, the differential equations derived in this section - in which differentiation relies upon the continuum limit - will be used to determine (i) the propensities for a discrete-space, continuous-time random walk on a non-uniform lattice and (ii) the rate at which the equations converge to the solution of the diffusion equation. In the limit of a large number of molecules, the scaled propensities ought to converge to the continuum equations [31] and this is satisfied, by construction, in this derivation.

The diffusion process on locally refined meshes will be derived in 2-D in order to alleviate the notation, although the 3-D derivation is analogous. Let \( \bar{u}^{(s)}(x) \) be the concentration of species \( s \), \( \bar{u}_i^{(s)} \) be the average concentration at volume element \( i \), to wit \( \bar{u}_i^{(s)} \triangleq h^{-2} \int_i u^{(s)} \, dV \), where \( h \) is the length of element \( i \), and \( U_i^{(s)} \) denotes the number of molecules in a volume element; videlicet \( U_i^{(s)} \triangleq \int_i \bar{u}_i^{(s)} \, dV = \bar{u}_i^{(s)} h^2 \). It can be shown that an infinite number of molecules that are subject to Brownian motion is equivalent to the continuum diffusion equation [60]. The diffusion equation consists of the equations for the conservation of mass

\[
\frac{\partial u^{(s)}}{\partial t} = -\nabla \cdot \mathbf{J},
\]

and Fick’s law for the flux \( \mathbf{J} \):

\[
\mathbf{J} = -D^{(s)} \nabla u^{(s)},
\]
where $D^{(s)}$ is the macroscopic diffusion coefficient of the species $s$. It is assumed that $D^{(s)}$ is constant throughout this derivation, although this need not be the case. It has been shown in \cite{77, 78} that spatially-dependent diffusion coefficients can be used with finite volume schemes. Specifically, the harmonic average is used to determine the diffusion coefficient across interfaces.

Integrating equation (4.54) over a volume element $i$ and applying the divergence theorem on the right-hand-side yields

$$\frac{dU^{(s)}}{dt} = -\int_{\partial i} \mathbf{J} \cdot \mathbf{n} \, dS. \quad (4.56)$$

Decomposing the surface integral into faces, $\gamma_a$, $a = 1, \ldots, 4$, gives \cite{28}

$$\frac{dU^{(s)}_i}{dt} = -\sum_{a=1}^{4} \int_{\gamma_a} \mathbf{J} \cdot \mathbf{n} \, dS. \quad (4.57)$$

**Uniform Mesh Interfaces**

The flux $\mathbf{J}$ between neighboring volume elements that have the same length, i.e. not at an interface, is obtained by virtue of a centered second-order finite-difference scheme (see Figure 4.6). For the face $\gamma_1$, the flux is

$$\mathbf{J} \left( x - \frac{h}{2}, y \right) = -D^{(s)} \left( \frac{\bar{\bar{u}}^{(s)}_i - \bar{u}^{(s)}_i}{h} \right) + \mathcal{O}(h^2). \quad (4.58)$$

Employing a quadrature on the interface and substituting $U^{(s)}_i$ for $\bar{u}^{(s)}_i$ yields

$$\frac{dU^{(s)}_i}{dt} = -\int_{\gamma_1} \mathbf{J} \left( x - \frac{h}{2}, y \right) \cdot \mathbf{n} \, dS$$

$$= D^{(s)} \left( \frac{\bar{u}^{(s)}_1 - \bar{u}^{(s)}_i}{h} \right) h + \mathcal{O}(h^2)$$

$$= \frac{D^{(s)}}{h^2} \left( U^{(s)}_1 - U^{(s)}_i \right) + \mathcal{O}(h^2). \quad (4.59)$$

The diffusion propensities are the rates per unit time \cite{28}:

$$a_i^{(s)} = \frac{D^{(s)}}{h^2} U^{(s)}_i, \quad (4.60)$$

$$a_{i1}^{(s)} = \frac{D^{(s)}}{h^2} U^{(s)}_1, \quad (4.61)$$

where these represent the unscaled probabilities of a molecule that moves from volume element $1$ to $i$ and vice versa, respectively.
Locally Refined Mesh Interfaces using Cell Averaging

For coarse-fine interfaces, series expansions will be used to approximate the gradient by fitting a plane through the known values, e.g. cells $i$, $j$, and $k$ across the interface $\gamma_2$ in Figure 4.6. Three second-order Taylor’s series will be used to approximate $J(x, y + \frac{2}{3}\delta y)$, namely: $\tilde{u}_i^{(s)}$, $\tilde{u}_j^{(s)}$, and $\tilde{u}_k^{(s)}$ about the point $(x, y + \frac{2}{3}\delta y)$:

\[
\begin{align*}
\tilde{u}_i^{(s)} &= u^{(s)}(x, y + \frac{2}{3}\delta y) - \frac{2\delta y}{3} \frac{\partial}{\partial y} u^{(s)}(x, y + \frac{2}{3}\delta y) + O(h^2), \\
\tilde{u}_j^{(s)} &= u^{(s)}(x, y + \frac{2}{3}\delta y) - \frac{\delta x}{\partial x} u^{(s)}(x, y + \frac{2}{3}\delta y) + O(h^2), \\
\tilde{u}_k^{(s)} &= u^{(s)}(x, y + \frac{2}{3}\delta y) + \frac{2\delta y}{3} \frac{\partial}{\partial y} u^{(s)}(x, y + \frac{2}{3}\delta y) + O(h^2),
\end{align*}
\]

(4.62)

(4.63)

(4.64)

where $\delta x \equiv h/4$ and $\delta y \equiv 3h/4$. Multiplying equation (4.62) by two and subtracting equations (4.63) and (4.64) gives

\[-2\delta y \frac{\partial}{\partial y} u^{(s)}(x, y + \frac{2}{3}\delta y) = 2\tilde{u}_i^{(s)} - \tilde{u}_j^{(s)} - \tilde{u}_k^{(s)} + O(h^2).\]

(4.65)
Chapter 4. Accelerated Stochastic Simulations in Space

Isolating the derivative along the y-axis requires dividing by $\delta y$ and renders the approximation first-order:

$$ \frac{\partial}{\partial y} u^{(s)}(x, y + \frac{2}{3} \delta y) = \frac{1}{2 \delta y} \left( \tilde{u}_{i}^{(s)} + \bar{u}_{k}^{(s)} - 2 \bar{u}_{i}^{(s)} \right) + O(h). \quad (4.66) $$

The flux for the cell averages is therefore

$$ \mathbf{J} \left( x, y + \frac{2}{3} \delta y \right) = -\frac{D^{(s)}}{2 \delta y} \left( \tilde{u}_{i}^{(s)} + \bar{u}_{k}^{(s)} - 2 \bar{u}_{i}^{(s)} \right) + O(h). \quad (4.67) $$

Using a quadrature, substituting in $h$ for $\delta y$, $U_{i}^{(s)}$ for $\bar{u}_{i}^{(s)}$ and analogously for $\tilde{u}_{i}^{(s)}$ and $\bar{u}_{i}^{(s)}$, and simplifying yields the following difference equation across the interface:

$$ \frac{dU_{i}^{(s)}}{dt} = -\int_{\gamma_{2}} \mathbf{J} \left( x, y + \frac{2}{3} \delta y \right) \cdot \mathbf{n} \, dS = \frac{4D^{(s)}}{3h^{2}} \left( 2 \left( U_{i}^{(s)} + U_{k}^{(s)} \right) - U_{i}^{(s)} \right) + O(h). \quad (4.68) $$

The derivations for the fine volume elements, $j$ and $k$, use the fluxes

$$ \mathbf{J} \left( x \pm \delta x, y + \frac{2}{3} \delta y \right) = \mathbf{J} \left( x, y + \frac{2}{3} \delta y \right), \quad (4.69) $$

which result in

$$ \frac{dU_{i}^{(s)}}{dt} = \frac{dU_{j}^{(s)}}{dt} = \frac{2D^{(s)}}{3h^{2}} \left( U_{i}^{(s)} - 2 \left( U_{j}^{(s)} + U_{k}^{(s)} \right) \right) + O(h), \quad (4.70) $$

where the quadratures are performed with an interface length of $h/2$. Additionally, the following conservative requirement across the interface is satisfied:

$$ \frac{dU_{i}^{(s)}}{dt} = - \left( \frac{dU_{j}^{(s)}}{dt} + \frac{dU_{k}^{(s)}}{dt} \right). \quad (4.71) $$

Analogously to the propensities defined in equations (4.60) and (4.61), the propensities across the interface are

$$ a_{ij}^{(s)} = a_{ik}^{(s)} = \frac{4D^{(s)}}{3h^{2}} \left( U_{j}^{(s)} + U_{k}^{(s)} \right), \quad (4.72) $$

$$ a_{ji}^{(s)} = a_{ki}^{(s)} = \frac{2D^{(s)}}{3h^{2}} U_{i}^{(s)}. \quad (4.73) $$
4.3. Multiresolution Stochastic Reaction-Diffusion Systems using AMR

Locally Refined Mesh Interfaces using Interpolation

Another method can be used for diffusion, namely interpolating the values in cell $i$ by virtue of the interpolation scheme presented below in Subsection 4.3.6. As shown in Figure 4.7, the cell $i$ is split into four cells. This enables the use of standard rates (i.e. $D/h^2$ as in equations (4.6o)-(4.6i)) since the mesh is now uniform. The error introduced by this interpolation will be inquired into below.

![Figure 4.7: Stencil for the adaptive multiresolution mesh using interpolation, where $\hat{u}_1$ and $\hat{u}_2$ are interpolated values. A standard 5-point finite difference stencil can then be used for cell $j$.]

4.3.5 Reactions and the Homogeneity Condition

The reaction propensities depend only on the number of each species in each volume element. A reaction $m$ in volume element $i$ can be written as

$$\sum_s r_i^{(s)} U_i^{(s)} \xrightarrow{k_+} \sum_s g^{(s)} U_i^{(s)},$$

(4.74)

where $k_+$ is related to the cross-section of a collision of the required molecules $[9]$; $r^{(s)}$ and $g^{(s)}$ are the numbers of reactants and products of species $s$, respectively. The propensity is
defined as
\[ q_i^{(m)} = \Omega_i k_+ \prod_s \left\{ \frac{((U_i^{(s)}))^{r(s)}}{\Omega_i^{r(s)}} \right\}, \quad (4.75) \]
where \((U_i^{(s)})^{r(s)} \models U_i^{(s)} (U_i^{(s)} - 1)(U_i^{(s)} - 2) \cdots (U_i^{(s)} - r(s) + 1)\). The validity of equation (4.75) rests on the assumption that the volume element \(i\) is homogeneous. This condition has been shown to be equivalent to [61]
\[ \frac{\tau_R}{\tau_D} \gg 1, \quad (4.76) \]
where \(\tau_R\) is the mean free time with respect to reactive collisions in a volume element and \(\tau_D\) is the mean time during which a molecule will remain in a volume element. An estimate of this ratio for a bimolecular reaction, irrespective of the dimensionality of the problem, is given by equations (4.44)-(4.45)
\[ \frac{\tau_R}{\tau_D} = \frac{D}{h^2 k_+}, \quad (4.77) \]
which shows that fine spatial resolutions are necessary to enforce the homogeneity condition. It should be noted that there is a lower-bound on the size of the volume elements (see [64,75,79]). In this work, however, the refinement procedure is halted after a predefined number of levels has been reached.

4.3.6 Adaptivity

Refinement Criterion

At every predefined number of iterations in the simulation algorithm, all of the refined levels are coarsened to the top level. Since the volume elements represent the number of molecules of a species and the refinement ratio is constant, the four (2-D) or eight (3-D) values are summed to produce the value at the next coarsest level.

Let \(\alpha, \beta, \gamma,\) and \(\delta\) denote the indices of the neighboring cells of element \(i\). The refinement criterion follows directly from the homogeneity assumption in that the objective is to minimize the gradients between volume elements. If the system were in equilibrium, the value of \(U_{\alpha}^{(s)}\) can be approximated by \(U_{\alpha}^{(s)} \sim \mathcal{N}(\Omega, \Omega)\), where the volume element indices for \(\Omega\) have been discarded since the values are on the same level and \(\mathcal{N}(\mu, \sigma^2)\) denotes the Gaussian distribution with mean \(\mu\) and variance \(\sigma^2\). The values of \(U_{\alpha}^{(s)}\) are expected to be between \(\Omega - C\sqrt{\Omega}\) and \(\Omega + C\sqrt{\Omega}\) with probability equal to \(\text{erf}(C/\sqrt{2})\), which is \(\approx 0.997\) if \(C = 3\). Therefore,
\[ p \left( U_{\alpha}^{(s)} \notin [\Omega - C\sqrt{\Omega}, \Omega + C\sqrt{\Omega}] \right) = 1 - \text{erf}(C/\sqrt{2}) = \text{erfc}(C/\sqrt{2}). \quad (4.78) \]
4.3. Multiresolution Stochastic Reaction-Diffusion Systems using AMR

To illustrate the method, the one-dimensional case will be considered as shown in Figure 4.8. Following Bell et al. [59], a volume element \( i \) is refined if

\[
|U^{(s)}_{\beta} - U^{(s)}_{\alpha}| > 2C\sqrt{\Omega},
\]

which will ensure that there are indeed gradients and not fluctuations. In this work, a specific species was monitored although this need not be the case. The value of \( C \) dictates how sensitive the criterion is to gradients. A low value of \( C \) would result in refinement in many regions, even with small or nonexistent gradients, and thus decrease the overall error of the diffusion process. Therefore, the value of \( C \) is related to the discretization error, where \( C = 3 \) has been used throughout this work. In terms of the errors defined in equation (4.53), \( C_3 \sim C \). In two dimensions, the maximum of the gradients will be used, i.e. a volume element \( i \) is refined if

\[
\max \left( |U^{(s)}_{\beta} - U^{(s)}_{\alpha}|, |U^{(s)}_{\delta} - U^{(s)}_{\gamma}| \right) > 2C\sqrt{\Omega}.
\]

It should be noted that problems in which small scales appear owing to the curvature of the field instead of the gradient, this criterion will fail to tag the proper volume elements for refinement. However, for the Fisher-Kolmogorov and Gray-Scott problems considered in this section, the gradient-based refinement criterion was adequate.
Stochastic Interpolation

The values of the new volume elements that did not exist in the previous iteration are determined by first analyzing the equilibrium case, which are known to be Poisson distributed random variables. Given that the coarse value is $U_i^{(s)}$, two (1-D), four (2-D), or eight (3-D) values, $U_1^{(s)}, \ldots$, need to be determined.

![Diagram](image)

Figure 4.9: Stochastic interpolation for refined volume elements in 1-D: The shaded region represents the volume element that is tagged for refinement. The values for the new volume elements, $\hat{U}_1^{(s)}$ and $\hat{U}_2^{(s)}$ are sampled from a binomial distribution. The weights for the distribution are determined from a strictly positive and conservative interpolation using the two neighboring values.

In 1-D, the weights that will be used for the interpolation are defined as

\[
\hat{p}_1 = \frac{m x_1 + a}{m(x_1 + x_2) + 2a}, \quad (4.81)
\]

\[
\hat{p}_2 = \frac{m x_2 + a}{m(x_1 + x_2) + 2a}, \quad (4.82)
\]

where $m$ is the slope of the line as shown in Figure 4.9. To preserve positivity and conservation, the following sampling procedure is performed: $\hat{U}_1 \sim B(U, \hat{p}_1)$ and $\hat{U}_2 = U - \hat{U}_1$. Note that $\hat{U}_1 + \hat{U}_2 = U$.

Generally, a multinomial distribution, which enforces that every volume element is Poisson distributed according to the size of the volume element, can be written as

\[
\hat{U}_i^{(s)} \sim B \left( U_i^{(s)} - \sum_{j=1}^{i-1} \hat{U}_j^{(s)}, \frac{\hat{p}_i}{1 - \sum_{j=1}^{i-1} \hat{p}_j} \right), \quad (4.83)
\]
Figure 4.10: Stochastic interpolation for refined volume elements: The shaded region represents the volume element that is tagged for refinement. The values for the new volume elements, $U_1^{(s)}, \ldots, U_4^{(s)}$, are sampled from a multinomial distribution. The weights for the distribution are determined from a strictly positive and conservative interpolation using the four neighboring values.

for $i = 1, \ldots, 4$ (2-D), where $\hat{p}_i$ are weights that can be determined by interpolation. In equilibrium, the interpolation scheme is exact, and the weights are $1/4$ (2-D) or $1/8$ (3-D). In the non-equilibrium case, interpolation was used to determine the weights. The result is a scheme that is strictly positive and conservative. In terms of the error, the scheme is $O(h^2)$ with respect to the first moment of the field. The weights in 2-D, which are determined by a tensor product of 1-D weights, are

\[
\begin{align*}
\hat{p}_1 &= Z^{-1}\phi_1\phi_2, \\
\hat{p}_2 &= Z^{-1}\phi_1\phi_4, \\
\hat{p}_3 &= Z^{-1}\phi_3\phi_2, \\
\hat{p}_4 &= Z^{-1}\phi_3\phi_4,
\end{align*}
\]

where $Z = 64(U_{\alpha}^{(s)} + U_{\beta}^{(s)})(U_{\gamma}^{(s)} + U_{\delta}^{(s)})$, $\phi_1 = (5U_{\alpha}^{(s)} + 3U_{\beta}^{(s)})$, $\phi_2 = (5U_{\gamma}^{(s)} + 3U_{\delta}^{(s)})$, $\phi_3 = (3U_{\alpha}^{(s)} + 5U_{\beta}^{(s)})$, $\phi_4 = (3U_{\gamma}^{(s)} + 5U_{\delta}^{(s)})$, and $U_{\alpha}^{(s)}, \ldots, U_{\delta}^{(s)}$ are the neighboring volume elements as shown in Figure 4.10.
Section 4.3.7 Numerical Results

Validation

The numerical method is validated by verifying that (i) equations (4.59), (4.68), and (4.70) converge to the solution of the heat equation and (ii) the properties defined in equations (4.60), (4.61), (4.72), and (4.73) - by virtue of equations (4.59), (4.68), and (4.70) - reproduce the distributions associated with a random walk.

Figure 4.11: Relative $L_2$ error versus the maximum volume element length for simulations of the diffusion equation. Uniform mesh, ■, and a composite mesh that contained a coarse and fine mesh, ∙.

Equations (4.59), (4.68), and (4.70) govern the stochastic diffusion process, and can be integrated using the aforementioned stochastic simulation algorithms (equations (4.49)-(5.7)). In the limit of an infinite number of molecules, it can be shown that stochastic simulation algorithms tend to the deterministic reaction rate equations [9, 31]. Invoking this limit will facilitate the analysis of convergence with respect to equations (4.59), (4.68), and (4.70).

A convergence analysis of equations (4.59), (4.68), and (4.70) was performed using the following function for the initial condition: $u_{init}(x, y) = \sin(\pi x) \sin(\pi y)$, $x, y \in [0, 1] \times [0, 1]$ for a uniform mesh and a composite mesh that included a refined region in the center of the domain, i.e. in $[0.3, 0.7] \times [0.3, 0.7]$. Simulations were performed with the explicit Euler method in $t \in [0, 5 \cdot 10^{-2}]$ using homogeneous Dirichlet boundary conditions such that 60% of the initial function decayed. The time-step was chosen in accordance with the stability restriction of the explicit Euler method, namely

$$\Delta t = \frac{F_o C_3 h_{\min}^2}{\max_s (D^{(s)})} \quad (4.88)$$

where $C_3 < 1$, $D^{(s)} = 1$ here, and $F_o = 1/(2d)$ is the so-called Fourier number, where
4.3. Multiresolution Stochastic Reaction-Diffusion Systems using AMR

Figure 4.12: Relative $L_2$ error versus the maximum volume element length for simulations of the diffusion equation. Uniform mesh, - - , and a composite mesh that contained a coarse and fine mesh, - - - .

$d$ is the dimension of the problem. The field for the composite mesh is shown in Figure 4.11. Figure 4.12 reveals that the composite mesh had a lower relative $L_2$ error, and, although the local errors of equations (4.68) and (4.70) are formally first-order, global second-order convergence is achieved. This phenomenon is indicative of *supraconvergence*, which is an increase in the rate of convergence of finite volume approximations based on conservation laws [28, 80].

To test whether the interpolation method would also be correct, another convergence study was performed where the mesh Fourier number $F_o$ was varied. The interpolation method accrued larger errors as more time-steps were performed, i.e. when a smaller $F_o$ was used, as shown in Figure 4.13. We speculate that this is because the interpolated value has an associated error of $O(h^2)$, which is then divided by another $O(h^2)$ term when using the differencing scheme. The local error at that point is therefore an unacceptable $O(1)$. Thus, the method was deemed unsuitable.

Another example was performed for validation, namely $u(x, y, t = 0) = \sin(2\pi x) \cos(2\pi y)$, where $(x, y) \in D$ with periodic boundary conditions, the results of which are shown in Figure 4.14.

A run in 3-D was also used to check the convergence of the method. The following initial and boundary conditions were used:
Figure 4.13: $L_2$ error versus the maximum volume element length for simulations of the diffusion equation. Uniform mesh, $\square$, and a composite mesh that contained a coarse and fine mesh using cell averaging $\triangle$, and using interpolation $\bullet$.

Figure 4.14: Top, snapshot of solution; Left, point-wise error in space using a uniform mesh; Right, point-wise error using a multiresolution mesh and cell averaging for diffusion.
4.3. Multiresolution Stochastic Reaction-Diffusion Systems using AMR

Figure 4.15: Error versus the maximum volume element length for simulations of the diffusion equation. Uniform mesh, −■−, and a composite mesh that contained a coarse and fine mesh, −→. Right, snapshot of the simulation.

\[ u(x, y, z, t) = e^{-3\pi^2 t} \sin(\pi x) \sin(\pi y) \sin(\pi z), \quad (x, y, z) \in \mathcal{D} \]  \hspace{1cm} (4.89)

\[ u(x, y, z, t) = 0, \quad (x, y, z) \in \partial \mathcal{D} \]  \hspace{1cm} (4.90)

Shown in Figure 4.15 is a convergence plot of the method along with a snapshot of the solution in 3-D.

The values \( U_i^{(s)} \) in equilibrium are, theoretically, Poisson distributed, videlicet [61]: \( U_i^{(s)} \sim \mathcal{P}(\Omega_i) \), where \( \Omega_i \triangleq Q \delta V_i \) such that \( Q \) is a constant that represents the total number of molecules in a volume element at the coarsest level and \( \delta V_i \) is the relative size of the volume element \( i \). Stochastic diffusion was simulated in equilibrium with \( Q = 960 \) using a 3-level composite mesh that consisted of one mesh per level. Shown in Figure 4.16 are discrete probability density functions for three cells that were located in the bulk, at the interface, and at a corner of each of the two refined meshes. A slight departure from the correct distribution can be seen for the cells that lie on the interface between two different sized meshes as suggested by the aforementioned error analysis.
Figure 4.16: Discrete probability density functions for stochastic diffusion in equilibrium for a 3-level composite mesh: level 3 (left) and level 2 (right) are shown using a cell from the bulk ▲, from the interface •, and from a corner ■ of each of the two refined meshes. The black lines correspond to the Gaussian distribution.

Figure 4.17: Diffusive fluctuations at all levels.
4.3. Multiresolution Stochastic Reaction-Diffusion Systems using AMR

Numerical Examples

The Fisher–Kolmogorov equation, which was proposed as a model for the spatial propagation of an advantageous gene in a population, is [33]

\[
\frac{\partial u^{(1)}}{\partial t} - D^{(1)} \Delta u^{(1)} = \lambda u^{(1)} u^{(2)},
\]

where \( u^{(s)} = u^{(s)}(x, t). \) In equation (4.92), the conservation relation \( u^{(1)} + u^{(2)} = 1 \) has been used to eliminate \( u^{(2)}. \) An analytical solution can be derived in 1-dimension [44]:

\[
u^{(1)}(z) = \frac{1}{(1 + ae^{bz})^2},
\]

where \( z = x - ct, a = \sqrt{2} - 1, b = 80 \sqrt{2/3}, \) and the wavespeed \( c = \frac{\lambda}{\sqrt{\nu}} \sqrt{D^{(1)} \lambda} = 1/(32 \sqrt{6}). \) The discrete model takes the form

\[
U_i^{(1)} + U_i^{(2)} \xrightarrow{\lambda} 2U_i^{(1)},
\]

where \( U_i^{(1)} \) and \( U_i^{(2)} \) are both diffusive species with the same diffusion coefficient \( D^{(1)}. \) This system was simulated stochastically using the \( \tau \)-Leaping [14] algorithm in a unit square using the present adaptive multiresolution method with \( D^{(1)} = 1/160^2, \lambda = 1, h_{\text{min}} = 1/800, h_{\text{max}} = 1/100, C = 3, \) and \( \tau_R/\tau_D = 25 \) using the following initial condition:

\[
U_i^{(1)} = \left[ \frac{\Omega_i}{(1 + a e^{k||\mathbf{x} - \mathbf{c}|| - r)^2}} \right],
\]

where \( \mathbf{x} \) is the center of volume element \( i, \mathbf{c} = (1/2, 1/2)^T, r = 1/5, [\ldots] \) denotes the rounding operator, \( Q = 960, \) and again \( \Omega_i = Q \delta V_i, \) which means that the number of molecules at the finest level is on the order of 15. \( U_i^{(2)} \) is defined as the complement of \( U_i^{(1)} \) such that \( U_i^{(1)} + U_i^{(2)} = \Omega_i. \) A simulation was performed in \( t \in [0, 20], \) the results of which are shown in Figures 4.18 and 4.19. On the right panel in Figure 4.18, a disk is shown that expands with a speed of \( c \) per unit time. The numerical wavespeed for the 2-D problem is 0.9436c as shown in Figure 4.20(a). Additionally, the position of the front with respect to the angle at time \( t = 6 \) is shown in Figure 4.20(b). The contour value 1/5 was used for both panels of Figure 4.20.

Implementation

The code has been written in C++ and made use of the AMR methods and data-structures in the Overture library [81,82]. The time-integration method that was used was \( \tau \)-Leaping.
Figure 4.18: Stochastic simulation of the Fisher-Kolmogorov propagating front at various times. The left panel shows the concentration of $U_i^{(1)} \in [0.45, 0.55]$ as $\bullet$, and the right panel depicts the adaptive multiresolution mesh and a radial projection the r-D solution for reference (white circle).
Figure 4.19: Enlargement of a region of the domain for the concentration of $U^{(1)}_4$ at time $t = 6.0$ for a stochastic simulation of the Fisher-Kolmogorov front.

Figure 4.20: Analysis of simulation of the Fisher-Kolmogorov propagating front. The left panel (a) shows the concentration of mean radius of the propagating front over time, $r(t)$, where $-$ is the numerical value averaged over all angles $\theta$ and the black line is the 1-D analytically-derived wavespeed. The right panel (b) depicts values with respect to $\theta$ at time $t = 6$ as $\bullet$, where the black line is the mean value.

[14] in order to have explicit control over the time-step. Specifically, the time-step is chosen such that it obeys the stability condition imposed by the diffusion process, namely equation (4.88). This was done since the error control parameter, $\epsilon$, does not necessarily enforce stability. Additionally, since the diffusion propensities should be larger than the reactions propensities due to local homogeneity (see Section 4.3.5), it is reasonable to impose a bound
Chapter 4. Accelerated Stochastic Simulations in Space

Figure 4.21: Efficiency of Fisher-Kolmogorov and Gray Scott simulations. The value $E(t')$, which is a measure of computational efficiency, is shown for the Fisher-Kolmogorov and Gray-Scott problems. The time axis has been rescaled to be in the interval $t' \in [0, 1]$ for both problems.

on the diffusion process. It should be noted that the methods developed here are independent of the time-integration scheme and therefore can be combined with other stochastic simulation algorithms [68].

The running time for a $d$ dimensional spatially-dependent simulation is $O(L)$, where $L = h^{-d}$, $h = 1/N_e$, and $N_e$ is the number of elements per dimension. In other words, $L$ is the total number of volume elements in the simulation. The objective of using adaptive mesh refinement can be seen as either (i) reducing the error of a simulation by effectively introducing more volume elements into regions of the domain where the error is expected to be the largest (e.g. regions where the gradients are large) whilst keeping the running time essentially constant or (ii) minimizing the number of volume elements so as to reduce the running time of the simulation whilst keeping the error essentially constant. Assuming that the objective is (ii), a computational efficiency variable can be defined as

$$E(t) = \frac{L_{\text{amr}}(t)}{L_{h_{\text{min}}}}; \quad (4.96)$$

where $L_{\text{amr}}(t)$ is the number of volume elements in the simulation at time $t$ and $L_{h_{\text{min}}}$ is the number of volume elements under the assumption that $h_{\text{min}}$ is used throughout the domain. The values of $L_{h_{\text{min}}}$ are $800^2$ and $400^2$ for the Fisher-Kolmogorov (Section 4.3.7) and Gray-Scott (Section 7.3) problems, respectively. A value of $E(t) = 1$ describes the situation
in which all of the volume elements are refined to the finest level. Figure 4.21 shows the efficiency of the two model problems considered in this work.

4.3.8 Conclusion

A spatially adaptive simulation method for the reaction-diffusion master equation has been presented. The diffusion rates on locally refined meshes have been derived and validated. Furthermore, refinement and coarsening criteria and a stochastic interpolation scheme have been proposed for stochastic simulations based on adaptive mesh refinement. The method was shown to be useful for problems that exhibit a localization of spatial scales such as in wave propagation and pattern formation processes.
Chapter 5

Accelerated Stochastic Simulations in Time

5.1 Processes with Delays

5.1.1 Introduction

Several reactions in eukaryotic cells are not instantaneous, but rather their reactants are subject to transcription, processing, and synthesis before they can react with other chemical species [83, 84]. These biochemical processes can be modeled via a system of reactions with delays. Delayed reactions can also be used as models of spatially dependent stochastic processes [85] when proteins may need to diffuse to a distant compartment in the cell in order to react with other proteins.

These reactions can be formulated as a continuous-time Markov process with a discrete set of states that can be expressed by the so-called Master Equation. (2.59) (ME) [9, 69]. Exact realizations of the ME can be obtained via the Stochastic Simulation Algorithm. [12, 13, 48] (SSA, see Section 3.6). The connections between SSA and Molecular Dynamics, as well as the classical Langevin, Fokker-Planck equation (2.71), and Reaction-Rate equations, have been recently reviewed in [68]. The SSA is exact as it independently simulates all reaction events but it can be computationally expensive for large systems.

In order to accelerate the SSA, several approximate algorithms have been proposed. These algorithms accelerate the SSA by either prescribing a larger time-step [14, 16, 85–87] or the number of reactions per time-step [15]. Recently, there has been interest in extending the SSA to incorporate delays. Delays in the stochastic process render it, by definition, non-Markovian, and suitable modifications to the SSA are necessary in order to produce the
correct dynamics [88–90]. Cai in [88] and Anderson in [90] have proposed exact, delayed SSAs and, additionally, Leier et al. in [91] have developed a delayed, accelerated, approximate, SSA (DAA-SSA).

In this chapter, a time-adaptive generalization of DAA-SSA is proposed (D-Leaping). D-Leaping is shown to converge to the Delay Differential Equation (DDE) and preserve the correct statistical fluctuations. Furthermore, the algorithm is adaptive in time, and is shown to be, for certain chemical reactions, orders of magnitude faster than the algorithm presented in [91]. The D-Leaping algorithm can be combined with R-Leaping [15] and τ-Leaping [14] as described in this work.

5.1.2 Continuum Chemical Kinetics and Delay Differential Equations

Continuum models of chemical kinetics systems with large numbers of molecules describe the evolution of concentrations with respect to time, instead of the evolution of discrete species. The concentration is defined as \( \chi_i(t) := X_i(t)/\Omega \), where \( X_i(t) \) is the number of molecules of species \( i \) in the volume \( \Omega \) at time \( t \). Under the assumption of a well stirred chemical system in thermal equilibrium, the evolution of concentrations can be expressed as

\[
\frac{d\chi_i(t)}{dt} = f_i(t, \chi_1(t), \ldots, \chi_N(t)), \quad i = 1, \ldots, N,
\]

where the functions \( f_i \) depend on the reactants of the chemical reaction, and \( N \) is the number of different species.

When the functions \( f_i \) depend not only on the current concentrations of the system, but also on concentrations in previous times, the time-delay differential equation for \( \chi_i(t) \) can be written as

\[
\frac{d\chi_i(t)}{dt} = f_i(t, \chi_1(t), \chi_1(t - \tau_{d,1}), \ldots, \chi_N(t), \chi_N(t - \tau_{d,N})), \quad \tau_{d,i} > 0, \quad i = 1, \ldots, N,
\]

where \( \tau_{d,i} \), for \( i = 1, \ldots, N \) are the delay times that, without loss of generality, are assumed to be constant throughout this chapter.

5.1.3 SSA and Accelerated, Approximate Algorithms

As in Subsection 3.6, we consider a system of \( N \) species that react through \( M \) reaction channels. The number of molecules of species \( i \) at time \( t \) is a random variable
5.1. Processes with Delays

Figure 5.1: A) Comparison of approximate algorithms for the treatment of delayed reactions, time on the x-axes and their corresponding queues on the right. DAA-SSA in [01], top: the delayed reactions are uniformly distributed in the interval \([t_\alpha + \tau_d, t_\alpha + \tau + \tau_\alpha]\), \(r_i \sim U(0, 1)\), for \(i = 1, \ldots, k_d\). D-Leaping, bottom: the delayed reaction is stored with the number of executions, \(k_d = 5\) (in this example), as well as the earliest possible execution time for the delayed reaction, \(q_{d,\alpha} = t_\alpha + \tau_d\), and the time-step which generated the reaction, \(\text{span}_d = \tau_\alpha\). B) Perspective when the delayed reaction ought to be executed in the current time-step \(\tau\). If \(((t + \tau - q_{d,\alpha})/\text{span}_d) < 1\), then a partial number of the \(k_d\) delayed reactions are executed, specifically \(\hat{k_d} \sim B(k_d, (t + \tau - q_{d,\alpha})/\text{span}_d)\).
\( \mathbf{X}(t) = (X_1(t), \ldots, X_i(t), \ldots, X_N(t)) \), as random molecular collisions give rise to chemical transformations described by the reaction channels \( \{R_1, \ldots, R_M\} \). A propensity function \( a_j(\mathbf{X}) \) (see equation (2.135)) and state-change vector \( \nu_j = (\nu_{1j}, \ldots, \nu_{N_j}) \) specify the dynamics of a reaction channel \( R_j \) for \( j = 1, \ldots, M \). The quantity \( a_j(\mathbf{X}) \tau \) represents the probability that a reaction of type \( R_j \) occurs in the infinitesimal time interval \([t, t+\tau)\) and \( \nu_j \) denotes the change induced on the number of molecule \( i \). The following vector is defined

\[
\Theta \triangleq \Theta(t) = (a_1(t), a_2(t), \ldots, a_M(t), \epsilon),
\]

(5.4)

where \( \epsilon \) is an error control parameter such that \( 0 \leq \epsilon \ll 1 \) [16]. All SSAs sample a time-step and the number of reactions that occurred within that time-step. Exact and accelerated SSAs can be expressed as:

\[
\tau \sim \zeta(\Theta),
\]

(5.5)

\[
k_j \sim \Psi(\Theta, \tau),
\]

(5.6)

\[
\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} k_j \nu_j,
\]

(5.7)

where \( \tau \) is the time-step, \( k_j \) are the number of reactions of type \( j \), and the distributions \( \zeta \) and \( \Psi \) vary depending on the algorithm used. In \( \tau \)-Leapin (Section 3.7.1) [14] the time-step is prescribed and the total number of reactions is a random variable; \( \zeta \) is a Delta distribution and \( \Psi \) is a Poisson distribution. Conversely in the case of \( R \)-Leapin (Section 3.7.2) [15], the total number of reactions per time-step is prescribed and the time-step is sampled from a probability distribution. More specifically, \( \zeta \) is a Gamma distribution and \( \Psi \) is a Multinomial distribution. The algorithms proceed by iterating through equations (5.4)-(5.7) until the predefined final time is reached.

### 5.1.4 Accelerated, Approximate SSA for delayed reactions

In consuming delayed reactions the reactants are instantaneously annihilated, but the products do not manifest, in reactive form, until a constant delay time \( \tau_d \) after the initial reaction. In turn for non-consuming reactions the reactants are not annihilated until the products appear [89].

Leier et al. [91] were the first to propose an accelerated scheme for SSA of delayed reactions. Their method extends the generic formulation presented in Section 3.6 by introducing a check, at every iteration, for delays. We let \( R_d \) denote a delayed reaction, and \( k_d \) the corresponding number of reactions executed at a given time-step. The algorithm proceeds by maintaining a queue structure of the delayed reactions (Figure 5.8, A). The queue is checked
at every iteration to determine whether the products of the delayed reactions ought be manifested. If this is indeed the case, the products are created and the reaction is removed from the queue. Reactions are added to the queue as a result of equation (5.6), where $k_d$ reactions are placed on the queue, which have uniformly distributed execution times in the interval $[t_x + \tau_d, t_x + \tau_d + \tau_a]$, where $t_x$ is the current time and $\tau_a$ denotes the time-step (Figure 5.8, A).

![Graph showing relative speed-up compared with the treatment of the delayed reactions in [91] for the Hest model. Logscale plot, where $\epsilon$ is varied from $10^{-4}$ to $10^{-1}$ while $\beta = 10^3$.](image)

**Figure 5.2:** Relative speed-up compared with the treatment of the delayed reactions in [91] for the Hest model. Logscale plot, where $\epsilon$ is varied from $10^{-4}$ to $10^{-1}$ while $\beta = 10^3$.

### 5.1.5 D-Leaping

The error in an accelerated stochastic simulation algorithm is $C_1 \epsilon + C_2 n^{-1/2}$, where $C_1, C_2$ are constants, and $n$ is the number of samples. The terms represent the time integration and statistical errors, respectively. The algorithm presented in [91] does not treat delayed reactions in an accelerated fashion since they are queued as individual events. The reason for doing so is that we have no a priori knowledge of the time-step in the future, i.e. whether the time-step when the delayed reaction is to be executed is on the same order of magnitude as the current time-step. If the time-step, at a later point, is substantially smaller than that used when the delayed reaction was queued, the simulation of a smaller time-step, which possibly executes single delayed reactions, would be necessary to preserve $O(\epsilon)$ for the time integration error.

Therefore, single-reaction resolution would be necessary if the current time-step is smaller
Figure 5.3: Convergence with respect to the number of protein molecules for the Hest model [83] (left, entire simulation; right, region where $t \in [5, 12]$). Shown on the $y$-axis is the scaled number of protein molecules ($P \times s$), where $s = 0.03/\beta$, and on the $x$-axis time in hours ($\theta$); $\beta$ is related to the number of molecules and was varied by factors of 10 and $\epsilon$ was fixed to $5 \times 10^{-4}$; $\beta = 10^3$, ‘-.’; $\beta = 10^4$, ‘-’; $\beta = 10^5$, ‘-’. delay differential equation (DDE), ‘-’.

than that used to queue the delayed reaction. Here we propose a time-adaptive algorithm (D-Leaping) which achieves single-reaction resolution when necessary. The algorithm does not produce an error additional to the errors made by an accelerated stochastic simulation algorithm. Its computational savings are obtained by not sampling a uniform distribution for every delayed reaction. Therefore large computational savings can be procured when a substantial number of delayed reactions are queued over a time-step, i.e. when the number of molecules is large or when $\epsilon$, the error parameter in an accelerated stochastic simulation algorithm, is large.

The time and time-step ($t_\alpha, \tau_\alpha$) at which a delayed reaction would be queued is denoted here with a subscript $\alpha$, whereas no subscript indicates the time and time-step ($t, \tau$) at which a delayed reaction ought to be executed. Rather than queueing $k_d$ delayed reactions, D-Leaping queues a reaction that ought to be executed $k_d$ times, as well as the first possible execution time for the delayed reaction, $q_{d,\alpha} = t_\alpha + \tau_d$, and the time-step, here called the span, which generated the reaction, $\text{span}_d = \tau_\alpha$ (Figure 5.8, A). In other words, $q_{d,\alpha}$ is the earliest possible execution time and $q_{d,\alpha} + \text{span}_d$ is the last possible execution time of the $k_d$ delayed reactions.

Once the earliest possible execution time of a set of $k_d$ delayed reactions is in the interval of the current time-step of the simulation, namely $q_{d,\alpha} \in [t, t+\tau)$, some of the $k_d$ reactions may be executed. The essence of the D-Leaping algorithm is that only a partial number of the $k_d$ reactions will be executed if the current time-step does not encapsulate the execution times of all $k_d$ reactions, i.e. the span of the delayed reactions. If however the current time-step
does indeed encapsulate the execution times, all of $k_d$ reactions will be executed at once. The partial number of executions for the former case can be determined by considering a partitioning of the time domain.

The $k_d$ reactions in $\text{span}_d$ follow the distribution:

$$ k_d \sim \mathcal{P}(a_{d,\alpha} \text{ span}_d), $$

where $a_{d,\alpha}$ was the propensity of the delayed reaction at time $t_a$ and $\mathcal{P}(\lambda)$ is the Poisson distribution with parameter $\lambda$. If the span is partitioned into two arbitrary subintervals denoted by $\gamma_1$ and $\gamma_2$ such that $\gamma_1 + \gamma_2 = \text{span}_d$, it follows that the number of reactions in each subinterval, denoted by $\hat{k}_{\gamma_i}, i = 1, 2$ is

$$ \hat{k}_{\gamma_i} \sim \mathcal{P}(a_{d,\alpha} \gamma_i) \quad i = 1, 2 \text{ such that } \hat{k}_{\gamma_1} + \hat{k}_{\gamma_2} = k_d. $$

It can be shown that the number of reactions in the first subinterval, namely $\hat{k}_{\gamma_1}$, is

$$ \hat{k}_{\gamma_1} \sim \mathcal{B} \left( k_d, \frac{\gamma_1}{\text{span}_d} \right), $$

where $\mathcal{B}(N, p)$ represents a Binomial distribution of $N$ trials with probability $p$. The conditional distribution of $\hat{k}_{\gamma_2}$ given $\hat{k}_{\gamma_1}$ reduces to

$$ \hat{k}_{\gamma_2} \sim \mathcal{B} \left( k_d - \hat{k}_{\gamma_1}, \frac{\gamma_2}{\text{span}_d - \gamma_1} \right) $$

$$ \sim \mathcal{B} \left( k_d - \hat{k}_{\gamma_1}, \frac{\gamma_2}{\gamma_2} \right) \Rightarrow \hat{k}_{\gamma_2} = k_d - \hat{k}_{\gamma_1}. $$

The subinterval $\gamma_1$ contains the delayed reactions which will be executed in the current time-step and is defined as $\min(t + \tau - q_{d,\alpha}, \text{span}_d)$ (Figure 5.8, B). $\gamma_2$ represents the remaining time in which the delayed reactions can be executed. The queued event representing $\gamma_2$ has the updated parameters $\text{span}_d = \text{span}_d - (t + \tau - q_{d,\alpha})$, $q_{d,\alpha} = t + \tau$, and $k_d = k_d - \hat{k}_{\gamma_1}$. We note that the next time-step can subject $\gamma_2$ to further partitioning.

Specifically, two cases need to be considered in the algorithm:

Case 1, encapsulated. The time-step covers all possible execution times for all of the $k_d$ reactions, namely $((t + \tau - q_{d,\alpha})/\text{span}_d) \geq 1$. The reactions do not need to be partitioned and can simply be executed all at once.

Case 2, not encapsulated. If $((t + \tau - q_{d,\alpha})/\text{span}_d) < 1$, then the portion of $k_d$ reactions that fall in the interval $[q_{d,\alpha}, t + \tau)$ will be executed (Figure 5.8, B). Any unexecuted reactions among the potential $k_d$ reactions will remain on the queue.
The reactions are separated into disjoint sets of delayed and non-delayed reactions, which are denoted by the indices $d$ and $j$, respectively. The method can be expressed by the following pseudocode:

```
1: while $t < t_{\text{final}}$ do
2:   $\tau \sim \zeta(\Theta)$
3:   $X(t + \tau) = X(t)$
4:   for all $d$ such that $q_{d,\alpha} \in [t, t + \tau)$ do
5:     $\hat{k}_d \sim \mathcal{B}\left(k_d, \frac{\min(\tau-q_{\alpha,\text{span}_d})}{\text{span}_d}\right)$
6:     $\text{span}_d = \text{span}_d - (t + \tau - q_{d,\alpha})$
7:     $k_d = k_d - \hat{k}_d$
8:   end if
9:   if $R_d == \text{consuming}$ then
10:      $X(t + \tau) = X(t + \tau) + \sum_d \hat{k}_d \nu_d^{\text{products}}$
11:   else
12:      $X(t + \tau) = X(t + \tau) + \sum_d \hat{k}_d \nu_d$
13:   end if
14: end for
15: $k_j \cup_d \sim \Psi(\Theta, \tau)$
16: for all $d$ such that $k_d \neq 0$ do
17:    Queue.insert( $[R_d, q_{d,\alpha}, k_d, \text{span}_d] \)$
18: end for
19: $X(t + \tau) = X(t + \tau) + \sum_j k_j \nu_j$
20: for all $d$ such that $R_d == \text{consuming}$ do
21:    $X(t + \tau) = X(t + \tau) + \sum_d k_d \nu_d^{\text{reactants}}$
22: end for
23: $t = t + \tau$
24: end while
```

where $\nu_d^{\text{reactants}}$ and $\nu_d^{\text{products}}$ denote the stoichiometric changes for the reactants and products of delayed reaction $d$, respectively.
5.1.6 Numerical Results

Hes1 Biochemical Model

Monk in [83] has shown that the oscillatory expression of mRNA and protein sequences can be the result of a feedback inhibition model involving delays. The delays are believed to be a result of transcription, transcript splicing, transcript processing, and protein synthesis. The delay differential equations for the so-called Hes1 model are:

\[
\frac{dm(t)}{dt} = \alpha_m \Lambda(p(t - \tau_d)) - \mu_m m(t),
\]

\[
\frac{dp(t)}{dt} = \alpha_p H(t - \tau_d)m(t) - \mu_p p(t),
\]

\[
\Lambda(p(t)) = \frac{1}{1 + \left(\frac{p(t)}{p_0}\right)^h},
\]

where \(m(t)\) and \(p(t)\) are the mRNA and protein concentrations, respectively; \(\mu_m\) and \(\mu_p\) are the degradation rates of the mRNA and protein, respectively; \(\alpha_m\) is the rate of transcription initiation in the absence of the protein; \(\alpha_p\) is the rate at which the Hes1 protein is produced; \(H(.)\) is the Heaviside function, which here has been added to the model of Monk in order to initially inhibit the delays; \(\Lambda(.)\) is the so-called Hill function, with a Hill coefficient, \(h\); \(p_0\) is the initial value of the protein [83].

The discrete form of the delay differential equations are the following elementary reactions [89]:

\[
\emptyset \xrightarrow{\beta \alpha_m \Lambda(P(t-\tau))} M \] (5.15)

\[
M \xrightarrow{\mu_m} \emptyset \] (5.16)

\[
\emptyset \xrightarrow{\alpha_p M(t)} P \] (5.17)

\[
P \xrightarrow{\mu_p} \emptyset \] (5.18)

where \(M\) and \(P\) denote the mRNA and protein molecules, respectively, and \(\beta\) is a scaling parameter. The delayed reaction is represented by equation (5.15). The initial condition is defined as, \(X(0) = (M(0), P(0)) = (3\beta, 100\beta)\). The following parameters were used in the simulations:

\[
(\alpha_m, \alpha_p, \mu_m, \mu_p, h, \tau_d) = \\
(1 \text{ min}^{-1}, 1 \text{ min}^{-1}, 0.029 \text{ min}^{-1}, 0.031 \text{ min}^{-1}, 19.7 \text{ min}),
\]

and the simulation time was \(t \in [0, 12 \text{ hours}]\).
Figure 5.4: Comparison of $D$-Leaping with the exact delayed SSA [88]. Shown on the $y$-axis is the scaled number of protein molecules ($P \times s$), where $s = 0.03/\beta$, and on the $x$-axis time in hours ($t$); The mean over $10^3$ runs ($D$-Leaping, ‘$\cdot$’, delay SSA, ‘$\cdot$’$\cdot$’) and ($\pm$) standard deviation ($D$-Leaping, ‘$\cdot$’, delay SSA, ‘$\cdot$’$\cdot$’) are shown for both methods, where $\beta = 10$, $\epsilon = 0.01$ was used for the $D$-Leaping method.

Figure 5.5: $\tau$ versus the iteration number for a single run of the Hes1 model for the $R$-Leaping, ‘$\cdot$’, [15] and $\tau$-Leaping, ‘$\cdot$’, [14] methods. The parameters $\epsilon$ and $\beta$ were fixed to $5 \times 10^{-2}$ and $10^3$, respectively. $\tau$ in the $R$-Leaping algorithm is a random variable, thereby producing a distribution in the $y$-axis, whereas $\tau$ is prescribed in $\tau$-Leaping.
5.1. Processes with Delays

Figure 5.6: Synchronous and asynchronous oscillations of the hert proteins. Shown on the $y$-axis is the scaled number of protein molecules ($P_{\text{hert}} \times s$), where $s = 1/\beta$, and time ($t$) on the $x$-axis; ‘$-$’ and ‘$-$’ denote the number of molecules in each of the two cells. (left) Using the delays times specified in $\tau^d$ results in synchronous oscillations, whereas (right) $\neq^d$ results in asynchronous oscillations.

Performance

Figure 5.2 shows the speed-up in running times compared with the treatment of the delayed reactions proposed in [91]. Simulations were performed by leaving $\beta$ invariant and varying $\epsilon$ (Figure 5.2). The results indicate a substantial speed-up because the delayed reactions need not always be distributed, and in the case that they are, efficient use of the binomial distribution is employed instead of a uniform distribution. Specifically, large computational savings can be seen if the number of molecules ($\beta$) is large, since, in effect, the number of delayed reactions that would be executed is also large.

Validation

The accuracy of an approximate stochastic simulation algorithm can be evaluated by checking its convergence to a differential equation as the number of molecules becomes larger, and by its statistical correctness with respect to the exact SSA in the case of a small number of molecules.

In order to determine if the proposed algorithm converges to the delay differential equation, the parameter $\beta$ was increased by factors of 10. The resulting simulations were compared to a numerical integration of equations (5.12)-(5.13) using the DDE23 routine in Matlab 7.5.0 (R2007b). Figure 5.3 shows the convergence of the protein concentration for three different simulations, where $\beta = 10^2, 10^3, 10^4$, and with an invariant $\epsilon = 0.005$ that bound the relative change for each species [16] using the $R\text{-Leaping}$ [15] and $\tau\text{-Leaping}$ [14] methods. As expected, the process converged to the differential equation as the number of molecules
was increased [9].

Simulations were also performed by prescribing $\beta = 10$ and comparing the $D$-Leaping algorithm to the exact, delayed SSA [88]. Figure 5.4 shows the mean of $10^5$ runs for both methods as well as the standard deviation. The similarity reveals that the accelerated, adaptive method preserves the correct statistical dynamics of the system.

Furthermore, $\tau$ was plotted against the iteration number for the $R$-Leaping and $\tau$-Leaping methods. Figure 5.5 shows that the time-steps, and in effect the spans of the delayed reactions vary during the simulation. It should be noted that, since the $\tau$-Leaping method does not sample the time-step, it does not produce a distribution in the $y$-axis as opposed to $R$-Leaping.

**Delta Notch 2-Cell Pathway**

The Notch signalling pathway has been proposed as a mechanism by which oscillating gene expression of somatic cells occurs during embryonic development (for example in zebrafish [92]). The hert and here genes encode for gene regulatory proteins, which are, in turn, regulated by another protein called the delta Notch protein. A 2-cell model proposed by Lewis in [92] was simulated using $D$-Leaping. The system consists 3 mRNA and 3 protein species, denoted by the index $i$, and 2 adjacent cells, denoted by $j$. The indices $i = 1, 2, 3$ correspond to hert, here, and delta, respectively. In total, there are 12 species subject to 24 reactions.

The discrete model for $i = 1, 2, 3$ and $l = 1, 2$ is as follows:

\[
\begin{align*}
\emptyset & \xrightarrow{k \rho \mathbf{P}^d(t_{i,l})} M_{i,l} & (5.20) \\
M_{i,l} & \xrightarrow{c} \emptyset & (5.21) \\
\emptyset & \xrightarrow{a_{M_l(t - \tau_{i,l}^d)}} P_{i,l} & (5.22) \\
P_{i,l} & \xrightarrow{b} \emptyset & (5.23)
\end{align*}
\]

where $\tau_{i,l}^d$ denote delay times, and the mRNA species are denoted by

\[
\mathbf{M}(t) = (M_{i,l}(t)) ,
\]

and the protein species are

\[
\mathbf{P}(t) = (P_{i,l}(t)) .
\]

and the species evaluated at the originating times are

\[
\mathbf{P}^d(t) = (P_{i,l}(t - \tau_{i,l}^d)) .
\]
5.1. Processes with Delays

The function $\rho$ is defined as

$$
\rho(p^d(t), i, l) = \gamma_i + \gamma_i^2 \frac{\phi_{3,i}}{1 + \phi_{3,i}} + \gamma_{1,i} \frac{1}{1 + \phi_{1,i} \phi_{2,i}} + \gamma_{4,i} \frac{1}{1 + \phi_{3,i} \phi_{3,i} \phi_{2,i}},
$$

(5.24)

where $\phi_{i,l} = \frac{P_{i,l}^d}{P_{i,\text{critical}}}$, where $P_{i,\text{critical}}$ is constant, $l$ is the current cell, and $\hat{l}$ is the adjacent cell.

Initially $M(0) = P(0) = 0$, $k = 33(2\beta)$, $a = 4.5$, $b = c = 0.23$, $P_{i,\text{critical}} = 2\beta \cdot (40, 40, 1000)$, where $\beta = 5$ is a scaling parameter and

$$
\gamma = (\gamma_{i,l}) = \begin{pmatrix}
0.01 & 0 & 0.99 \\
0.01 & 0 & 0.99 \\
0 & 0 & 1 & 0
\end{pmatrix},
$$

and

$$
\tau^d = (\tau_{i,l}^d) = \begin{pmatrix}
11.4 & 12.6 \\
6.745 & 7.455 \\
16 & 20.5
\end{pmatrix},
$$

Additionally, another set of delay times are defined:

$$
\tilde{\tau}^d = (\tilde{\tau}_{i,l}^d) = \begin{pmatrix}
11.4 & 12.6 \\
6.745 & 7.455 \\
16 & 40.5
\end{pmatrix}.
$$

Figure 5.6 shows a simulation using D-Leaping for $t \in [0, 2000]$. Depending on the delay times, the system can produce synchronous or asynchronous oscillations, as shown in [92]. When using the delay times defined by $\tau^d$, the system undergoes synchronous oscillations (the her1 protein for both cells is shown in Figure 5.6). Asynchronous oscillations are the result of using $\tilde{\tau}^d$, where the delta Notch species has a longer delay time. The results indicate D-Leaping reproduces the correct dynamics of the system.

5.1.7 Conclusion

We presented D-Leaping, a time-adaptive method for the accelerated simulation of stochastic processes with delays. The method was shown to be as fast as existing works for certain benchmark problems and in certain cases orders of magnitude computationally more efficient. Furthermore, D-Leaping does not introduce any additional errors in the accelerated stochastic simulation when incorporating delayed reactions.
Chapter 5. Accelerated Stochastic Simulations in Time

The accuracy of the method was verified by numerically showing convergence to the delay differential equation as the number of molecules in the system increased and by comparing the fluctuations in the system to those produced by the exact, delayed SSA. The algorithm was also shown to be capable of simulating complex systems of delayed reactions, i.e. the delta Notch signalling pathway. Additionally, the method is general in that it can be combined with different accelerated stochastic simulation algorithms.
5.2 Flow-Averaging Algorithms

5.2.1 Introduction

Stiff equations are defined as processes that have different time-scales. In order to resolve a fast process, an explicit numerical method is forced to take prohibitively small time-steps. The eigenvalues of a system of equations are generally used to measure the amount of stiffness present. For example, consider the set of chemical reactions:

\[ S_1 \xleftarrow{1/\epsilon} S_2 \xrightarrow{1} S_3. \tag{5.25} \]

Under the assumption that the concentration of the species \( S_3 \) is 1, the system can be described by the following system of equations:

\[ \frac{d}{dt} \mathbf{s} = \mathbf{A} \cdot \mathbf{s}, \tag{5.26} \]

where \( \mathbf{s} = (s_1, s_2)^T \) and

\[ \mathbf{A} = \begin{pmatrix} \frac{1}{\epsilon} & \frac{1}{\epsilon} \\ \frac{1}{\epsilon} & -1 - \frac{1}{\epsilon} \end{pmatrix}, \tag{5.27} \]

the eigenvalues of which are

\[ \lambda_i = \frac{-2 - \epsilon \pm \sqrt{1 + \epsilon^2}}{2\epsilon}. \tag{5.28} \]

Stiffness is formally characterized by the one of the following indices:

\[ L = \max |\text{Re} (\lambda_i)|, \tag{5.29} \]
\[ S = \frac{\max |\text{Re} (\lambda_i)|}{\min |\text{Re} (\lambda_i)|}. \tag{5.30} \]

Shown in Figure 5.7 are the eigenvalues with respect to \( \epsilon \). It is clear that as \( \epsilon \) becomes smaller, one of the eigenvalues becomes increasingly large and thereby renders the problem stiff, irrespective of the index used (namely equations (5.29) and (5.30)).

In Subsection 5.2.2 we review a recently proposed numerical method for the integration of stiff ordinary differential equations, and in Subsection 5.2.3 we present our method for stiff Markov processes.
5.2.2 Flow Averaging Integration for Ordinary Differential Equations

This section recapitulates the FLow AVeraging integratORS (called FLAVORS) presented by Tao, et al. in [17]. Consider the following stiff system of ordinary differential equations

\[ \dot{u} = G(u) + \frac{1}{\epsilon} F(u), \]

where \( \epsilon \ll 1 \). Let \( \bar{u}_t \) denote the numerical solution at time \( t \). An explicit numerical scheme is generally of the form

\[ \bar{u}_{t+\tau} = \Phi_{1/\epsilon}^\tau(\bar{u}_t), \]

where \( \Phi_{1/\epsilon}^\tau \) denotes an arbitrary legacy integrator with time-step \( \tau \) and with the parameter \( 1/\epsilon \). Because of numerical stability and accuracy, we will choose \( \tau \) such that \( \tau \ll \epsilon \). The flow averaging integration scheme is defined as the composition of two legacy integration schemes:

\[ \bar{u}_{t+h} = \left( \Phi_{1/\epsilon}^{h_{\text{st}} - \tau} \circ \Phi_{1/\epsilon}^{\tau} \right)^M(\bar{u}_t), \]

where \( M \) is the ‘number of samples’ used to average the flow. Note that in \( \Phi_{1/\epsilon}^{h_{\text{st}} - \tau} \), the parameter \( 1/\epsilon = 0 \), i.e. the function \( F(u) \) is disregarded. The numerical algorithm proceeds by alternating between two time-steps, \( \tau \) and \( \delta - \tau \) (see Figure 5.8), where \( \delta = \frac{h_{\text{st}}}{M} \) and \( 0 < \tau \ll \epsilon \ll \delta \ll 1 \). The following two conditions are needed on \( \delta \):
Figure 5.8: Schematic of the flow averaging algorithm: the algorithm proceeds by alternating between the two time-steps and switching the parameter $\alpha$ on ($\alpha = 1/\epsilon$) and off ($\alpha = 0$). (Image from [17])

$$\left(\frac{\tau}{\epsilon}\right)^2 \ll \delta$$

which is the control of the error on the slow dynamics induced by the inaccuracy of the numerical integration scheme, and

$$\delta \ll \frac{\tau}{\epsilon}$$

which is needed for the averaging of the hidden slow dynamics with respect to the hidden fast variables.

**Numerical Example: Van Der Pol Oscillator**

Consider van der Pol's oscillator, which is governed by the following equations:

$$\frac{dx}{dt} = \frac{1}{\epsilon} \left( x + y - \frac{1}{3} x^3 \right)$$

$$\frac{dy}{dt} = -\epsilon x$$

where the term $1/\epsilon$ will be turned on and off. Figure 5.9 shows the FLAVOR algorithm compared to the explicit Euler method. It should be noted that the FLAVOR algorithm was about $10^2$ times faster than the explicit euler method.

### 5.2.3 Flow Averaging Integration for Markov Processes

**Introduction**

A straightforward numerical method will be presented for the simulation of stiff, discrete-space, continuous-time Markov processes. The method is in the spirit of *Flow Averaging for*
the integration of stiff ordinary and stochastic differential equations. The algorithm itself is ultimately an effortless modification to the well-known \textit{Stochastic Simulation Algorithm} (see Section 3.6). Furthermore, a cutoff phenomenon with respect to the error of the numerical method is shown, which is used to determine the optimal compromise between error and speed-up. Two numerical examples from chemical kinetics are also provided to illustrate the feasibility and efficiency of the method.

The stochastic simulation algorithm is an inefficient numerical method if the Markov process is stiff. Stiffness in these systems arises from either disparate reaction rates or large variations in the molecular populations. Recently, many algorithms have been proposed to simulate such processes. Some of the methods that have been developed rely upon the assumption that the fast processes are in so-called \textit{quasi-equilibrium}, namely the work by Cao et al. [93–95]. Alternatively, Peles et al. [96] have formulated an efficient algorithm for the direct solution of the master equation with multiple time-scales. In a separate line of work, Weinan et al. [97, 98] have used singular perturbation and averaging theory to accelerate stochastic simulations. Although the method presented here relies upon the same averaging principle, it is formulated differently and results in a simple and noninvasive algorithm. The objective of this subsection is to provide a preliminary description of a novel method for the simulation of stiff discrete-space, continuous-time Markov processes. It shall be shown

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5_9.png}
\caption{Explicit Euler (solid lines) and FLAVOR (dashed lines) methods.}
\end{figure}
that ultimately the algorithm that is nothing more than a rescaling of the fast propensities at every other time-step.

**Stochastic Simulation Algorithms**

Under the assumption that there is a clear separation of fast and slow processes, the total propensity can be decomposed as follows:

\[ a_0(t) = \frac{1}{\epsilon} \sum_i a_i^{(fast)}(t) + \sum_j a_j^{(slow)}(t), \]

where \( a_i^{(fast)}(t) = \frac{1}{\epsilon} a_i^{(fast)}(t) \) and \( \epsilon \ll 1 \) is the stiffness parameter that is either specified by the problem (a fast reaction rate) or emerges during the evolution of the system (disparities in the propensities). It should be noted that this dimensionless parameter has been introduced only for the clarity of the presentation and the analysis of the method and in practice it specific value is not strictly needed. The fast processes evolve on a timescale of \( \mathcal{O}(\epsilon) \) and the slow processes are on a timescale of \( \mathcal{O}(1) \). The inefficiency of SSA is incident to the fact that \( a_0 \approx \mathcal{O} \left( \bar{A}^{(fast)} \epsilon^{-1} \right) \), where \( \bar{A}^{(fast)} = \sum_i a_i^{(fast)}(t) \) has units \([1/time]\) and is \( \mathcal{O}(1) \) and consequently \( \mathbb{E}[\tau] \approx \mathcal{O}(\epsilon) \).

Assume that the propensities are in ascending order and indexed by \( k \). Then, \( \epsilon \) can be determined by

\[ \epsilon = \min_k \left\{ \frac{a_{k-1}}{a_k} \right\}. \]

Partitioning of the propensities can be performed by indexing the fast propensities by \( i \geq k' \) and the slow propensities \( j < k' \), where \( k' \) is the arg min (argument of the minimum) of equation (5.39). We note that this partitioning can also be performed hierarchically for systems with multiple time-scales, although in this article such systems are not investigated. Moreover equation (5.39) assumes that the time-scales are clearly separated. For example, if \( a_k = 2a_{k-1} \) for \( k = 2, \ldots, K \), then although there is a large disparity between \( a_1 \) and \( a_K \), there is not a clear separation between fast and slow processes.

Letting \( \Phi^{(1/\epsilon)} \) denote one iteration of the SSA, the numerical method can be written as

\[ (X_n, t_n) = \left( \Phi^{(1/\epsilon)} \right)^n (X_0, t_0), \]

which is equivalent to performing \( n \) iterations of SSA.

**FLAVORized-SSA**

Tao et al. in [17] presented a class of numerical methods, called \textit{FLow AVeraging integrat-ORS} (called FLAVORS), for the integration of stiff ordinary and stochastic differential
equations. The method is based on averaging the instantaneous flow of a dynamical system with hidden fast and slow variables. The advantage of the method is that the computational cost is determined by the slow processes instead of the fast ones. In order to briefly outline FLAVOR we consider the stiff ordinary differential equation

\[
\dot{\mathbf{u}} = \mathbf{G}(\mathbf{u}) + \frac{1}{\epsilon} \mathbf{F}(\mathbf{u})
\]  

(5.41)

Assume that there exists a possibly unknown and nonlinear diffeomorphism \( \eta \) such that \((x, y) = \eta(\mathbf{u})\) is the solution of the following stiff set of ordinary differential equations:

\[
\dot{x} = g(x, y), \\
\dot{y} = \frac{1}{\epsilon} f(x, y).
\]

(5.42)

(5.43)

Let \( \phi^{(1/\epsilon)} \) be the flow of a first order accurate legacy integrator for equation (5.41), i.e. \( \mathbf{u}_{t+h} = (\phi^{(1/\epsilon)}_h)(\mathbf{u}_t) \), where \( \mathbf{u} \) is a numerical approximation of the solution of equation (5.41). FLAVOR works by averaging the flow of equation (5.41) (instead of equations (5.42)-(5.43)) by splitting and resynchronization. By splitting we refer to a composition \( (\phi^{(0)}_{\delta-\tau} \circ \phi^{(1/\epsilon)}_\tau) \), where \( \phi^{(0)}_{\delta-\tau} \) denotes the flow with the stiffness parameter \( 1/\epsilon = 0 \) and \( \delta \gg \tau \) are the time-steps. By resynchronization we refer to the distinct time-steps \( \delta \) and \( \tau \), the effects of which are to advance the internal clock of fast variables by \( \tau \) every time-step of length \( \delta \). The averaging applied \( M \) times means that \( (\phi^{(0)}_{\delta-\tau} \circ \phi^{(1/\epsilon)}_\tau)^M \approx (\phi^{(1/\epsilon)}_\tau)^L \), where \( L \tau = M \delta \). The variable \( M \) denotes the number of samples used to average the flow. The approximate scheme is therefore \( \mathbf{u}_t = (\phi^{(0)}_{\delta-\tau} \circ \phi^{(1/\epsilon)}_\tau)^M(\mathbf{u}_0) \), where \( M \delta = t \). It is shown in [17] that the proposed flow averaging is accurate (in a strong sense with respect to (possibly hidden) slow variables and in the sense of measures with respect to (possibly hidden) fast variables) provided that fast variables are locally ergodic and \((\tau)^2 \ll \delta \ll (\bar{\tau}) \). The condition \( \delta \ll (\bar{\tau}) \) ensures that the slow dynamic has been averaged with respect to that of the fast variables. The condition \((\tau)^2 \ll \delta \) ensures that the error caused by fast variables on slow ones (when \( \eta \) is nonlinear) remains small.

Analogously to the FLAVOR algorithm, the FLAVORzed-SSA method is defined as the composition of two SSA steps:

\[
(\mathbf{X}_n, t_n) = (\Phi^{(\xi/\epsilon)}(\xi^{(1/\epsilon)}) \Phi^{(\xi/\epsilon)})(\mathbf{X}_{n-1}, t_{n-1}),
\]

(5.44)

where \( \xi \in [0, 1] \). The SSA step \( \Phi^{(\xi/\epsilon)} \) advances the simulation clock by a time interval of size \( \delta \), which is an exponential random variable with parameter

\[
\tilde{a}_0(t, \xi) := \frac{\xi}{\epsilon} \sum_i a_i^{(\text{fast})}(t) + \sum_j a_j^{(\text{slow})}(t),
\]

(5.45)
It also selects a fast reaction $i$ with probability $\frac{\xi \beta^{(\text{fast})}(t)}{a_0(t)}$ or slow reaction $j$ with probability $\frac{\xi \beta^{(\text{slow})}(t)}{a_0(t)}$. This is simply equivalent to identifying fast reactions and multiplying fast propensities by $\xi$ at every other step. If $\xi = 1$, the algorithm reduces to the standard formulation of the SSA. The only difference between the FLAVORized-SSA and the SSA is that the fast (stiff) propensities are rescaled by $\xi$ at every other iteration. Intuitively, this amounts to slowing down the fast processes to be on the same time-scale as the slow processes, thus permitting a larger time-step to be taken. For $\xi/\epsilon \gg 1$, fast variables have the time to converge in law towards their equilibrium distribution before any significant change on slow variables. Conversely, if $\xi/\epsilon$ is of $O(1)$, then fast variables do not have the time to converge in law towards their equilibrium distribution before significant changes on the slow variables. Hence, speed-up and error are both increasing as a function of $1/\xi$. However, although speed-up is proportional to $1/\xi$, we will show that the error exhibits a sharp transition with respect to the value of $\xi$, i.e. a cutoff phenomenon.

**Cutoff Phenomenon**

Stiff systems, and in particular fast processes, force stochastic simulation algorithms to take prohibitively small time-steps as discussed above. These fast processes are in several occasions not as important as the slow processes in determining the overall dynamics of the system. Hence, if one were to artificially slow down the fast reactions by a factor $0 \leq \xi < 1$ then the stiffness would be alleviated and the problem of small time-steps would be mended. Naturally, an error would be introduced since one is not simulating the original system, but rather a rescaled version. It will be shown below that the error introduced by this artificial slowing down of fast processes is not linear with respect to $\xi$, but rather changes significantly when $\xi$ approaches $\epsilon$. In other words, when $\xi > \epsilon$ virtually no error is made, yet when $\xi < \epsilon$ a large error is made. This sharp transition of the error around a certain value, in this case $\xi \approx \epsilon$, can be likened to a so-called cutoff phenomenon as studied by Diaconis in [99]. Diaconis proved that fewer than six riffle shuffles of a deck of playing cards is not sufficient to bring the deck to its stationary distribution (sc. a deck with a random distribution of cards), yet - rather surprisingly - seven shuffles suffice. In the case of chemical kinetics, the fast processes (i.e. reactions) can be artificially slowed down without substantially modifying the dynamical properties of the system, the result of which is a decrease in the computational time.

Sharp convergence in the long-time behavior of Markov chains is called cutoff phenomena. Diaconis in [99] proved that fewer than six riffle shuffles of a deck of playing cards is not sufficient to bring the deck to its stationary distribution, yet - rather surprisingly - seven shuffles suffice. Sharp convergence of the present numerical method with respect to $\xi$ will be
investigated next. Since the value of $\xi$ dictates both the error and speed-up of the simulation, it is important to enquire into its optimal value. Therefore, the problem is the following: given a tolerance $\alpha > 0$, how large should $\xi$ be such that

$$\| P^{\xi}(t') - \pi \| < \alpha, \quad (5.46)$$

where

$$\pi \triangleq \lim_{t \to \infty} P(t), \quad (5.47)$$

$P(t) \triangleq P(\{x\}, t)$ is the solution of the master equation, and $P^{\xi}(t') \triangleq P^{\xi}(\{x\}, t')$ is the FLAVORized master equation with the parameter $\xi$. For simplicity, the case in which the master equation has slow processes frozen is considered, sc.:

$$\frac{dP(t)}{dt} = \frac{1}{\epsilon}\mathbb{W} \cdot P(t), \quad (5.48)$$

where the transition matrix $\mathbb{W}$ is defined as [9]

$$\mathbb{W}(y, x) \triangleq \mathbb{W}(y \rightarrow x) \triangleq a_{x,y} - \delta_{x,y} \left( \sum_{y'} a_{y',x} \right), \quad (5.49)$$

where $a_{x,y}$ is the propensity for a transition from state $y$ to $x$, $a_{x,x} = 0$, $\delta_{x,y}$ is the Kronecker delta function, and $1/\epsilon$ has been factored out of the matrix. The solution of equation (5.48) is [9]

$$P(t) = e^{(t/\epsilon)\mathbb{W}} \cdot \mu, \quad (5.50)$$

where $\mu \triangleq P(0)$. The master equation for the FLAVORized-SSA defined in equation (5.44) is

$$P^{\xi}(t') = \left( e^{(\delta - \tau)(\xi/\epsilon)\mathbb{W}} \cdot e^{(\tau/\epsilon)\mathbb{W}} \right) M_2 \cdot \mu \quad (5.51)$$

$$= e^{(t'/\epsilon)\mathbb{W}} \cdot \mu, \quad (5.52)$$

where $t' = M_2(\tau + \xi(\delta - \tau))$. Note that equations (5.50) and (5.52) differ only by the definition of time. Let $\mathbb{K} \triangleq \mathbb{W} - I$, where $I$ is the identity matrix, and also let $\gamma$ be the spectral gap of $\mathbb{K}$ ($\gamma > 0$) [100]. The error of the solution with respect to the stationary distribution $\pi$ is known to decay exponentially with respect to the $l^2(\pi)$ norm (see e.g. [100], Lemma 20.5, Section 20.3), viz.

$$\|P(t) - \pi\|_2 < \exp \left\{ -\frac{t}{\gamma} \right\}, \quad (5.53)$$

Therefore, the following can be written for equation (5.52):
5.2. Flow-Averaging Algorithms

\[ ||P_{\xi}(\tilde{t}) - \pi||_2 \]

Figure 5.10: Cutoff Phenomenon at \( \xi = \epsilon \). Equation (5.57), \( ||P_{\xi}(\tilde{t}) - \pi||_2 \) versus \( \xi/\epsilon \). The error becomes relatively small once \( \xi/\epsilon > 1 \).

\[ ||P_{\xi}(t') - \pi||_2 < \exp \left\{ \frac{-\gamma t'}{\epsilon} \right\}, \quad (5.54) \]

which yields

\[ ||P_{\xi}(t') - \pi||_2 < \exp \left\{ -\gamma M_2 \left( \tau \left( \frac{1 - \xi}{\epsilon} \right) + \delta \frac{\xi}{\epsilon} \right) \right\}. \quad (5.55) \]

Since \( \delta \gg \tau \) and because

\[ \exp \left\{ -\gamma M_2 \left( \tau \left( \frac{1 - \xi}{\epsilon} \right) + \delta \frac{\xi}{\epsilon} \right) \right\} < \exp \left\{ -\gamma \tilde{t} \frac{\xi}{\epsilon} \right\}, \quad (5.56) \]

where \( \tilde{t} \equiv M_2 \delta \), then

\[ ||P_{\xi}(\tilde{t}) - \pi||_2 < \exp \left\{ -\gamma \tilde{t} \frac{\xi}{\epsilon} \right\}. \quad (5.57) \]

Equation (5.57) reveals a cutoff of the error when \( \xi = \epsilon \) as shown in figure 5.10. To ensure a relatively low error in numerical simulations, \( \xi \) should be chosen to be larger than \( \epsilon \).
Numerical Example 1: Minimal Stiff Chemical System

Following [197], the simple stiff chemical system described by equation (5.25) is considered, where $\epsilon \ll 1$. The propensities and stoichiometric vectors for this system are:

$$a^{(\text{fast})}_1 = \frac{1}{\epsilon} X_1, \quad \nu_1 = (-1, +1)^T,$$
$$a^{(\text{fast})}_2 = \frac{1}{\epsilon} X_2, \quad \nu_2 = (+1, -1)^T,$$
$$a^{(\text{slow})}_3 = X_2, \quad \nu_3 = (0, -1)^T,$$
$$a^{(\text{slow})}_4 = \Omega, \quad \nu_4 = (0, +1)^T.$$

For simplicity, the species $S_3$ is assumed to be constant. Simulations were performed in $t \in [0, 1]$, where $X(t) = (X_1(t), X_2(t))^T$. $X_i(t)$ denotes the number of molecules or particles of species $S_i$ at time $t$, $X(0) = (50, 50)^T$, $\Omega = 100$, $N = 2 \times 10^5$, and $\epsilon = 10^{-4}$. The value of $\xi$ was changed in order to study the convergence of the simulation. Figure 5.11 shows the cumulative distributions of $P^\xi(t' = 1)$ for the species $S_1$, $S_2$. A shift in distributions is noticeable when the value of $\xi$ becomes smaller than $\epsilon$. This shift is incident to the rescaling.
5.2. Flow-Averaging Algorithms

Figure 5.12: Error Analysis: $||P^\xi(t') - P(t = 1)||$, where $l_\infty$ ‘— ’; $l_1$ ‘— ’; $l_2$ ‘ — ’; $\xi = 10^{-j}$, $j = 0, 1, \ldots, 8$. The errors have been rescaled to be in the range [0, 1] and the $x$-axis is logscale.

Figure 5.13: Computational Efficiency: speed-up compared to SSA, $S(\epsilon, \xi)$, with respect to the scaling parameter, $\xi$, and the stiffness parameter, $\epsilon$. All axes are logscale.
of the time variable (cf. equations (5.50) and (5.52)). The error analysis in figure 5.12 shows a clear cutoff around $\xi = \epsilon = 10^{-4}$. The errors have been rescaled to be in the range $[0, 1]$. The value of $\epsilon$ was then varied in order to determine the speed-up of the numerical method compared to SSA. Figure 5.13 shows the speed-up with respect to $\xi$ and $\epsilon$. Note that to ensure accuracy, the value of $\xi$ should be larger than $\epsilon$. It is therefore possible to obtain a speed-up of more than 10x if $\epsilon$ is very small. As expected, the speed-up is largest when $\xi \to 0$ and $\epsilon \to 0$.

**Numerical Example 2: Emergence of Time-Scales**

![Graph](image)

Figure 5.14: Optimal Value of $\xi$: error of the probability density function, $E_P(\xi)$, versus speed-up, $S(\xi)$. $l_\infty \leftarrow \leftarrow \leftarrow$; $l_1 \leftarrow \leftarrow \leftarrow$; $l_2 \leftarrow \leftarrow \leftarrow$. $\xi = 10^{-j}$, where $j = 0, 1, \ldots, 5$ and where $\xi = 1$ is the leftmost point and $\xi = 10^{-5}$ is the rightmost point. The maximum error has been rescaled to 1 and the error at $\xi = 1$ is the sampling error. The plot markers denote simulated points and the lines are a result of second-order interpolation between simulated points. The optimal value is $\xi = 10^{-2}$ where $S(\xi) \approx 7$ and the error is negligible.

The following chemical system, which was investigated in [97], exhibits a disparity in time-
5.2. Flow-Averaging Algorithms

scales at \( t \approx 1 \):

\[
\begin{align*}
S_1 & \rightarrow S_2, \\
S_1 & \rightarrow S_3, \\
2S_2 + S_3 & \rightarrow 3S_4.
\end{align*}
\]

(5.62)  
(5.63)  
(5.64)

This system does not have an explicit stiffness parameter \( \epsilon \), but rather the stiffness arises owing to the physics of the problem. Moreover, this chemical system is indicative of real-world problems in which the value of \( \epsilon \) is not known a priori. The reversible reaction shown in equation (5.64) at time \( t \approx 1 \) has propensity values of roughly \( 10^{3} \) times larger than the propensities of the other reactions. The value of \( \epsilon \) varies in the range of \( 10^{-3} \) and \( 10^{-4} \) when \( t > 1 \). The corresponding propensities and stoichiometric vectors are:

\[
\begin{align*}
a_1^{(\text{slow})} & = X_1, & \mathbf{\nu}_1 = (-1, +1, 0, 0)^T \\
a_2^{(\text{slow})} & = X_2, & \mathbf{\nu}_2 = (+1, -1, 0, 0)^T \\
a_3^{(\text{slow})} & = X_3, & \mathbf{\nu}_3 = (-1, 0, +1, 0)^T \\
a_4^{(\text{slow})} & = X_4, & \mathbf{\nu}_4 = (+1, 0, -1, 0)^T \\
a_5^{(\text{fast})} & = 2X_2(X_2 - 1), & \mathbf{\nu}_5 = (0, -1, -2, +3)^T \\
a_6^{(\text{fast})} & = 2X_4(X_4 - 1)(X_4 - 2), & \mathbf{\nu}_6 = (0, +1, +2, -3)^T
\end{align*}
\]

(5.65)  
(5.66)  
(5.67)  
(5.68)  
(5.69)  
(5.70)

Simulations were performed with \( \mathbf{X}(t) = (X_1(t), X_2(t), X_3(t), X_4(t))^T \) and \( \mathbf{X}(0) = (100, 3, 3, 3)^T \) in \( t \in [0, 4] \). The FLAVORized-SSA was turned on once there was a disparity in the system, namely at time \( t = 1 \). A total of \( N = 2 \times 10^5 \) samples were performed.

A parametric plot of the error of the probability density function versus the speed-up compared to SSA is shown in figure 5.14. It should be noted that the speed-up is relatively modest.
since SSA is performed in the interval $0 \leq t < 1$. The value of $\xi$ was varied in the range $\xi = 10^{-j}$, where $j = 0, 1, \ldots, 5$. The optimal value for $\xi$ in terms of the error and speed-up is $\xi = 10^{-2}$ for this particular system. This confirms the cutoff analysis above that showed that the error was minimized once $\xi > \epsilon$. Additionally, a comparison of the probability density functions for $\xi = 10^{-2}$ and $\xi = 10^{-5}$ is shown in figure 5.15. The larger error for $\xi = 10^{-5}$ is a result of the shift of the distribution.

**Conclusion**

A straightforward numerical method was presented for the simulation of stiff Markov processes. Additionally, a cutoff phenomenon with respect to the error was shown analytically and numerically. This cutoff was used to determine the optimal trade-off between error and speed-up.
Part III

Applications
Chapter 6

A Stochastic Model of Microtubule Motors

Owing to the inherent discreteness of many biological phenomena, stochastic processes have been thought to play a crucial role in many aspects of cell biology [101–104]. In this chapter, a stochastic processes is formulated for the random binding and unbinding of motors to viruses, as well as the uncertain and hither and thither movement of virus capsids after being engulfed by a cell.

6.1 Introduction

Cytoplasmic transport of organelles, nucleic acids, and proteins on microtubules is usually bidirectional with dynein and kinesin motors mediating the delivery of cargoes in the cytoplasm. Here we combine live cell microscopy, single virus tracking and trajectory segmentation to systematically identify the parameters of a stochastic computational model of cargo transport by molecular motors on microtubules. The model parameters are identified using an evolutionary optimization algorithm to minimize the Kullback-Leibler divergence between the in silico and the in vivo run length and velocity distributions of the viruses on microtubules. The present stochastic model suggests that bidirectional transport of human adenoviruses can be explained without explicit motor coordination. The model enables the prediction of the number of motors active on the viral cargo during microtubule-dependent motions as well as the number of motor binding sites, with the protein hexon as the binding site for the motors.

The function of eukaryotic cells relies on the transport of macromolecules and organelles throughout the cytoplasm. Pathogenic viruses can exploit a cell’s cytoplasmic transport
mechanisms [105, 106] in order to reach their site of replication. Cytoplasmic transport involves three types of molecular motors. Kinesin and dynein motors use microtubule tracks to move cargo throughout the cytoplasm, while myosin motors interact with actin filaments to move their cargoes [107, 108]. Microtubule-based transport is usually bidirectional and its mechanism can be explained by the exclusive binding of dynein and kinesin motors to the cargo, motor cooperation and regulation, or a stochastic tug-of-war [109–112]. Exclusive binding of motors has not been reported in cells, while in systems with cooperating motors, additional factors such as on/off switches or coordinators between motors have been postulated for bidirectional transport of large cargo, such as vesicles [110]. The mechanism of bidirectional motor transport by non-coordinated motors of opposite polarity has been the basis of tug-of-war models [110, 113].

In this work we propose a stochastic model for motor transport on microtubules and we systematically identify its parameters using virus trajectories obtained by in vivo imaging. Trajectories are obtained by live cell microscopy of fluorescently labelled human adenovirus type 2 (Ad2) using confocal microscopy. Motility information extracted through single virus tracking [114], and trajectory segmentation [115] are implemented in order to study the properties of virus transport by employing a systems identification process [116] for a stochastic model of cargo transport on microtubules.

### 6.2 A Stochastic Model for Cargo Transport

The small number of motor proteins involved in microtubule transport implies a system where the fluctuations in the behavior of motors and the randomness of molecular reactions are essential characteristics [104] suggesting a stochastic modeling of the governing processes. Here we propose a stochastic representation of the main events involved in motor transport, namely stepping along microtubules and binding and unbinding of molecular motors to the cargo.

The proposed model has six parameters, namely the binding, unbinding and stepping rates of plus-end and minus-end motors (herein presumed to be dynein and kinesin, respectively). The step sizes of the motors were set to ~8/4 nm for dynein/kinesin as suggested by the results of single molecule experiments [117, 118]. We note that we do not impose any geometrical information on the motors and their binding sites on the virus capsid. The motor protein binding sites on the adenovirus capsid are not known even though a recent cryo-EM image of the structure of the human adenovirus type 2 temperature sensitive mutant revealed the organization of the surface of the virus capsid [119].

The six model parameters are inferred through a system identification process using the
6.3 Results

velocity and displacement distributions of segmented trajectories as the cost function of our optimization. An evolutionary algorithm, capable of handling noisy cost functions, is used to obtain the rates that minimize the distance between the velocity and displacement distributions of the in silico and in vivo trajectories.

The velocity distribution in virus trajectories has led to several suggestions regarding the cooperation or lack thereof between molecular motors. High velocities, in the order of a few microns per second, were observed for intracellular viruses (Figure 6.2 E) [120]. Similar high speeds have been observed for vesicles moving along microtubules such as peroxisomes [121] and endosomes [122]. These velocities are above the maximum velocities measured for single motors without load (3μm/s for dynein, [117]; 0.4μm/s for kinesin 1, [123]; 3μm/s for kinesin 1, [124]; 0.8μm/s for kinesin-1 [125], 0.8μm/s kinesin-1 and 0.5μm/s kinesin-2, [126] in in vitro experiments. It has also been reported for drosophila lipid droplets, that multiple processive motors do not move cargoes faster [127]. It is likely that yet unknown mechanisms account for the high velocities measured in in vivo biological systems. These mechanisms may involve motors which are able to increase their velocities up to few microns per second or motors are able to act additively to achieve higher speeds. Both assumptions have not been experimentally validated or discarded in in vivo experiments. Additive behavior of motors is an underlying assumption in our model (Figure 6.2 A). The additive behavior is inherent to the Stochastic Simulation Algorithm [13] used herein to simulate the model, since the time step to the next event depends on the total propensity (numbers and event rates). The proposed stochastic model does not impose any explicit coordination between motor proteins, e.g. a switching mechanism that selects a certain motor protein type to be active.

We emphasize that our model does not aim at a mechanistic description at the motor level. Forces are known to affect motor properties, but it is not clear how they are distributed among multiple motors [128]. Furthermore while it is possible to obtain data relating forces for certain motors in vitro, there is no such data for in vivo experiments. In the present model the forces between molecular motors and cargo are implicitly taken into account through the binding/unbinding/stepping rates of the stochastic models.

6.3 Results

The simulation of the stochastic model produces cargo trajectories that depend on the parameter settings. The model contains no a-priori assumptions on the existence of either a tug-of-war or coordination between molecular motors. In turn, the model parameters are systematically identified with a derandomized evolution strategy that minimizes the dif-
Figure 6.1: Imaging, tracking and trajectory segmentations of single adenoviruses. (A) HeLa cells were infected with fluorescent adenovirus type 2 for 30 min, and imaged by spinning disc confocal fluorescence microscopy [129]. Virus tracks (black lines) recorded by a single particle tracking algorithm [114] using the nucleus (red square) as a reference point are displayed over a phase contrast image of the infected cell. (B) Two-dimensional projection of a single virus trajectory with directed motion segments in red. (C) Distance travelled along the trajectory shown in Figure 6.1 B plotted as a function of time. Reduction to 1-D is justified, since in cultured cells microtubules are largely flat and straight over distances of many micrometers [130]. Negative/positive values indicate displacements towards the cell center/periphery.

Comparison between the length and velocity distributions of directed motions (fast microtubule dependent runs [115]) produced by the model and those of fluorescently labelled human adenovirus type 2 (Ad2) as measured by confocal microscopy at 25 Hz temporal resolution. The two-dimensional virus trajectories are extracted by a single particle tracking algorithm [114] (Figure 6.1 A, B). Directed motions along microtubules are classified by trajectory segmentation [115] and the distance travelled along the microtubule is determined as a function of time (1-D trajectory shown in Figure 6.1 C). The same analysis is applied to trajectories obtained by the simulation of our model using SSA [13] (see Section 3.6). These trajectories are also subsequently segmented to classify directed motions [115]. An optimization algorithm is then used to identify the parameters of the model [131].
Figure 6.2: A stochastic model for microtubule-dependent movements of human adenovirus. The stochastic model reproduces directed motion length and velocity distributions of human adenovirus, and predicts the optimal number of either common or separate binding sites for dynein and kinesin motors on the capsid. (A) Dynein (D, blue) and kinesin (K, orange) bind to, and unbind from the capsid and transport it along a microtubule (green). Equations [1]-[6] describe the dynamics of the model: [3]/[4] dynein binding/unbinding and decrease/increase of the number \( \rho \) of the available motor binding sites on the virus capsid with \( \frac{da}{d\delta} \) binding/unbinding rates, [5]/[6] kinesin binding/unbinding and decrease/increase of the number \( \rho \) of the available motor binding sites on the virus capsid with \( k_a/k_\delta \) binding/unbinding rates, and [1]/[2] dynein/kinesin motor stepping with \( d\mu/k\mu \) stepping rates. Cost function (B) and parameter values (C) (blue= \( d\mu \), green= \( k\mu \), red= \( da \), cyan= \( k\alpha \), yellow= \( d\delta \), magenta= \( k\delta \)) versus number of evaluations during the optimization of the 14 common binding sites model. Probability distribution of directed motion length (D) and velocity (E) for the in vivo in silico (black/red) trajectories. (F) Plot of the cost function versus the number of motor binding sites for the common (red) and the separate binding sites model (blue, grey, dark blue, black colors). The separate binding sites have a total number of 8 (blue), 10 (grey), 12 (dark blue) and 14 (black) binding sites for dynein plus kinesin motors. The central dot in each curve represents 50% dynein and 50% kinesin occupancy (e.g. black curve: 7+7). The remaining dots denote permutations with decreasing/increasing dynein binding sites (e.g. 6 dynein + 8 kinesin on the right and 8 dynein + 6 kinesin on the left of the central dot of the black curve). (G, H) examples of segmented in silico 1D trajectories for the 14 common (G) or 7+7 separate (H) binding sites models. The distance (\( \mu m \)) traveled along the 1D microtubule is plotted against the time in seconds, and the directed motions are depicted in red.

Here the six model parameters (binding, unbinding and stepping for both kinesin and dynein, Figure 6.2 A) were identified by minimizing the Symmetric Kullback-Leibler divergence between the in silico and in vivo length and velocity distributions using an Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [132] (Figure 6.2 B,C). The proposed de-randomized optimization algorithm is an essential aspect of our method. CMA-ES samples the six-dimensional multivariate normal distribution involving the parameters of this problem at each iteration and it encodes relations between the parameters of the model and the objective that is being optimized without requiring explicitly the gradients of the cost function [132]. The CMA-ES is a method capable of optimizing noisy cost functions (such as those from the present stochastic model) and its efficiency, reliability and robustness were demonstrated over a number of benchmark problems [133,134]. The CMA-ES is particularly suitable to this optimization problem as it is know to perform best [132] in problems that are low dimensional (here six parameters), inherently noisy (here a stochastic model),
6.3. Results

multimodal and computationally expensive (for each parameter set thousands of trajectories are generated and segmented to collect reliable statistics). The algorithm identifies an optimal set of parameters and at the same time provides a sensitivity analysis of the model. The standard deviations of the six principal axes are shown to converge in Figure 6.5, thus yielding a minimum.

After the convergence of the optimization process we found that the directed motion length and velocity distributions of the in silico trajectories, under the optimal set of parameters, matched with high accuracy the experimental data (Fig. 6.2 D, E).

The maximum number of motors attached to the viral cargo is limited by the number of the binding sites on the virus. The present model enables predictions on the number of motor binding sites on the viral capsid, a quantity that is difficult to determine experimentally but important for understanding the mechanisms of transport. We first estimated the number (between 2 and 20) of motor binding sites on the virus by an optimization procedure (Figure 6.2 F). In models with 6–16 binding sites, the cost function values were almost constant around the minimum value obtained for 14 binding sites. For less than 6 motor binding sites, the optimization process did not converge to the experimentally observed directed motion length and velocity distributions. Above 16 binding sites, an unbalanced configuration of motors was feasible only at low binding and unbinding rates, and yielded largely unidirectional trajectories due to infrequent motor binding to the virus. We concluded that 14 common binding sites for dynein and kinesin correspond optimally to the experimental data.

Since it is not possible to differentiate between common and separate binding sites, we additionally investigated the possibility that the experimental data support separate binding sites for the different motors. We optimized a model where dynein and kinesin have distinct binding sites, namely 4+4, 5+5, 6+6, 7+7 binding sites, and various permutations thereof (Figure 6.2 F), and found that an equal number of motor binding sites was optimal in all cases. This is consistent with the observation that center and periphery directed length and velocity distributions were almost symmetric (Figure 6.2 D, E). We note that the optimal number of binding sites, i.e., 14, is the same for the models with common and separate binding sites (Figure 6.2 F, black curve).

Molecular motors carrying cargo on microtubules operate as individuals or as an ensemble. We found that, on average, during virus directed motions, 1.56 ± 0.56 dynein or kinesin (for minus-end and plus-end directed motions, respectively) motors, and 0.15 ± 0.22 motors of opposite polarity were bound to the virus (Figure 6.3 A). The probability of binding more than four motors to one virus particle was below 10^{-3}, and most often only one type of motor was bound (Figure 6.3 A, B). These data are in agreement with low number of motors
estimated on vesicular cargo in squid axoplasm by cryo-EM [110]. For other organelles, the estimates for motor numbers range from a few to dozens based on immunological detections in chemically fixed cells.

In order to quantify the correlation between the number of bound motors and the directed motion length, the Sample Pearson Product Moment correlation coefficient (with a range of 0 to 1, where 1 is maximal correlation) between motor numbers and directed motion length was computed to be 0.51 for dynein and 0.49 for kinesin for minus-end and plus-end directed motions, respectively. This implies a weak correlation between the number of bound motors and the directed motion length, showing that long runs do not necessarily require many motors, as two or three already account for lengths in the order of micrometers (Figure 6.3 B). This result is consistent with the recent in vitro observation that two motors are sufficient to move a cargo over several micrometers [135]. The velocities, derived from optimized stepping rates, for single dynein and kinesin motors were 866 nm/s and 833 nm/s, respectively, consistent with values reported for dynein and conventional kinesin-1 or kinesin-2 [124–126, 136]. Although kinesins are currently not known to be involved in cytoplasmic transport of adenovirus [106], the model makes a clear prediction for a plus end directed motor in cytoplasmic transport of adenovirus.

Our findings indicate that microtubule-based motility of adenovirus requires a low number of bound motors compared to the number of binding sites on the capsid. This allows configurations where only one motor type is bound, and thereby produce directed motions. Low numbers of motors allow fast switches between directions and therefore, bidirectional motion. Importantly, the binding and unbinding rates were much smaller than the stepping rates, which is key for directed motion runs (Figure 6.2 C). Small perturbations of binding and unbinding rates greatly affect the model dynamics. The susceptibility of motor based cargo transport to these parameters has been reported in other theoretical studies [128] and hints to a possible mechanism to regulate the run length of the motors [135].

The present results enabled an assessment on the virus binding sites for motor proteins. The outer surface of adenovirus particles is composed of five polypeptides, three of which are still present on cytosolic viruses that have undergone stepwise disassembly [137]. Cytosolic particles contain the major protein hexon, a large fraction of the pentameric penton base at the icosahedral vertex, and protein IX (pIX), which stabilizes hexon. By considering the size (90 nm in diameter) and icosahedral geometry of the virus and the cylindrical microtubule (25 nm in diameter), we can postulate that the maximum number of microtubule motor-capsid interactions occurs along the edge of a capsid facet, in this case on hexon (Figure 6.4 A, B). This arrangement implies that 9 hexon trimers are aligned with the microtubule, giving a maximum of 27 hexon binding sites for the motors. This is above the value of 14 binding sites predicted from the simulations. If we assume, however, that the motor protein
6.4. Discussion

Figure 6.3: A low number of dynein and kinesin motors mediate directed motions of adenovirus. (A) In silico probabilities of the number of bound dynein (black) and kinesin (red) motors during periphery directed motions. Similar results were observed for center directed motions with dynein prevalence. Phase diagrams of the length (B) of directed motions versus the average number of bound dynein (black) and kinesin (red). Positive (negative) lengths correspond to periphery (center) directed motions. The results were obtained with the optimized model with 14 common binding sites.

binding sites are located at the interface of two trimeric hexons, one microtubule filament could cover 1-15 sites (Figure 6.4 B, red lines), which is within the predicted range of 6-16. In addition to hexon, 6 to 8 binding sites were available for pIX, and less than 5 for penton base which detaches to a significant extent from the incoming virions before reaching the cytosol [137]. We analyzed trajectories of pIX-deficient adenoviruses to distinguish between hexon and pIX binding sites for motor proteins [138]. The directed motion length and velocity distributions of pIX-deleted adenovirus were similar to those from wild-type viruses without significant deviations or asymmetries, indicating that pIX may not provide a binding site for microtubule dependent motors during cytoplasmic transport (Figure 6.4 C, D). Therefore, we predict that hexon harbours the binding sites for dynein and kinesin motors.

6.4 Discussion

In this study, we use in vivo imaging to identify a stochastic model of cargo transport by molecular motors on microtubules. The model parameters were systematically identified using live imaging data of virus trajectories and a de-randomized optimization algorithm to minimize the Kullback-Leibler divergence between the length and velocity distributions
Figure 6.4: Hexon not protein IX is the predicted to be motor binding site on the adenovirus capsid. (A) Schematic model of the icosahedral adenovirus capsid with the major capsid protein hexon (blue hexagonal structures representing trimers), and pIX (yellow lines representing trimers). An icosahedral facet is enlarged in (B), where hexon-hexon trimer interfaces are depicted in red and in light green the microtubule orientation that maximizes the overlap with one facet. Note that the fourth trimer of pIX is covered by the red lines depicting hexon-hexon interfaces. (C, D) Directed motion length and velocity distributions for protein IX deficient adenovirus recorded in HeLa cells 30 to 90 min post infection.

of adenovirus directed motions on microtubules with the in silico trajectories produced by the model. The model accounts for directed motions at μm/s speeds, processive stepping over hundreds of nanometers, and periods of stationary behavior. The results show that the stochastic model can result in bidirectional support without an explicit coordination mechanism.

In our work kinetic rates of a stochastic model are determined via an evolutionary optimization approach using experimental data. The identified model enables a number of predic-
6.4. Discussion

Figure 6.5: Convergence of the standard deviation of the principal axes. Convergence is shown for the six axes of the distribution which CMA-ES samples from. The evolution of the standard deviations during the optimization procedure is shown for the non-competing binding sites model with 14 receptors.

...tions. First, it shows that one to four motors are active on virus particles during microtubule-dependent motions, although the number of motor binding sites is estimated to be 6-16. The observation that the cost function value is constant within this range suggests that the virus may align with the microtubule in different orientations (Figure 6.4 B) and still preserve its motility. This range is consistent with the maximum of 15 hexon trimer-trimer interfaces along the edge of a capsid facet. The low number of motors involved in directed motions supports an emerging concept from wet lab experiments and in silico simulations, that key events of cell functions are in many cases executed by only a few polypeptides [139].

Second, if equal numbers of opposite motors are attached, the cargo oscillates and eventually stops, or remains confined to small areas. This may be an important mechanism for fine-tuning the subcellular velocity to achieve localized delivery of the cargo. We anticipate that viral transport is tuned by the binding and unbinding rates of motors to microtubules or the cargo, rather than by additional regulatory factors. Such tuning could be cell-type specific [120], and could control the number of engaged motors and motor configuration, and also provide specific segregation or guidance cues for traffic. In support of this, it has been suggested that the microtubule binding protein Tau can fine-tune the distance that the cargo travels by reducing microtubule binding of kinesin in distal parts of neuronal axons [110, 140]. In addition, motor properties can be tuned by post-translational modifications, such as phosphorylation of dynein or kinesin binding partners, which could affect their enzymatic
functions and hence their stepping rates [110].

We close by noting that besides the results on motor transport on microtubules, the algorithm taken here is in line with reverse engineering and systems identification approaches [131,141–143] which are gaining significance as discovery and model validation tools in systems biology. The CMA-ES algorithm is capable of handling noisy and multimodal cost functions that are inherently associated with stochastic models. The CMA-ES optimization algorithm along with image analysis of in vivo systems can be a robust process to help identify parameters of stochastic models employed in several areas of systems biology.
Chapter 7

Pattern Formation

7.1 Introduction

As discussed in the introduction of this thesis, pattern-formation equations have been proposed as models for morphogenesis [4]. It has been postulated that these simple reaction-diffusion systems are sufficient for describing the imperative characteristics of biological processes. Depending on how the parameters are chosen, and the size of the domain, one can obtain a multitude of patterns that may mimic natural phenomena.

The Gray-Scott model is an example of self-organization in non-equilibrium, chemically reacting systems [5]. The partial differential equations for this model are

\[
\frac{\partial u^{(1)}}{\partial t} - D^{(1)} \Delta u^{(1)} = -\rho \, u^{(1)} u^{(2)}^2 + F(1 - u^{(1)}),
\]

\[
\frac{\partial u^{(2)}}{\partial t} - D^{(2)} \Delta u^{(2)} = \rho \, u^{(1)} u^{(2)}^2 - (F + \kappa)u^{(2)},
\]

where \( u^{(s)} \) denotes the \( s \)th species, and \( D^{(s)} \) the diffusion coefficient for the \( s \)th species. The following chemical reactions represent the discrete model:

\[
U_i^{(1)} + 2U_i^{(2)} \xrightarrow{F} 3U_i^{(2)},
\]

\[
\emptyset \xrightarrow{F} U_i^{(1)},
\]

\[
U_i^{(1)} \xrightarrow{F} \emptyset,
\]

\[
U_i^{(2)} \xrightarrow{F+\kappa} \emptyset,
\]

where \( U_i^{(s)} \) is the number of molecules of species \( s \) in volume element \( i = (i_x, i_y, i_z) \). The values of \( F, \rho, \kappa \), the diffusion coefficients, as well as the size of the domain determine what kind of pattern will appear.
Figure 7.1: Analysis of the role of the number of particles for the Gray-Scott equations solved with a $300 \times 300$ discretization with $F = 0.04, \kappa = 0.06, t = 1000$. From left to right the number of particles per unit of concentration is increased from 100, 1000, 1000, 5000, 10000, continuum, respectively. The methods used to solve the equations were the following (from left to right): $S\tau$-Leaping, $S\tau$-Leaping, $H\tau$-Leaping, $S\tau$-Leaping, $H\tau$-Leaping, deterministic.

### 7.2 Uniform-Mesh Simulations

We performed numerical simulations of the Gray-Scott equations in 2 and 3-dimensions with periodic boundary conditions using deterministic, $H\tau$-Leaping (Section 4.1.3), and $S\tau$-Leaping approaches (Section 4.1.3) with varying levels of particles in order to determine whether we obtain qualitatively different patterns. Figure 7.1 shows results for 2-dimensional simulations of the Gray-Scott equations. We varied the number of particles in each cell while keeping $F = 0.04, \kappa = 0.06$ and $\rho = 1$, and integrated from $t = 0$ to $t = 1000$. There are notable differences in the solutions: as the number of particles increases, the stochastic simulations converge to the pattern observed by purely deterministic simulations of reactions and diffusion.

Lastly, we solved the Gray-Scott equations in 3-dimensions using a discretization of $128 \times 128 \times 128$ with $F = 0.04, \kappa = 0.06$ and $\rho = 1$, and integrated from $t = 0$ to $t = 1000$ (Figure 7.2). In 3-dimensions, the noise from the low numbers of particles makes itself apparent and the solution notably differs from the deterministic solution.

### 7.3 Adaptive Simulations

Gray-Scott simulations were also performed using the methods presented in Section 4.3. The initial conditions for the were species were

$$U_i^{(1)} = \begin{cases} 
\Omega_i \left( \frac{1}{2} + \xi \cdot 10^{-2} \right) & \text{if } x_i \in G_{\text{int}} \\
\Omega_i (1 + \xi \cdot 10^{-2}) & \text{otherwise}
\end{cases},$$

(7.7)
Figure 7.2: 3-dimensional solutions of the Gray-Scott equations using (left) deterministic solver and $H\tau$-Leaping solver (right) on a $128 \times 128 \times 128$ discretization with $F = 0.04, \kappa = 0.06, \rho = 1, t = 1000$. The $H\tau$-Leaping method was performed with 1000 particles per unit of concentration.

and

$$U_i^{(2)} = \begin{cases} \frac{[\Omega_i(\frac{1}{4} + \xi \cdot 10^{-2})]}{[\Omega_i(\xi \cdot 10^{-2})]} & \text{if } \mathbf{x}_i \in \mathcal{G}_{\text{int}} \\ \text{otherwise} & \end{cases},$$

(7.8)

where $\xi \sim \mathcal{U}(0, 1)$ and $\mathcal{G}_{\text{int}} = [1/4, 3/4] \times [1/4, 3/4]$. Figure 7.3 shows the concentration of $U_i^{(1)}$ along with mesh at various times for a simulation using the $\tau$-Leaping [14] algorithm in a unit square domain, where $t \in [0, 1000], \rho = 1, F = 2.8 \cdot 10^{-2}, \kappa = 6.1 \cdot 10^{-2}, D^{(1)} = 12.5 \cdot 10^{-5}, D^{(2)} = 6.25 \cdot 10^{-5}, C = 3$, and $Q = 10^4$, which results in 625 molecules at the finest level if the concentration is 1. The species $U_i^{(1)}$ was used to monitor the gradient in equation (4.80). The resolution of the simulation was $h_{\text{min}} = 1/400$ and $h_{\text{max}} = 1/100$. Although the stochastic simulation of the Gray-Scott equations was qualitatively similar to a deterministic simulation, the symmetry is not necessarily preserved as shown in Figure 7.4.
Figure 7.3: Stochastic simulation of the Gray-Scott model at various times with the adaptive mesh (left to right, top to bottom): $t = 0, 75, 115, 255, 435, 700, 870, 1000$. 
Figure 7.4: Comparison of stochastic (left column) and deterministic (right column) multi-resolution simulations of the Gray-Scott equations. Time from top to bottom: $t = 250, 500, 750$. 
Chapter 8

Simulations of Brain Tumors

8.1 Introduction

Cancerous tumors are caused by the mutation of one or more cells that exhibit uncontrolled and rapid growth. The effect of this is an impairment of normal tissue physiology. There are many different types of cancer and quite often they are distinguishable [144]. However, this also means that different types of cancer behave differently, thus making treatment difficult. The main characteristics or hallmarks of cancer are [144]:

1. Evading apoptosis
2. Self-sufficiency in growth signals
3. Insensitivity to anti-growth signals
4. Sustained angiogenesis
5. Limitless replicative potential
6. Tissue invasion and metastasis

As described in [144], most - if not all - cancerous cells exhibit this behavior.

The type of tumor considered here is the glioblastoma (glioma), which is the most malignant and most common brain tumor. The tumor is known to disseminate quickly throughout the brain and for this reason they are tumors with, as J. D. Murray states [145], “a depressingly dismal prognosis for recovery”. Indeed, if a glioma is left untreated, the median survival time is roughly 6 to 12 months [145]. Surgical removal of the tumor is presently the most effective treatment, thereby increasing the median survival time by 2.5 months [146].
The human brain consists of grey and white matter, the former of which is composed of neuronal and glial cell bodies that are responsible for controlling brain activity, while the latter is composed of fiber tracts of neuronal axon bundles. Shown in Figure 8.1 are the grey and white matter regions, as well as the corpus callosum that connects the left and right hemispheres of the brain. Since white and grey matter are fundamentally different, it is not surprising that the rate of dissemination is different in the white matter than in the grey matter [145].

Figure 8.1: Grey and white matter of a cross-section of a brain (image from [145]).

### 8.2 Numerical Method

We are interested in modelling the dissemination and proliferation of tumor cells in the brain. In order to do this, we begin by modelling the dissemination of cells with a diffusive term and will deal with the proliferation of cells later. The diffusion process is modeled by the following partial differential equation:

\[
\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( D(x) \frac{\partial u}{\partial x} \right), \quad x \in \mathcal{D},
\]

\[
\frac{\partial u}{\partial x} = 0, \quad x \text{ on } \partial \mathcal{D},
\]

where \( \mathcal{D} = [0, 1] \). At the moment we shall confine ourselves to the 1-D situation. The diffusion coefficient depends on \( x \) since it has been shown that proliferation is faster in the
8.2. Numerical Method

Figure 8.2: Dashed lines denote initial condition \( \cdots \), solution \( \rightarrow \), \( D_{i,j}^{(\text{mean})} \) \( \bigtriangleup \), \( D_{i,j}^{(\text{simple})} \) \( \circ \), and \( D_{i,j}^{(\text{harmonic})} \) \( \blacksquare \).

white matter than the grey matter, viz:

\[
D(x) = \begin{cases} 
D_g & \text{if } x \in D_{\text{grey}}, \\
D_w & \text{if } x \in D_{\text{white}},
\end{cases}
\]

where \( D = D_{\text{grey}} \cup D_{\text{white}} \), and where \( D_g \) and \( D_w \) are constants.

8.2.1 Inhomogeneous Diffusion

Let \( u_i(t) \triangleq u(x_i, t) \) where \( i \) is the index of a node. Using an explicit Euler method for the time-integration, the numerical method becomes

\[
u_i(t + \Delta t) = \frac{\Delta t}{h^2} \sum_{\{j: \ j \in N(i)\}} D_{i,j} \left( u_j(t) - u_i(t) \right),
\]

where \( N(i) \) denotes the set of indices that are neighbors of cell \( i \). The diffusion coefficient across the interface of cells \( i \) and \( j \), denoted by \( D_{i,j} \), needs to be defined. Following [77], a
Figure 8.3: Zoom into the discontinuity. Dashed lines denote initial condition ——, solution −−−, $D_{i,j}^{(\text{mean})}$ $\triangle$, $D_{i,j}^{(\text{simple})}$ $\circ$, and $D_{i,j}^{(\text{harmonic})}$ $\square$. Note the discrepancy between $D_{i,j}^{(\text{mean})}$ and $D_{i,j}^{(\text{harmonic})}$ at $x \approx 1/2$.

harmonic mean can be used, namely

$$D_{i,j}^{(\text{harmonic})} = 2 \left( \frac{1}{D_i} + \frac{1}{D_j} \right)^{-1}.$$  
(8.6)

Additionally, we will also define the diffusion coefficient as an average,

$$D_{i,j}^{(\text{mean})} = \frac{D_i + D_j}{2},$$  
(8.7)

or simply

$$D_{i,j}^{(\text{simple})} = D_i.$$  
(8.8)
8.2. Numerical Method

Figure 8.4: \( O(h^2) \) convergence using \( D_{i,j}^{(\text{harmonic})} \), equation (8.6). Black lines have a slope of one and two, and the error measure is the relative \( l_\infty \) norm.

8.2.2 Boundary Conditions

To handle the Neumann boundary conditions, we use a ghost point method. Consider the stencil at the left boundary:

\[
u_0(t + \Delta t) = \frac{\Delta t}{h^2} \left( D_{0,1} u_1(t) - 2D_{0,0} u_0(t) + D_{0,-1} u_{-1}(t) \right) + O(h^2), \quad (8.9)
\]

where there is a ghost point at \( x_{-1} = -h \). The derivative across the boundary needs to be zero, in which case we may use a central finite difference scheme for the gradient across the boundary:

\[
\frac{u_1(t) - u_{-1}(t)}{2h} + O(h^2) = 0, \quad (8.10)
\]
or simply $u_1(t) = u_{-1}(t)$. Plugging this into the stencil yields the scheme at the boundary:

$$u_0(t + \Delta t) = \frac{2\Delta t}{h^2} (D_{0,1} u_1(t) - D_{0,0} u_0(t)) + \mathcal{O}(h^2). \quad (8.11)$$

### 8.2.3 Numerical Validation

We consider an example where $D(x)$ is

$$D(x) = \begin{cases} 1 & \text{if } x \leq \frac{1}{2}, \\
5 & \text{if } x > \frac{1}{2}. \end{cases} \quad (8.12)$$

The solution of equations (8.1)-(8.2) with a diffusion coefficient defined by equation (8.12) is [28]

$$u(x, t) = 1 + \frac{1}{4} f_2(x) \exp \left\{ -\lambda_2^2 t \right\}, \quad (8.13)$$

where $\lambda_2 \approx 9.0065$, $T = \ln(x)/\lambda_2^2 \approx 0.0085$, and

$$f_2(x) = \begin{cases} \rho_2 \cos(\lambda_2 x) & \text{if } x \leq \frac{1}{2}, \\
\cos(\lambda_2(1 - x)/\sqrt{5}) & \text{if } x > \frac{1}{2}, \end{cases} \quad (8.14)$$

where $\rho_2 \approx 2.0651$.

Figures 8.2 and 8.3 show the analytical and numerical solution using different definitions for the diffusion coefficient (equations 8.6)-(8.8)). Note that $D_{i,j}^{(\text{simple})}$ is clearly incorrect since we are solving $\frac{\partial u}{\partial t} = D(x) \frac{\partial^2 u}{\partial x^2}$ and not $\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( D(x) \frac{\partial u}{\partial x} \right)$. Furthermore, as shown in Figure 8.3, $D_{i,j}^{(\text{mean})}$ is incorrect at the discontinuity, $x \approx 1/2$. Figure 8.4 shows $\mathcal{O}(h^2)$ convergence for $D_{i,j}^{(\text{harmonic})}$. We note that the rate of convergence for $D_{i,j}^{(\text{mean})}$ is $\mathcal{O}(h)$.

### 8.3 3-D Simulations using MRAG

We consider the inhomogeneous Fisher-Kolmogorov reaction-diffusion equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(x) \nabla u) + \rho u(1-u), \quad (8.15)$$

where $u = u(x, t), x \in \mathcal{D}$, the term $\rho u(1-u)$ represents the proliferation of cells, and

$$D(x) = \begin{cases} D_g & \text{if } x \in \mathcal{D}_{\text{grey}}, \\
D_w & \text{if } x \in \mathcal{D}_{\text{white}}, \\
0 & \text{if } x \notin \mathcal{D}_{\text{grey}} \cup \mathcal{D}_{\text{white}}. \end{cases} \quad (8.16)$$
8.3. 3-D Simulations using MRAG

8.3.1 Biological Data

Equation (8.15) will be solved inside a realistic model of the human brain. The anatomy of the human brain comes from the biological database Brain Web [147]. The Brain Web database was created using a Magnetic Resonance Imaging (MRI) simulator and defines the distributions and locations of various elements of the brain on a 3-D grid. At each voxel a concentration of grey and white matter is provided (along with fat, muscle/skin, skull, etc.), which will be used to define the geometry of a human brain.

The value of the diffusion coefficients at each voxel are

\[ D_i \triangleq p_i^{(w)} D_i^{(w)} + p_i^{(g)} D_i^{(g)}, \quad (8.17) \]

where \( p_i^{(w)} \) and \( p_i^{(g)} \) are the relative fractions of white and grey matter, respectively, from the Brain Web database such that inside of the brain \( p_i^{(w)} + p_i^{(g)} = 1 \). The values of \( D_i^{(g)} = 1.3 \cdot 10^{-3} \text{cm}^2/\text{day} \), \( D_i^{(w)} = 5D_i^{(g)} \), and \( \rho = 1.2 \cdot 10^{-2}/\text{day} \) were taken from [145]. These rates are used to model highly invasive tumor cells.

8.3.2 Stochastic Model

In order to define a stochastic process, we must define the drift process and multiplicative factor for the fluctuations. Moreover, we will use a multiresolution wavelet based framework (MultiResolution Adaptive Grids, MRAG, [148]) to solve the 3-D equations that we will formulate in this section. The MRAG framework operates on blocks of meshes that locally have uniform resolutions and exploits parallel computing architectures. The equations must therefore be formulated independent of neighboring cells. Moreover, since the framework is based on wavelet interpolation, the fluxes will not be conservative. With this in mind, we will formulate a non-conservative and local stochastic differential equation to model the dissemination and proliferation of a highly invasive brain tumor.

The drift of the stochastic process, \( \mu \), is defined as

\[
\mu(u_i) = h^{-2} \sum_{\{j: j \in N(i)\}} D_{i,j}^{(\text{harmonic})}(u_j - u_i) + \rho u_i(1 - u_i),
\]

(8.18)

\[
= \Delta_+ + \Delta_- + \rho u_i(1 - u_i),
\]

(8.19)

where the Laplace operator has been split into positive \( \Delta_+ \) and negative components \( \Delta_- \), i.e. the incoming and outgoing fluxes.

Equation (8.18) can be written as

\[
\frac{du_i}{dt} = \mu(u_i),
\]

(8.20)
or
\[ \text{du}_i = \mu(u_i) \text{dt}. \] (8.21)

A Brownian motion term is added to construct a stochastic differential equation
\[ \text{du}_i = \mu(u_i) \text{dt} + \sigma(u_i) \text{dB}_i, \] (8.22)

where we must now define the multiplicative factor \( \sigma(u_i) \) for the fluctuations. As discussed in Chapter 4, the diffusion process is modeled as transitions to neighboring cells, where the fluctuations are transitions from or into a cell. Here we formulate the fluctuations as being proportional to the incoming transitions, namely
\[ \sigma(u_i(t)) = \sqrt{\Omega \Delta}. \] (8.23)

As discussed in Section 3.8, the numerical method is
\[ u_i(t + \Delta t) = u_i(t) + \Delta t \mu(u_i(t)) + \Omega^{-1} \sqrt{\Delta t} \sigma_1(u_i(t)) \xi + \Omega^{-1} \sqrt{\Delta t} \sigma_2(u_i(t)) \eta, \] (8.24)

where \( \xi \) and \( \nu \) are random variates from a standard normal distribution and \( \sigma_2(u_i(t)) = \sqrt{\Omega u_i(t)(1 - u_i(t))} \). Because of the instability of Fisher’s equation, we use
\[ u_i(t + \Delta t) = u_i(t) + \Delta t \mu(u_i(t)) + F \left[ \Omega^{-1} \sqrt{\Delta t} \sigma_1(u_i(t)) \xi + \Omega^{-1} \sqrt{\Delta t} \sigma_2(u_i(t)) \eta \right] \] (8.25)

where \( F[.] \) is a rounding operator that rounds to the nearest \( n/\Omega \), where \( n \in \mathbb{Z} \), i.e. the fluctuations are on the order of particles in the system so as to not spuriously heat up the leading edge of the front. In principle, \( \Omega \) should be the number of tumor cells, where the number of tumor cells in a real tumor is \( 10^{11} \) [145]. The value of \( \Omega \) was set to \( 10^7 \) per unit of concentration per node which is, however, lower than a total of \( 10^{11} \). We note that the fluctuations are on the order of \( 10^{-3} \) or \( 10^{-4} \) (i.e. \( \Omega^{-1/2} \)), which if we model single cells, the fluctuations are relatively small.

A simulation over a time period of two years is shown in Figures 8.5-8.7. The initial conditions was a point source at an arbitrary position in the brain. It can be seen that the growth pattern of the tumor is nontrivial and is highly dependent on the anatomy of the brain and initial position of the tumor. Specifically, the location of the white and grey matter tissues dictate the growth process.
Figure 8.5: Virtual glioma at time $t = 0$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue).

Figure 8.6: Virtual glioma at time $t = 364$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue).
Figure 8.7: Virtual glioma at time $t = 720$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue).
Part IV

Conclusion & Outlook
Chapter 9

Conclusion

In Chapter 4 two numerical methods were presented for the efficient simulation of reaction-diffusion processes as described by stochastic and hybrid models. In $S\tau$-Leaping, a unified $\tau$-Leaping procedure was used for both the reaction and the diffusion processes whereas in $H\tau$-Leaping, diffusion was handled deterministically and the reactions stochastically. We validated the methods using the analytical solution of Fisher’s equation and we investigated the role of the number of molecules in pattern forming Gray-Scott equations. The algorithms were shown to exhibit significant computational improvements over the exact spatial SSA.

Additionally, a spatially adaptive simulation method for the reaction-diffusion master equation was presented. The diffusion rates on locally refined meshes were derived and validated. Furthermore, refinement and coarsening criteria and a stochastic interpolation scheme were proposed for stochastic simulations based on adaptive mesh refinement. The method was shown to be useful for problems that exhibit a localization of spatial scales such as in wave propagation and pattern formation processes.

However, the effect of reactant size was not investigated, which has been shown to have a significant influence on simulations (vide [149,150]). The reactant size can become especially important in adaptive simulations, where the number of molecules per cell is small and size of the reactants with respect to the cell size cannot be considered negligible.

In Chapter 5, a time-adaptive method was presented for the accelerated simulation of stochastic processes with delays. The method was shown to be as fast as existing works for certain benchmark problems and in certain cases orders of magnitude computationally more efficient. The accuracy of the method was verified by numerically showing convergence to the delay differential equation as the number of molecules in the system increased and by comparing the fluctuations in the system to those produced by the exact, delayed SSA. The algorithm was also shown to be capable of simulating complex systems of delayed
reactions, i.e. the Delta-Notch signalling pathway.

One caveat that needs to be mentioned is that the speed-up of the method is notable when the number of molecules becomes large. However, at a certain point - which depends on the system being simulated - the deterministic delay-differential equations may suffice. The system must therefore have enough molecules to warrant the use of an accelerated scheme whilst not having enough molecules to warrant a deterministic simulation.

Also in Chapter 5, a straightforward numerical method was presented for the simulation of stiff Markov processes. The theory of cutoff phenomenon was used to determine the optimal trade-off between error and speed-up. This trade-off was shown both as was shown analytically and numerically. Moreover, the use of cutoff theory - which has essentially been confined to theoretical work on probability theory - was shown to be beneficial in applied, computational work.

The algorithm, however, has not yet been applied to a realistic biochemical process that exhibits disparities in time-scales. Without a thorough comparison with other methods against a benchmark problem, the practicability and merits of the method will remain dubious.

In Chapter 6 we combined live cell microscopy, single virus tracking and trajectory segmentation to systematically identify the parameters of a stochastic computational model of cargo transport by molecular motors on microtubules. The model parameters were identified using an evolutionary optimization algorithm to minimize the Kullback-Leibler divergence between the in silico and the in vivo run length and velocity distributions of the viruses on microtubules.

The stochastic model suggested that bidirectional transport of human adenoviruses could be explained without explicit motor coordination. The model enabled the prediction concerning the number of motors active on the viral cargo during microtubule-dependent motions as well as the number of motor binding sites, with the protein hexon as the binding site for the motors.

In Chapter 7, pattern formation results were presented for the aforementioned spatially-dependent simulation methods that were presented in Chapter 4. Convergence results were presented for the Grey-Scott equations as the number of molecules becomes large. Moreover, results were shown for the adaptive mesh refinement algorithm, which highlighted the practical use of the algorithm.

In Chapter 8 3-D simulations of brain tumor growth were presented using a stochastic differential equation to model the spread of tumorous cells inside of a brain. Using experimental data of grey and white matter inside of a brain, non-spherical growth patterns were observed.
The model was based on macroscopic processes (reaction and diffusion), but it must be stated that the model leaves much to be desired in terms of modeling physical processes. The deformation of the brain tissue caused by the tumor was not modeled, nor was the increased pressure in the skull caused by the expansion of the tumor considered. For realistic simulations of brain tumors, the inclusion of these processes will be necessary.
Chapter 10

Outlook

Speculation on the outlook of the work contained herein will follow.

Ω-Expansion

The analytical expansions of the master equation is Section 2.4.4 are not of great practical use for most problems because of the complexity of the resulting equations. Even for low order approximations, the expansions become tedious and therefore, it is useful to exploit computer algebra systems (CAS). Moreover, using CAS, arbitrary order expansions can be performed that will yield a hierarchy of time-dependent moment equations [9], i.e., a set of ordinary differential equations. These equations can then be solved numerically to obtain accurate solutions of the master equation. This combination of symbolic manipulation and numerical analysis (using e.g., GiNaC [151]) may provide an efficient method for solving the master equation.

Convection-Diffusion Equations

In this thesis we confined ourselves to time-dependent equations (reactions only) and reaction-diffusion equations. There are, however, other areas where stochastic simulations (and in general Monte Carlo methods) can be applied. Convection-Diffusion equations in 2- and 3-D could, in principle, also be simulated using some of the spatially-adaptive algorithms presented in this thesis (see Breuer’s and Petruccione’s work on Burger’s equation: [152–154]), which could be promising if there is reason to believe that a stochastic model will yield qualitatively different results compared to deterministic simulations.

Chaotic Systems

The authors of a recent publication on an exact, accelerated stochastic simulation algorithm used a chaotic system, the so-called Williamowski-Rössler model, to validate their method [76]. It was shown that apart from their method’s results, the approximate leaping algorithms [14, 15] did not accurately capture the chaotic process (Figure 4 in [76]).
on the validity of approximate leaping algorithms in chaotic systems is one area that is both
desired and lacking since these methods, by virtue of being approximate, are significantly
faster than even exact, accelerated algorithms.

**Stiff Systems**

In Chapter 5 stiffness was assumed to be present in all of the examples. Detecting nontrivi-
also stiffness in dynamical systems, and in particular stochastic systems, is something which
would be of great benefit. Furthermore, the setting is complicated when stochastic systems
are far from equilibrium, in which case the assumption that the fluctuations are approxi-
mately Gaussian is generally invalid.
Part V

Appendix & Curriculum Vitæ
## List of Tables

2.1 Markov’s results after analyzing 20,000 letters from *Eugeny Onegin*: $C$ denotes a consonant, $V$ a vowel, $V|C$ denotes a consonant followed by a vowel [18]. ................................................................. 9

2.2 Example of a random variable: the number of consonants in a probabilistic model of two letters. ................................................................. 12
List of Figures

1.1 Brownian motion: the trajectory of a pollen granule in suspension [1]. . . . . . 2
1.2 Images from experiments with the Belousov-Zhabotinsky reactions [2, 3]. . . 3

2.1 Shown in 2.1(a) is the jump process with discrete states labelled by \( n \). Figure 2.1(b) depicts the ansatz and the typical evolution of the density function (image from [9]). ................................. 21
2.2 Comparison of the result of the \( \Omega \)-Expansion (top) with the solution of the master equation (bottom). Note the change of the state space from discrete, \( S \), to continuous, \( S' \) ......................................................... 23
2.3 The solution as a result of the expansion. The mean of probability density function \( \Pi(\xi, t) \) follows the deterministic trajectory \( \Omega \phi(t) \), whereas the variance is \( \sim \Omega^{1/2} \) ......................................................... 25
2.4 Reversible isomerization reaction: \([20, 0]\) denotes the state with 20 molecules of species \( A \) and 0 molecules of species \( B \). Shown is a comparison of the result of the \( \Omega \)-Expansion (top) with the solution of the master equation (bottom). 26
2.5 Example of the chemical state space: \([0, \Omega]_d\) lattice, which is a subset of \( \mathbb{Z}^d \). 30
2.6 Evolution of the Reversible Isomerization Reaction. The steady-state is shown in Figure 2.4 ......................................................... 30

3.1 Convergence of Monte Carlo methods. In 3.1(a), the variance exists and results in the well-known convergence rate, whereas in 3.1(b) the variance does not exist. The black lines have a slope of \(-1/2\), the red lines are simulations, and the blue lines are fit slopes of the simulations. ................................. 37
3.2 Comparison of the Random Walk method with the Diffusion Equation. . . 39
3.3 Surface plots of Poisson's equation, deterministic (left) and stochastic (right) for the functions \( f_1 \) and \( f_2 \). \( x- \) and \( y- \) axes show the index of the grid node. 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Contour plots of the broken disk Poisson equation.</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Point-wise error in space of the Monte Carlo solution.</td>
<td>42</td>
</tr>
<tr>
<td>3.6</td>
<td>Convergence of the Stochastic Simulation Algorithm with respect to $N$, the number of samples. The chemical example is the reversible isomerization (vide equations (2.105)-(2.106)).</td>
<td>45</td>
</tr>
<tr>
<td>3.7</td>
<td>Shown in 3.7(a) are the streamlines of the differential equations and the two steady-state solutions. In 3.7(b) convergence with respect to $\Omega$ is clear, however, the discrepancy when $\Omega$ is small is not due to an inadequate number of samples.</td>
<td>47</td>
</tr>
<tr>
<td>3.8</td>
<td>When $\Omega = 10$, the probability density is Gaussian-like and centered around the solution of the ordinary differential equations, whereas when $\Omega = 1$ results in a nontrivial bimodal distribution with peaks around the stable and unstable states of the differential equations.</td>
<td>48</td>
</tr>
<tr>
<td>3.9</td>
<td>Leaping Algorithms. For 3.9(a), light blue is the integral computed by using leaping algorithms, dark blue is the exact integral. The shift in the distributions in 3.9(b) signifies the incorrect computation of the first moment.</td>
<td>50</td>
</tr>
<tr>
<td>3.10</td>
<td>Weak and strong errors: the slope is $\gamma$, simulation (red), and theoretical (black).</td>
<td>53</td>
</tr>
<tr>
<td>3.11</td>
<td>The probability density function 3.11(a) produced from equation (3.104). In 3.11(b) $</td>
<td>e</td>
</tr>
<tr>
<td>4.4</td>
<td>Multiresolution mesh for the Fisher-Kolmogorov simulation: cell size $h_\lambda$ against position, where $\cdot \rightarrow \cdot$ is the resolution at $t = 0$ and $\cdot \rightarrow \cdot$ at $t = t_{end} = 19.6$. Wavefront center at $t_{end}$ is $x = 1/4$.</td>
<td>71</td>
</tr>
<tr>
<td>4.15</td>
<td>Error versus the maximum volume element length for simulations of the diffusion equation. Uniform mesh, $\mathbf{\square}$, and a composite mesh that contained a coarse and fine mesh, $\mathbf{\bullet}$. Right, snapshot of the simulation.</td>
<td>89</td>
</tr>
<tr>
<td>4.18</td>
<td>Stochastic simulation of the Fisher-Kolmogorov propagating front at various times. The left panel shows the concentration of $U_i^{(1)} \in [0.45, 0.55]$ as $\bullet$, and the right panel depicts the adaptive multiresolution mesh and a radial projection the 1-D solution for reference (white circle).</td>
<td>92</td>
</tr>
</tbody>
</table>
4.20 Analysis of simulation of the Fisher-Kolmogorov propagating front. The left panel (a) shows the concentration of mean radius of the propagating front over time, $r(t)$, where $\circ$ is the numerical value averaged over all angles $\theta$ and the black line is the 1-D analytically-derived wavespeed. The right panel (b) depicts values with respect to $\theta$ at time $t=6$ as $\bullet$, where the black line is the mean value. ........................................ 93

5.1 A) Comparison of approximate algorithms for the treatment of delayed reactions, time on the $x$-axes and their corresponding queues on the right. DAA-SSA in [91], top: the delayed reactions are uniformly distributed in the interval $[t_a+\tau_d, t_a+\tau_d+\tau_a]$, $r_i \sim \mathcal{U}(0,1)$, for $i = 1, \ldots, k_d$. D-Leaping, bottom: the delayed reaction is stored with the number of execution, $k_d = 5$ (in this example), as well as the earliest possible execution time for the delayed reaction, $q_{d,a} = t_a + \tau_d$, and the time-step which generated the reaction, $\text{span}_d = \tau_a$. B) Perspective when the delayed reaction ought to be executed in the current time-step $\tau$. If $((t+\tau-q_{d,a})/\text{span}_d) < 1$, then a partial number of the $k_d$ delayed reactions are executed, specifically $\hat{k}_d \sim \mathcal{B}(k_d, (t+\tau-q_{d,a})/\text{span}_d)$. ................................. 99

5.2 Relative speed-up compared with the treatment of the delayed reactions in [91] for the Hes1 model. Logscale plot, where $\epsilon$ is varied from $10^{-4}$ to $10^{-1}$ while $\beta = 10^3$. ......................................................... 101

5.3 Convergence with respect to the number of protein molecules for the Hes1 model [83] (left, entire simulation; right, region where $t \in [5, 12]$). Shown on the $y$-axis is the scaled number of protein molecules $(P \times s)$, where $s = 0.03/\beta$, and on the $x$-axis time in hours ($\theta$); $\beta$ is related to the number of molecules and was varied by factors of 10 and $\epsilon$ was fixed to 5 $10^{-4}$; $\beta = 10^2$, $\epsilon\text{-\tiny$\blacksquare$}$; $\beta = 10^3$, $\epsilon\text{-\tiny$\blacksquare$}$; $\beta = 10^4$, $\epsilon\text{-\tiny$\blacksquare$}$; delay differential equation (DDE), $\epsilon\text{-\tiny$\blacksquare$}$ .... 102

5.4 Comparison of D-Leaping with the exact delayed SSA [88]. Shown on the $y$-axis is the scaled number of protein molecules $(P \times s)$, where $s = 0.03/\beta$, and on the $x$-axis time in hours ($\theta$); The mean over $10^3$ runs (D-Leaping, $\epsilon\text{-}$, delay SSA, $\epsilon\text{-}$) and ($\pm$) standard deviation (D-Leaping, $\epsilon\text{-}$, delay SSA, $\epsilon\text{-}$) are shown for both methods, where $\beta = 10$. $\epsilon = 0.01$ was used for the D-Leaping method. ......................................................... 106
5.5 \( \tau \) versus the iteration number for a single run of the Hes1 model for the R-Leaping, \( \cdot^o \), [15] and \( \tau \)-Leaping, \( \cdot^\tau \), [14] methods. The parameters \( \epsilon \) and \( \beta \) were fixed to \( 5 \times 10^{-4} \) and \( 10^3 \), respectively. \( \tau \) in the R-Leaping algorithm is a random variable, thereby producing a distribution in the \( y \)-axis, whereas \( \tau \) is prescribed in \( \tau \)-Leaping. .................................................. 106

5.6 Synchronous and asynchronous oscillations of the hert proteins. Shown on the \( y \)-axis is the scaled number of protein molecules \( (P_{her1} \times s) \), where \( s = 1/\beta \), and time \( (t) \) on the \( x \)-axis; \( \cdot^o \) and \( \cdot^\tau \) denote the number of molecules in each of the two cells. (left) Using the delays times specified in \( \tau \)-d results in synchronous oscillations, whereas (right) \( \pi \)-d results in asynchronous oscillations. .................................................. 107

5.7 Example of stiffness as defined by disparities in eigenvalues with respect to a parameter, \( \epsilon \), in the system of equations (5.26)-(5.27). ...................... 112

5.8 Schematic of the flow averaging algorithm: the algorithm proceeds by alternating between the two time-steps and switching the parameter \( \alpha \) on \( (\alpha = 1/\epsilon) \) and off \( (\alpha = 0) \). (Image from [17]) ...................... 113

5.9 Explicit Euler (solid lines) and FLAVOR (dashed lines) methods. ........... 114

5.10 Cutoff Phenomenon at \( \xi = \epsilon \). Equation (5.57), \( ||P^\xi(t) - \pi||_2 \) versus \( \xi/\epsilon \). The error becomes relatively small once \( \xi/\epsilon > 1 \). ...................... 119

5.11 Mind the Shift: shift from left to right of the cumulative distribution functions of \( P^\xi(t' = 1) \) for the species \( S_2 \) as the value of \( \xi \) becomes smaller. Orange lines denote FLAVORized-SSA, black SSA. \( \xi = 10^{-j} \), where \( j = 1, \ldots, 8 \). The shift becomes prominent when \( \xi = \epsilon = 10^{-4} \) (solitary orange line) and is a result of the rescaling of the time variable (cf. equations (5.50) and (5.52)). ...................... 120

5.12 Error Analysis: \( ||P^\xi(t' = 1) - P(t = 1)|| \), where \( l_\infty \leftarrow \cdot--; l_1 \leftarrow \cdot--; l_2 \leftarrow \cdot--; \xi = 10^{-j}, j = 0, 1, \ldots, 8 \). The errors have been rescaled to be in the range \([0, 1]\) and the \( x \)-axis is logscale. ...................... 121

5.13 Computational Efficiency: speed-up compared to SSA, \( S(\epsilon, \xi) \), with respect to the scaling parameter, \( \xi \), and the stiffness parameter, \( \epsilon \). All axes are logscale. ...................... 121
5.14 Optimal Value of $\xi$: error of the probability density function, $E_P(\xi)$, versus speed-up, $S(\xi)$. $l_\infty \leftarrow \circ \rightarrow$; $l_1 \leftarrow \square \rightarrow$; $l_2 \leftarrow \diamond \rightarrow $. $\xi = 10^{-2}$, where $j = 0, 1, \ldots, 5$ and where $\xi = 1$ is the leftmost point and $\xi = 10^{-5}$ is the rightmost point. The maximum error has been rescaled to 1 and the error at $\xi = 1$ is the sampling error. The plot markers denote simulated points and the lines are a result of second-order interpolation between simulated points. The optimal value is $\xi = 10^{-2}$ where $S(\xi) \approx 7$ and the error is negligible.  

5.15 Comparison of Solutions: orange denotes FLAVORized-SSA, black SSA. $P^{\xi=10^{-2}}(t' = 4)$ (left) and $P^{\xi=10^{-5}}(t' = 4)$ (right) for the species $S_i$. Note that the large error for $P^{\xi=10^{-5}}(t' = 4)$ results from a shift in the distribution.  

6.1 Imaging, tracking and trajectory segmentations of single adenoviruses. (A) HeLa cells were infected with fluorescent adenovirus type 2 for 30 min, and imaged by spinning disc confocal fluorescence microscopy [129]. Virus tracks (black lines) recorded by a single particle tracking algorithm [114] using the nucleus (red square) as a reference point are displayed over a phase contrast image of the infected cell. (B) Two-dimensional projection of a single virus trajectory with directed motion segments in red. (C) Distance travelled along the trajectory shown in Figure 6.1 B plotted as a function of time. Reduction to 1-D is justified, since in cultured cells microtubules are largely flat and straight over distances of many micrometers [130]. Negative/positive values indicate displacements towards the cell center/periphery.  

171
6.2 A stochastic model for microtubule-dependent movements of human adenovirus. The stochastic model reproduces directed motion length and velocity distributions of human adenovirus, and predicts the optimal number of either common or separate binding sites for dynein and kinesin motors on the capsid. (A) Dynein (D, blue) and kinesin (K, orange) bind to, and unbind from the capsid and transport it along a microtubule (green). Equations [1]-[6] describe the dynamics of the model: [3]/[4] dynein binding/unbinding and decrease/increase of the number ($\rho$) of the available motor binding sites on the virus capsid with $d\alpha/d\delta$ binding/unbinding rates, [5]/[6] kinesin binding/unbinding and decrease/increase of the number($\rho$) of the available motor binding sites on the virus capsid with $k_\alpha/k_\delta$ binding/unbinding rates, and [1]/[2] dynein/kinesin motor stepping with $d\mu/k\mu$ stepping rates. Cost function (B) and parameter values (C) (blue= $d\mu$, green= $k\mu$, red= $dx$, cyan= $k\alpha$, yellow= $d\delta$, magenta= $k\delta$) versus number of evaluations during the optimization of the 14 common binding sites model. Probability distribution of directed motion length (D) and velocity (E) for the in vivo in silico (black/red) trajectories. (F) Plot of the cost function versus the number of motor binding sites for the common (red) and the separate binding sites model (blue, grey, dark blue, black colors). The separate binding sites have a total number of 8 (blue), 10 (grey), 12 (dark blue) and 14 (black) binding sites for dynein plus kinesin motors. The central dot in each curve represents 50\% dynein and 50\% kinesin occupancy (e.g. black curve: 7+7). The remaining dots denote permutations with decreasing/increasing dynein binding sites (e.g. 6 dynein + 8 kinesin on the right and 8 dynein + 6 kinesin on the left of the central dot of the black curve). (G, H) Examples of segmented in silico 1D trajectories for the 14 common (G) or 7+7 separate (H) binding sites models. The distance ($\mu m$) traveled along the 1D microtubule is plotted against the time in seconds, and the directed motions are depicted in red. 131

6.3 A low number of dynein and kinesin motors mediate directed motions of adenovirus. (A) In silico probabilities of the number of bound dynein (black) and kinesin (red) motors during periphery directed motions. Similar results were observed for center directed motions with dynein prevalence. Phase diagrams of the length (B) of directed motions versus the average number of bound dynein (black) and kinesin (red). Positive (negative) lengths correspond to periphery (center) directed motions. The results were obtained with the optimized model with 14 common binding sites. 135
6.4 Hexon not protein IX is the predicted to be motor binding site on the adenovirus capsid. (A) Schematic model of the icosahedral adenovirus capsid with the major capsid protein hexon (blue hexagonal structures representing trimers), and pIX (yellow lines representing trimers). An icosahedral facet is enlarged in (B), where hexon-hexon trimer interfaces are depicted in red and in light green the microtubule orientation that maximizes the overlap with one facet. Note that the fourth trimer of pIX is covered by the red lines depicting hexon-hexon interfaces. (C, D) Directed motion length and velocity distributions for protein IX deficient adenovirus recorded in HeLa cells 30 to 90 min post infection. ........................................ 136

6.5 Convergence of the standard deviation of the principal axes. Convergence is shown for the six axes of the distribution which CMA-ES samples from. The evolution of the standard deviations during the optimization procedure is shown for the non-competing binding sites model with 14 receptors. . . . 137

8.1 Grey and white matter of a cross-section of a brain (image from [145]). . . . 146

8.2 Dashed lines denote initial condition --- , solution —, $D_{ij}^{(mean)} \bigtriangleup$, $D_{ij}^{(simple)} \circ$, and $D_{ij}^{(harmonic)} \square$. .......................... 147

8.3 Zoom into the discontinuity. Dashed lines denote initial condition --- , solution —, $D_{ij}^{(mean)} \bigtriangleup$, $D_{ij}^{(simple)} \circ$, and $D_{ij}^{(harmonic)} \square$. Note the discrepancy between $D_{ij}^{(mean)}$ and $D_{ij}^{(harmonic)}$ at $x \approx 1/2$. .......................... 148

8.4 $O(h^2)$ convergence using $D_{ij}^{(harmonic)}$, equation (8.6). Black lines have a slope of one and two, and the error measure is the relative $l_\infty$ norm. .......... 149

8.5 Virtual glioma at time $t = 0$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue). ......................... 153

8.6 Virtual glioma at time $t = 364$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue). ......................... 153

8.7 Virtual glioma at time $t = 720$ days: tumor density from low (white) to high (red), grey matter (blue), white matter (light blue). ......................... 154
Bibliography


[56] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Stiffness in stochastic 

[57] M. J. Ablowitz and A. Zeppetella. Explicit solutions of fishers equation for a special 

    154(1):134–155, Sep 1999. 74

[59] A. L. Garcia, J. B. Bell, and J. Foo. Algorithm refinement for the stochastic burgers’ 

    15(1):0001–0089, Jan 1943. 74, 77


[63] M.A. Burschka. Fluctuations in diffusion reaction systems. i: Adiabatic elimination 
    of transport modes from a mesoscopic n-body system and the omega-expansion. *J. 

[64] F. Baras and M. M. Mansour. Reaction-diffusion master equation: A comparison with 

[65] B. Bayati, P. Chatelain, and P. Koumoutsakos. Multiresolution stochastic simulations 

[66] S. Engblom, L. Ferm, A. Hellander, and P. Lotstedt. Simulation of stochastic reaction-
    diffusion processes on unstructured meshes. *SIAM Journal on Scientific Computing*, 
    31(3):1774–1797, 2009. 74, 75


Curriculum Vitae

Name  Basil Bayati

Citizen of  U.S.A.

Born  July 27th, 1982

2000-2004  B.S. Information and Computer Science  
University of California, Irvine, U.S.A.

2004-2006  M.S. Bioinformatics  
Chalmers University of Technology, Gothenburg, Sweden

2006-2011  Research and teaching assistant  
ETH Zürich, Switzerland  
Computational Science and Engineering Laboratory  
under the supervision of Prof. Dr. Petros Koumoutsakos