On the Stochastic Modelling of Reaction-Diffusion Processes

Aidan Twomey

Linacre College

University of Oxford

A thesis submitted for the degree of

Master of Science in Mathematical Modelling & Scientific Computing

Trinity 2007

Abstract

Traditionally, the law of mass action has been used to deterministically model chemical reactions. There are, however, fundamental limitations to this approach when the number of molecules involved in the reactions is small. This is often the case in biological cells, and many authors have reported that deterministic models do not adequately describe processes such as gene expression. This dissertation shows circumstances in which stochastic simulation describes non-linear reactions more accurately than deterministic models. Exact stochastic algorithms due to Gillespie and Gibson-Bruck, based on the chemical master equation, are introduced. A compartment based model is described and analysed for reaction-diffusion systems with low copy numbers of molecules. Finally, this approach is used to produce Turing patterns that may be useful in describing pattern formation in developmental biology.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Dr Radek Erban for his considerable patience and helpful suggestions during the course of this dissertation. What mistakes remain would surely have been eliminated if I had listened more carefully.

Finally, I would also like to thank my infinitely supportive wife Helen, a non-mathematician who allowed me to believe that my conversation wasn't boring her. My next project is to convince her that x can be useful in the supermarket.

Chapter 1

Introduction

Chemical reactions occur because of the discrete nature of matter; they arise due to interactions between molecules that are constantly moving within a confined domain. It is well known from elementary observations by chemists that, for a sufficiently large number of molecules, the random interactions always combine in such a way that the results are deterministic and repeatable. This is a dissertation about modelling the outcomes of reactions when the numbers of molecules are low and the domain is a biological cell.

At one level, modelling of biochemical reactions is an old and well understood phenomenon. One extremely well known system is the enzyme reaction mechanism described by Michaelis and Menten in 1913. Suppose we have a spoonful each of a substrate S and an enzyme E that react according to the equation

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \xrightarrow{k_2} E + P \tag{1.1}$$

to produce an intermediate chemical C that then breaks down to produce a product P. If we know the rate constants k_1, k_{-1} and k_2 we can predict the amount of P that will be produced at any time using the *law of mass action* which states that, in a well mixed system, the rate at which two chemicals combine to form a third is proportional to the product of their concentrations. Thus a system of ordinary differential equations (one for each chemical) can be formed and solved for P. Fowler [9] gives full details of the calculation.

The key to this model is that the law of mass action is justified by "averaging" arguments, which may be roughly summarised as follows. Consider the reaction $S + E \xrightarrow{k_1} C$, and denote the concentrations of the chemical species S, E and C as s, e and c respectively. A particular S molecule will trace out a certain "collision" volume in time interval $[t,\ t + \delta t)$ as it travels with random motion through the domain. The number of C molecules created by a reaction will be proportional to the number of E molecules that have their centre within this collision volume and to the rate constant k_1. Assuming that the collision volume is sufficiently large we surmise that the change in c is equal to $k_1 se\delta t$. Letting δt tend towards zero, we can form a differential equation for the rate of change of c due to the reaction $S + E \xrightarrow{k_1} C$ in the form $\mathrm{d}c/\mathrm{d}t = k_1 se$. Unfortunately, as Gillespie pointed out [11], this is a physically bogus argument, since as δt goes towards zero, the number of E in the (small) collision volume, being either zero or one, is no longer proportional to concentration e. To maintain the fiction requires that the number of molecules involved is sufficiently large that "averaging" the number of molecules E in the collision volume (corresponding to a molecule of S) over many molecules of S gives an illusion of continuity and deterministic dynamics of the species concentrations.

This reasoning has broad applicability in applied mathematics to systems comprising essentially discrete events, such as traffic on a busy road, trading at a financial exchange or gas dynamics. Bogus argument or not, models such as the law of mass action that in some sense average the individual events give excellent results at the macroscopic level, and many successful models involving systems of PDEs arise this way[18]. However, when the number of events is small and essentially random, there is no reason to expect our derived equations to predict the correct answer at any particular time, because there are not enough events to make averaging meaningful.

We have so far used the words "large" and "small" with abandon. To convince the reader that the shortcomings of the law of mass action are problematic we should quantify these notions and identify certain systems for which we need to develop new ways of modelling chemical reactions. As a rule of thumb, Van Kampen suggests [28] that the relative fluctuations of random interactions in a system of N molecules will be of the order $N^{-\frac{1}{2}}$. If this ratio between the size of the "random component" and the size of the "deterministic component" is significant we cannot use the deterministic arguments outlined above. So for a reaction involving 32g of oxygen, that is 6×10^{23} molecules, the fluctuations have an imperceptible affect on the rate (which is why the law of mass action is so successful). However, in cells, copy numbers of molecules are low. For example, Guptarsarma [12] describes the genome of the bacterium *Escherichia coli*, in which about 1700 important proteins are present in small numbers: there are between 10 and 30 molecules of the lac repressor involved in gene expression, and only 50 to 100 of the primase dnaG that initiates DNA replication. In this case, $N^{-\frac{1}{2}}$ becomes a significant concern, and we will show in Chapter 2 that for non-linear systems the interaction between fluctuating molecules gives results that cannot be modelled by the law of mass action.

The first illustrative problem is motivated by gene regulation in biological cells. Cells have many functions, including movement, signalling and reproduction[15]. Fundamental to cell reproduction is the process by which the genetic material is transmitted to a new cell. A gene is a sequence of DNA corresponding to a single protein that is assembled when the base pairs of sugars of an existing strand of DNA are replicated, transcribed and translated [1]. This process is the same for all living cells: first, segments of DNA are used as templates to produce a polymer mRNA (*messanger RNA*), which directs the way in which amino acids are brought together to form proteins. The manner in which different proteins are assembled is known as *gene expression*. We see that the proteins are the essential component of this process, and as well as being its product also tightly regulate gene expression by acting as catalysts and breaking down to provide products to fuel it. In Section 2.1 we shall consider a toy problem introduced by Paulsson [23] which demonstrates that deterministic modelling may not adequately describe gene regulatory processes. We continue Chapter 2 with a brief description of self-induced stochastic resonance, which demonstrates that small intrinsic fluctuations may have a decisive impact on the dynamics of a chemical system. We finish Chapter 2 by formulating the chemical master equation from the theory of chemical kinetics.

In Chapter 3 we shall introduce the tools we shall need to model chemical reactions, exact stochastic algorithms due to Gillespie and Gibson-Bruck which are equivalent to solving the master equation. In Chapter 4 we shall apply these tools to linear and non-linear 2D reaction-diffusion problems, which we shall model using a compartment based approach. In Chapter 5 we shall test the conditions in which our compartment based approach to reaction-diffusion is accurate and finally in Chapter 6 see how patterns may be formed in cells.

Chapter 2

Stochastic Modelling

To motivate the use of stochastic modelling in cells, we demonstrate that stochastic modelling can explain qualitative behaviour in cells that deterministic models can not. In Section 2.1 we study Stochastic Focusing proposed by Paulsson[23]. In Section 2.2 we present an example of self-induced Stochastic Resonance introduced by Muratov[19].

2.1 Stochastic Focusing

Paulsson et al [23] describe a phenomenon they call Stochastic Focusing, which shows how the fluctuations in one chemical species can drive a system to a different steady state than that predicted by the deterministic model. Consider a chemical system:

$$\emptyset \xrightarrow{k_1} I \xrightarrow{k_2} P \xrightarrow{k_3} \emptyset \tag{2.1}$$

$$I + S \xrightarrow{k_4} S \tag{2.2}$$

where some product molecule P is derived from I in the presence of a signal S which is subject to the reactions:

$$\emptyset \underset{k_6}{\overset{k_5}{\rightleftarrows}} S \tag{2.3}$$

This system could be a caricature of the manufacture of a protein outlined in the introduction, with P the protein end product, mRNA playing the role of I and S being a (protein) catalyst. Equation (2.1) describes how I is produced (from some molecule that does not concern us) with rate k_1 and P is produced from I with rate k_2, and degrades with rate k_3. This reaction will be affected by the degradation of I, which is catalysed by S with rate k_4, described by equation (2.2). This in turn is affected by the production (with rate k_5) and degradation (with rate k_6) of S described by equation (2.3). By the law of mass action, the time evolution of the

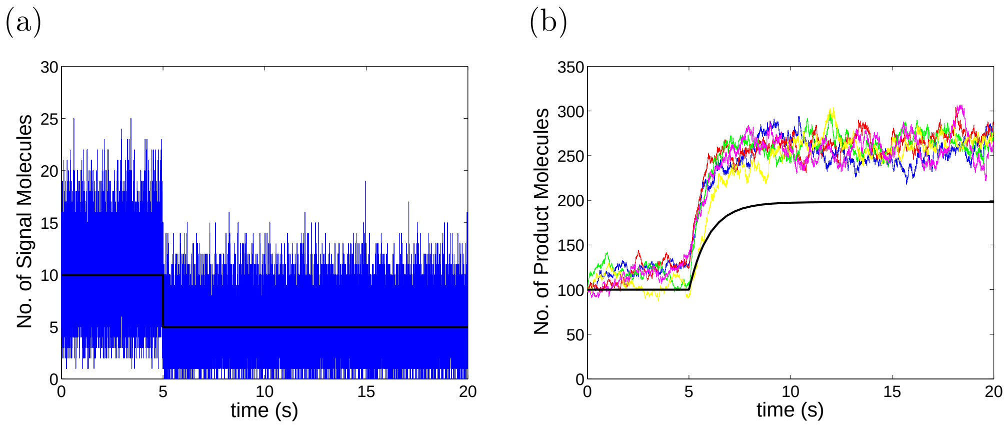

Figure 2.1: *Stochastic Focusing.* (a) *If the rate of production of S in equation* (2.3) *is abruptly halved from $k_5 = 10000s^{-1}$ (for time $t \leq 5s$) to $k_5 = 5000s^{-1}$ (for time $t > 5s$) the number of molecules fluctuates with mean $\langle S \rangle = 10$ for time $t \leq 5s$ and with mean $\langle S \rangle = 5$ for time $t > 5s$.* (b) *The numbers of product molecules P, as simulated by the Gillespie algorithm (coloured lines), are significantly larger than the deterministic solution (black line) allows. The rate constants are $k_1 = 10000s^{-1}$, $k_2 = 1000s^{-1}$, $k_3 = 1s^{-1}$, $k_4 = 9900s^{-1}$, $k_6 = 1000s^{-1}$. Initial copy numbers are $I = 0$, $S = 10$, $P = 100$.*

concentrations of I, P and S can be described by the system of ODEs

$$\frac{\mathrm{d}I}{\mathrm{d}t} = k_1 - k_2 I - k_4 S I, \tag{2.4}$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = k_2 I - k_3 P, \tag{2.5}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = k_5 - k_6 S. \tag{2.6}$$

To determine whether the system has any equilibrium points we let the derivatives in equations (2.4)-(2.6) equal zero and solve for I, S and P. We find that the number of P molecules produced depends upon the rate constants like this:

$$P = k_6 \frac{k_1 k_2}{k_3 \left(k_2 k_6 + k_4 k_5\right)}$$

Now if the value of the rate constants were to change we would expect the number of P molecules at steady state to vary according to this equation. As a simple case, let us suppose that the degradation of the catalyst S halves at a particular point in time, that is the rate constant k_5 is halved. Now if $k_4 k_5 \gg k_2 k_6$ we see that

$$P \approx k_6 \frac{k_1 k_2}{k_3 k_4 k_5}. \tag{2.7}$$

Thus the deterministic model predicts that number of P molecules will at most double. We can see this to be the case in Figure 2.1(b), where the black line is a solution of the deterministic system (2.4)-(2.6).

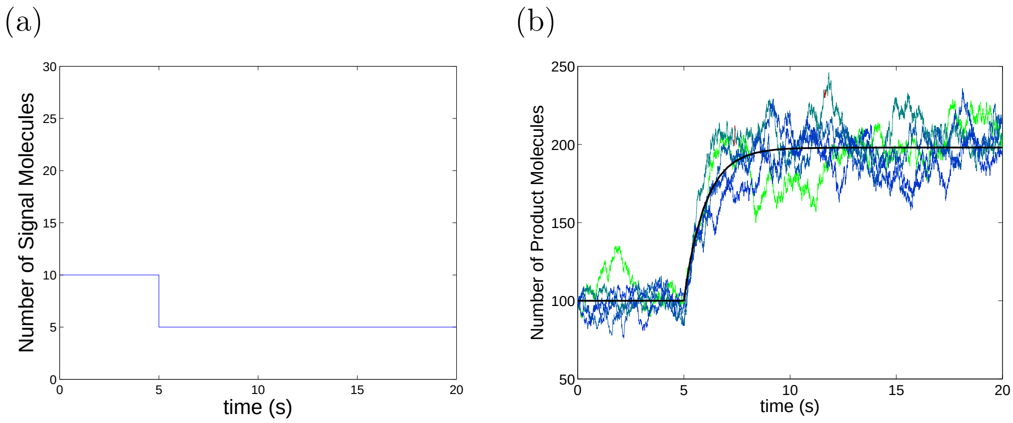

Figure 2.2: *Stochastic Focusing - linear model.* (a) *Abrupt change in the constant signal molecules in a deterministic and stochastic models.* (b) *Five realisations of resulting P numbers calculated by the Gillespie algorithm (coloured lines) and the equivalent deterministic solution to system of chemical reactions* (2.1)-(2.2) *(black line) are shown. The rate constants are* $k_1 = 10000s^{-1}$, $k_2 = 1000s^{-1}$, $k_3 = 1s^{-1}$, $k_4 = 9900s^{-1}$.

We now examine the system's response to a sudden change using a Monte Carlo algorithm developed by Gillespie (described in Chapter 3) which calculates the copy numbers of each molecule on a reaction-by-reaction basis. Simulating the system of six reactions (given in equations (2.1)-(2.3)) in this way we see that the number of molecules of S fluctuates around the initial values of S (see Figure 2.1(a)). Halving k_5 as before we see that the mean value of S halves but, surprisingly, the number of molecules of P produced is significantly greater than the steady state solution of the deterministic system (2.1) predicts: P more than doubles.

To see that the dramatic increase in P is due to the stochastic fluctuation of S, we hold S steady at the mean values. We simulate this simplified system of four reactions (ignoring equations (2.3)) using the Gillespie algorithm. We see (Figure 2.2) that the simulation results match the deterministic solutions very well. This observation can also be rigorously proven: if S does not fluctuate, then system (2.4)-(2.5) is essentially a linear system of two ODEs for I and P for constant S. It is well known that for *linear* systems the deterministic ODEs give the time evolution of mean values of stochastic simulation[8].

For the rate constants and initial values given in Figure 2.1, after 15 seconds we calculate (using 100 simulations) that the mean number of P molecules created is 264 (standard deviation 17.3). A solution of the deterministic equations (2.4)-(2.6) (using a Matlab ODE solver) predicts that 198 molecules of P should be created. On the other hand, simulating equations (2.1)-(2.2) with 100 runs of the Gillespie algorithm,

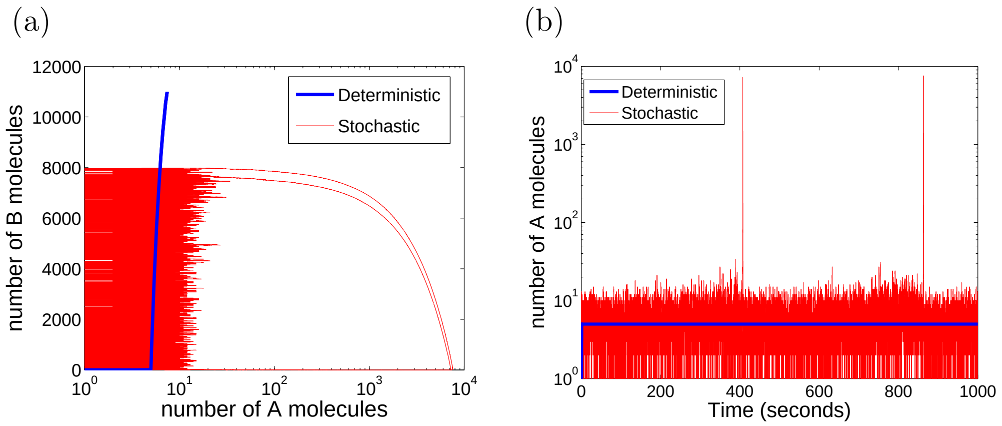

Figure 2.3: *Stochastic Resonance. The fluctuation in A molecules allows the system to reach an excited state even though the deterministic system reaches equilibrium.* (a) *The stochastic and deterministic trajectories in the A-B plane. Note that the A axis is logarithmic.* (b) *Time evolution of A. To show the small fluctuations, a logarithmic scale is used for the A axis.*

we now find that the number of P molecules matches the deterministic prediction - see Figure 2.2(b) - and that the average number calculated by the Gillespie algorithm (198) matches the deterministic value exactly.

We conclude that the occasional high value of S allows the P value to jump to a higher level than a deterministic equation would allow. So, if a cell responds to a certain threshold number of P molecules, this stochastic mechanism dependent on the random kinetic properties of the molecules explains how a cell can be more sensitive to signal than a deterministic analysis would allow.

2.2 Stochastic Resonance

It is important to realise that even small noise terms in an evolving system can help the stochastic solution to reach qualitatively different states than its deterministic counterpart. For example, a process known as Stochastic Resonance, where the random fluctuations of a signal chemical drive a system into a limit cycle whereas a noise-free model would reach an equilibrium, is thought to occur in many systems [29] including sensory neurons, which are hypersensitive to weak signals but ignore false stimuli. Following Muratov [19], consider an excitable system of reactions

$$2A + B \xrightarrow{k_1} 3A \qquad \emptyset \underset{k_3}{\overset{k_2}{\rightleftarrows}} A \qquad \emptyset \xrightarrow{k_4} B \tag{2.8}$$

This system has an “excited” state where large number of A molecules are produced. Solving the associated deterministic system, we see that it reaches a steady state where only a few molecules of A are produced, and will never reach the excited state. However, simulating the system with the Gillespie algorithm, we see that when there are random fluctuations in A copy numbers, the system can "hop" the energy barrier and reach the excited state. The result is a limit cycle whose amplitude and frequency depend on the noise terms (see Figure 2.3).

Although the system (2.8) does not have as satisfactory a physical interpretation as the Paulsson system (the trimolecular reaction $2A + B \xrightarrow{k_1} 3A$ would require the physically unrealistic simultaneous collision of three molecules), these two examples are an indication that the law of mass action is unsatisfactory and that we must develop different ways of modelling the chemical reactions that occur in biological cells.

2.3 Definitions, Terminology and Notation

We make the forthcoming discussion easier to follow by now introducing the necessary terminology and notation. We consider a chemical system with N species A_j, $j = 1, \dots, N$. The copy number of A_j molecules at time t is denoted by $A_j(t)$. We let R denote a reaction. For a system of M reactions we denote the i^{th} reaction by R_i for $1 \leq i \leq M$.

For a reaction R denote the set of reactant molecules as Depends(R) and the set of all species that have their copy numbers altered when the reaction occurs as Affects(R). Let

> $\boldsymbol{\nu}_i \in \mathbb{R}^N$ be the vector of coefficients that adjusts the copy numbers of each chemical species in the system according to the reaction R_i.

For example, for reaction (2.2), Depends(R) $= \{I, S\}$, Affects(R) $= \{I\}$, $\boldsymbol{\nu} = [\Delta I, \Delta S, \Delta P] = [-1, 0, 0]$.
For a system comprising a number of reacting chemicals, let

> $\mathbf{v} \equiv \mathbf{v}(t) = [A_1(t), A_2(t), \dots, A_N(t)] \in \mathbb{R}^N$ be a state vector with the number of molecules of each chemical present in the system at time t as co-ordinates.

For example, for the Paulsson system (2.1)-(2.3), we have $\mathbf{v}(t) = [I(t), S(t), P(t)]$. Finally, we denote by $\mathbb{P}(\mathbf{v}', t' | \mathbf{v}, t)$ the conditional probability that the system is in state $\mathbf{v}'$ at time t' given that it was in state $\mathbf{v}$ at time t.

2.4 Chemical Kinetics and The Master Equation

At the mesoscopic level, Einstein tells us [5, 20] that molecules in a fluid possess kinetic energy which causes them to travel in uncorrelated directions according to a velocity distribution. We can use the random movements to develop a probabilistic model for chemical reactions that we may use to simulate a system of reactions.

Intuitively we can accept that chemical reactions occur when molecules collide. If we assume that the system is in thermal (but not chemical) equilibrium so that the molecules are uniformly distributed, they will collide with each other at random, reacting when they have appropriate chemical properties (which do not concern us). Two molecules may only react when their centres are separated by a distance not greater than the sum of their radii. Even then, they may only react if they collide in a certain way, for example if certain "binding sites" on the surface of the molecules meet. We incorporate this into our assumptions by assigning a certain probability to the event that molecules react upon collision.

We can use these assumptions to introduce the important concept of the *propensity* of a chemical reaction R_i, $i = 1, \ldots, M$. Given that there is a certain probability of reactant molecules colliding and another (independent) probability that they will react upon this collision, we can conclude that the likelihood of R_i occurring is proportional to the number of combinations of reactant molecules. The constant of proportionality is the mesoscopic rate constant k. Furthermore, the *propensity function*, $\alpha_i(\mathbf{v})$, of a reaction R_i is defined so that $\alpha_i \mathrm{d}t$ is the probability that R_i will occur in the time interval $[t, t + \mathrm{d}t)$ given that the system was at state $\mathbf{v}$ at time t. It is determined by the rate constant and the state of the system. For example, the reaction (2.2) depends on the copy numbers of I and S, as well as the rate constant k_4. Thus the propensity function is given by

$$\alpha = k_4 I(t) S(t)$$

A list of the propensity functions used in Chapter 2 can be found in Table 2.1.

Let us consider a system of molecules of species A subject to chemical reactions R_i, $i = 1, \ldots, M$. Let us suppose that initially the state of the system was $\mathbf{v}_0$ at time $t = 0$. We wish to write an equation for the quantity $\mathbb{P}(\mathbf{v}, t|\mathbf{v}_0, 0)$, which is the probability that the system is in state $\mathbf{v}$ at time t given that it was in state $\mathbf{v}_0$ at time $t = 0$. Such an equation is called a *master equation*[*]. Let us consider the time

[*] The term "master equation" was coined in [21] to calculate the energy distribution of cosmic rays, from which their probable arrival rate could be calculated. The term has stuck to describe a specific probability balance equation.

System	Reaction	Propensity	Affects	Depends
Stochastic Focusing	$\emptyset \xrightarrow{k_1} I$	k_1	I	-
	$I \xrightarrow{k_2} P$	$k_2 I(t)$	I, P	I
	$P \xrightarrow{k_3} \emptyset$	$k_3 P(t)$	P	P
	$I + S \xrightarrow{k_4} S$	$k_4 I(t) S(t)$	I	I, S
	$\emptyset \xrightarrow{k_5} S$	k_5	S	-
	$S \xrightarrow{k_6} \emptyset$	$k_6 S(t)$	S	S
Stochastic Resonance	$2A + B \xrightarrow{k_1} 3A$	$k_1 A(t) \left(A(t) - 1\right) B(t)$	A, B	A, B
	$\emptyset \xrightarrow{k_2} A$	k_2	A	-
	$A \xrightarrow{k_3} \emptyset$	$k_3 A(t)$	A	A
	$\emptyset \xrightarrow{k_4} B$	k_4	B	-

Table 2.1: *Propensity functions for reactions appearing in Chapter 2. The notation $S(t)$ denotes the number of molecules of species S at time t.*

interval $[t,\ t + \mathrm{d}t\ )$. The probabilities that *no* reaction occurs in $[t,\ t + \mathrm{d}t\ )$, *exactly one* reaction occurs or *more than one* reactions occur are mutually exclusive and so can be combined additively (by the laws of probability):

$$\begin{aligned}
\mathbb{P}\left(\mathbf{v}, t + dt | \mathbf{v}_0, 0\right) &= \mathbb{P}\left(\mathbf{v}, t | \mathbf{v}_0, 0\right) \left[1 - \sum_{i=1}^{M} \alpha_i(\mathbf{v}) dt + o(dt)\right] \\
&+ \sum_{i=1}^{M} \mathbb{P}\left(\mathbf{v} - \boldsymbol{\nu}_i, t | \mathbf{v}_0, 0\right) \left[\alpha_i(\mathbf{v} - \boldsymbol{\nu}_i) dt + o(dt)\right] \\
&+ o(dt)
\end{aligned}$$

where $\boldsymbol{\nu}_i$ is the change to the system initiated by one reaction R_i so that the state vector $\mathbf{v}(t)$ will go to state $\mathbf{v}(t + \mathrm{d}t)$ as required. As a shorthand we introduce the notation $p(\mathbf{v}) = \mathbb{P}\left(\mathbf{v}, t | \mathbf{v}_0, 0\right)$ to denote the probability that the system is in state $\mathbf{v}$ at time t. Subtracting this probability from each side, dividing by dt and letting dt go to zero we arrive at the chemical master equation

$$\frac{\partial p(\mathbf{v})}{\partial t} = \sum_{i=1}^{M} \left[\alpha_i(\mathbf{v} - \boldsymbol{\nu}_i) p(\mathbf{v} - \boldsymbol{\nu}_i) - \alpha_i(\mathbf{v}) p(\mathbf{v})\right] \tag{2.9}$$

Looking at the Paulsson system we see that for $\mathbf{v} = [I, S, P]$ the master equation is

$$\begin{aligned}
\frac{\partial p(I,S,P)}{\partial t} &= k_1\, p\,(I-1,S,P) - k_1\, p\,(I,S,P) \\
&+ k_2(I+1)\, p\,(I+1,S,P-1) - k_2 I\, p\,(I,S,P) \\
&+ k_3(P+1)\, p\,(I,S,P+1) - k_3 P\, p\,(I,S,P) \\
&+ k_4(I+1)S\, p\,(I+1,S,P) - k_4 IS\, p\,(I,S,P) \\
&+ k_5\, p\,(I,S-1,P) - k_5\, p\,(I,S,P) \\
&+ k_6(S+1)\, p\,(I,S+1,P) - k_6 S\, p\,(I,S,P)
\end{aligned}$$

The non-linear term makes an analytic solution of this equation difficult (we shall see a method for resolving this kind of problem in Chapter 4), so we turn our efforts to the numerical simulation of algorithms equivalent to master equations.

Chapter 3

Stochastic Simulation Algorithms

We introduce three algorithms that produce simulations equivalent to solving the master equation. In Section 3.1 we present the original exact stochastic simulation algorithm due to Gillespie[11]. In Section 3.2 we present refinements to this algorithm, first by Gillespie and then by Gibson and Bruck[10], the algorithm we shall use for the remainder of the dissertation.

3.1 The Gillespie Algorithm

The master equation has the virtue of being an exact formulation of the evolution of a system through time, but is solvable only in simple cases, and is of little practical use when there are many reactions or the reactions are non-linear. For example, a model for the development of the virus phage[†] λ studied by Arkin et al. [3] contains 75 chemical reactions involving 57 species. Even writing down the master equation would be a formidable problem, let alone solving it.

Instead of trying to solve for *all* possible trajectories by the master equation, we can simulate a *particular* evolution of a system by using a stochastic algorithm developed by Gillespie[11] in 1977 that is exact in the sense that one simulation is a particular realisation of a random evolution trajectory governed by the master equation. Thus repeated execution of the exact algorithm will give an average that corresponds to the solution of the master equation.

The rationale of the algorithm is this. Suppose that we have a system of N different chemicals which we denote A_j, $j = 1, \ldots, N$ which are subject to the reactions R_i, $i =$

[†] λ is a virus consisting of a single strand of DNA wrapped in a protein. When it infects an *Escherichia coli* bacterium, it either reproduces until the host cell bursts (phage) or it inserts itself into the cell nucleus where it lies dormant (prophage). The host cell reproduces the λ DNA, which is reactivated at a later point by the action of ultra-violet light. Further details can be found in [22].

$1, \ldots, M$. At a given time, we ask two questions: at what time does the next reaction occur, and which reaction is it? The probability that any reaction occurs in the time interval $[t+\tau, t+\tau+d\tau)$ is given by (see [8] for a straightforward derivation)

$$P(\tau)\,\mathrm{d}\tau = \alpha_0 \exp(-\alpha_0\tau)\mathrm{d}\tau. \tag{3.1}$$

where α_0 is the sum of the individual propensities

$$\alpha_0 = \sum_{k=1}^{M} \alpha_k. \tag{3.2}$$

Suppose that $\tau \in [0, \infty)$ is the random number such that $t+\tau$ is the time that the next reaction occurs given the state $\mathbf{v}(t)$ at t. It is a random number distributed according to (3.1). Then

$$F(\tau) = \exp(-\alpha_0\tau) \tag{3.3}$$

is a random number uniformly distributed in $(0,1)$ (see [8]). Hence to find τ, we can generate a (uniformly distributed) random number r_1 for $F(\tau)$ and then solve (3.3) for τ to find that the time until the next occurrence of a reaction is

$$\tau = \frac{1}{\alpha_0}\ln\left[\frac{1}{r_1}\right]. \tag{3.4}$$

Since the reactions partition the interval $(0,1)$ according to the size of their propensity function, we can decide which reaction has occurred at $t+\tau$ by generating a second random number r_2 uniformly distributed in $(0,1)$ and deciding in which partition r_2 lies. So R_μ is the reaction which occurs at the time $t+\tau$ where μ is the number for which

$$r_2 \geq \frac{1}{\alpha_0}\sum_{k=1}^{\mu-1}\alpha_k \text{ and } r_2 < \frac{1}{\alpha_0}\sum_{k=1}^{\mu}\alpha_k. \tag{3.5}$$

Hence we can summarise the Gillespie algorithm as follows:

GILLESPIE DIRECT ALGORITHM

1. Initialise: set the initial molecule copy numbers, set time $t = 0$.
2. Calculate the propensity function α_i for each reaction, and the total propensity according to equation (3.2), $i = 1, \ldots, M$.
3. Generate two uniformly distributed random numbers r_1 and r_2 from the range $(0, 1)$.
4. Compute the time τ to the next reaction using equation (3.4).
5. Decide which reaction R_μ occurs at the new time using equation (3.5).
6. Update the state vector $\mathbf{v}$ by adding the update vector $\boldsymbol{\nu}_\mu$:
$\mathbf{v}(t+\tau) = \mathbf{v}(t) + \boldsymbol{\nu}_\mu$
7. Set $t = t + \tau$. Return to step 2 until t reaches some specified limit t_{MAX}.

Each execution of the Gillespie algorithm will produce a calculation of the evolution of the system. However, any one execution is only a probabilistic simulation, and the chances of being the same as a particular reaction is vanishingly small. Therefore to garner any useful information from the algorithm, it should be run many times in order to calculate a stochastic mean and variance that tells us about the behaviour of the system.

3.2 The Gibson-Bruck Next Reaction Algorithm

The Gillespie algorithm is an elegant and useful tool, but it suffers from the drawback that as the the computational time increases with the number of reactions, the so-called “curse of dimensionality”. This is aggravated by the need for two random numbers at each step, since pseudo-random number generation is a relatively slow process. To get around this we use an algorithm developed by Gibson and Bruck. To introduce it we first present Gillespie’s second algorithm, the so-called First Reaction Method[11], which answers the “which reaction occurs next” question as follows. Let $\mathbb{P}(R_i, \tau)\mathrm{d}\tau$ be the probability that the reaction R_i occurs in the time interval $[t+\tau, t+\tau+d\tau)$ provided that no other reaction has occurred beforehand. It has the density

$$P\left(R_i, \tau\right) \mathrm{d}\tau = \alpha_i \mathrm{exp}(-\alpha_i \tau)\mathrm{d}\tau \tag{3.6}$$

where α_i is the propensity function of the reaction R_i. Since the reactions are mutually exclusive, we can use the same argument as in the previous section to arrive at a formula for the relative time to the next occurrence of each reaction

$$\tau_i = \frac{1}{\alpha_i} \mathrm{ln}\left[\frac{1}{r_i}\right] \qquad i = 1, \ldots, M, \tag{3.7}$$

using a random number r_i for each reaction. Then we can select the reaction with the lowest reaction time to update the system.

GILLESPIE FIRST REACTION ALGORITHM

1. Initialise: set the initial molecule copy numbers, set time $t = 0$.
2. Calculate the propensity function α_i for each reaction, $i = 1, \ldots, M$.
3. For each i generate a putative time τ_i to the next occurrence of R_i according to equation (3.7).
4. Let R_μ be the reaction with the least τ_i, $i = 1, \ldots, M$.
5. Update the state vector $\mathbf{v}$ by adding the update vector $\boldsymbol{\nu}_\mu$:
   $\mathbf{v}(t + \tau_\mu) = \mathbf{v}(t_\mu) + \boldsymbol{\nu}_\mu$
6. Set $t = t + \tau_\mu$. Return to step 2 until t reaches some specified limit t_{MAX}.

This is quicker than the Gillespie direct algorithm since it saves on a great deal of looping, but it introduces many redundant calculations and requires a large number of random numbers to be generated. This is undesirable because of accuracy (see the note in Section 3.2.1) and because it is relatively slow; both versions of the Gillespie algorithm are $O(M)$. A more efficient algorithm has been presented by Gibson and Bruck [10], which we will use for our simulations. First we need to introduce two data structures.

The *dependency graph* is a directed graph with a reaction at each vertex. There is an edge between two reactions R_1 and R_2 only if the intersection of Affects$(R_1) \cap$ Depends(R_2) is not null. Thus each R_i has a set of edges that start at the R_i and terminate at those reactions whose propensity is changed when R_i occurs. When a reaction occurs the only propensity functions that need to be updated are those in this set, minimising the number of operations required to update the system. A dependency graph for the Paulsson system of reactions can be seen in Figure 3.1.

An *indexed priority queue* is a tree structure of floating point values such that each node has a smaller value than its two child nodes. This means that the smallest value is in position zero. The Gibson-Bruck algorithm stores the absolute time to the next occurrence of reaction in this structure, negating the need that the Gillespie first reaction algorithm has to find the minimum putative reaction time. We outline the algorithm for a system with M reactions (note that we distinguish relative times to the next reaction τ from absolute times t):

GIBSON-BRUCK NEXT REACTION ALGORITHM

1. Initialise: set initial copy numbers of molecules. Set time $t = 0$. Generate a dependency graph of reactions. Calculate the propensity function α_i for each reaction $i = 1, \ldots, M$. Draw a random putative time τ_i for each reaction from an exponential distribution using equation (3.8). Store the absolute time $t_i = t + \tau_i$ in an indexed priority queue.

2. Let R_μ be the reaction whose putative time is least (this will always be the first member of the indexed priority queue).

3. Update the molecule copy numbers according to the reaction R_μ:
$\mathbf{v}(t_\mu) = \mathbf{v}(t) + \boldsymbol{\nu}_\mu$. Set $t = t_\mu$.

4. For each edge in the dependency graph leaving the node corresponding to R_μ, update the corresponding α_i that has changed as a result of R_μ.

5. For R_μ, generate a new random number, calculate the next τ_μ (as in step 1) and set $t_\mu = t + \tau_\mu$. For the set of reactions in the dependency graph connected to μ by an edge leaving from R_μ, update the absolute time to the next reaction using equation (3.9).

6. Go to step 2 until t reaches some specified stopping time t_{MAX}.

The nub of the algorithm is this: the randomly generated *absolute* (rather than relative) time to each reaction is stored in a tree structure. After the reaction with the least time value is executed, its next reaction time is randomly generated using

$$t_i = t + \frac{1}{\alpha_i}\log\left(\frac{1}{r_i}\right). \tag{3.8}$$

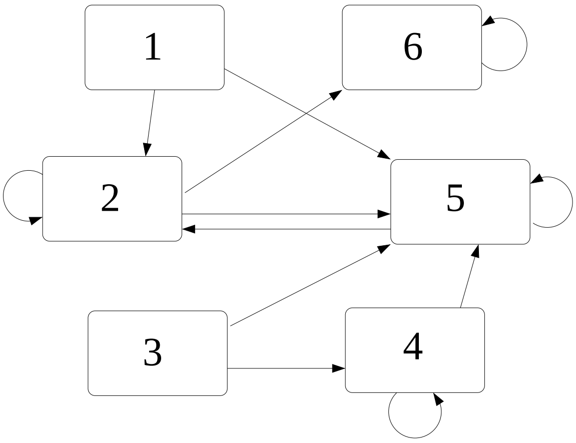

Figure 3.1: *Dependency Graph for the Paulsson system (equations* (2.1)-(2.3)*). Reactions are:* $\boxed{1}\ \emptyset \xrightarrow{k_1} I$, $\boxed{2}\ I \xrightarrow{k_2} P$, $\boxed{3}\ \emptyset \xrightarrow{k_5} S$, $\boxed{4}\ S \xrightarrow{k_6} \emptyset$, $\boxed{5}\ I + S \xrightarrow{k_4} S$ *and* $\boxed{6}\ P \xrightarrow{k_3} \emptyset$.

This is justified formally in [10]. Heuristically, we accept that we can transform from exponentially distributed relative times to exponentially distributed absolute times by adding the elapsed time t to the relative time. All the other affected absolute reaction times can be transformed to new values without requiring a new random number. This is the most innovative part of the Gibson-Bruck algorithm and needs to be justified.

To do this, suppose we are executing the algorithm at absolute time t. Now the putative next reaction time for each reaction is t_i, and the corresponding relative times are $\tau_i = t_i - t$, $i = 1, \ldots, M$. Without loss of generality suppose that reaction R_1 has occurred first, which means that t_1 is the smallest putative reaction time. This is now a sure variable and so to generate its next reaction time requires a fresh random number. Now, the remaining τ_i, $i = 2, \ldots, M$, are no longer exponentially distributed, but we have additional information about them, that $\tau_i > \tau_1$. If we could use this information to transform[1] the reaction times $\{t_2, \ldots, t_M\}$ from the interval $[t_1, \infty)$ to the set of uniformly distributed random numbers in $(0, 1)$ without introducing any bias into the sample, we could use these new values to calculate the next reaction times, since it would not matter if these numbers were randomly generated or calculated by this transformation. This was done rigorously by Gibson and Bruck. They arrive at a formula which transforms the absolute next reaction times of the reactions $R_2, \ldots, R_M$ from their current values $t_{i,old}$ to the values after

[1] This transform performs the same role as the function F in equation (3.3) of the Gillespie algorithm, which transformed the reaction times from the interval $(0, \infty)$ to (0,1).

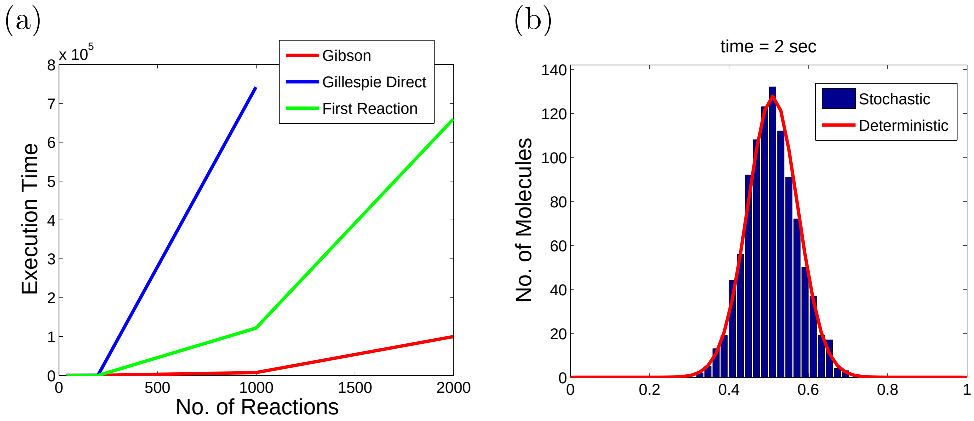

Figure 3.2: (a) *Comparative average execution times (in clock cycles) for exact stochastic simulation algorithms, using the compartment based model of diffusion of Section 4.1.* (b) *1D diffusion simulated by Gibson-Bruck algorithm (blue bars), with a finite difference solution of the diffusion equation (red line) for comparison. The initial condition is 1000 molecules at $x = 0.5mm$ and the diffusion constant is $0.001mm^2s^{-1}$.*

R_1 occurs $t_{i,new}$:

$$t_{i,new} = \frac{\alpha_{i,old}}{\alpha_{i,new}}(t_{i,old} - t) + t \tag{3.9}$$

where $\alpha_{i,old}$ is the propensity of R_i before the copy numbers are updated and $\alpha_{i,new}$ is the propensity afterwards. Those reactions that have not had their propensity changed (that is, $\alpha_{i,old} = \alpha_{i,new}$) by the occurrence of R_1 are independent of R_1 and therefore their absolute next reaction times remain unchanged.

To complete the discussion of the algorithm, we must account for the scenario where the copy number of a species falls to zero. In this case the propensity of all reactions depending on this species becomes zero and it is not possible to use equations (3.9) to calculate the time to the next reaction. This can be overcome by generating a random relative time to the next reaction using the last non-zero propensity (using equation (3.9)) and storing it in memory. Letting the absolute time to the next reaction be infinite so that the reaction drops to the bottom of the indexed priority queue, the generated reaction time will be needed only when the copy number of the species becomes non-zero once more (thanks to some other reaction). Then we transform to a new absolute time using

$$t_{i,new} = \frac{\alpha_{i,old}}{\alpha_{i,new}}(t_{i,old} - t_0) + t_0' \tag{3.10}$$

where t_0 is the time at which α_i became zero, t_0' the time at which it stopped being zero.

We implement the Gibson-Bruck algorithm* in C++, as a comparison with the Gillespie algorithm. Firstly we note that it is a much more complicated algorithm to code, as it requires the custom data structures. Secondly we note that it is designed for sparse systems of reactions. That is systems where the protagonist molecules in each reaction intersect with (at most) few other reactions. For the Paulsson system, which is not sparse, we see that the Gibson-Bruck algorithm is slower than the Gillespie algorithm. Averaging over 10 runs we find the execution times to be roughly 3 times slower. However, for a 1-D diffusion system (described in Chapter 4.1) with 78 reactions, the Gibson-Bruck algorithm takes half the execution time. We would expect the performance advantage of the Gibson-Bruck algorithm to increase with the number of reactions. Indeed this is the case and is shown in Figure 3.2(a).

To see why this is so, consider the reaction with the highest number of connecting reactions in the dependency graph (i.e. the reaction that affects the highest number of propensity functions). Let the number of edges be m. Neglecting constant terms, at each iteration of the Gibson-Bruck algorithm, m operations are required to update the m affected reactions' next reaction time. After each operation the indexed priority queue must be re-sorted. Since it is held as a binary tree, the maximum number of operations to complete this task is $\log_2 m$. Thus the Gibson-Bruck algorithm completes in $O\left(m \log_2 m\right)$ time. If $m \ll M$ the Gibson-Bruck algorithm gives a considerable speed-up over the Gillespie algorithms. For the diffusion systems we will study now, m does not grow with M: for a 2D system it is at most 8. As M is typically of the order 10^3 or greater, the Gibson-Bruck algorithm is ideal.

3.2.1 A Note on Random Numbers

The algorithms used in this dissertation require large amounts of random numbers. Of course we can generate only pseudo-random numbers, and many authors (including Gibson and Bruck) caution against exhausting the capacities of the generators, the majority of which produce a sequence of random numbers with a large period. Although exhausting the period is not fatal for these algorithms (by the time a random number is repeated the system has completely changed), we must pay attention to the randomness of our numbers. Developing in C++ has the added disadvantage that quality of random numbers varies according to compiler. Press et al[24] claim that “the implementation in many, if not most, ANSI C libraries is flawed; quite a number of implementations are in the category ‘totally botched ’.”. For this reason

*All algorithms in this dissertation were coded using the GCC C++ compiler. The graphs were generated using Matlab.

we have implemented the `ran1` generator from their book with a period of about 2^{10}. Having tested the resulting distributions we are satisfied that they are sufficiently random.

For the Gibson-Bruck algorithm, which only generates one random number per iteration, this period is clearly sufficient for the simulations in this dissertation. However, to simulate a 2D reaction-diffusion on a square of 50×50 compartments using Gillespie's First Reaction algorithm would require of the order of 10^3 random numbers per iteration. We would be much less confident that the results were truly random.

Chapter 4

Stochastic Modelling of Reaction-Diffusion

We have seen that chemicals need to come into contact in order to react. In a cell's aqueous environment, this happens naturally as the chemicals diffuse throughout the cell. In this chapter we shall present a simple method for modelling diffusion. We ignore the complex geometries that can arise, for example, as diffusion occurs through the nucleus wall[13]. We also ignore any active transport that may occur in the cell and consider diffusive mechanisms only. We also assume that the diffusion constant is constant in time. This may not be a good assumption if the reactions are thermal, but it does allow us to investigate diffusion using the techniques we have already seen. In Section 4.1 we model diffusion by dividing the domain into compartments, and in Section 4.2 we shall add chemical reactions to this model to explore reaction-diffusion systems.

4.1 Compartment Based Simulation of Diffusion

One way of modelling diffusion is to divide the domain into (imaginary) compartments. Firstly suppose that we have a 1D domain of length 1mm (see Figure 4.1(b)), which we subdivide into L sections (or *compartments*), each with length $h = 1/L$ mm. Each compartment can then be considered a chemical species, with the number of molecules in the compartment as the species copy number. We assume that the molecules are well-mixed within compartments, but not necessarily well-mixed between compartments. As molecules travel they will naturally cross from one imaginary compartment to another. This "jump" can be considered as a chemical reaction. So for a particle going from compartment i to $i+1$ we have:

$$A_i \xrightarrow{d} A_{i+1}$$

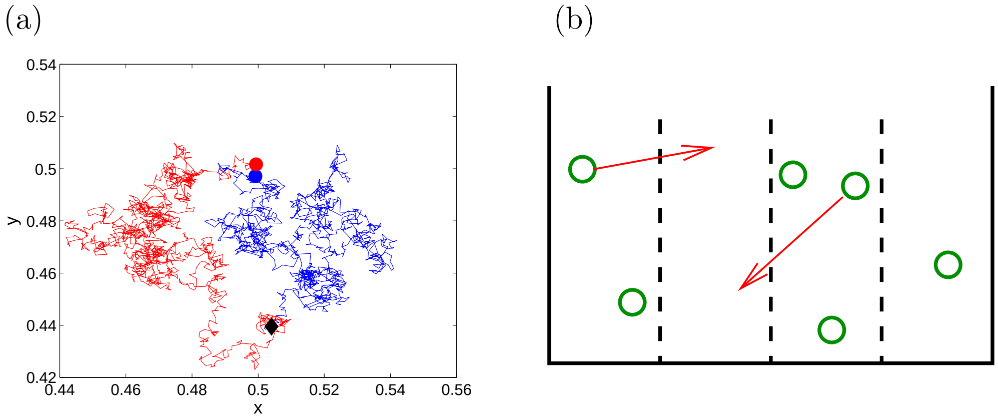

Figure 4.1: *Models of diffusion.* (a) *Two diffusing molecules with trajectories calculated by equations* (4.3)-(4.4). (b) *The computational domain divided into four (imaginary) compartments containing some (or no) molecules.*

where A_i, $i = 1, \ldots, L$ is the species representing the number of molecules in the i^{th} compartment. These reactions can be chained together to form a series of "reactions" that simulate particles diffusing throughout the domain:

$$A_1 \underset{d}{\overset{d}{\rightleftarrows}} A_2 \underset{d}{\overset{d}{\rightleftarrows}} A_3 \underset{d}{\overset{d}{\rightleftarrows}} A_4 \quad \cdots \quad \underset{d}{\overset{d}{\rightleftarrows}} A_L. \tag{4.1}$$

Furthermore, we can easily extend this model to 2D (or 3D) by dividing the domain into squares (or cubes, respectively), and allowing the molecule to jump into any of the adjacent four (six) compartments. The only parameter that needs to be matched to the physical scenario is the rate constant d. It can be shown[8] that the stochastic mean for this chain of reactions will be equivalent to a discretisation of the diffusion equation $a_t = D\nabla^2 a$ with Neumann boundary conditions when we select the rate constant to be

$$d = \frac{D}{h^2} \tag{4.2}$$

regardless of dimension. We are using $a \equiv a(\mathbf{x}, t)$ to represent the concentration of A, which we suppose to be continuous, at a point $\mathbf{x}$ in the domain at time t. Obviously we require that the results of the stochastic simulation do not depend on h. For the moment we shall accept that this is the case. We shall examine this assumption in Chapter 5.

In two dimensions, for a $[0, 1\text{mm}] \times [0, 1\text{mm}]$ domain sub-divided into square compartments of area $h \times h$ (where $h = 1/L$mm again), there are eight possible reactions for each (interior) compartment: four jumps into the compartment from adjacent compartments and four jumps out of the compartment. Neumann boundary conditions are implemented by allowing only jumps within the domain. Since each jump

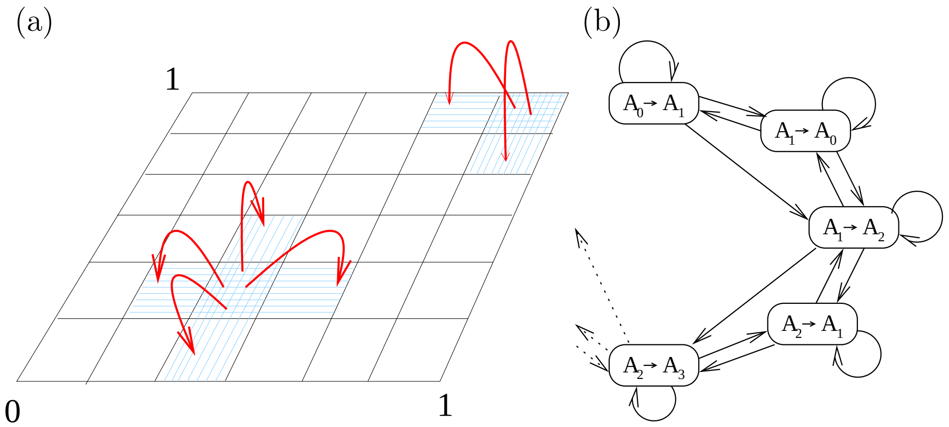

Figure 4.2: *Implementation of the Gibson-Bruck algorithm for compartment based diffusion.* (a) *Schematic of 2D compartment based diffusion model. A molecule may jump into any adjacent compartment. When it does so, the propensity of all reactions involving the shaded compartments need to be updated.* (b) *The dependency graph for 1D diffusion. The dependency graph for 2D diffusion is an extension of this picture, with each reaction involving an interior compartment will have eight edges leaving from it to the reactions it affects (including itself).*

affects the propensity of only seven other reactions (there are $4\left[L^2 - L\right]$ reactions in total in two dimensions), the dependency graph for the Gibson-Bruck algorithm is sparse, and this algorithm is ideal for simulating diffusion equations. To see that this simulation gives the expected results, we show the solution of the 1D diffusion equation in Figure 3.2(b), with a finite difference solution of the corresponding diffusion equation. In Figure 4.3 we compare the stochastic simulation of diffusion in the square domain $[0, 1\text{mm}] \times [0, 1\text{mm}]$ to a finite difference solution of the diffusion equation. The results confirm that the compartment based model is a suitable method for simulating diffusion.

We have chosen to implement Neumann boundary conditions. These are a special case of the general boundary principle. That is, when a molecule reaches the boundary, which may be a cell wall or cell nucleus in the context we are studying, the molecule will be reflected with a certain probability (which we choose to be 1) or adsorbed onto the boundary with a certain probability (which we choose to be zero). The more general case can be represented with a Robin boundary condition $D\nabla a(\mathbf{x}, t) = Ba(\mathbf{x}, t)$, where $\mathbf{x}$ lies on the boundary and the parameter B describes the reactivity of the boundary. Moreover, the reactive properties of the boundary may change as molecules reach it, for example if binding sites on the cell wall become saturated or large molecules block others from reaching the boundary. We ignore these effects and consider only reflective interactions with the boundary. To do this

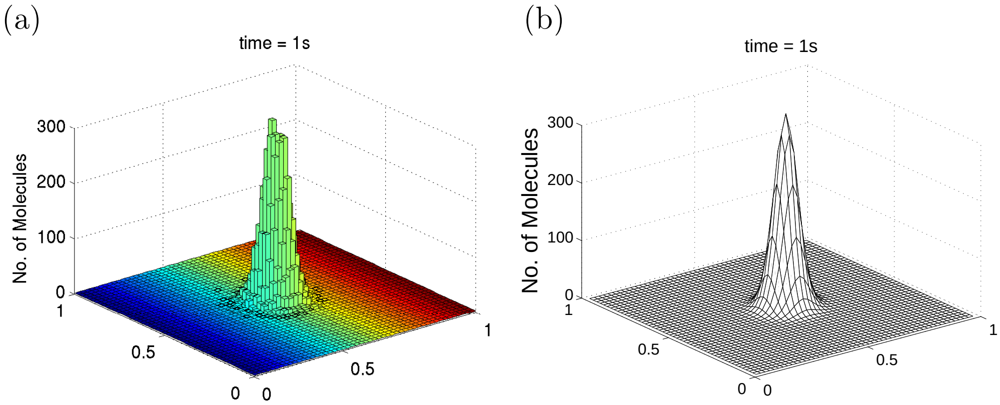

Figure 4.3: *2D Diffusion with $D = 10^{-3}\,mm^2 s^{-1}$ and initial condition 10,000 molecules at $x = y = 0.5mm$.* (a) *Simulation by dividing the domain into 50×50 compartments and evolving the compartment based diffusion model using the Gibson-Bruck algorithm.* (b) *Finite difference solution of $a_t = D\nabla^2 a$ where a is the concentration of molecules.*

we must be satisfied that the chemical reactions we are going to add in the next section do not influence the reflection. Erban and Chapman[7] have shown that the boundary conditions of stochastic models of reaction-diffusion processes are the same as the corresponding model of diffusion only. Thus we can implement the reflective boundary condition with the added chemical reactions using our diffusion model.

4.1.1 Other Models of Diffusion

There are different stochastic techniques for simulating diffusion. Here we briefly outline one based on the approach of von Smoluchowski. We calculate the trajectory of each molecule individually by stochastically calculating the component of the position of the molecule in each dimension. To do this in 2D we assume that a molecule follows a Brownian motion with its positions in time given by random variables $X(t)$ and $Y(t)$. Then under our assumption of Brownian motion the change in each co-ordinate in a timestep Δt is normally distributed with mean 0 and variance $2D\Delta t$. Hence we can update the molecule's position by generating normally distributed random numbers ξ_x and ξ_y and calculating

$$X(t + \Delta t) = X(t) + \sqrt{2D\Delta t}\xi_x \tag{4.3}$$

$$Y(t + \Delta t) = Y(t) + \sqrt{2D\Delta t}\xi_y. \tag{4.4}$$

Such trajectories for two molecules are shown in Figure 4.1(a). One problem with this approach to stochastic simulation is that it is not easy to include bimolecular chemical

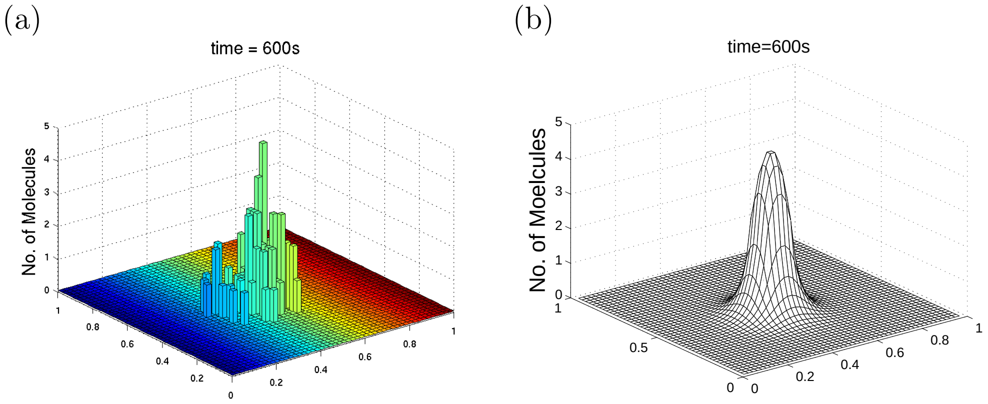

Figure 4.4: *Reaction-diffusion model.* (a) *Stochastic simulation using the Gibson-Bruck algorithm.* (b) *Numerical solution of the corresponding reaction-diffusion equation using a finite difference scheme. Details are given in the text.*

reactions in the model. To simulate a collision, it will be necessary to decide upon a physically realistic binding radius such that if two molecules are within this distance they are considered to have met, and a realistic probability of a reaction occurring once a collision has occurred. Furthermore, it will be necessary to determine that if the molecules are "close" at a particular timestep, what is the probability that they might have collided in the time since the previous timestep?

One approach to overcoming these problems is that of Andrews and Bray[2]. They define the binding radius for bimolecular reactions to be the value such that the simulated rate equals the experimental rate. Then they use the diffusion equation to calculate the binding radius σ by $\sigma = k/4\pi D$, where k is the diffusion-limited rate constant. In Figure 4.1(a) we started two molecules, one at the red dot the other at the blue. These molecules were traced until the distance between them was less than some arbitrary σ (we chose 10^{-6}mm). Then a chemical reaction could be performed (see [2] for details) in order to evolve the system. However, as the number of molecules grows this method becomes computationally expensive and for the rest of this dissertation we choose to concentrate on the compartment based approach instead.

4.2 Adding Chemical Reactions

The benefit of the compartment based simulation is that it is easy to include chemical reactions in the model. Here we show this for production and degradation. Since we already have a system of chemical reactions we can simply add more reactions to represent production:

$$\emptyset \xrightarrow{k_p} A_i \tag{4.5}$$

or degradation:

$$A_i \xrightarrow{k_d} \emptyset \tag{4.6}$$

in whichever compartments we wish. All we need to do is to update the dependency graph, which is trivial, and work out the correct rate constants for each compartment. The degradation constant k_d does not depend on the size of the compartment, only on the number of molecules within it, so it does not scale depending on the number of compartments and we may use it as it is. However, the production constant k_p depends on the area of the compartment, so we choose a macroscopic value and divide it by the number of compartments in order to obtain the correct rate for the reaction.

We demonstrate the results of a simple reaction-diffusion simulation in Figure 4.4. For this simulation, molecules were degraded according to equation (4.6) in all compartments in the domain with rate $k_d = 0.005\text{s}^{-1}$, and diffusion occurred as before. Molecules were produced according to equation (4.5) in the square with corner points $(x_1, y_1) = (0.4\,\text{mm}, 0.4\,\text{mm})$, $(x_2, y_2) = (0.5\,\text{mm}, 0.4\,\text{mm})$, $(x_3, y_3) = (0.5\,\text{mm}, 0.5\,\text{mm})$, $(x_4, y_4) = (0.4\,\text{mm}, 0.5\,\text{mm})$, with rate $\overline{k}_p = 1\text{mm}^{-2}\text{s}^{-1}$ so all compartments within this square were subject to production, degradation and diffusion. Production rate in each compartment of this square is $k_p = \overline{k}_p h^2$, where h^2 is the compartment area. Molecules outside this square were subject to degradation and diffusion only. Initially, there were no molecules in the domain.

Chapter 5

Choice of the Compartment Size h

So far we have seen efficient algorithms for stochastic simulation of systems of chemical reactions that are exact in the sense that averaging over many realisations of the algorithm gives the same result as solving the master equation. We have used the most appropriate of these, the Gibson-Bruck algorithm, to simulate reaction-diffusion processes using the compartment based approach. However, we now need to convince ourselves that the compartment size - which is a contrivance that we have used to investigate the systems - does not affect the results that we have obtained. Intuitively, we would expect that if the compartment is "too large" then the molecules within would not be well mixed enough to justify our assumptions. This is analogous to solving PDEs numerically for reaction-diffusion problems: to achieve decent spatial resolution it is necessary to consider sufficient number of grid points. In a similar fashion, for stochastic simulation it is necessary to have a sufficiently large number of compartments. Thus there is a restriction on the compartment size h from above. However, unlike PDEs, there may also be a restriction from below. In this chapter we shall demonstrate the conditions in which this may be so.

We shall consider two model problems. Model problem 1 is a linear problem. We consider a chemical A in the cubic domain of volume $V = 1\text{mm}^3$ subject to two chemical reactions:

$$\emptyset \xrightarrow{k_p} A \qquad\qquad A \xrightarrow{k_{ld}} \emptyset \tag{5.1}$$

where the rate constants have values $k_p = 1\text{mm}^{-3}\text{s}^{-1}$ and $k_{ld} = 0.1\text{s}^{-1}$. This model will be analysed in Chapter 5.1. Model problem 2 is a non-linear problem. Again, we consider a chemical A in the cubic domain of volume $V = 1\text{mm}^3$. The reaction equations for this problem are

$$\emptyset \xrightarrow{k_p} A \qquad\qquad A + A \xrightarrow{k_{nd}} \emptyset \tag{5.2}$$

where the rate constants have values $k_p = 1\text{mm}^{-3}\text{s}^{-1}$ and $k_{nd} = 0.1\text{mm}^3\text{s}^{-1}$. We analyse this problem in Chapter 5.2. In both cases we will discover that there are approximately 10 molecules on average in the domain. In Chapter 5.3 we shall use these models to test the dependence of the results of the compartment based approach on the compartment size h. To do this, we shall divide the domain into many compartments and consider the diffusion jumps between them. We shall then compute the average number of molecules in the domain. Ideally, this number should be constant (approximately 10) regardless of h, but we shall see that this may not necessarily happen for the non-linear problem.

5.1 Analysis of Model Problem 1

Let $p(n,t)$ be the probability that there are n molecules of A present in the domain at time t. The time evolution of $p(n,t)$ is given by the master equation (see equation(2.9)):

$$\frac{\partial p(n,t)}{\partial t} = k_{ld}\left\{(n+1)p(n+1,t) - np(n,t)\right\} + k_p\left\{p(n-1,t) - p(n,t)\right\}. \tag{5.3}$$

The mean number of molecules in the domain at time t is given by the stochastic mean M:

$$M(t) = \sum_{n=0}^{\infty} np(n,t). \tag{5.4}$$

The variance in the number of molecules in the domain at time t is given by :

$$V(t) = \sum_{n=0}^{\infty} (n - M(t))^2 p(n,t). \tag{5.5}$$

Multiplying (5.3) through by n and summing over n we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{n=0}^{\infty} np(n,t) &= k_{ld}\sum_{n=0}^{\infty} n(n+1)p(n+1,t) - k_{ld}\sum_{n=0}^{\infty} n^2 p(n,t)\\ &+ k_p\sum_{n=1}^{\infty} np(n-1,t) - k_p\sum_{n=0}^{\infty} np(n,t),\end{aligned}$$

then changing the indices so that we may collect the terms[8] we see that

$$\frac{\mathrm{d}M}{\mathrm{d}t} = -k_{ld}M + k_p. \tag{5.6}$$

The mean number of molecules at steady state is defined as M_s

$$M_s = \lim_{t\to\infty} M(t). \tag{5.7}$$

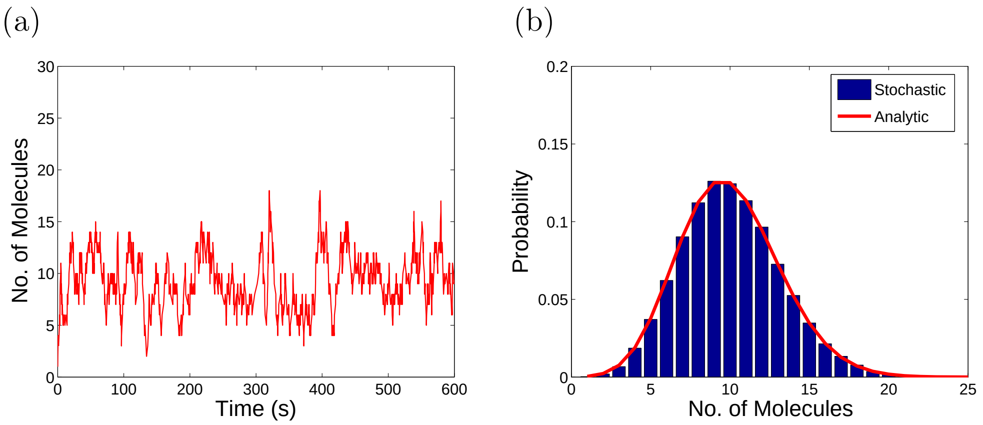

Figure 5.1: *The linear production-degradation system* (5.1) *with* $k_p = 1mm^{-3}s^{-1}$ *and* $k_{ld} = 0.1mm^3s^{-1}$*.* (a) *Time evolution of the number of molecules of* A*.*(b) *Stationary probability distribution calculated analytically[8] (red line) and the histogram computed by the stochastic simulation (blue bars).*

Similarly, we can define the steady state stochastic variance V_s as

$$V_s(t) = \lim_{t\to\infty} V(t).$$

Using (5.6) and a similar procedure for the variance[8], we find that

$$M_s = \frac{k_p}{k_{ld}} \qquad \text{and} \qquad V_s = \frac{k_p}{k_{ld}}. \tag{5.8}$$

For our parameter values $k_p = 1\text{mm}^{-3}\text{s}^{-1}$ and $k_{ld} = 0.1\text{s}^{-1}$ we find that $M_s = V_s = 10\text{mm}^{-3}$. In Figure 5.1(a) we present the time evolution of this linear model system (5.1), and in Figure 5.1(b) we compare the stationary distribution calculated by stochastic simulation with the analytic one derived from the master equation[8]. We see that there is excellent agreement between the two approaches.

5.2 Analysis of Model Problem 2

Now we work out the master equation for the non-linear system (5.2). Again, let $p(n, t)$ be the probability that there are n molecules of A present in the domain at time t. We see that for there to be n molecules of A present at time $t + dt$, then either there were n molecules at time t and neither reaction has occurred, or there were $n + 2$ molecules at time t and the second reaction has occurred, or there were $n - 1$ molecules at time t and the first reaction has occurred. Since no other scenarios are possible, the probability that there are n molecules of A present at time $t + dt$ is the sum

$$\begin{aligned} p(n,t+dt) &= p(n,t)\left[1-n\left(n-1\right)k_{nd}dt-k_p dt\right] \\ &+ p(n+2,t)k_{nd}\left(n+2\right)\left(n+1\right)dt \\ &+ p(n-1,t)k_p dt. \end{aligned}$$

Using simple algebra we find

$$\begin{aligned} \frac{p(n,t+dt)-p(n,t)}{dt} &= -k_{nd}\,n\left(n-1\right)p(n,t)-k_p p(n,t) \\ &+ k_{nd}\left(n+2\right)\left(n+1\right)p(n+2,t)+k_p p(n-1,t). \end{aligned}$$

Letting dt tend towards zero we derive the master equation

$$\begin{aligned} \frac{\partial p(n,t)}{\partial t} &= k_p\left[p(n-1,t)-p(n,t)\right] \\ &+ k_{nd}\left[\left(n+2\right)\left(n+1\right)p(n+2,t)-n\left(n-1\right)p(n,t)\right]. \end{aligned} \tag{5.9}$$

This gives us a system of equations that we could use to work out the evolution of the system. However, the non-linearity makes equation (5.9) analytically intractable. Although we cannot derive the time evolution for the stochastic mean $M(t)$ defined by equation (5.6), we can find an analytic formula for the steady state mean M_s given by (5.7) using the method outlined in [6]. We introduce the probability generating function F:

$$F(z,t)=\sum_{n=0}^{\infty}z^n p(n,t) \tag{5.10}$$

Multiplying the master equation (5.9) by z^n and summing over $n \geq 0$ we see that:

$$\begin{aligned} \frac{\partial}{\partial t}\sum_{n=0}^{\infty}z^n p(n,t) &= k_p\sum_{n=1}^{\infty}z^n p(n-1,t)-k_p\sum_{n=0}^{\infty}z^n p(n,t) \\ &+ k_{nd}\sum_{n=0}^{\infty}(n+2)(n+1)z^n p(n+2,t)-k_{nd}\sum_{n=0}^{\infty}n(n-1)z^n p(n,t). \end{aligned}$$

Changing indices we see that

$$\begin{aligned} \frac{\partial}{\partial t}\sum_{n=0}^{\infty}z^n p(n,t) &= k_p\left(z\sum_{n=0}^{\infty}z^n p(n,t)-\sum_{n=1}^{\infty}z^n p(n,t)\right) \\ &+ k_{nd}\left(\sum_{n=0}^{\infty}n(n-1)z^{n-2}p(n,t)-z^2\sum_{n=0}^{\infty}n(n-1)z^{n-2}p(n,t)\right). \end{aligned}$$

Using the definition (5.10) and differentiating it twice we see that

$$\frac{\partial F}{\partial t}=-k_p\left(1-z\right)F+k_{nd}\left(1-z^2\right)\frac{\partial^2 F}{\partial z^2}. \tag{5.11}$$

This PDE has no obvious solution, but we are interested in the steady state solution, which is given by the ordinary differential equation in z:

$$F_s''(z) - \frac{k_p}{k_{nd}} \frac{1-z}{1-z^2} F_s(z) = 0, \tag{5.12}$$

where $F_s \equiv F_s(z)$ is

$$F_s(z) = \lim_{t\to\infty} F(z,t) = \sum_{n=0}^{\infty} z^n p_s(n),$$
$$p_s(n) = \lim_{t\to\infty} p(n,t),$$

and the prime denotes differentiation with respect to z. Using simple algebra we see that (5.12) read as follows:

$$F_s''(z) = \frac{k_p}{k_{nd}} \frac{1}{1+z} F_s(z). \tag{5.13}$$

We show that the solution of (5.13) is given by

$$F_s(z) = C^{-1}\sqrt{1+z}\, I_1\Big(2K\sqrt{1+z}\Big)\,, \tag{5.14}$$

where I_1 is the modified Bessel function[16], and

$$K = \sqrt{\frac{k_p}{k_{nd}}}, \qquad C = \frac{1}{\sqrt{2}\, I_1\big(2\sqrt{2}K\big)}.$$

The modified Bessel function I_1 is a solution of the equation

$$x^2 I_1''(x) + x I_1'(x) - \big(x^2+1\big)\, I_1(x) = 0. \tag{5.15}$$

To verify that (5.14) is a solution of (5.13) we choose $x = 2K\sqrt{1+z}$ and substitute this value into equation (5.15):

$$\begin{aligned} 4K^2\,(1+z)\; I_1''\Big(2K\sqrt{1+z}\Big) + 2K\sqrt{1+z}\, I_1'\Big(2K\sqrt{1+z}\Big) - \\ \Big(4K^2\,(1+z)+1\Big)\, I_1\Big(2K\sqrt{1+z}\Big) &= 0 \end{aligned}$$

and dividing through by $4(1+z)^{\frac{3}{2}}$ we see that

$$\begin{aligned} K^2 \frac{1}{\sqrt{1+z}} I_1''\Big(2K\sqrt{1+z}\Big) + \frac{K}{2}\frac{1}{1+z} I_1'\Big(2K\sqrt{1+z}\Big) \\ -\frac{4K^2\,(1+z)+1}{4\,(1+z)^{3/2}} I_1\Big(2K\sqrt{1+z}\Big) = 0. \end{aligned} \tag{5.16}$$

Now differentiating $F_s(z)$ (given by (5.14)) twice with respect to z we obtain

$$F_s''(z) = \frac{K^2}{C}\frac{I_1''\big(2K\sqrt{1+z}\big)}{\sqrt{1+z}} + \frac{K}{2C}\frac{I_1'\big(2K\sqrt{1+z}\big)}{1+z} - \frac{I_1\big(2K\sqrt{1+z}\big)}{4C(1+z)^{\frac{3}{2}}}.$$

Using (5.16), we see that

$$F_s''(z) = \frac{K^2}{C\sqrt{1+z}}\, I_1\Big(2K\sqrt{1+z}\Big)$$

$$= \frac{k_p}{k_{nd}}\frac{1}{1+z}C^{-1}\sqrt{1+z}I_1\Big(2K\sqrt{1+z}\Big) = \frac{k_p}{k_{nd}}\frac{1}{1+z}F_s(z).$$

Thus we confirm that (5.14) is a solution of (5.13). To get the normalisation constant C we notice that

$$F_s(1) = \sum_{n=0}^{\infty} p_s(n) = 1.$$

Evaluating $F_s(1)$ using (5.14) we obtain

$$C = \sqrt{2}\, I_1\Big(2\sqrt{2}K\Big).$$

To find an expression for $p_s(n)$ we notice that if we allow $z = 0$ in (5.10) all the terms disappear except for the term corresponding to $n = 0$. If we differentiate m times and then evaluate at $z = 0$, all the terms disappear except for the term corresponding to $n = m$. Thus we can conclude that for the steady state

$$p_s(n) = \frac{F_s^{(n)}(0)}{n!} = C\frac{K^n}{n!}I_{n-1}(2K), \tag{5.17}$$

where I_{n-1} is the modified Bessel function of order $n-1$. Now to calculate the mean M_s for the non-linear reaction (5.2), we differentiate the moment generating function $F_s(z)$ at the point $z = 1$ to get:

$$F_s'(z)\mid_{z=1} = \left(\sum_{n=0}^{\infty} z^n p_s(n)\right)'\Bigg|_{z=1}$$

$$= \left(\sum_{n=0}^{\infty} n z^{n-1} p_s(n)\right)\Bigg|_{z=1} = \sum_{n=0}^{\infty} n p_s(n) = M_s.$$

Hence using the formula for F_s given by (5.14)

$$M_s = F_s'(1) = \frac{1}{4} + \frac{2\sqrt{2}}{4}\,\frac{I_1'\big(2\sqrt{2}K\big)}{I_1\big(2\sqrt{2}K\big)}. \tag{5.18}$$

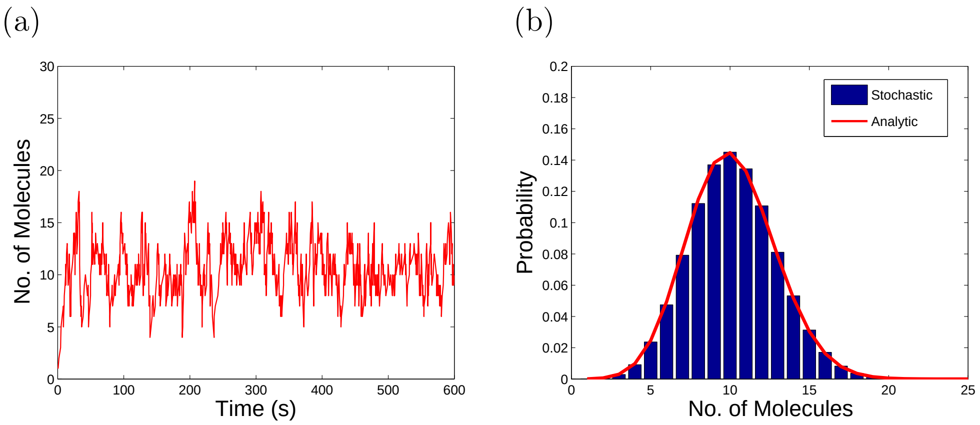

Figure 5.2: *Non-linear production-degradation system* (5.2) *with* $k_p = 1mm^{-3}s^{-1}$ *and* $k_{nd} = 0.005mm^3s^{-1}$. (a) *Time evolution of the system.* (b) *Stationary probability distribution calculated analytically using equation* (5.17) *(red line) and the histogram computed by the stochastic simulation (blue bars).*

From our definition (5.5) we see that the steady state variance is

$$\begin{aligned} V_s(t) &= \sum_{n=0}^{\infty} (n - M_s)^2 \, p_s(n) \\ &= \sum_{n=0}^{\infty} n^2 p_s(n) + M_s^2 \sum_{n=0}^{\infty} p_s(n) - 2M_s \sum_{n=0}^{\infty} n p_s(n) \\ &= \sum_{n=0}^{\infty} n^2 p_s(n) - M_s^2. \end{aligned}$$

Differentiating the moment generating function twice and evaluating at $z = 1$ we see that

$$\begin{aligned} F_s''(1) &= \sum_{n=0}^{\infty} n(n-1)p_s \\ &= \sum_{n=0}^{\infty} n^2 p_s(n) - M_s. \end{aligned}$$

Hence we see that

$$V_s = F_s''(1) + M_s - M_s^2. \qquad (5.19)$$

Using Matlab's `besseli` function and the identity $2I_1' = I_0 + I_1$ we can easily calculate the mean and variance for this non-linear reaction. For $k_{nd} = 1\text{mm}^3\text{s}^{-1}$ and $k_p = 0.005\,\text{mm}^{-3}\text{s}^{-1}$ as before we calculate that $M_s = 10.13\,\text{mm}^{-3}$ and $V_s = 7.57\,\text{mm}^{-3}$. Figure 5.2(a) shows that the value of M_s agrees well visually with the results from a Gibson-Bruck simulation. To test this, we sampled data points from the evolution of

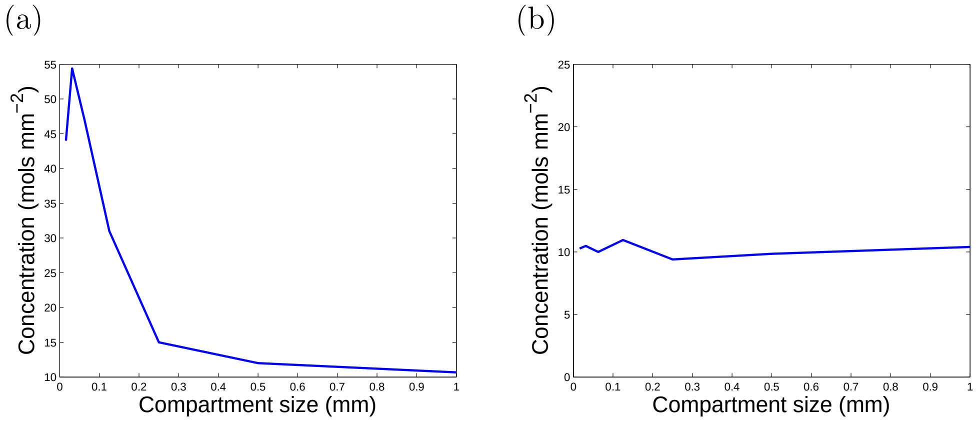

Figure 5.3: *Concentration of A molecules at steady state of reaction system* (5.2) *depends upon compartment size h.* (a)*concentration as a function of h* (b) *for very small compartments ($h = 1/64$), because the propensity of production is so small, the value in compartments is either 1, because the propensity of degradation is still zero, or 0, where either no production has occurred or two molecules have been produced and quickly degraded.*

a system that had already reached steady state by sampling the number of molecules of A every 10 seconds and calculating the mean. We found that the mean converged to $10.13\,\text{mm}^{-3}$ and the variance was $7.57\,\text{mm}^{-3}$, as expected. It is important to note that this is a different equilibrium value than that obtained from the ODE for the number of A molecules

$$\frac{\mathrm{d}A}{\mathrm{d}t} = k_{nd} - 2k_pA^2$$

derived using the law of mass action. This would predict that exactly 10 molecules should exist in the steady state. We see that, unlike the linear Model Problem 1, this is not the case.

Figure 5.2(b) compares the stochastically calculated stationary probability distribution (obtained by sampling 10^6 data points from one realisation at steady state, counting the number of occurrences of each copy number and dividing by the number of data points) with the analytic Equation (5.17). We now test the accuracy of the compartment model over a 2D domain.

5.3 Investigating the Effect of Compartment Size

To test the effect of compartment size we start with the linear model problem (5.1). We divide the cubic domain into L^3 compartments of volume $h^3 = 1/L^3\,\text{mm}^3$. Then the problem (5.1) becomes (in each compartment)

$$\emptyset \xrightarrow{k_p h^3} A_{i,j,k} \qquad\qquad A_{i,j,k} \xrightarrow{k_{ld}} \emptyset \tag{5.20}$$

where $A_{i,j,k}$ is the "chemical species" in the compartment $[(i-1)h, ih] \times [(j-1)h, jh] \times [(k-1)h, kh]$, $i = 1, \dots, L$, $j = 1, \dots, L$, $k = 1, \dots, L$. Moreover, we have to also consider diffusion jumps between each compartment with the rate D/h^2, where D is the macroscopic diffusion constant. Let $M_{i,j,k}$ be the mean number of molecules in the compartment $[(i-1)h, ih] \times [(j-1)h, jh] \times [(k-1)h, kh]$. Following [8], we can write the master equation for this model and use it to obtain the evolution equation for $M_{i,j,k}$ in the following form (away from the boundary):

$$\begin{aligned} \frac{\partial M_{i,j,k}}{\partial t} &= \frac{D}{h^2}\Big(M_{i+1,j,k} + M_{i-1,j,k} + M_{i,j+1,k} + M_{i,j-1,k} + M_{i,j,k+1} + M_{i,j,k-1} \\ &- 6M_{i,j,k}\Big) + k_p h^3 - k_{ld} M_{i,j,k}. \end{aligned}$$

We are interested in the stationary state

$$M^s_{i,j,k} = \lim_{t\to\infty} M_{i,j,k}(t).$$

It satisfies

$$\begin{aligned} 0 &= \frac{D}{h^2}\Big(M^s_{i+1,j,k} + M^s_{i-1,j,k} + M^s_{i,j+1,k} + M^s_{i,j-1,k} + M^s_{i,j,k+1} + M^s_{i,j,k-1} - 6M^s_{i,j,k}\Big) \\ &+ k_p h^3 - k_{ld} M^s_{i,j,k}. \end{aligned}$$

Solving this equation we see that the average number in any compartment at the steady state is

$$M^s_{i,j,k} = \frac{k_p h^3}{k_{ld}}.$$

Therefore in the whole domain there are

$$M^s(h) = \sum_{i=1}^{L}\sum_{j=1}^{L}\sum_{k=1}^{L} M^s_{i,j,k} = \sum_{i=1}^{L}\sum_{j=1}^{L}\sum_{k=1}^{L} \frac{k_p h^3}{k_{ld}} = L^3 \frac{k_p h^3}{k_{ld}} = \frac{k_p}{k_{ld}}$$

molecules. Thus we conclude that the number of molecules in the compartment $M^s(h)$ is independent of h and equal to (5.8) for any value of D. We conclude that there is no restriction on h for a linear model.

We repeat a similar exercise for the non-linear model (5.2). Again we divide the cubic domain into L^3 compartments of volume $h^3 = 1/L^3\,\text{mm}^3$. Then the problem (5.2) becomes (in each compartment)

$$\emptyset \xrightarrow{k_p h^3} A_{i,j,k} \qquad\qquad A_{i,j,k} + A_{i,j,k} \xrightarrow{k_{nd} h^{-3}} \emptyset \tag{5.21}$$

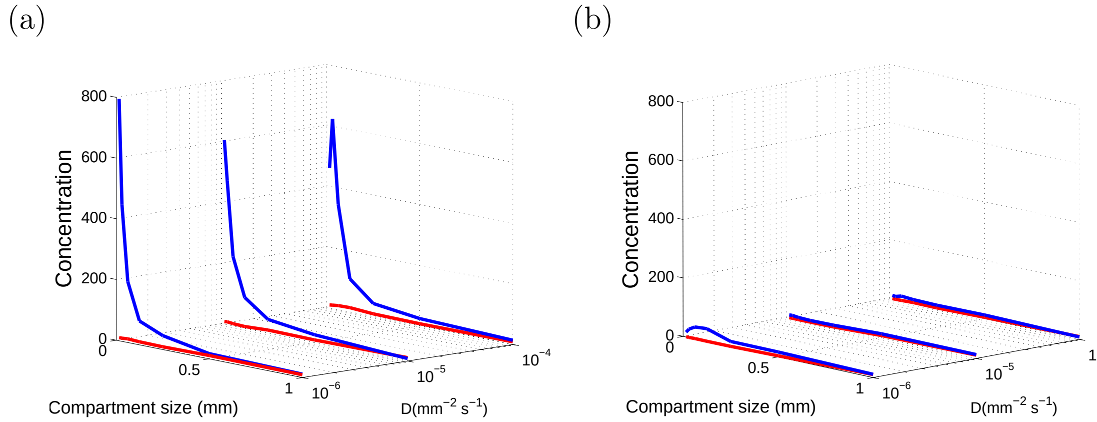

Figure 5.4: *Concentration of A molecules at steady state of compartment based reaction-diffusion systems varies with D. Red lines indicate the linear system* (5.1) *(*$k_p = 1$*,* $k_{ld} = 0.1$*), blue lines indicate the non-linear system* (5.2) ($k_p = 1$, $k_{nd} = 0.005$) (a)k_p *scaled as* $1/h^2$*.* (b) k_p *scaled as* $1/h^3$*.*

where the notation is the same as above. Our goal is to find how $M^s(h)$ depends on h and D. We cannot use the master equation in the same manner, so we test the dependence numerically using our stochastic algorithm.

We test the h dependence by calculating the average concentration of A molecules in the domain, for a range of compartment sizes. As $h \to 0$, at steady state we expect that each small compartment will have either one or zero molecules, for if two molecules were produced the resulting non-zero propensity would cause degradation back to zero. Thus the average concentration in a compartment tends towards $1/2h^2$. We use the Gibson-Bruck algorithm to calculate the average concentration for h ranging from 2^{-7}mm to 1mm, then calculate the average concentration in the domain. We have worked in 2D to all the computations to run quickly enough to be able to take the average of many realisations. That is,

$$\frac{1}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{n_{i,j}}{h^2} = \sum_{i=1}^{L}\sum_{j=1}^{L} n_{i,j}$$

where $n_{i,j}$ is the number of molecules in the compartment in row i and column j, averaged over 10 simulations. In the absence of diffusion, the results can be seen in Figure 5.3(a). We see that below $h = 0.25$ mm the concentration is much too high, so simulations should not be attempted with compartment sizes smaller than this.

We note in passing that, as expected, these limitations do not apply to the linear case. Repeating the above experiment with the linear reaction of system (5.20) we find that the same number of molecules is produced in the domain, up to noise, regardless of h. Thus the concentration is constant for different h (see Figure 5.3(b)). This is important since the compartment based diffusion simulation is built on this pair of reactions. Thus we can have confidence that all the results obtained in Chapter 4, which all used a 50×50 grid (that is, $h = 0.02\,\text{mm}$) are accurate.

Having seen the behaviour in the absence of diffusion, we now include diffusion reactions in our tests. For the linear problem we see in Figure 5.4(a) that the linear problem behaves as we would expect, regardless of h or D. However, we see for our non-linear problem that for small h the concentration is very inaccurate. Although we note that as D increases this appears to improve slightly, as the more frequent jumps between compartments as D increases encourage more degradation reactions. We shall now turn to trying to apply the compartment based model to a more complicated system. For these production and degradation systems we have either run our simulations with zero molecules in the domain at time $t = 0$, or started with 10 molecules randomly distributed throughout the domain in order that the system should reach steady state more quickly. Bearing in mind the restrictions on h from below, henceforth we shall run our simulations with slightly higher copy number of molecules in order that the non-linear reactions occur even in small compartments.

Chapter 6

Stochastic Modelling of Pattern Formation

We finish this dissertation with an application of the previous material to biological pattern formation. We shall consider a model system of two reacting and diffusing chemical species, which we shall call the *activator* and the *inhibitor*, that displays Turing instability[18]. This mechanism for the formation of patterns in reaction-diffusion systems can be explained as follows. In a well mixed system (without diffusion) the concentrations of activator and inhibitor converge towards a steady state. However, if the inhibitor possesses a higher diffusion constant than the activator, then the homogeneous steady state is unstable, and spatial patterns may develop in the following way. A random perturbation within a high concentration of the activator introduces a local instability. As a result, the inhibitor diffuses quickly and tends to gather in an inhibiting region away from the concentration of activators. This has the effect of containing the activator in one region, which provides conditions for the process to repeat. This continues until a steady state is reached when the activator and inhibitor have separated into a pattern. Thus the system can be locally stable but globally unstable. This runs counter to our intuition about diffusion, which we usually think of as a smoothing process. This process is commonly called *diffusion driven instability.*

The classical method for modelling this phenomenon is by a pair of reaction-diffusion PDEs, and conditions for the requisite instability in the presence of spatial inhomogeneity of the concentrations can be found in [18]. Remembering the discussion in Chapter 1, we are happy to accept this model when the copy number of molecules is large. However in situations where this assumption is not valid, for example in developing embryos, we are more circumspect. Sick et al[26] have recently proposed that the reaction-diffusion model can possibly be applied to the development of hair spacing in mice, with the gene WNT playing the role of the activator and DKK the inhibitor. Since WNT is a much larger protein than DKK it should diffuse more slowly than the inhibitor. They show that these proteins satisfy the conditions for diffusion driven instability and are known to be present in murine embryos as follicles develop, and that suppression of WNT signalling increases the spacing between hair follicles. If this reaction-diffusion mechanism can be reproduced experimentally with these proteins it would be an important demonstration of a Turing pattern in nature.

Having seen the potential applications of the Turing mechanism, we shall try to simulate a particular reaction stochastically. Modelling a real biochemical process often involves determining a large number of chemical reactions and measuring their rate constants, which can be a challenging task. We shall not consider a genuine biological system. Instead we consider the following hypothetical chemical reaction system that has been explored in depth in the literature of this subject[25, 17, 27]:

$$\begin{aligned} 2U + V &\xrightarrow{k_1} 3U && (6.1)\\ \emptyset \xrightarrow{k_2} U &\xrightarrow{k_3} \emptyset && (6.2)\\ \emptyset &\xrightarrow{k_4} V && (6.3) \end{aligned}$$

where U diffuses with constant D_u and V with D_v. Note that this is the same system as the Stochastic Resonance system studied in Section 2.2. We are extending it by adding diffusion. We shall first present the analysis of this system in 2D, and then proceed to apply a compartment based stochastic simulation algorithm to it.

6.1 Analysis

We denote the concentration of U and V by $u \equiv u(\mathbf{x}, t)$ and $v \equiv v(\mathbf{x}, t)$, and we consider the evolving system in the 2D domain $[0, 1\,\mathrm{mm}] \times [0, 1\,\mathrm{mm}]$. We apply the law of mass action and diffusion to get a pair of reaction-diffusion equations:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u + k_1 u^2 v + k_2 - k_3 u && (6.4)\\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v - k_1 u^2 v + k_4 && (6.5) \end{aligned}$$

We introduce the notation $\mathbf{u} = [u, v]^T$, $\mathbf{0} = [0, 0]^T$ and the function

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} f_1(u, v) \\ f_2(u, v) \end{bmatrix} = \begin{bmatrix} k_1 u^2 v + k_2 - k_3 u \\ -k_1 u^2 v + k_4 \end{bmatrix}.$$

Ignoring diffusion, the system (6.4)-(6.5) is described by the ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u} = \mathbf{f}(\mathbf{u}) \qquad (6.6)$$

The homogeneous steady state (u_*, v_*) can be obtained by solving the system:

$$\mathbf{f}(\mathbf{u}) = \mathbf{0} \tag{6.7}$$

So we solve:

$$\begin{aligned} k_1u^2v + k_2 - k_4u &= 0 \qquad (6.8)\\ -k_1u^2v + k_4 &= 0 \qquad (6.9)\end{aligned}$$

Adding equations (6.8) and (6.9) and solving for u_* and v_* gives us the steady state values

$$u_* = \frac{k_2 + k_4}{k_3}, \qquad v_* = \frac{k_4}{k_1}\left(\frac{k_3}{k_2 + k_4}\right)^2$$

In what follows we choose the values of the rate constants to be $k_1 = 123.45\mu\mathrm{m}^4\mathrm{s}^{-1}$, $k_2 = 0.09\mu\mathrm{m}^{-2}\mathrm{s}^{-1}$, $k_3 = 2\mathrm{s}^{-1}$ and $k_4 = 0.27\mu\mathrm{m}^{-2}\mathrm{s}^{-1}$. Then the steady state is $u_* = 0.18\,\mu\mathrm{m}^{-2}$, $v_* = 0.065\,\mu\mathrm{m}^{-2}$. To examine the stability of this steady state[14], we linearise the system about these equilibria by taking the Jacobian of $\mathbf{f}$,

$$\mathcal{J}(\mathbf{u}) = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{u}} = \begin{bmatrix} 2k_1uv - k_3 & k_1u^2 \\ -2k_1uv & -k_1u^2 \end{bmatrix},$$

so that equation (6.6) is linearised by

$$\dot{\mathbf{w}} = \mathcal{J}(u_*, v_*)\mathbf{w}$$

where $\mathbf{w} = [u - u_*, v - v_*]^T$. Evaluating $\mathcal{J}(u, v)$ at (u_*, v_*) with our chosen rate constants we find that

$$\mathcal{J}(u_*, v_*) = \begin{bmatrix} 1 & 4 \\ -3 & -4 \end{bmatrix}.$$

The eigenvalues of this system are $\lambda = (-3 \pm \sqrt{23}\,i)/2$. Since the real parts are negative we conclude that the equilibrium points are stable spiral nodes.

Reintroducing diffusion with Neumann boundary conditions into our system, these equilibrium points are also a homogeneous solution of the system of PDEs (6.4)-(6.5):

$$u(\mathbf{x}, t) = u_*, \qquad v(\mathbf{x}, t) = v_*.$$

Now the linearised system (6.4)-(6.5) is

$$\dot{\mathbf{w}} = \begin{bmatrix} D_u & 0 \\ 0 & D_v \end{bmatrix} \nabla^2\mathbf{w} + \begin{bmatrix} 1 & 4 \\ -3 & -4 \end{bmatrix}\mathbf{w}.$$

Looking for the solution of this equation in cosine series

$$\mathbf{w}(x, t) = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \mathbf{a}_{k,l}(t)\,\cos(2\pi kx)\cos(2\pi ly)$$

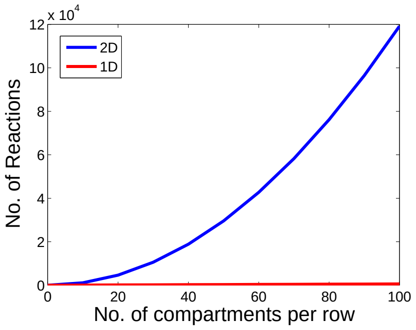

Figure 6.1: *The number of reactions for a compartment based simulation of the system* (6.1)-(6.3) *grows as* $6L-2$ *for 1D and* $8(L^2-L)+4L^2$ *for 2D*

the vector $\mathbf{a}_k$ has to satisfy the equation

$$\frac{\partial \mathbf{a}_k}{\partial t} = \left(-4\pi^2(k^2+l^2)\begin{bmatrix} D_u & 0 \\ 0 & D_v \end{bmatrix} + \begin{bmatrix} 1 & 4 \\ -3 & -4 \end{bmatrix}\right)\mathbf{a}_k. \tag{6.10}$$

We see that the steady state (u_*, v_*) will become unstable if at least one eigenvector of the matrix

$$-4\pi^2(k^2+l^2)\begin{bmatrix} D_u & 0 \\ 0 & D_v \end{bmatrix} + \begin{bmatrix} 1 & 4 \\ -3 & -4 \end{bmatrix}$$

has a positive real part for some $k, l > 0$. Writing $z = 4\pi^2(k^2+l^2)$, $c_1 = 3+z(D_u+D_v)$ and $c_2 = z^2D_uD_v + z(4D_u - D_v) + 8$ we see that the characteristic equation is

$$\lambda^2 + c_1\lambda + c_2 = 0.$$

The solution is

$$\lambda = \frac{-c_1 \pm \sqrt{c_1^2 - 4c_2}}{2}.$$

We see that the only way that λ can become positive is if $c_2 < 0$. The minimum value of λ occurs when c_2 is also minimal, so we find this c_2 by differentiating with respect to z to see that c_2 is minimal at the point

$$z_{min} = \frac{D_v - 4D_u}{2D_uD_v}$$

Substituting z_{min} into our definition of c_2 we see that the condition for a positive real part of an eigenvalue is

$$c_2(z_{min}) = \frac{(D_v-4D_u)^2}{4D_uD_v} - \frac{(D_v-4D_u)^2}{2D_uD_v} + 8 < 0.$$

Rearranging we find that

$$32\frac{D_v}{D_u} < \left(\frac{D_v}{D_u} - 4\right)^2. \tag{6.11}$$

Expanding the square and collecting the terms we find that for spatial patterns to develop we require

$$\frac{D_v}{D_u} > 39.6. \tag{6.12}$$

We shall choose $D_u = 5 \times 10^{-4}\text{mm}^2\text{s}^{-1}$ and $D_v = 0.06\text{mm}^2\text{s}^{-1}$ for our diffusion coefficients, so that $D_v/D_u = 120$. Thus condition (6.12) is satisfied, so conditions exist for the formation of a Turing pattern, which we now attempt to simulate stochastically.

6.2 Numerical Results

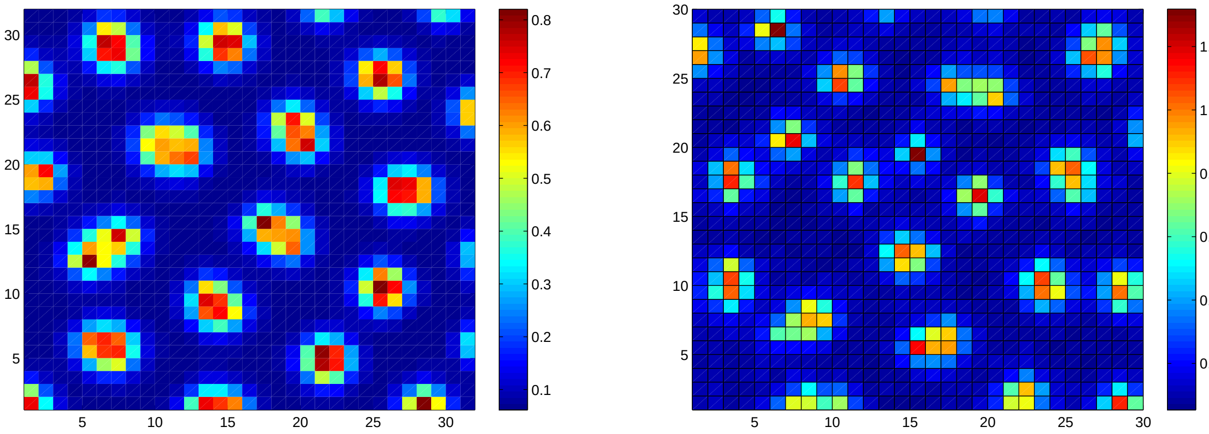

Figure 6.2: *2D Turing pattern derived from the system* (6.1)-(6.3)*. These figures show the concentration of U molecules.* (left) *deterministic solution of equations* (6.4)-(6.5)*, solved using a finite difference method on a 30 × 30 grid.* (right) *Stochastic simulation using 30 ×30 compartment model with the Gibson-Bruck algorithm.*

Now we try to use these parameters to create some patterns stochastically, using the Gibson-Bruck algorithm to simulate the reactions in our chosen system. We start with a 2D system and divide the domain $[0, 1\text{mm}] \times [0, 1\text{mm}]$ into L^2 square compartments of area $h^2 = 1/L^2$. We create all four reactions (6.1)-(6.3) in each compartment, plus diffusion jumps between adjacent compartments (with jumps restricted to the computational domain at the boundary). Denoting the numbers of molecules in the i^{th} compartment on the j^{th} row by $U_{i,j}$ and $V_{i,j}$, the reactions in each compartment are

$$2U_{i,j} + V_{i,j} \xrightarrow{k_1 h^{-4}} 3U_{i,j} \tag{6.13}$$

$$\emptyset \xrightarrow{k_2 h^2} U_{i,j} \xrightarrow{k_3} \emptyset \tag{6.14}$$

$$\emptyset \xrightarrow{k_4 h^2} V_{i,j} \tag{6.15}$$

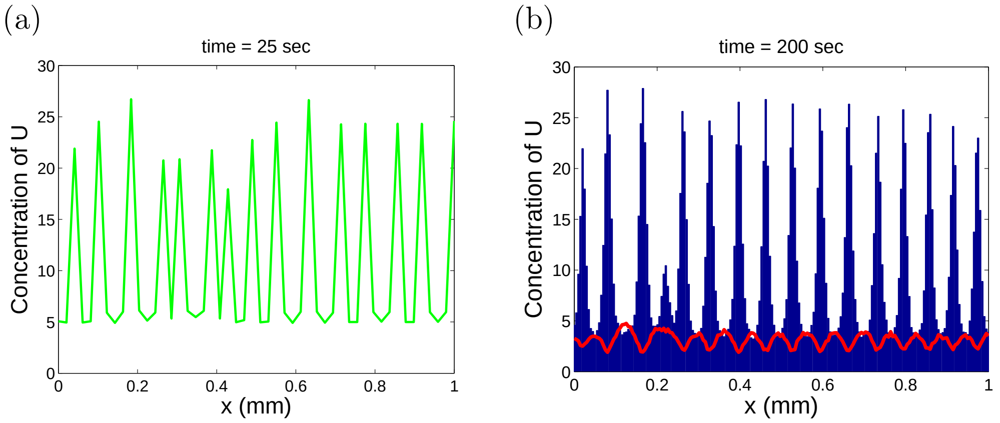

Figure 6.3: *1D Turing pattern derived from System* (6.1)-(6.3) *using 200 compartments. The blue bars show the concentration of U molecules, the red line the concentration of V molecules.* (a) *deterministic solution* (b) *Stochastic simulation using Gibson-Bruck algorithm with 200 compartment model*

In two dimensions, the number of reactions required for a compartment based simulation rises quickly with the number of compartments: Figure 6.1 shows that 1D increases linearly and 2D quadratically. For large numbers of compartments, even with the Gibson-Bruck algorithm, the simulation times are measured in days when run on PCs. Therefore we have not been able to simulate a Turing pattern with good resolution. However, we have produced a Turing pattern in 2D using a 30 × 30 compartment model. This is shown in Figure 6.2(b). The deterministic pattern, obtained by solving equations (6.4)-(6.5) using a finite difference scheme in Matlab are shown in Figure 6.2(a) for comparison.

Limited as we are in 2D, we turn to a 1D simulation for higher resolution Turing pattern. With the same system of reactions (6.13)-(6.13) we easily managed to compute the pattern arising with 200 compartments. The results are shown in Figure 6.3. We see that clearly defined peaks in the concentration of U are visible in the domain, which correspond to the dips in the concentration of V - a classic Turing pattern.

This system may be simple, but this system of chemical reactions does indeed give rise to Turing patterns, and may be part of a larger family of such systems. Tsai et al[27] have created patterns using a similar system of reactions that they call the Templator model, and suggest that there may be a class of two variable models that exhibit this behaviour.

Chapter 7

Summary and Discussion

We have shown in this dissertation that there is a need for techniques to model biochemical reactions when the copy number of molecules is small enough that the deterministic approach of the law of mass action is inadequate to deal with the non-linear reactions that arise in cells. To this end, we have formulated the master equation to study the time-evolution of systems of chemical reactions, based on chemical kinetics. We have introduced and implemented several algorithms from the literature that are exact in the sense that averaging over many realisations of an exact algorithm is equivalent to solving the master equation. We have then modelled reaction-diffusion processes using a compartment based approach using the most appropriate of them, the Gibson-Bruck algorithm. We tested the circumstances in which the compartment based model may be accurate by calculating analytically the moments of two model reactions, and comparing these figures to our numerical calculations. We found that for non-linear reactions there is a limit to the spatial resolution possible with this model if the compartments become so small that there are only single molecules in many compartments. Finally we have seen that these simulations really can produce interesting results by simulating the formation of a Turing pattern in 1D and 2D using our stochastic algorithm.

We have hinted at certain biological systems, such as gene expression and pattern formation in developmental biology where these techniques may be important. We might also point out that these techniques are not limited to biological applications. Recent advances in astrochemistry have shown that large molecules (commonly called “space dust”) exist in the interstellar medium, and that chemical reactions to form organic compounds can occur on the surface of these dust grains when atoms collide. Since the number of atoms is miniscule, deterministic models derived from mass action are quite unsuitable, and Charnley and Rodgers[4] have used the Gillespie algorithm to solve the chemical master equation to calculate abundance of certain hydrocarbons on dust grains.

However, it is in the field of cellular biology that these techniques have been most useful to date, and there are many examples from the literature that demonstrate that exact stochastic techniques have made possible many interesting experiments. Thus we can be satisfied that the techniques outlined in this dissertation are useful for the study of certain phenomena in cells, and we can be confident in their accuracy.

Bibliography

[1] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. *Molecular Biology of the Cell*, 4th ed. Garland Science, 2002.

[2] Andrews, S. S., and Bray, D. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical Biology 1* (2004), 137–151.

[3] Arkin, A., Ross, J., and McAdams, H. H. Stochastic kinetic analysis of developmental pathway bifurcation in Phage λ-infected Escherichia coli cells. *Genetics 149* (August 1998), 1633–1648.

[4] Charnley, S., and Rodgers, S. Pathways to molecular complexity. In *Astrochemistry: Recent Successes and Current Challenges* (2005), no. 231, International Astronomical Union, pp. 237–246. IAU Symposium.

[5] Einstein, A. Investigations on the theory of the Brownian movement. *Dover Publications Inc.*. translated by A.D.Cowper.

[6] Engblom, S. Computing the moments of high dimensional solutions of the master equation. *Appl.Math.Comput. 180*, 2 (2006), 498–515.

[7] Erban, R., and Chapman, S. Realistic boundary conditions for stochastic simulations of rection-diffusion processes. *Physical Biology 4*, 1 (2007), 16–28.

[8] Erban, R., Chapman, S., and Maini, P. K. A practical guide to stochastic simulations of reaction-diffusion processes. *http://arxiv.org 0704.1908* (2007).

[9] Fowler, A. C. *Mathematical Models in the Applied Sciences.* Cambridge University Press, 1997.

[10] Gibson, M. A., and Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J.Phys.Chem.A 104* (2000), 1876–1889.

[11] Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry 81*, 25 (1977), 2340–2361.

[12] Guptasarma, P. Does replication-induced transcription regulate synthesis of the myriad of low copy proteins of Escherichia coli? *Bioessays 17*, 11 (1995), 987–997.

[13] Isaacson, S. A., and Peskin, C. S. Incorporating diffusion in complex geometries into stochastic chemical kinetics simulatinos. *SIAM J.Sci.Comput. 28*, 1 (2004), 47–74.

[14] Jordan, D., and Smith, P. *Nonlinear Ordinary Differential Equations*, 3rd ed. Oxford University Press, 1999.

[15] Koshland, D. E., Goldbeter, A., and Stock, J. B. Amplification and adaptation in regulatory and sensory systems. *Science 217*, 4556 (July 1982), 220–225.

[16] Lebedev, N. *Special Functions and Their Applications.* Dover Publications Inc, 1972.

[17] Lin, K., Schirmer, S., and Camacho Wirkus, E. Chemical pattern formation in reaction-diffusion systems. Available online http://qubit.damtp.cam.ac.uk/users/sonia/research/notes/MSRI99.pdf, 1999.

[18] Markowich, P. A. *Applied Partial Differential Equations : A Visual Approach.* Springer, 2007.

[19] Muratov, C. B., Vanden-Eijnden, E., and Weinan, E. Self-induced stocastic resonance in excitable systems. *Physica D* (2005), 227–240.

[20] Nelson, E. *Dynamical Theories of Brownian motion.* Princeton University Press, 1967.

[21] Nordsieck, A., Lamb, W., and Uhlenbeck, G. On the theory of cosmic-ray showers I : The furry model and the fluctuation problem. *Physica*, 7 (1940), 344–360.

[22] Pashtne, M. *A Genetic Switch : Phage λ Revisited*, 3 ed. Blackwell Scientific Publications, 2002.

[23] Paulsson, J., Berg, O. G., and Ehrenberg, M. Stochastic focusing : Fluctuation-enhanced sensitivity of intracellular regulation. *Proc.Natl.Acad.Sci. USA 97*, 13 (June 2000), 7148–7153.

[24] Press, W. H., Teukolsky, S. A., T.Vetterling, W., and P.Flannery, B. *Numerical Recipes in C++*, 2nd ed. Cambridge University Press, 2002.

[25] Qiao, L., Erban, R., Kelly, C., and Kevrekidis, I. Spatially distributed stochastic systems: Equation-free and equation-assisted preconditioned computation. *Journal of Chemical Physics 125* (2006).

[26] Sick, S., Reinker, S., Timmer, J., and Schalke, T. WNT and DKK determine hair follicle spacing through a reaction-diffusion mechanism. *Science 314* (December 2006), 1447–1450.

[27] Tsai, L., Hutchison, G., and Peacock-Lopez, E. Turing patterns in a self-replicating mechanism with a self-complementary template. *Journal of Chemical Physics 113*, 5 (2000), 2003–2006.

[28] Van Kampen, N. *Stochastic Processes In Chemistry and Physics*, 3rd ed. North-Holland, 2007.

[29] Wiesenfeld, K., and Moss, F. Stochastic resonance and the benefits of noise: From ice ages to crayfish and SQUIDS. *Nature 373* (January 1995).