THE UNIVERSITY OF CALGARY

Stochastic Simulation of Pattern Formation: An Application of L-systems

by

Mikolaj Cieslak

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

July, 2006

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Stochastic Simulation of Pattern Formation: An Application of L-systems" submitted by Mikolaj Cieslak in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor, Dr. Przemyslaw Prusinkiewicz
Department of Computer Science

Co-supervisor, Dr. Christiane Lemieux
Department of Mathematics and Statistics
and Department of Computer Science

Dr. Sheelagh Carpendale
Department of Computer Science

Dr. Michael Surette
Department of Biochemistry and Molecular Biology

Date

Abstract

Turing proposed a mathematical model based on chemical reactions and diffusion of substances throughout the xy-plane, called reaction-diffusion, to study the emergence of patterns from a homogeneous medium [65]. Hammel and Prusinkiewicz used L-systems and reaction-diffusion in a one-dimensional medium to model the formation of sea shell patterns, and in an expanding one-dimensional medium to model heterocyst spacing in the cyanobacterium Anabaena catenula [25]. They considered the formation of these two patterns as a continuous deterministic process neglecting the noise which is inherent to such a process.

The approach taken in this work is to stochastically model reaction-diffusion in a spatial, and possibly expanding, linear structure using L-systems. The stochastic simulation method for chemical reaction kinetics which was developed by Gillespie [20] is used to study this model. On the basis of theoretical considerations, the L-system modelling language L+C [31] is extended to include a stochastic rewriting strategy based on Gillespie's algorithm.

Acknowledgements

I would like to thank both of my supervisors, Dr. Przemyslaw Prusinkiewicz and Dr. Christiane Lemieux, for helping me to find an interesting and diverse research project, and for helping me to reach a point in my research where I feel that I have made a contribution to biological modelling. The support and encouragement I received from both of them made writing this thesis an achievable goal for me. Also, their corrections and suggestions for improving my thesis were of great help.

For many helpful discussions of my work, I would like to thank Brendan Lane, Adam Runions and Richard Smith, who are all members of the Biological Modelling and Visualization Group of which I am a part. The programming work done by Radoslaw Karwowski was of great importance to my research as well. It made writing and presenting my models much easier, and was an integral part of my work.

My final thanks belong to my family and friends for patiently encouraging me to finish this thesis.

Table of Contents

List of Tables

List of Figures

List of Programs

Chapter 1

Introduction

Extensive research has been done on the mechanisms involved in the formation of biological patterns, such as mammalian coat patterns or butterfly wing patterns [48, chapters 2,3]. Much of the research has been focused on finding mathematical models to describe the formation of such patterns, and many of the proposed models have been based on a reaction-diffusion type mechanism. This is a mechanism first proposed by Turing in 1952 [65] to explain *morphogenesis*, which is the emergence of a complex form or pattern during an organism's development. In its basic form, reaction-diffusion involves two substances that interact with one another according to some chemical reactions (e.g., autocatalization of a substance) and that diffuse throughout a medium. A mathematical model of this mechanism was given by Turing as a set of differential equations, which describe the reaction kinetics and diffusion of the substances over time. Meinhardt further extended and investigated this type of model, and using it reproduced and explained a number of patterns found in nature [39]. To produce these patterns, he used numerical methods to solve the differential equations that represent reaction-diffusion.

The focus of the research presented in this thesis is on another computational approach for modelling pattern formation using reaction-diffusion. Two models of pattern formation are examined in this thesis: the pigmentation patterns on sea shells and the spacing of heterocysts (specialized cells) in the bacterium *Anabaena catenula*. A continuous differential equation model for both of these patterns has

been used by Fowler, Meinhardt, and Prusinkiewicz for the former [14], and by Hammel and Prusinkiewicz for the latter [25]. The problem with this approach is that it does not account for stochastic effects present in the formation of such patterns, and without considering these effects it is hard to justify the ability of the model to produce the desired pattern with respect to noise (i.e. the robustness of the model [70]). One possible solution is to model the reaction-diffusion mechanism as a (continuous time) discrete-event stochastic process, where the number of molecules of each substance is a random variable that changes according to the mechanism. The model can then be studied using one of several approaches: a stochastic simulation approach (Markov process), a τ-leaping approach (Poisson process), or a Langevin type stochastic differential equation approach (Gaussian process) [67]. Figure 1.1 shows these modelling possibilities in a discrete to continuous relation from left to right. The research presented in this thesis examines the reaction-diffusion mechanism in a growing structure using stochastic simulation (as a Markov process).

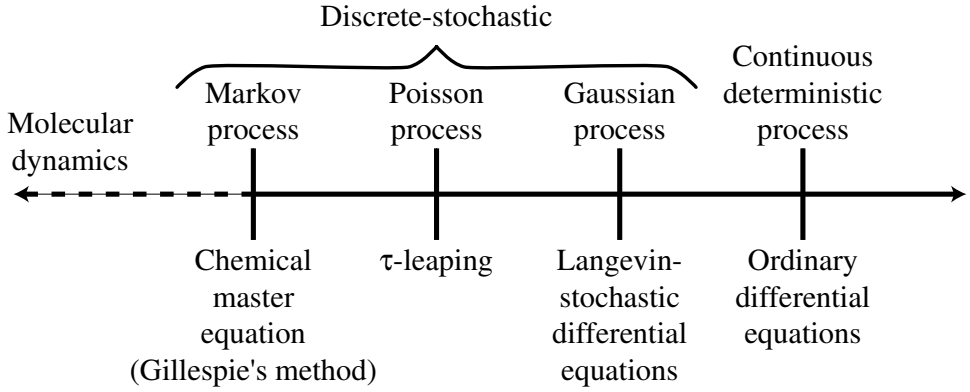


Figure 1.1: The chemical reaction kinetic modelling spectrum: on top the mathematical model, and on bottom the mathematical tool used to study the model.

The first study of the reaction-diffusion mechanism using a stochastic simulation method was done by Stundzia and Lumsden [63]. They presented a generalized version of a stochastic simulation algorithm used for studying models of chemical reactions. Elf, Doncic and Ehrengerb later improved the computation time of the generalized stochastic simulation algorithm by using a specialized data structure called a minimum heap [10]. Both works, however, focused on implementation details rather than on pattern formation, and therefore did not consider the reaction-diffusion mechanism in a growing structure or applications of their techniques to pattern formation.

There is computer software available to model the reaction-diffusion mechanism using any one of the three approaches: deterministic differential equations, stochastic differential equations, and the stochastic simulation approach. Some of this software requires almost no knowledge of computer programming, such as *XPPAUT* [11], and some requires at least high-level programming knowledge, such as *Octave* (`http://www.octave.org`). There is also software that offers a graphical user interface for inputting the reactions involved in the mechanism, such as *Cellware* [9] or *SmartCell* [1]. None of these, however, are nicely fitted towards models of spatial structures, and in particular growing structures, where the number of equations/reactions changes over time (e.g., through the process of cell division). To model growing organisms, more advanced software is needed. The simulation software, lpfg, is designed to handle developmental models of growing organisms that form linear or branching filaments through the use of the L-system formalism [31, 32]. It simulates models that are written in the L+C modelling language [31], which extends the C++ programming language with constructs inherent in L-systems [32].

An alternate L-system based simulator cpfg [26, 55] does exist, but it does not have the flexibility of the L+C modelling language. Two other alternatives to L-system based simulators, which are designed to handle developmental models of complex topological structures, are the modelling languages *MGS* [16] and *VV* [62]. Although it is possible to use one of these languages instead of an L-system based one, they would be excessive for modelling sea shell patterns or Anabaena. The final outcome of this research is a stochastic simulation approach using L-systems for models of pattern formation in sea shells and Anabaena.

1.1 Contributions of Research

A review of mathematical models for reaction kinetics and the methods which may be used to solve them is the first contribution of this research. The models considered are of two types: deterministic models that do not account for noise which may affect the pattern formation process, or stochastic models that do. The methods for solving the deterministic models are described briefly, while full attention is given to the stochastic models. There is also a discussion of approximation methods for the stochastic simulations.

The incorporation of a stochastic simulation method into L-system formalism and the related extension to the L+C modelling language is another contribution of this research. In principle, L+C uses a parallel rewriting strategy for deriving a new L-system string, so having a sequential stochastic rewriting strategy is a fundamental change to the L+C language. The idea to incorporate a stochastic simulation method into the L+C modelling language is due to Przemyslaw Prusinkiewicz. The design

of the language constructs that formalize the stochastic simulation method in the L+C language is due to Przemyslaw Prusinkiewicz and Mikolaj Cieslak. The first implementation was done by Radoslaw Karwowski, and subsequent work was done by Mikolaj Cieslak.

Another contribution of this research is the application of a stochastic simulation method to reaction-diffusion in the context of L-systems. A description of a stochastic approach to the activator-substrate equations, a type of reaction-diffusion equation, that can be used to generate pigmentation patterns on sea shells, is given. Also, two different stochastic models for the bacterium Anabaena are given: one is based on the original thresholding model by Lindenmayer, and the other is based on a more recent reaction-diffusion model by Hammel and Prusinkiewicz.

1.2 Organization of Thesis

In Chapter 2, we review the background knowledge which is necessary to stochastically simulate reaction-diffusion using L-systems. The review includes: chemical reaction kinetics (the basis for reaction-diffusion), a visualization technique that may be used for chemical systems, and the L-system formalism along with the modelling language L+C, which is based on that formalism. In Chapter 3, we review two methods for studying chemical kinetics. In the first method, ordinary differential equations are used to represent changes in concentrations of a chemical species, and in the other method, a master equation is used to represent discrete changes in the number of molecules of a chemical species.

Since the master equation is difficult to solve analytically or numerically, we

review a stochastic simulation method for chemical reactions in Chapter 4. Then in Chapter 5, Lotka's chemical system and a chemical system based on quasi-steady state enzyme kinetics are simulated using the stochastic approach. Also in Chapter 5, the extension of a sequential stochastic rewriting strategy to the L+C modelling language and the related simulator lpfg is introduced. Using the combination of the stochastic rewriting strategy and L-systems, we implement stochastic models of reaction-diffusion in Chapter 6. Reaction-diffusion models in a growing linear structure are also presented there.

Finally, the thesis concludes in Chapter 7 with an evaluation of the CPU time that is required to generate the patterns presented in this thesis, and with a discussion of possible further research directions.

Chapter 2

Background

2.1 Introduction

The background knowledge which is necessary to understand the work presented in this thesis is discussed in three parts: first, a deterministic and stochastic approach to modelling chemical reaction kinetics, second, a visualization technique for chemical systems, and last, the L-system formalism with a brief description of the L+C modelling language. The relevance of this background knowledge to the rest of the thesis is also discussed in this chapter.

2.2 Kinetics of Chemical Reactions

Modelling the kinetics of chemical reactions is important to discuss because most reaction-diffusion models are based on a set of chemical reactions. Thus, by reviewing mathematical models of chemical reactions, the task of modelling reaction-diffusion will be made much easier.

A chemical reaction of the following form [18, page 14],

$$n_a A + n_b B \xrightarrow{k} n_c C + n_d D, \tag{2.1}$$

is read as: n_a molecules of species A react with n_b molecules of species B to produce n_c molecules of species C and n_d molecules of species D. The species A and B are called the *reactants,* and the species C and D are called the *products.* The numbers

of molecules of each species, n_a, n_b, n_c and n_d, are small non-negative integers called *stoichiometric coefficients*. The value k above the reaction arrow is the *reaction constant*, which "is a way of specifying the amount of time the reaction takes [to complete]" [18, page 14]. More precisely, this constant specifies the probability that (at a fixed temperature) a collision between the reactant molecules will occur with sufficient energy to complete the reaction and give the products. As an example let us consider decay, which is a part of the reaction-diffusion model and involves only one simple reaction. The following chemical reaction describes the decay process:

$$A \xrightarrow{\mu} *. \tag{2.2}$$

This is just a degenerated case of reaction (2.1), because all but one of the stoichiometric coefficients are equal to zero. Several mathematical models have been developed for modelling such a process (an overview is given by Gibson and Mjolsness [18, page 20]). For our purposes, however, only two types of models are examined: a continuous model at the level of concentrations of the chemical species, and a discrete model at the level of the number of molecules.

2.2.1 Deterministic ODE Approach for Modelling Chemical Kinetics

To model a chemical reaction using the deterministic ordinary differential equation (ODE) approach, three assumptions are made [18, page 31]: (1) the system is well mixed so that the rate of collisions between molecules is greater than the rate of reactions, (2) the number of molecules is large enough so that discrete changes can be approximated by continuous changes in concentration, and (3) the fluctuations in concentration about the mean are small (i.e. the variance is low). These assumptions

allow us to express the chemical reactions using a set of ODEs, whose variables are the concentrations of the species in the system. As an example, consider the chemical reaction for decay (2.2). In the deterministic ODE approach this reaction is written as follows,

$$\frac{da(t)}{dt} = -\mu a(t), \tag{2.3}$$

where lowercase letters denote the concentration, $a(t)$, of the species A at time t. The ODE (2.3) expresses the decay of the species A over time with reaction rate $\mu a(t)$, i.e., the concentration a decreases with rate $\mu a(t)$. The reaction rate is found according to the *law of mass action* (first formulated by Guldberg and Waage in 1864 [6]), and says that the rate of reaction is directly proportional to the product of the concentration of the reactants (or in the case of decay, reactant). In other words, the reaction rate is a product of the reactants multiplied by the reaction constant. Now let us examine the deterministic ODE approach in the general case.

Recall the general form of a chemical reaction (2.1). If this type of reaction is to be modelled using ODEs, the change in concentration of each species over time must be found. Using the law of mass action, with the concentration of each reactant raised to the power of its stoichiometric coefficient [18, page 26], the change in concentration of each species is found. Hence for reaction (2.1) the following ODEs are obtained,

$$\frac{da(t)}{dt} = \frac{db(t)}{dt} = -k(a(t)^{n_a})(b(t)^{n_b}) \tag{2.4}$$

$$\frac{dc(t)}{dt} = \frac{dd(t)}{dt} = k(a(t)^{n_a})(b(t)^{n_b}). \tag{2.5}$$

The change in concentration of species A and B is multiplied by negative one because their concentration is decreasing. If there was more than one reaction, the change

in concentration of each species would be the sum of the contribution from each reaction. Let us, now, consider the problem that occurs when the assumptions of this approach cannot be satisfied.

2.2.2 The Limitations of the Deterministic ODE Method for Modelling Chemical Kinetics

The assumptions we make when modelling a chemical system using deterministic ODEs are not satisfied when only a small amount of molecules exist in the system. This is due to the noise inherent to such a system (i.e. high system fluctuations) and the non-continuous changes in the concentration of chemical substances. Also, Rényi [61] showed the law of mass action to be only approximately valid for small systems, because the variance of the number of molecules in the system is assumed to be negligible (see Bharucha-Reid [7, page 362]). An example of a biochemical system for which these assumptions are not satisfied is gene transcription, where there is only a single DNA molecule with a small number of binding sites for a particular gene. As a result, the chemical reaction which describes RNA polymerase binding to these sites occurs with a widely distributed reaction time [43, 37]. There is, however, another important implication of noise in a chemical system, which is that two identical systems with the same initial conditions may exhibit different behaviours. This phenomenon is due to the multiple stable states and bifurcations that are often a part of small chemical systems, due to stochastic effects that may drive the system to different behaviours [43]. This has been shown experimentally and computationally by McAdams and Arkin for phage λ-infected *E. coli* [2]. The importance of this phenomenon in the context of biological modelling has been discussed by Kerszberg

[33], and Federoff and Fontana [13]. Furthermore, experiments have been done on measuring the noise in several gene networks using green fluorescent protein (see Paulsson's summary of these studies [49]). Modelling chemical systems with noise is important enough for us to examine a stochastic approach to modelling these systems.

2.2.3 The Stochastic Approach to Chemical Kinetics

One limitation of the deterministic ODE approach is the assumption that there is a large number of molecules in the system, so that discrete changes in the number of molecules can be approximated by continuous changes in the concentration of those molecules. There are, however, systems in which we cannot make this assumption because the noise that is inherent to such systems plays a significant role in the system's behaviour. Therefore, a stochastic method of modelling chemical systems that deals with the exact number of molecules in the system and discrete changes in the system's state must be considered.

Stochastic Formulation

Let us start by recalling what is known about a system of chemical reactions. The volume V with N chemical species S_i ($i = 1, \ldots, N$), and an initial state $X^{(0)} = \{X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)}\}$ specifying the number of molecules of each species S_i are given. Also, M chemical reactions R_μ, for which there exist M reaction constants k_μ (recall the definition from Section 2.2) with $\mu = 1, \ldots, M$, are given. We cannot directly use the same reaction constants in the stochastic formulation of a chemical system because the stochastic approach deals with the probability of a reaction occurring

in the volume V, and not with the rate of a reaction. For this reason, the *stochastic reaction constants*, c_μ for reaction R_μ, are given [21, page 2342]. These constants characterize "reaction probabilities per unit time" [21, page 2342], which are defined as

$$\begin{aligned} c_\mu dt \quad &= \quad \text{the average probability with which a specific combination} \\ &\quad\quad \text{of molecules involved in reaction } R_\mu \text{ will react in the time} \\ &\quad\quad \text{interval } (t, t+dt). \end{aligned} \tag{2.6}$$

On its own, the product $c_\mu dt$ does not give the probability of a reaction occurring in the volume V, because it is independent of the number of molecules involved in the reaction R_μ. Before defining a probability that *is* dependent on the number of molecules, let us examine the meaning of $c_\mu dt$ more closely by calculating its value for the following bimolecular reaction,

$$R_\mu : S_1 + S_2 \rightarrow 2S_3. \tag{2.7}$$

The steps in calculating $c_\mu dt$ for this bimolecular reaction are shown in Figure 2.1, and are similar to Gillespie's calculation [20, page 407][21, page 2341].

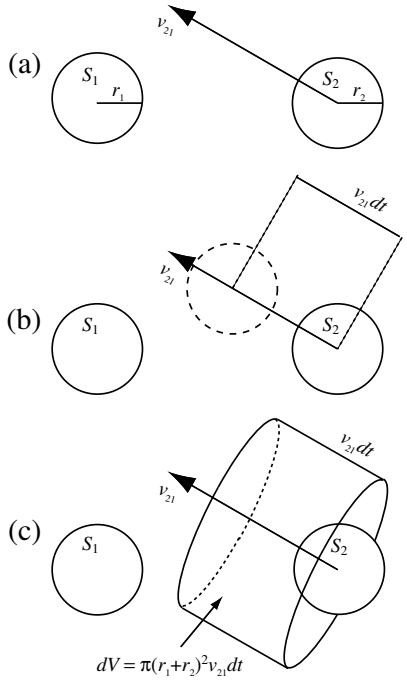


Figure 2.1: Calculating the collision volume for a bimolecular reaction: (a) Definition of molecule S_1 with radius r_1, molecule S_2 with radius r_2 and the velocity v_{21} of molecule S_2 relative to S_1. (b) The change in position of S_2 in dt. (c) The collision volume.

First, assume that molecule S_1 has radius r_1, molecule S_2 has radius r_2, and the velocity of molecule S_2 relative to molecule S_1 is $v_2 - v_1 = v_{21}$ (see Figure 2.1a). A collision of the two molecules will occur when the distance between their centres is less than or equal to $r_1 + r_2$. The collision volume of molecule S_2 relative to molecule S_1 can be swept out by moving molecule S_2 in the direction of v_{21} in the next small time interval dt (see Figure 2.1b). We call this new volume the *collision volume of molecule* S_2, and calculate it as $dV_{collision} = \pi(r_1 + r_2)^2 v_{21} dt$ (see Figure 2.1c). If molecule S_1 is in the collision volume of molecule S_2, then the two molecules will

collide within the time interval $(t, t + dt)$.

To obtain the probability with which a specific S_1 molecule and S_2 molecule will collide, we must make the assumption that all of the molecules are distributed randomly and uniformly in the volume, V [21, page 2343]. Notice that this is the first assumption made in the deterministic ODE approach. Thanks to this assumption, the probability that a specific S_1 molecule will be in the collision volume of a specific S_2 molecule is $dV_{collision}/V$. By averaging the velocity of all S_1-S_2 pairs of molecules in V, the average probability that a specific S_1-S_2 pair of molecules will collide in the time interval $(t, t + dt)$ [21, page 2342] is obtained:

$$\overline{dV_{collision}/V} = \pi(r_1 + r_2)^2 \overline{v_{21}} dt/V. \tag{2.8}$$

Assuming that when a pair of S_1-S_2 molecules collide a reaction between them occurs, Equation (2.8) gives the value of $c_\mu dt$ [22, page 412] for the bimolecular reaction we are considering, and the reason why the term *average probability* is used in the definition of $c_\mu dt$ should now be clear. Let us now find the probability of the reaction R_μ occurring in the volume V.

The Propensity Function

Since $c_\mu dt$ is defined as the average probability of a specific combination of molecules reacting in the time interval $(t, t + dt)$, what is the probability of the reaction R_μ occurring in this time interval? Notice that if $c_\mu dt$ is multiplied by all possible combinations of reactant molecules involved in reaction R_μ, the overall probability for reaction R_μ occurring in V in the time interval $(t, t+dt)$ is obtained. Let us define the function $h_\mu(X)$ as the number of distinct combinations of reactant molecules involved in reaction R_μ, where $X = (X_1^{(t)}, X_2^{(t)}, ..., X_N^{(t)})$ is a vector with components

set to the number of molecules of each species at time t [20, page 412]. Thus, the overall probability of a reaction R_μ occurring in V in the time interval $(t, t+dt)$ is given by,

$$a_\mu(X)dt = c_\mu h_\mu(X)dt. \tag{2.9}$$

Gillespie coined the function, $a_\mu(X)$, the *propensity function*, which gives the likelihood that reaction R_μ takes place in V in the time interval $(t, t+dt)$. As an example, reconsider the bimolecular reaction (2.7), $h_\mu(X)$ is equal to the number of distinct combinations of reactant molecules involved in this reaction (i.e. $h_\mu(X) = X_1^{(t)}X_2^{(t)}$), and therefore the propensity function is given by: $a_\mu(X) = c_\mu X_1^{(t)}X_2^{(t)}$. Using the propensity function, let us now consider the connection between the reaction constant k_μ and the stochastic reaction constant c_μ.

2.2.4 The k_μ-c_μ Connection

It is important to examine the connection between the reaction rate constant k_μ, and the stochastic reaction constant c_μ, because there will be times when we want to create a stochastic model from a deterministic ODE one and vice versa. For this reason, and considering the following statement made by Gillespie [21, page 2343],

> "...from a practical point of view, c_μ and k_μ will differ at most by only two simple constant factors. From a theoretical point of view, however, the difference between c_μ and k_μ is much more complicated...",

let us only examine the mathematical relationship between the two constants. We start by calculating k_μ for the bimolecular reaction (2.7).

By definition of $a_\mu dt$, the bimolecular reaction occurs in V in the time interval $(t, t+dt)$ with probability $X_1 X_2 c_\mu dt$. The average probability or rate at which the reaction occurs in V over a large time period, say $(0, T)$, is $\langle (X_1 X_2) c_\mu \rangle = \langle X_1 X_2 \rangle c_\mu$, where $\langle X_1 X_2 \rangle$ means the average number of S_1, S_2 molecules in V over $(0, T)$. Now for a bimolecular reaction, k_μ is calculated as the average rate of the reaction per unit volume, $\langle X_1 X_2 \rangle c_\mu / V$, divided by the average concentrations of the two molecular species, S_1 and S_2 [20, page 410]. Therefore, k_μ is given by,

$$k_\mu = \frac{\langle X_1 X_2 \rangle c_\mu / V}{\langle X_1 / V \rangle \langle X_2 / V \rangle}. \tag{2.10}$$

If we let $x_i = X_i / V$ be the molecular concentrations of species S_i, this equation can be rewritten as,

$$k_\mu = \frac{\langle x_1 x_2 \rangle c_\mu V}{\langle x_1 \rangle \langle x_2 \rangle}. \tag{2.11}$$

Since the deterministic ODE approach is used under the assumption that the variance of the two species' concentrations is zero, the average of the products is equal to the product of the averages, $\langle x_1 x_2 \rangle = \langle x_1 \rangle \langle x_2 \rangle$. This simplifies Equation (2.11) to

$$k_\mu = c_\mu V. \tag{2.12}$$

The relationship, Equation (2.12), between the two constants does not come as a surprise because k_μ is used in the deterministic ODE approach which deals with molecular concentrations (molecules per unit volume), and c_μ is used in the stochastic approach which deals with molecular quantities (total number of molecules in V). For reaction types other then the bimolecular type, the relationship between the two constants depends on the number of distinct combinations of reactant molecules,

and only differs by a power of V and a constant factor. Table 2.1 lists the value of $h_\mu(X)$ for several reaction types [20, page 405, 413] and the relationship between the constants k_μ and c_μ for these reactions.

Reaction R_μ	**Combinations** $h_\mu(X)$	k_μ**'s relationship to** c_μ
$* \rightarrow products$	1	c_μ
$S_1 \rightarrow products$	X_1	c_μ
$S_1 + S_2 \rightarrow products$	X_1X_2	$c_\mu V$
$2S_1 \rightarrow products$	$X_1(X_1-1)/2$	$c_\mu V/2$
$S_1 + S_2 + S_3 \rightarrow products$	$X_1X_2X_3$	$c_\mu V^2$
$S_1 + 2S_2 \rightarrow products$	$X_1X_2(X_2-1)/2$	$c_\mu V^2/2$
$3S_1 \rightarrow products$	$X_1(X_1-1)(X_1-2)/6$	$c_\mu V^2/6$

Table 2.1: The number of distinct molecular combinations of reactants for several reaction types and the relationship between k_μ and c_μ.

2.3 Petri nets

In 1962, Carl Petri proposed a mathematical modelling tool for describing many types of computer systems (e.g., discrete distributed systems) [46]. The tool he proposed is called a Petri-net graph, and has found many applications in modelling. It is defined as a directed bipartite graph consisting of two types of nodes: *places* and *transitions*. The nodes in the graph are connected by weighted arcs, with the only condition being that arcs connect places to transitions or transitions to places (and nothing else). A nonnegative integer, called a *marking*, is assigned to each place in the graph. In the visualization of a Petri net, places are drawn as circles, transitions as bars, arcs as arrows connecting the circles and bars, and markings are drawn as tokens. Several interpretations of place, transition, and marking can be made depending on the modelling application.

Consider the Petri-net visualization for the chemical reaction (2.1) shown in Figure 2.2. In a Petri-net visualization of a chemical system such as this one, places are interpreted as molecular species, transitions are interpreted as chemical reactions, and markings are interpreted as the number of molecules of each species. Specifically in this example, circles represent the molecular species A, B, C, and D, and the symbols inside represent the number of molecules of each species (here, symbols have been used instead of tokens). The chemical reaction, $n_a A + n_b B \longrightarrow n_c C + n_d D$, is represented by a transition bar, where the arrows from a circle to the bar represent the reactants, and the arrows from the bar to a circle represent the products. The number of reactants (n_a and n_b) and products (n_c and n_d) are represented by the symbols above the appropriate arrows.

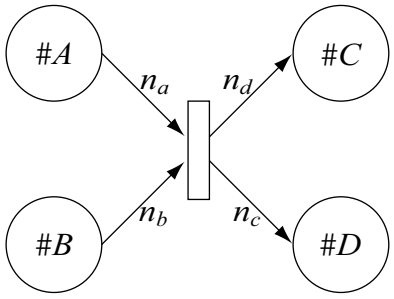


Figure 2.2: Stochastic Petri net for a simple chemical reaction, where $\#A$, $\#B$, $\#C$ and $\#D$ are the number of molecules of species A, B, C, and D, respectively.

The Petri-net is a good visualization technique to use because it provides a clear and unambiguous description of the model, and allows us to focus on the model's meaning rather than its implementation. Petri nets also make it possible "to describe more complex models as a composition of simpler ones" [57, page 17], which will greatly simplify the description of the chemical reactions involved in the reaction-

diffusion mechanism.

2.4 Review of L-systems and the L+C Language

L-systems were introduced by Aristid Lindenmayer in 1968 as a ruled-based formalism for modelling growing structures that form linear or branching filaments [34]. Among his examples was a model of the development of Anabaena, which nicely illustrates the power of the formalism [56]. The development of such a structure, from an initial structure (called the *axiom*), is described by *rewriting rules* (or productions) that are applied in parallel steps to the *modules* (or components) of the structure. Each rewriting rule specifies how a *predecessor* module is replaced by any number of *successor* modules. For example, a rewriting rule for cell division where cell A is replaced by two daughter cells B and C is given by $A \longrightarrow BC$. The application of a rewriting rule to a module can be either context free or context-sensitive depending on the rule's purpose. In the context-free case, a module is only matched to the predecessor of the rewriting rule, but in the context-sensitive case, a module and its neighbouring modules are matched to the predecessor of the rule. A context-sensitive rewriting rule is written as,

$$\textit{left neighbour} \; < \; \textit{predecessor} \; > \; \textit{right neighbour} \; \longrightarrow \; \textit{successor} \; . \qquad (2.13)$$

In a *parametric* L-system, each module is associated with a number of parameters that store additional information (e.g., a module with one parameter can be expressed as $A(x)$, where the parameter x may represent the size of a cell). Context-sensitive rules combined with parametric L-systems are useful for simulating information transfer between neighbouring modules in a developing structure. Thus, the

parameters of a successor module may depend on the predecessor's parameters and on the neighbouring modules' parameters. The combination of these concepts has led to many extensions of L-systems for calculating new parameter values (mostly in cpfg [26] and its modelling language [55]), but the most flexible extension is the L+C modelling language.

The L+C modelling language is an extension to the C++ programming language with constructs inherent in L-systems [31]. The advantage of this language over its predecessors is its increased expressiveness, which among other things allows for typed module parameters (primitive and compound data types) and productions with multiple successors (e.g., through the use of conditional statements in the productions) [32]. This makes the language well suited to modelling the reaction-diffusion mechanism using either a deterministic or stochastic approach. As a first look at the L+C language, let us consider a model of a growing filament with polarity dependent asymmetric cell division based on the age of walls that separate cells, which is similar to a model given by Karwowski and Prusinkiewicz [32, page 2]. The L+C code for this model is given by,

```
1  module Cell (float);
2  module Wall (float);
3
4  axiom: Wall(AGE0) Cell(SIZE0) Wall(AGE0);
5
6  Wall(age_left) < Cell (size) > Wall(age_right):
7  {
8    if (size >= MAX_SIZE)
9    {
10     if (age_left >= age_right)
11       produce Cell(SIZE0) Wall(0) Cell(size-SIZE0);
12     else
13       produce Cell(size-SIZE0) Wall(0) Cell(SIZE0);
```

```
14    }
15    else
16      produce Cell(size*GROWTH_RATE);
17  }
18
19  Wall(age):
20  {
21    produce Wall(age+1);
22  }
```

This example illustrates the basic notion of L-system programming using L+C rather well. In L+C, modules are declared using the keyword `module`, with the parameter types specified for each parameter in the parentheses immediately following the declaration of a module. Two modules, with one floating point parameter each, are declared in this example: `Cell` has a parameter for the cell's size, and `Wall` has a parameter for the wall's age. The initial structure is declared after the keyword `axiom`, and in this case is a single cell between two walls, with each module's parameter initialized using a constant: `AGE0` for the walls and `SIZE0` for the cell. A rewriting rule, delimited by curly braces, is given by a predecessor followed by a colon (instead of a right-handed arrow) with any number of `produce` statements denoting the possible successors of that predecessor. The flexibility of L+C is that these rules may contain any valid C++ constructs, such as conditional statements. The first rewriting rule in this example, for the `Cell` module, is context-sensitive. The purpose of this rule is to grow each cell in the structure until it is necessary for this cell to divide, which happens when its size reaches the maximum value, `MAX_SIZE`. The cell division is asymmetrical (if `SIZE0` is not equal to half of `MAX_SIZE`), and the two daughter cells are given two different sizes that depend on the age of the left and right wall. The second rewriting rule, for the `Wall` module, is context-free. Its

purpose is to advance the age of each wall in the structure. There are more features of the L+C modelling language that are not given in this example. For instance, several rewriting rules may be arranged into subsets using the keyword `group`, and one of these subsets may be invoked using the predefined function `UseGroup(n)`, where `n` is the group number. There are also predefined functions that are executed at the beginning/end of the simulation, `Start/End`, and at the beginning/end of each derivation step, `StartEach/EndEach`.

Figure 2.3 is a visualization of the cellular structure that develops when the rewriting rules given in the example above are applied to an initial structure roughly 300 times. It was produced using the simulation program lpfg [54], which makes it possible to execute models written using the L+C language and to visualize them. The visualization shows cells as red coloured rounded rectangles, and the walls as green and brown-coloured vertical bars for young and old walls, respectively.

Figure 2.3: A visualization of asymmetrical cell division, with cells visualized as rounded rectangles and the walls separating them as vertical bars.

Before using the L+C modelling language and lpfg simulator to simulate models of reaction-diffusion based pattern formation, let us review the deterministic and stochastic approaches that can be used.

Chapter 3

Studying the Kinetics of Chemical Reactions

3.1 Introduction

This chapter discuss some methods for studying the kinetics of chemical reactions. For both the deterministic and stochastic approaches, an analytical and numerical solution is considered. The chapter's main purpose is to motivate the use of the stochastic simulation approach for modelling reaction-diffusion.

3.2 Solution to Deterministic ODEs for Chemical Reactions

Let us first consider the solution to the deterministic ODE for decay. The solution is a function that describes the concentration of the species A at time t. To find this function the (very basic) ODE for the decay process must be integrated. Start by separating the variables in ODE (2.3) to get the following equation,

$$\frac{da(t)}{a(t)} = -\mu dt. \tag{3.1}$$

This equation is integrated over the time interval 0 to t as follows (let $dt = ds$):

$$\begin{aligned}
\int_0^t \frac{da(s)}{a(s)} &= \int_0^t -\mu ds \\
\ln a(s) \Big|_0^t &= -\mu s \Big|_0^t \\
\ln \frac{a(t)}{a(0)} &= -\mu t \\
a(t) &= a(0)e^{-\mu t}, \qquad (3.2)
\end{aligned}$$

where $a(0)$ is the initial concentration of species A at time $t = 0$. Since $a(t)$ is the concentration of A at time t, the concentration of the species A decreases exponentially according to Equation (3.2). This is the solution to the deterministic ODE for decay. Now let us examine the solution to a deterministic ODE for any chemical reaction.

Finding an analytic solution to the coupled ODEs given in Equations (2.4) and (2.5) for the chemical reaction $n_a A + n_b B \longrightarrow n_c C + n_d D$ is not as easy as in the case of the decay reaction. A numerical solution to these coupled ODEs must be found instead of an analytical solution. In this example, a numerical method, such as the forward Euler method, which gives an update formula for the concentration of each species over time is used. The ODEs (2.4) and (2.5) have the following update formulae,

$$\begin{aligned} a(t+\Delta t) &= a(t) + (-ka(t)^{n_a} b(t)^{n_b})\Delta t \\ b(t+\Delta t) &= b(t) + (-ka(t)^{n_a} b(t)^{n_b})\Delta t \\ c(t+\Delta t) &= c(t) + (ka(t)^{n_a} b(t)^{n_b})\Delta t \\ d(t+\Delta t) &= d(t) + (ka(t)^{n_a} b(t)^{n_b})\Delta t, \end{aligned} \tag{3.3}$$

where $f(t)$ denotes the concentration of species $f \in \{a, b, c, d\}$ at time t. Using these formulae, the concentration of each species is updated with a fixed time step (i.e. of regular time intervals Δt). This numerical method is not the best choice of integrators because the time step must be very small in order to ensure an accurate solution. There are several other integrators which are more accurate than the forward Euler method but will not be required for our purposes (see Ermentrout's book for a summary of all these methods [11, page 256]). Almost any chemical

system modelled by deterministic ODEs can be solved using these methods. For chemical systems that cannot be modelled using deterministic ODEs, the stochastic approach must be used.

3.3 Solution to the Stochastic Process for Chemical Reactions

Before finding a solution to the stochastic process for chemical reactions in the general case, let us examine the solution to the stochastic process for the decay reaction. The intention is to show how difficult it is to find the solution even for such a simple reaction.

3.3.1 Solution to the Decay Process

Let us consider the solution to the stochastic process for the decay reaction found by Bartholomay [5]. In his approach, Bartholomay defines $A(t)$ to be a (discrete) random variable that represents the number of molecules of a species A at time t (i.e., a stochastic process), and he assumes that in an infinitesimal time Δt the random variable will change according to the following transition probabilities,

$$Prob\{A(t) \rightarrow A(t) - 1\} = \mu A(t)\Delta t - o(\Delta t) \tag{3.4}$$

$$Prob\{A(t) \rightarrow A(t)\} = 1 - \mu A(t)\Delta t \tag{3.5}$$

$$Prob\{A(t) \rightarrow A(t) - j\} = o(\Delta t), \tag{3.6}$$

where μ is the decay rate constant, $j > 1$, and $o(\Delta t)/\Delta t$ tends to zero as $\Delta t \rightarrow 0$ [38, page 7]. Equation (3.4) is the probability of one molecule decaying in the time

interval $(t, t+\Delta t)$. Equation (3.5) is the probability of no molecules decaying in the time interval $(t, t+\Delta t)$. Equation (3.6) is the probability of more than one molecule decaying in the time interval $(t, t+\Delta t)$. Using the notation $Prob\{A(t) = a\} = P_a(t)$, the probability for the number of molecules of species A at time $t + \Delta t$ is given by

$$P_a(t + \Delta t) = \mu(a + 1)\Delta t P_{a+1}(t) + (1 - \mu a \Delta t)P_a(t) - o(\Delta t). \tag{3.7}$$

The first term is the probability of the system's transition to state a in $t + \Delta t$ multiplied by the probability that the system is in state $a + 1$ at time t (i.e. the previous state). The second term is the probability of the system's state not changing in $t+\Delta t$ multiplied by the probability the system is already in state a. The last term is the probability of more than one molecule decaying in the time interval $(t, t+\Delta t)$.

To solve this probability equation, we need to consider the time evolution of $P_a(t)$, which is given by

$$\begin{aligned} \lim_{\Delta t \to 0} \frac{P_a(t + \Delta t) - P_a(t)}{\Delta t} &= \lim_{\Delta t \to 0} \frac{\mu(a + 1)\Delta t P_{a+1}(t) - \mu a \Delta t P_a(t) - o(\Delta t)}{\Delta t} \\ \frac{dP_a(t)}{dt} &= \mu(a + 1)P_{a+1}(t) - \mu a P_a(t), \end{aligned} \tag{3.8}$$

where $\lim_{\Delta t \to 0} o(\Delta t)/\Delta t = 0$ by definition. Solving Equation (3.8) gives us insight into the behaviour of the decay process, e.g., through the expected value and variance of the random variable $A(t)$. The expected value of this random variable is defined as

$$E\{A(t)\} = \sum_{a=0}^{\infty} P_a(t)a, \tag{3.9}$$

and the variance is defined as

$$Var\{A(t)\} = E\{(A(t) - E\{A(t)\})^2\}. \tag{3.10}$$

To solve Equation (3.8) for $P_a(t)$, Bartholomay uses a *probability-generating function* for $A(t)$ [38, page 7]. This function is useful in finding the probability distributions and other properties (e.g., expected value and variance) of integer-valued random variables [69, page 123]. It is defined by

$$\begin{aligned} F(s,t) &= E\{s^a\} \\ &= P_0(t) + P_1(t)s + P_2(t)s^2 + \ldots \\ &= \sum_{a=0}^{\infty} P_a(t)s^a, \end{aligned} \tag{3.11}$$

where $|s| < 1$. Finding the value of this probability-generating function is difficult since $P_a(t)$ is not known. Therefore, we rewrite Equation (3.11) as the following partial differential equation (PDE),

$$\frac{\partial F(s,t)}{\partial t} = \sum_{a=0}^{\infty} \frac{\partial P_a(t)}{\partial t} s^a.$$

If $\partial P_a(t)/\partial t$ is substituted from Equation (3.8), the following is obtained,

$$\begin{aligned} \frac{\partial F(s,t)}{\partial t} &= \sum_{a=0}^{\infty} [\mu(a+1)P_{a+1}(t) - \mu a P_a(t)]s^a \\ &= \mu \left[\sum_{a=0}^{\infty} (a+1)P_{a+1}(t)s^a - \sum_{a=0}^{\infty} a P_a(t)s^a \right] \\ &= \mu \left[\sum_{a=1}^{\infty} a P_a(t)s^{a-1} - \sum_{a=0}^{\infty} a P_a(t)s^{a-1}s \right] \\ &= \mu(1-s) \sum_{a=0}^{\infty} a P_a(t)s^{a-1}, \end{aligned} \tag{3.12}$$

where the last factorization is possible because the first term of the summation is zero when $a = 0$. Since

$$\frac{\partial F(s,t)}{\partial s} = \sum_{a=0}^{\infty} P_a(t) a s^{a-1},$$

Equation (3.12) can be rewritten as:

$$\frac{\partial F(s,t)}{\partial t} = \mu(1-s)\frac{\partial F(s,t)}{\partial s}. \tag{3.13}$$

To solve for $F(s,t)$, first we make the substitution $1-s = e^z$ [15, page 239], such that $F(s,t) = G(z,t)$. Then Equation (3.13) becomes,

$$\frac{\partial G(z,t)}{\partial t} + \mu\frac{\partial G(z,t)}{\partial z} = 0, \tag{3.14}$$

since

$$\frac{\partial F(s,t)}{\partial t} = \frac{\partial G(z,t)}{\partial t},$$

and

$$\begin{aligned}\frac{\partial F(s,t)}{\partial s} &= \frac{\partial G(z,t)}{\partial z}\cdot\frac{\partial z}{\partial s} \\ &= \frac{\partial G(z,t)}{\partial z}\cdot\frac{-1}{e^z}.\end{aligned}$$

Equation (3.14) is called the first-order wave equation and it is easy to verify that its solution is an arbitrary function of $(z-\mu t)$ (by the method of characteristics) [50, page 25]. The solution has the following form:

$$G(z,t) = H(z-\mu t). \tag{3.15}$$

This function must satisfy the initial condition that at time $t=0$ there are $A(0) = A_0$ molecules in the system. That is, it must satisfy the following initial condition:

$$\begin{aligned}F(s,0) &= \sum_{a=0}^{\infty} P_a(0)s^a \\ &= s^{A_0},\end{aligned} \tag{3.16}$$

since the probability that there are A_0 molecules in the system at time $t = 0$ is only equal to one when $a = A_0$ and is zero otherwise. Solving $G(z,t)$ at this initial condition yields

$$F(s,0) = G(z,0) = (1 - e^z)^{A_0},$$

since $s = 1 - e^z$. This equation can be evaluated for $G(z,t)$ at $(z - \mu t)$ to give,

$$G(z,t) \;=\; (1 - e^{z-\mu t})^{A_0},$$

and substituting $z = \ln(1-s)$ into this equation gives

$$\begin{aligned} F(s,t) &= (1 - e^{\ln(1-s)-\mu t})^{A_0} \\ &= (1 - (1-s)e^{-\mu t})^{A_0}. \end{aligned} \tag{3.17}$$

The function, $F(s,t)$, can now be expanded using the binomial theorem to give,

$$\begin{aligned} F(s,t) &= [(1 - e^{-\mu t}) + se^{-\mu t}]^{A_0} \\ &= \sum_{a=0}^{A_0} \binom{A_0}{a} s^a e^{-\mu t a} [1 - e^{-\mu t}]^{A_0 - a}. \end{aligned} \tag{3.18}$$

Equation (3.11) can now be solved, which gives the following value of $P_a(t)$ [21, page 2347],

$$\begin{aligned} \sum_{a=0}^{\infty} P_a(t) s^a &= \sum_{a=0}^{A_0} \binom{A_0}{a} s^a e^{-\mu t a} [1 - e^{-\mu t}]^{A_0 - a} \\ P_a(t) &= \frac{A_0!}{a!(A_0 - a)!} e^{-\mu t a} [1 - e^{-\mu t}]^{A_0 - a}, a = 0, 1, \ldots, A_0. \end{aligned} \tag{3.19}$$

Not surprisingly, this is the form of the binomial probability function with parameters $(n = A_0, p = e^{-\mu t})$, where n is the sample size and p is the probability of a successful event (in terms of the binomial probability distribution). The expectation

and variance are known for this function, and therefore these values for $A(t)$ are given by [38, page 7],

$$E\{A(t)\} = A_0 e^{-\mu t} \tag{3.20}$$

$$Var\{A(t)\} = A_0 e^{-\mu t}(1 - e^{-\mu t}). \tag{3.21}$$

As an example, Figure 3.1 shows the expectation and variance for the decay process with $A_0 = 100$ and $\mu = 0.5$.

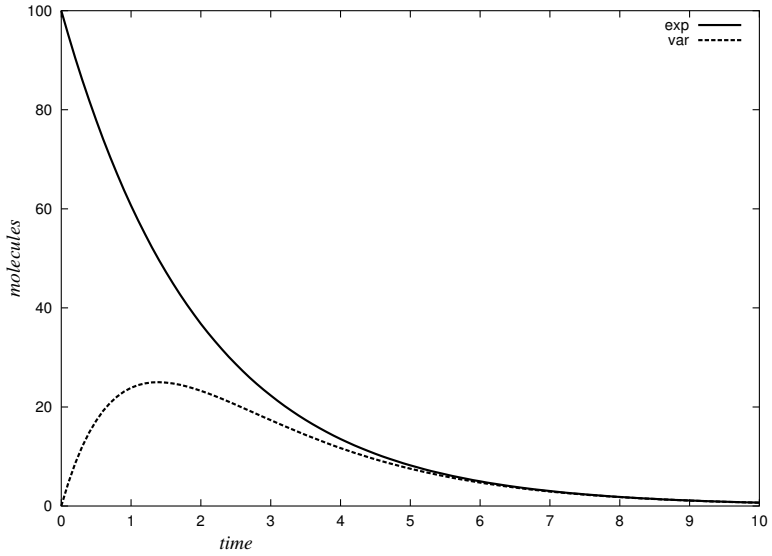


Figure 3.1: Expectation and variance for the decay process with $A_0 = 100$ and $\mu = 0.5$.

The solution to the stochastic process for decay can now be compared with the solution to the deterministic ODE for decay. The expected value (Equation (3.20)) Bartholomay found in his stochastic approach to the decay process is the deterministic ODE solution (Equation (3.2)), which shows that the two approaches are "consistent in the mean" [38, page 8]. The stochastic approach, however, also describes the variance (Equation (3.21)) of the number of molecules over time. This

gives us a better idea of the system's behaviour than just the expected value by itself, because it indicates the variability of the number of molecules over time. For example, in the decay process there is an initially increasing variance, but once there are only half the number of initial molecules left, the variance starts decreasing (see Figure 3.1). Thus the stochastic approach is indeed a more comprehensive approach to modelling biological systems than the deterministic ODE approach. For this reason, we wish to consider the stochastic formulation for any system and not just the simple decay process.

3.3.2 Solving the Chemical Master Equation

The fundamental method for stochastically modelling the state of a chemical system over time is to define a *master equation* for finding molecular populations at time $t + \Delta t$ given the molecular populations at time $t = 0$. The key element of this equation is the *grand probability function* [20, page 411], which is defined as

$$\begin{aligned} P(X,t) \;=\; & \text{probability that the chemical system will be in state} \\ & X = (X_1^{(t)}, X_2^{(t)}, ..., X_N^{(t)}) \text{ at time } t \text{ given that the system is} \\ & \text{in state } X = (X_1^{(0)}, X_2^{(0)}, ..., X_N^{(0)}) \text{ at time } t = 0. \end{aligned} \tag{3.22}$$

The time evolution of this function gives a complete characterization of the evolution of the system state, because it gives the probability of being in state X at time t. It can be derived by summing the number of different ways the system arrives at state X in time $t + \Delta t$. Hence, if there are M reactions that will change the system's state, then there are M ways in which the system arrives in a new state, but since there is also the possibility of no reactions occurring, there are $M + 1$ ways in which

the system's state can change at time $t + \Delta t$. Since the propensity function gives the likelihood of a reaction occurring in $(t, t + \Delta t)$, the time evolution of the grand probability function is given by,

$$\begin{aligned} P(X, t + \Delta t) &= P(X, t) \left[1 - \sum_{\mu=1}^{M} a_\mu(X) \Delta t \right] \\ &\quad + \sum_{\mu=1}^{M} [P(X - v_\mu, t) a_\mu(X - v_\mu) \Delta t], \end{aligned} \tag{3.23}$$

where v_μ is the change in the number of molecules produced by reaction R_μ. For example, for the chemical reaction (2.1), v_1 is given by

$$v_1 = \{-n_a, -n_b, n_c, n_d\}. \tag{3.24}$$

The first term in Equation (3.23) can easily be explained by noting that, if the system was in the state X at time t, the only possible way the system can be in state X at time $t + \Delta t$ is if no reaction occurs in the time interval $(t, t + \Delta t)$. Assuming that dt is small enough so that the probability of two or more reactions occurring over the time interval is negligible, the probability of this event is one minus the probability of a reaction occurring. The second term adds up, for each reaction R_μ, the probability of being in state $X - v_\mu$ at time t multiplied by the probability of reaction R_μ occurring in the time interval $(t, t + \Delta t)$. We can then finish our derivation of the time evolution of the grand probability function by writing it in the

following form,

$$\lim_{\Delta t \to 0} \frac{P(X, t+\Delta t) - P(X,t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{-P(X,t) \sum_{\mu=1}^{M} a_\mu(X)\Delta t}{\Delta t} + \lim_{\Delta t \to 0} \frac{\sum_{\mu=1}^{M} [P(X - v_\mu, t) a_\mu(X - v_\mu)\Delta t]}{\Delta t}$$

$$\frac{\partial P(X,t)}{\partial t} = \sum_{\mu=1}^{M} [P(X - v_\mu, t) a_\mu(X - v_\mu) - P(X,t) a_\mu(X)]. \tag{3.25}$$

This is the *chemical master equation*, which gives us the probability of the system being in state X at time t. We would like to solve this equation for any chemical system.

Bartholomay's stochastic solution to the decay process, where he gave the expectation and variance of the number of molecules in the system at time t, cannot be generalized to solve the chemical master equation. The reason is that, in the decay process, the number of ways in which the system's state can change in time $t + \Delta t$ is only two (a molecule either decays or it does nothing). We end up with a binomial probability function which exactly describes the behaviour of the decay process. In the chemical master equation, since there are $M + 1$ ways for the system's state to change at time $t + \Delta t$, it is difficult to find a probability function that provides the correct distribution [20, page 412]. McQuarrie [38, chapter 3] and Gardiner [15, chapter 7] provide some solutions to other elementary reactions besides the decay process, but even for simple reactions, Gardiner uses approximating methods to find a solution [15, page 268].

In some cases, it may be possible to numerically solve the chemical master equation, but usually the numerical solution is computationally intractable. Consider the

tree in Figure 3.2, which at its root has the probability of being in the initial state X at time $t = 0$, and at each child node has the probability of transition to a new state (a parent node has $M + 1$ child nodes). Thus, for K time steps of size Δt the number of probabilities that must be computed is given by:

$$\sum_{i=0}^{K} (M+1)^i = \frac{(M+1)^{K+1} - 1}{M}. \tag{3.26}$$

It is only possible to compute and store this many probabilities for very small values of M and K, and therefore, instead of numerically solving the chemical master equation, we must try another approach.

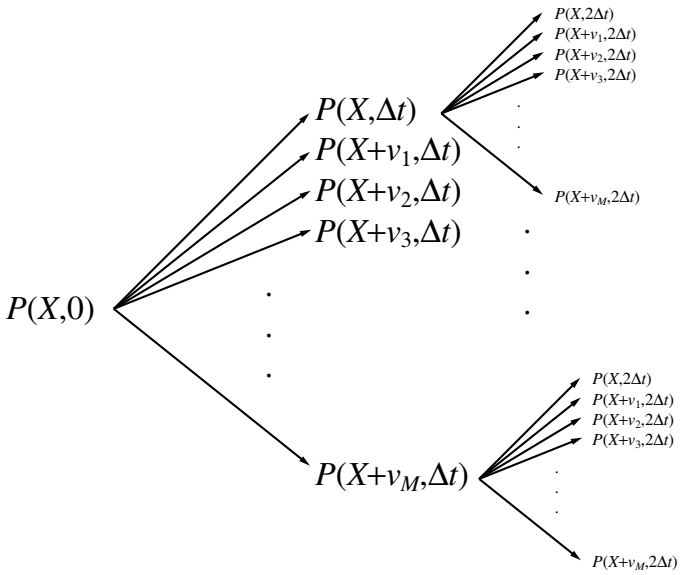


Figure 3.2: Tree representation of numerical solution to chemical master equation.

Chapter 4

Stochastic Simulation Approach to Kinetics of Chemical Reactions

4.1 Gillespie's Method

In 1976, Daniel Gillespie developed a method for exactly simulating the state of a chemical system over time under the assumption that the chemical system is well mixed [20, 21]. The distinction between Gillespie's method and the chemical master equation is that the former is a numerical simulation of the Markov process that the chemical master equation describes (since the probabilities for the state of the system at time $t+dt$ only depend on the system's state at time t and not on previous states, the process is Markov [17, page 3]). In other words, Gillespie's method is only one path through the tree shown in Figure 3.2. Thus, the method provides the state X of a chemical system over time t instead of the probability of being in state X at time t. To simulate the state of the chemical system over time, given the initial state $X^{(0)}$, Gillespie developed the following algorithm:

1. Initialize the simulation time to zero, $t = 0$.

2. Specify the initial numbers of molecules of N species, $X^{(0)}$.

3. Specify M stochastic reaction constants $c_1, c_2, c_3, \ldots, c_M$.

4. Calculate $h_1(X)c_1, h_2(X)c_2, h_3(X)c_3, \ldots, h_M(X)c_M$ the propensities for M re-

actions.

5. While molecular populations are all > 0 or time t has not reached some maximum value do:

   (a) Monte-Carlo step: generate the inter-reaction time τ and the reaction R_μ that takes place next.

   (b) Advance t by τ, and change the state X according to the reaction R_μ.

   (c) Recalculate, $h_\nu(X)c_\nu$, the propensities for all reactions R_ν that involve chemical species in R_μ.

The Monte-Carlo step is the heart of Gillespie's method and once it is examined, the relationship between Gillespie's method and the chemical master equation will be more clear.

The time of the next reaction and the reaction that takes place next are generated according to the joint probability density function, $p(\tau, \mu|X, t)$, which is defined by [20, page 412],

$$p(\tau, \mu|X, t)d\tau = \text{probability that the next reaction in the chemical system will occur in the time interval } (t+\tau, t+\tau+d\tau) \text{ and will be the reaction } R_\mu, \text{ given the system is in state } X \text{ at time } t. \quad (4.1)$$

To generate one random pair (τ, μ) according to the probability density function $p(\tau, \mu|X, t)$, an expression for it must first be derived (see [20, page 413] and [22, page 423]). The starting point in finding this expression is to calculate the probability $p(\tau, \mu|X, t)d\tau$. It can be calculated by subdividing the time interval $(t, t+\tau)$ into

subintervals of equal length, where each subinterval has length $\epsilon = \tau/k$ such that $k > 1$ (see Figure 4.1), and finding the probability of no reaction occurring in that time interval. Then, finding the probability of a reaction occurring in the time interval $(t + \tau, t + \tau + d\tau)$.

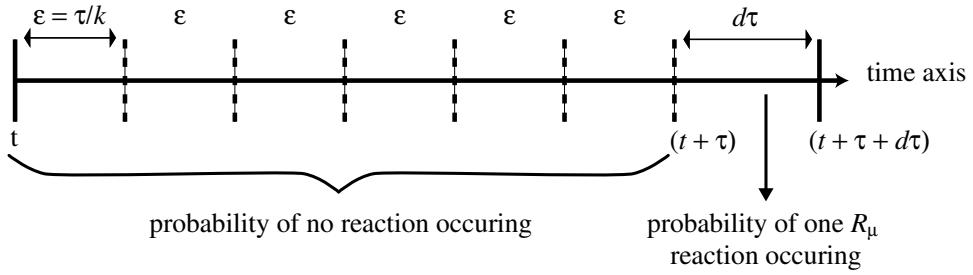


Figure 4.1: Finding the value of $p(\tau, \mu)d\tau$ by subdividing the time interval $(t, t + \tau)$ into $k = 6$ subintervals

The probability of no reaction occurring in a specific k subinterval is $[1-\sum_{\nu=1}^{M} a_\nu(X)\epsilon + o(\epsilon)]$, and the probability of one R_μ reaction occurring in the last interval, given that none occurred in $(t, t+\tau)$, is $[a_\mu(X)d\tau + o(d\tau)]$, where $o()$ is the probability of more than one reaction occurring in the time interval $(t, t+\tau+d\tau)$. By the *multiplication law of probabilities* [69, page 50], for the entire time interval $(t, t+\tau+d\tau)$ we obtain:

$$p(\tau, \mu|X, t)d\tau = [1 - a_0\epsilon + o(\epsilon)]^k[a_\mu(X)d\tau + o(d\tau)], \text{ where } a_0 = \sum_{\nu=1}^{M} a_\nu(X). \quad (4.2)$$

If this equation is divided by $d\tau$ on both sides and is taken to the limit $d\tau \to 0$, the following expression for the probability density function, $p(\tau, \mu|X, t)$, is obtained:

$$\begin{aligned} \lim_{d\tau\to 0} \frac{p(\tau, \mu|X, t)d\tau}{d\tau} &= \lim_{d\tau\to 0} \frac{[1 - a_0\epsilon + o(\epsilon)]^k[a_\mu(X)d\tau + o(d\tau)]}{d\tau} \\ p(\tau, \mu|X, t) &= [1 - a_0\epsilon + o(\epsilon)]^k[a_\mu(X)]. \end{aligned} \quad (4.3)$$

This expression can be simplified by writing the first term on the right-hand-side as,

$$[1 - a_0\epsilon + o(\epsilon)]^k = \left[1 - \frac{a_0 k\epsilon + o(\epsilon)k}{k}\right]^k = \left[1 - \frac{a_0\tau + o(\epsilon)\tau/\epsilon}{k}\right]^k \text{ since } \tau = k\epsilon.$$

Taking the limit with $k \to \infty$, we obtain:

$$\lim_{k\to\infty}\left[1 - \frac{a_0\tau + o(\epsilon)\tau/\epsilon}{k}\right]^k = e^{-a_0\tau}. \tag{4.4}$$

By substituting this value into Equation (4.3), the final form of the joint probability density function is obtained:

$$p(\tau, \mu|X, t) = a_\mu(X)e^{-a_0\tau}. \tag{4.5}$$

Given this equation, there are two ways to generate the inter-reaction time, τ, and the reaction, R_μ, that takes place at time $t + \tau$: the direct method and the first reaction method. The key difference between these methods is that in the direct method a value for τ and μ is generated according to $p(\tau, \mu|X, t)$, but in the first reaction method, a putative time τ_ν ($\nu = 1, \ldots, M$) is generated for each reaction, with the smallest putative time giving τ and μ.

4.1.1 The Monte-Carlo Step: Direct Method

The direct method of generating τ and R_μ involves rewriting the joint probability density function, $p(\tau, \mu|X, t)$, as the product of a single variable probability density function and a conditional probability. Then, independently generating τ from the single variable probability function, and R_μ from the conditional probability. We obtain the following equation [20, page 418],

$$p(\tau, \mu|X, t) \;=\; p_1(\tau|X, t) \cdot P_2(\mu|\tau, X, t). \tag{4.6}$$

The probability density function, $p_1(\tau|X,t)$, is defined by $p_1(\tau|X,t)d\tau$, which is the probability that a reaction will occur in the time interval $(t+\tau, t+\tau+d\tau)$. The conditional probability, $P_2(\mu|\tau,X,t)$, is defined by the probability that the next reaction will be R_μ, if the time of that reaction is $t+\tau$.

An expression for $p_1(\tau|X,t)$ can be obtained by summing, over all reactions, the probability of a particular reaction occurring in $(t+\tau, t+\tau+d\tau)$. It is given by the equation

$$p_1(\tau|X,t) = \sum_{\mu=1}^{M} p(\tau,\mu|X,t) = \sum_{\mu=1}^{M} a_\mu(X)e^{-a_0\tau} = a_0 e^{-a_0\tau}, \tag{4.7}$$

which describes an exponential distribution with mean $1/a_0$. The expression for $P_2(\mu|\tau,X,t)$ can now be easily derived from Equation (4.6), and is given by,

$$\begin{aligned} P_2(\mu|\tau,X,t) &= p(\tau,\mu|X,t)/p_1(\tau|X,t) \\ &= a_\mu(X)/a_0. \end{aligned} \tag{4.8}$$

To complete the Monte-Carlo step in Gillespie's algorithm, a random τ and a random μ are generated according to $p_1(\tau|X,t)$ and $P_2(\mu|\tau,X,t)$, respectively, by using the inversion method. This is a method that can be used to generate a random variable according to some cumulative distribution function, F, from a uniform distribution (provided that the function F^{-1} exists). The cumulative distribution function of $p_1(\tau|X,t)$ is given by (let $d\tau = dx$):

$$F_1(\tau) = \int_0^\tau a_0 e^{-a_0 x} dx = 1 - e^{-a_0\tau}, \tag{4.9}$$

and the cumulative distribution function of $P_2(\mu|\tau,X,t)$ is given by:

$$F_2(\mu) = \sum_{\nu=1}^{\mu} a_\nu(X)/a_0. \tag{4.10}$$

Clearly, the inverse for both of these distribution functions do exist (see [20, page 431]), so if two independent uniform random numbers r_1 and r_2 in the interval $[0, 1)$ are generated, then the following equations are used to generate τ and μ,

$$\tau = (1/a_0)\ln(1/r_1), \text{ and} \tag{4.11}$$

$$\mu = \text{the integer that satisfies } \sum_{\nu=1}^{\mu-1} a_\nu(X) < r_2 a_0 \leq \sum_{\nu=1}^{\mu} a_\nu(X). \tag{4.12}$$

The direct method gives a fast and efficient way to perform the Monte-Carlo step in Gillespie's algorithm, but before using it to simulate Lotka's chemical system, let us examine an alternate method.

4.1.2 The Monte-Carlo Step: First Reaction Method

The relevance of this alternative to the direct method for generating the time and type of the next reaction will become more clear when a speed-up technique for the first reaction method is discussed, but for now let us consider it as just an alternative. In the first reaction method, a putative inter-reaction time, τ_ν is generated for each reaction R_ν, where $\nu = 1, \ldots, M$. The reaction with the smallest putative time is the one that occurs first, and therefore it is chosen as the next reaction. The probability that reaction R_ν will occur in the time interval $(t + \tau, t + \tau + d\tau)$ is given by [20, page 419]:

$$P_\nu(\tau|X, t)d\tau = a_\nu(X)e^{-a_\nu(X)\tau}d\tau \text{ for } \nu = (1, 2, \ldots, M). \tag{4.13}$$

The expression for this probability can be derived similarly to the one which was derived for $p(\tau, \mu|X, t)d\tau$, where $a_\nu(X)d\tau$ is the probability reaction R_μ will occur in the time interval $(t + \tau, t + \tau + d\tau)$, and $e^{-a_\nu(X)\tau}$ is the probability that reaction

R_μ will not occur before that time interval. The cumulative distribution function for $P_\nu(\tau)$ is an exponential distribution with mean $1/a_\nu(X)$. Again by the inversion method, a random τ_ν is generated using the following equation,

$$\tau_\nu = (1/a_\nu(X)) \ln(1/r_\nu) \text{ for } \nu = (1, 2, \ldots, M), \tag{4.14}$$

where each r_ν is an independently generated uniform random number. To complete the Monte-Carlo step, the smallest τ_ν is chosen as the inter-reaction time and R_ν as the reaction that takes place next. That is the following values for τ and μ are used,

$$\tau = \text{smallest } \tau_\nu \text{ for } \nu = (1, 2, \ldots, M), \text{ and} \tag{4.15}$$

$$\mu = \text{the same } \nu \text{ which } \tau_\nu \text{ was smallest for.} \tag{4.16}$$

This method is not as fast or efficient as the direct method because M random numbers must be generated per simulation step instead of just 2, so in further discussions of Gillespie's algorithm we only consider the direct method for performing the Monte-Carlo step.

4.2 Stochastic Simulation of the Decay Process

By running Gillespie's algorithm on a chemical system formulated stochastically (as in Section 2.2.3), one possible outcome of the system's behaviour over time 0 to time t is obtained. That is one realization of the stochastic process involving the random variable X over time t is obtained. Since it is not possible to directly calculate the expectation and variance of the process over time from this one realization, these values must be estimated by running several independent trials of the process under

the same initial conditions [20, page 416]. First, define the state of the chemical system as the following,

$$X_j(t) = \text{the state of the system at time } t \text{ for run } j, \tag{4.17}$$

where $j = 1, 2, \ldots, J$, and J is the total number of simulation runs. The expectation and variance of the system's state at time t are then estimated as [69, page 326],

$$\widehat{E}\{X(t)\} = \frac{1}{J}\sum_{j=1}^{J} X_j(t) \tag{4.18}$$

$$\widehat{Var}\{X(t)\} = \frac{1}{J-1}\left[\sum_{j=1}^{J}(X_j(t))^2 - \frac{1}{J}\left(\sum_{j=1}^{J} X_j(t)\right)^2\right]. \tag{4.19}$$

The estimated expected value, $\widehat{E}\{X(t)\}$, can then be compared to the solution obtained from the deterministic ODE model or if the actual expected value and variance are available from the chemical master equation, the values can be compared. Let us use Gillespie's algorithm to simulate the decay process and compare the estimated expected value and variance.

In the case of the decay process, the estimated expected value and variance may be compared to the calculated values from the chemical master equation. This is not always possible because of the difficulty in solving the chemical master equation. In Section 3.3.1, the expected value and variance of the decay process were found to be $A_0e^{-\mu t}$ and $A_0e^{-\mu t}(1 - e^{-\mu t})$, respectively. Figure 3.1 showed a plot of these values for the initial number of molecules $A_0 = 100$ and the decay rate $\mu = 0.5$. Gillespie's method can be used to generate a realization of the decay process over time under these conditions. First, consider the visualization of the decay process shown as a stochastic Petri net, a variant of a Petri net, in Figure 4.2.

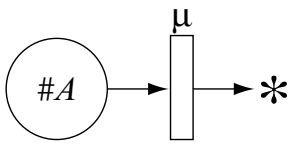


Figure 4.2: Stochastic Petri net for the decay process, with $\#A$ representing the number of molecules of species A and μ the stochastic reaction constant.

There are only two differences between standard Petri nets and stochastic Petri nets. In a stochastic Petri net, a randomly generated time t is assigned to each transition event similarly to Gillespie's first reaction method, where a randomly generated time is assigned to each reaction. This is in contrast to the asynchronous application of transitions in standard Petri nets. Also, in a stochastic Petri net a symbol is drawn above the transition bar that represents the likelihood of that transition being applied, which is equivalent to the stochastic reaction constant, c_μ. Thus, let us use stochastic Petri nets to visualize a chemical system that is simulated using Gillespie's method, similarly to Goss and Pecaud [24].

The decay process described by the stochastic Petri net in Figure 4.2 is visualized using the standard techniques (see Section 2.3) except for the μ symbol, which is the constant used in calculating the propensity of a molecule decaying. Also, the * symbol is used instead of a circle to represent a molecule *disappearing*. The visualization makes the task of writing a program to simulate the decay process more clear. The pseudo code that may be used to generate one realization of this process is given by:

```
1 mu = 0.5 // the decay rate
2 A  = 100 // the initial number of molecules
3 t  = 0   // the time of the last reaction
4
5 while (A > 0) do
```

```
6    A = A - 1                 // decay one molecule
7    t = t - ln(1-ran(1))/mu   // update time of last reaction
```

The first line in the pseudo code sets the stochastic reaction constant μ to 0.5. In the next two lines, the number of molecules of species A and the time of the last reaction t are initialized. The last three lines are used to simulate the decay of the species A according to Gillespie's method. Since there is only one type of reaction, using a random number to pick this reaction as the one that takes place next is excessive. Updating the time of the last reaction, however, still requires a uniform random number in the interval $[0, 1)$. In the pseudo code, this random number is obtained by calling the function `ran(1)`.

Using equations (4.18)-(4.19) and running the above program many times, an estimate of the expected value and variance for the decay process can be obtained. Figure 4.3a shows five realizations of the process, and Figure 4.3b shows an estimated expected value and variance of the process over 1,000 runs. A minor detail in making these estimates is that since t is a continuous variable, the estimated expected value and variance is computed in small discrete time steps over t. In Figure 4.3b the time step is 0.01 units.

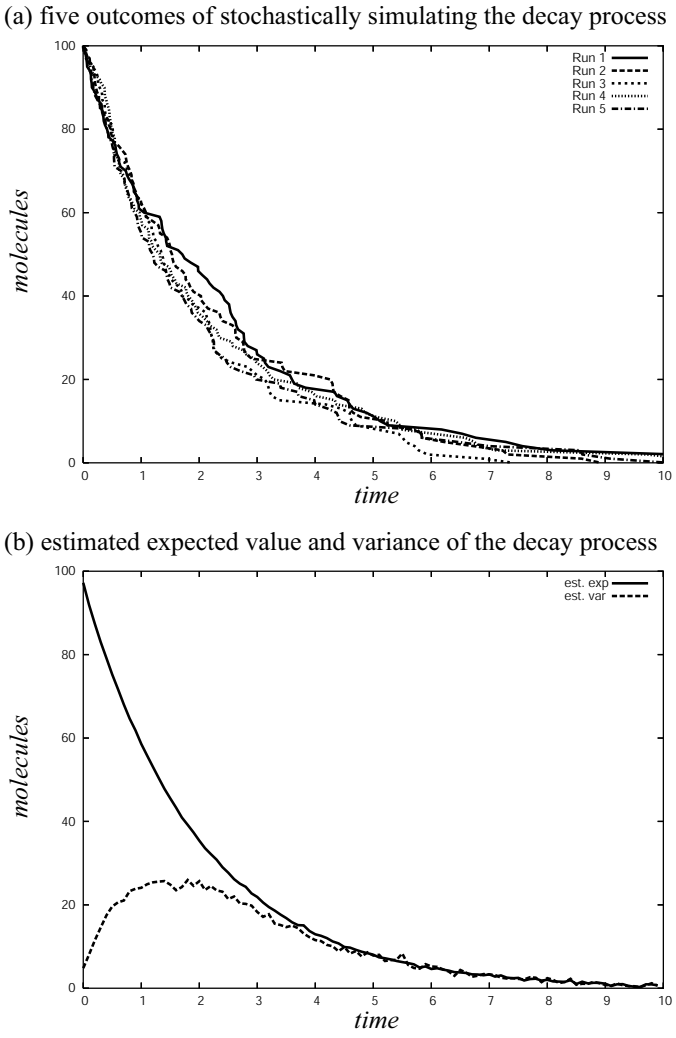


Figure 4.3: Stochastic simulation of the decay process: (a) five realizations, and (b) estimated expected value and variance.

The estimated expected value and variance for $\mu = 0.5$ and $A_0 = 100$ are similar to the actual values found from the chemical master equation (compare Figure 3.1 with Figure 4.3b). To reduce the error in these estimates, more simulation runs would need to be performed keeping in mind that the convergence rate is $1/\sqrt{J}$, i.e., quadrupling the number of runs J will halve the error. Even though this convergence

rate implies that to obtain a good estimate of the expected value and variance of the number of molecules in a chemical system many runs are required, using the stochastic simulation approach to find these values is still easier than solving the chemical master equation. Thus, let us now consider some chemical systems for which the chemical master equation cannot be solved.

Chapter 5

Simulating Chemical Systems using Gillespie's Method in L+C

5.1 Lotka's Chemical System

A good example of a chemical system that can be modelled using the chemical master equation, but is difficult to solve, consists of three hypothetical chemical reactions proposed by Lotka in 1920 [36] (the basis for the Lotka-Volterra equations). It also nicely reveals a shortcoming of the inability of the deterministic ODE approach to account for random fluctuations that are inherent to a chemical system [21, page 2351]. Before modelling these reactions using the stochastic approach, let us review the Lotka-Volterra equations.

5.1.1 Lotka-Volterra equations

Lotka [36] and Volterra [68] independently proposed the same model for the oscillatory and periodic behaviour of two different systems. Lotka's model was of three hypothetical chemical reactions (as was mentioned above), and Volterra's model was of several fish catches in the Adriatic Sea [47, page 80]. An experiment done by Huffaker in 1958 to confirm Volterra's model [29] showed the oscillatory and periodic behaviour of a system involving two species of mites. One of the species of mites is called the *predator* because it consumes the other species of mites called

the *prey*. The experiment was set up in a way that ensured the predator species did not consume the entire prey population [30, page 53]. The differential equations that describe this model are similar to the ones which can be derived from Lotka's hypothetical chemical reactions.

Lotka's chemical system is given by the following reactions [21, page 2350],

$$X + Y_1 \xrightarrow{k_1} 2Y_1 \tag{5.1}$$

$$Y_1 + Y_2 \xrightarrow{k_2} 2Y_2 \tag{5.2}$$

$$Y_2 \xrightarrow{k_3} *, \tag{5.3}$$

where X, Y_1 and Y_2 are the chemical species, and k_1, k_2 and k_3 are the reaction rates. The first two reactions are autocatalytic because the species Y_1 and Y_2 are both products and promoters of the reactions. The third reaction is just the decay of the species Y_2. In terms of ecological models, these reactions may be interpreted as the prey species reproducing by feeding on some food source X that does not deplete over time (Equation (5.1)), the predator species Y_2 reproducing by feeding on the prey species Y_1 (Equation (5.2)), and the predator species dying by natural causes (Equation (5.3)). This interpretation, however, is only useful for forming an impression of the nature of these reactions, and therefore let us return to modelling the reactions.

The deterministic ODE model of Lotka's chemical system consists of the following equations,

$$\frac{dy_1(t)}{dt} = k_1 x y_1(t) - k_2 y_1(t) y_2(t) \tag{5.4}$$

$$\frac{dy_2(t)}{dt} = k_2 y_1(t) y_2(t) - k_3 y_2(t), \tag{5.5}$$

where lower case letters are used to denote concentrations (see Section 2.2.1). Figure 5.1a shows a solution to these deterministic ODEs, which was found by using the forward Euler method. It is evident from the figure that, with certain parameters, the concentrations of both species oscillate, and the concentration of Y_2 generally follows the concentration of Y_1 (which is to be expected since the number of molecules of Y_2 can only increase/decrease when the number of molecules of Y_1 increases/decreases). The phase plot of this solution, shown in Figure 5.1b, suggests that the oscillatory and periodic behaviour of Lotka's reactions will continue forever. The inability of the deterministic ODE approach to account for random fluctuations that are inherent to Lotka's chemical system may lead to such an erroneous conclusion. To show more clearly why this is a drawback of the deterministic ODE approach, let us examine the case when the chemical system reaches a steady state.

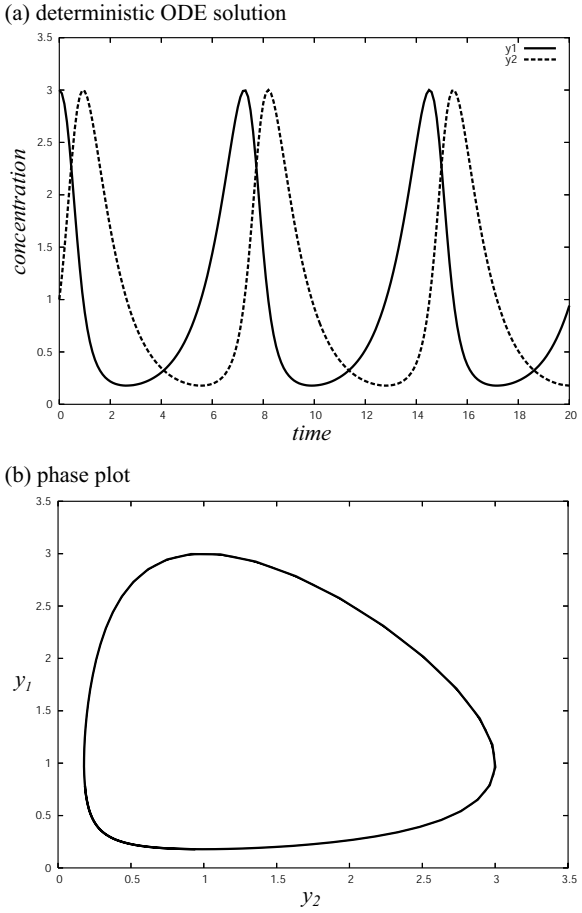


Figure 5.1: A solution to Lotka's chemical system: (a) The deterministic ODE solution with initial conditions $y_1(0) = 3$, $y_2(0) = 1$ and $k_1 x = 1$, $k_2 = 1$ and $k_3 = 1$. (b) The phase plot for the solution in (a).

Lotka's chemical system reaches a steady state when the deterministic differential equations (5.5) satisfy the following condition,

$$\frac{dy_1(t)}{dt} = \frac{dy_2(t)}{dt} = 0. \tag{5.6}$$

There are only two cases that satisfy this condition, which can be found by solving

for the zeros in the equations (5.5). They are given by:

$$y_1(t) = y_2(t) = 0, \text{ and} \tag{5.7}$$

$$y_1(t) = k_3/k_2,\ y_2(t) = k_1 x/k_2. \tag{5.8}$$

The first case is obvious, but the second one shows that if $y_1(0) = k_3/k_2$ and $y_2(0) = k_1 x/k_2$ are the initial concentrations used in the deterministic differential equations (5.5), the solution to these equations will not show an oscillatory and periodic behaviour. Furthermore, the concentrations of the two species will never change [21, page 2350]. To show that this prediction from the deterministic ODE approach is incorrect, let us model Lotka's chemical system using the stochastic approach.

5.1.2 Stochastic Approach to Lotka's Chemical System

In the stochastic formulation of Lotka's chemical reactions, there are $N = 2$ chemical species represented by the set $S = \{Y_1, Y_2\}$ which gives the species' types. There are also $M = 3$ reactions that are described by the following set,

$$\begin{aligned} R \;=\; \{&X + Y_1 \longrightarrow 2Y_1, \\ &Y_1 + Y_2 \longrightarrow 2Y_2, \\ &Y_2 \longrightarrow *\}. \end{aligned} \tag{5.9}$$

The chemical species X is not considered in the set S because its value is assumed constant. The values of the propensity functions are given by,

$$\begin{aligned} a_1(Y_1, Y_2) &= c_1 X Y_1 \\ a_2(Y_1, Y_2) &= c_2 Y_1 Y_2 \\ a_3(Y_1, Y_2) &= c_3 Y_2. \end{aligned} \tag{5.10}$$

In the case of Lotka's chemical system, the stochastic reaction constants c_1, c_2 and c_3 are equivalent to the reaction rate constants k_1, k_2 and k_3 (i.e. we do not divide by V) because all three reactions can be interpreted as monomolecular. The reason is that Y_1 appears as a reactant and product in the first reaction, and similarly Y_2 appears as a reactant and product in the second reaction.

The stochastic Petri net visualization of Lotka's chemical system is given in Figure 5.2. There are three transitions with each one representing one of the chemical reactions. A place for the species, X, is not included in the stochastic Petri net because we assume it is a constant.

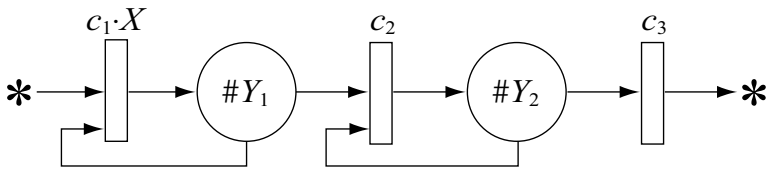


Figure 5.2: Stochastic Petri-net for Lotka's system, where $\#Y_1$ and $\#Y_2$ is the number of molecules of species Y_1 and species Y_2, respectively.

Using the same initial conditions as for Equations (5.8), the initial number of Y_1, Y_2 molecules is given by $X_i^{(0)} = \{c_3/c_2, c_1X/c_2\}$, i.e., at time $t = 0$ there are c_3/c_2 number of Y_1 molecules and c_1X/c_2 number of Y_2 molecules. Let us now address the question of how the number of Y_1 and Y_2 molecules change over time.

5.1.3 Lotka-Volterra Chemical Master Equation

It is easy to write the chemical master equation for any chemical system. In Lotka's chemical system, there are $M + 1 = 3 + 1$ ways in which the system's state can change from time t to $t + dt$. Thus, there are four values which give the probability

of a transition from the current state $\{Y_1, Y_2\}$ to a new state, which depends on the reaction type. These are the values:

$$P(\{Y_1, Y_2\} \to \{Y_1 + 1, Y_2\}) = c_1 X Y_1 dt \tag{5.11}$$

$$P(\{Y_1, Y_2\} \to \{Y_1 - 1, Y_2 + 1\}) = c_2 Y_1 Y_2 dt \tag{5.12}$$

$$P(\{Y_1, Y_2\} \to \{Y_1, Y_2 - 1\}) = c_3 Y_2 dt \tag{5.13}$$

$$P(\{Y_1, Y_2\} \to \{Y_1, Y_2\}) = 1 - [c_1 X Y_1 + c_2 Y_1 Y_2 + c_3 Y_2]dt. \tag{5.14}$$

By summing these transition probabilities, the chemical master equation for Lotka's system is given by the following equation [15, page 10],

$$\begin{aligned} \frac{\partial P(\{Y_1, Y_2\}, t)}{\partial t} = {} & c_1 X (Y_1 - 1) P(\{Y_1 - 1, Y_2\}, t) + \\ & c_2 (Y_1 + 1)(Y_2 - 1) P(\{Y_1 + 1, Y_2 - 1\}, t) + \\ & c_3 (Y_2 + 1) P(\{Y_1, Y_2 + 1\}, t) - \\ & [c_1 X Y_1 + c_2 Y_1 Y_2 + c_3 Y_2] P(\{Y_1, Y_2\}, t). \end{aligned} \tag{5.15}$$

This equation gives us the exact behaviour of Lotka's system over time, and by solving it (to get a formula for $P(\{Y_1, Y_2\}, t)$) we could compare our results with the deterministic ODE model's results. Unfortunately, Equation (5.15) is difficult to solve analytically or numerically: approximating methods are needed to find an analytical solution (as Gardiner has done [15, page 268]), and the numerical solution is computationally intractable (after only ten time steps, about one million probabilities must be computed). Figure 5.3 shows the number of probabilities that must be computed after only two time steps is 16. Therefore, instead of solving the chemical master equation for Lotka's system, let us use the stochastic simulation approach which was developed by Gillespie [20].

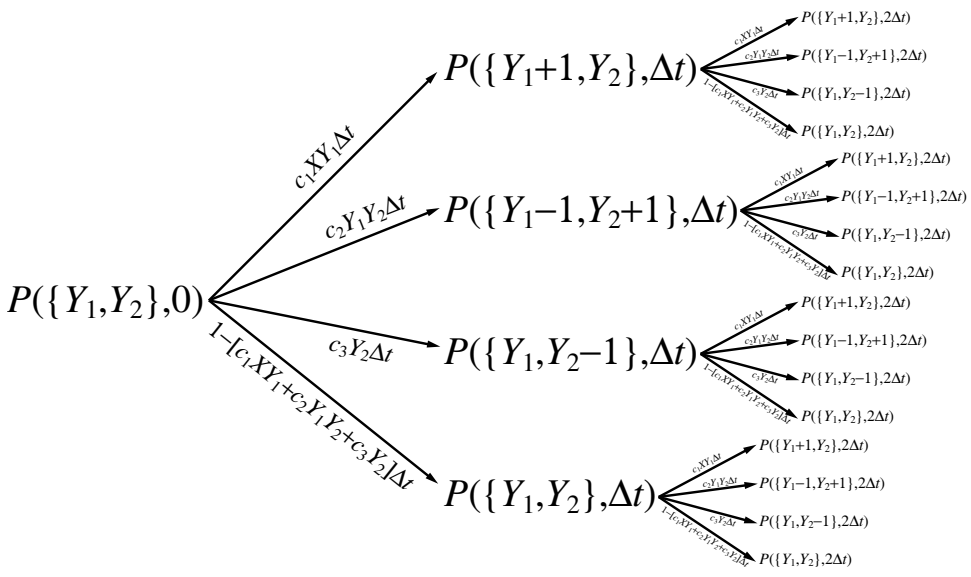


Figure 5.3: Numerical solution to the Lotka-Volterra chemical master equation, which shows that 16 probabilities must be computed after two time steps.

5.2 Stochastic Simulation of Lotka's Chemical System

Recall, in the case under consideration the deterministic ODE model predicts that the concentrations of the two chemical species in Lotka's system will not change over time. To show this prediction is false, a program that stochastically simulates Lotka's chemical system over time must be created. The L-system modelling language L+C is used for this purpose, and lpfg is used to simulate the model (the reason for using L+C and lpfg was given in Section 1). Program 1 gives the L+C source code for the stochastic model of Lotka's chemical system.

Program 1: A stochastic model of Lotka's chemical system implemented using Gillespie's method in lpfg

```
1 #include <lpfgall.h>
2
```

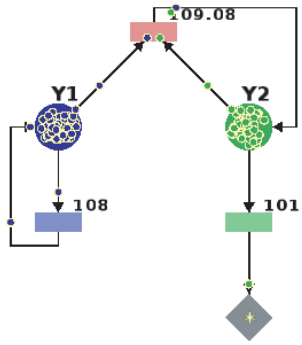
```
3  const float c1 = 0.001;
4  const float c2 = 0.01;
5  const float c3 = 10.0;
6  const float X = 10000.0;
7
8  float gillespie_time;
9  Start: {gillespie_time = 0.f;}
10
11 module Cell(int,int);
12 derivation length: 0;
13 Axiom: Cell(1000,1000);
14
15 Cell(Y1, Y2):
16 {
17   float propensity[3];
18   float current_sum, propensity_sum, next_reaction;
19   int reaction_index;
20
21   // calculate propensities and then their sum
22   propensity[0] = ((float)(X * Y1)) * c1;
23   propensity[1] = ((float)(Y1 * Y2)) * c2;
24   propensity[2] = ((float)(Y2)) * c3;
25
26   propensity_sum = propensity[0] +
27                    propensity[1] +
28                    propensity[2];
29
30   // using direct method, pick next reaction and time of reaction
31   if (propensity_sum > 0.f)
32   {
33     next_reaction = ran(1) * propensity_sum;
34
35     reaction_index = 0;
36     current_sum = propensity[reaction_index];
37     while (current_sum < next_reaction && reaction_index < 3)
38       current_sum += propensity[++reaction_index];
39
40     // depending on which reaction was picked,
41     // update number of molecules involved in the reaction
42     if (reaction_index == 0) { ++Y1; }
```

```
43      else if (reaction_index == 1) { --Y1; ++Y2; }
44      else if (reaction_index == 2) { --Y2; }
45
46      // update the global simulation time
47      gillespie_time += -log(1.0-ran(1)) / propensity_sum;
48    }
49    produce Cell(Y1, Y2);
50  }
```

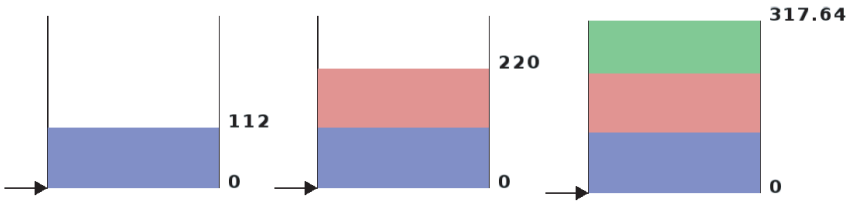
This implementation of Lotka's chemical system in lpfg follows Gillespie's algorithm exactly. First, the stochastic reaction constants are set in lines 3 to 6, the time variable is set to zero in line 9 and the initial conditions are set in line 13. For each derivation step of the L-system (or for each simulation step in Gillespie's algorithm), the propensities are calculated in lines 22 to 24 and their sum is calculated in line 26. The reaction that takes place next is found in lines 33 to 38, and the number of molecules of `Y1` and `Y2` are updated according to that reaction in lines 42 to 44. Finally, the time of the next reaction is found in line 47. Figure 5.4 is a visualization of how Gillespie's algorithm runs on Lotka's chemical system.

By recording the `Y1` and `Y2` values at time, `gillespie_time`, for each derivation step, the state X of Lotka's system over time t is found. Running this program J times, provides a means of estimating the expectation and variance of `Y1` and `Y2`, but this is not done in the program. Before estimating these values, let us examine a better implementation of Lotka's chemical system.

(a) Changes in number of molecules for Lotka's chemical system



(b) Selection of next reaction according to propensities



(c) Selection of the time of next reaction

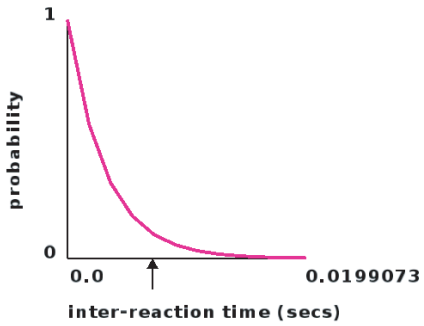


Figure 5.4: Visualization of Gillespie's algorithm for Lotka's chemical system: (a) the change in the number of molecules visualized as tokens moving through the stochastic Petri net (propensity of each reaction is given by the number shown in the upper right corner of a transition box), (b) selection of which reaction takes place next according to the propensity of each reaction, where the sum of propensities is shown on the right-hand side of the graph (the arrow points to the next reaction), and (c) generating the time of the next reaction according to an exponential distribution (the arrow points to the time of the next reaction).

If we consider which lines in the above program are actually a part of the Lotka system's model, we find that only the propensity calculations and the lines that change the number of molecules fit this part. It is reasonable then to suggest that the other generic parts of the program could be handled internally by lpfg. For this reason, lpfg has been extended to include Gillespie's algorithm and L+C has been extended to include the necessary constructs (see Section 1.1 for a list of contributions).

5.2.1 Gillespie's Method as a Rewriting Strategy in L-Systems

One reason for extending the L+C modelling language and the lpfg simulator to include Gillespie's method is that most of the algorithm can be handled internally by lpfg. The inclusion of this method makes the task of writing models for chemical systems less complicated, especially for modelling filamentous cellular structures. A stochastic rewriting strategy, called *stochastic L-systems* [56, page 28], has already been introduced for L-systems in cpfg. It was necessary to introduce this strategy because all of the visualizations generated from a model are identical otherwise, which is an undesirable effect for biological modelling (e.g., of plants).

In a stochastic L-system, a production is randomly selected from several matching productions according to the probability factor assigned to each matching production. The probability factor, in cpfg, is an arithmetic expression given behind the successor in a production following a colon. It is equivalent to the propensity of a reaction occurring in terms of Gillespie's algorithm. Consider, for example, the

stochastic L-system for asymmetrical cell division given below.

$$\begin{aligned} Cell(x) &\rightarrow Cell(x * 0.45)Cell(x * 0.55) : 9 \\ Cell(x) &\rightarrow Cell(x * 0.55)Cell(x * 0.45) : 6 \end{aligned} \tag{5.16}$$

The first rule in this example is for cell division of a mother cell into a shorter left daughter cell and a longer right daughter cell. The second rule is just the reverse cell division of the first rule. The probability factors have been assigned in such a way that applies the first rule more often than the second one (60% for rule 1, and 40% for rule 2). Because of the clear connection between probability factors and propensities, it is tempting to use stochastic L-systems in cpfg for modelling chemical systems. Therefore, using stochastic L-systems with Gillespie's direct method, Lotka's chemical system can be modelled by the following rules:

$$\begin{aligned} Cell(Y_1, Y_2, t) &\longrightarrow Cell(Y_1 + 1, Y_2, t - ln(1 - ran(1))/p) : c_1 X Y_1 \\ Cell(Y_1, Y_2, t) &\longrightarrow Cell(Y_1 - 1, Y_2 + 1, t - ln(1 - ran(1))/p) : c_2 Y_1 Y_2 \\ Cell(Y_1, Y_2, t) &\longrightarrow Cell(Y_1, Y_2 - 1, t - ln(1 - ran(1))/p) : c_3 Y_2, \end{aligned} \tag{5.17}$$

where $p = c_1 X Y_1 + c_2 Y_1 Y_2 + c_3 Y_2$. Each of these rewriting rules represents one of Lotka's chemical reactions, and the probability factor given at the end of the rule is equal to the propensity for that reaction. The third parameter in each module, t, is the time of the last reaction. It is advanced by the inter-reaction time, τ, which is given by the equation, $\tau = -ln(1 - ran(1))/p$, where $ran(1)$ is a uniform random number in the interval $[0, 1)$.

The drawback of implementing Gillespie's method using stochastic L-systems in cpfg becomes apparent when there are at least two modules interacting through

signalling (e.g., by diffusion of molecules between cells). The problem is that the rewriting rules are applied to each module separately, without taking into consideration the rules that may be applied in neighbouring modules. This is a subtle difference compared to Gillespie's method, where only one reaction is chosen from the set of all possible reactions. In other words, for each derivation step one reaction is chosen per module in a stochastic L-system while one reaction for all modules is chosen in Gillespie's method. Also, applying rewriting rules to each module separately introduces the difficulty of synchronizing the time t between different modules. Thus, let us examine a stochastic rewriting strategy for L-systems which matches Gillespie's method.

Extension of the L+C Language with Gillespie's Method

In L+C, we can use Gillespie's method to simulate chemical and biological L-system models by specifying a *Gillespie group*. In each module of the Gillespie group, the propensity of a reaction is specified following the key word `propensity` and the change in the system's state is given as a successor following the key word `produce`. For each derivation step, lpfg will randomly choose the reaction that takes place next and the time of the next reaction. It will pick the next reaction according to the propensities specified in each production of *all* the modules in a Gillespie group, so that the reaction with greatest propensity will more likely be picked. Then, lpfg will generate the time of the next reaction according to the equation $\tau = -ln(1-\xi)/p$, where ξ is a uniform random number in the interval $[0, 1)$ and p is the sum of the propensities of all the modules. As an example of using Gillespie's method in L+C, consider the implementation of Lotka's chemical system given in Program 2.

Program 2: A stochastic model of Lotka's chemical system implemented using Gillespie groups in lpfg

```
1  #include <lpfgall.h>
2
3  const float c1 = 0.001;
4  const float c2 = 0.01;
5  const float c3 = 10.0;
6  const float X = 10000.0;
7
8  float gillespie_time = 0.0;
9  Start: {UseGroup(1);}
10 EndEach: {gillespie_time = GillespieTime();}
11
12 module Cell(int,int);
13 derivation length: 0;
14 axiom: Cell(1000,1000);
15
16 ggroup 1:
17 Cell(Y1, Y2) :
18 {
19   propensity (X * Y1 * c1) produce Cell(Y1+1,Y2);
20   propensity (Y1 * Y2 * c2) produce Cell(Y1-1,Y2+1);
21   propensity (Y2 * c3) produce Cell(Y1,Y2-1);
22 }
```

In this example, there is a module `Cell` with two parameters `Y1` and `Y2` that specifies the three reactions in the system and the propensity of each reaction. Each one of the `propensity...produce...` statements represents one of the chemical reactions in the system. There is also a global variable `gillespie_time` that represents the time at which the last reaction occurred in the system. In each derivation step, lpfg will execute one of the `propensity...produce...` statements, which will update the values of `Y1` and `Y2`, and lpfg will update the last reaction time, which is stored in the `gillespie_time` variable. Comparing this program to the previous

one, it should be immediately obvious that having Gillespie groups in lpfg is a huge benefit. This is true in terms of readability (describing the L-system model to others is far easier) and implementation of new models (this version of the program is only half as long as the previous version).

The stochastic rewriting strategy for Gillespie's method in L+C and the strategy in stochastic L-systems are similar except for the way in a which a rewriting rule is chosen. The first one chooses one successor from all modules and the second one chooses one per module. The benefit of the Gillespie strategy in L+C over the stochastic strategy in cpfg will become more clear (in the next chapter) when it is used to simulate a model of diffusion between cells. For now, however, let us return to simulating Lotka's chemical system over time t using Gillespie's method.

5.2.2 Expectation and Variance of Lotka's Chemical System

Running Program 2 in Section 5.2.1, which is a model of Lotka's chemical system, will give one possible outcome of its behaviour over time. The program uses the initial conditions $Y_1 = c_3/c_2$ and $Y_2 = c_1X/c_2$, for which the deterministic ODE model would predict that there is no change in the concentration of the two species. Figure 5.5 shows the state of Lotka's system from time $t = 0$ to $t = 10$ produced by two runs of the simulation. It is easy to see that even with the initial conditions $Y_1 = c_3/c_2$ and $Y_2 = c_1X/c_2$ the stochastic model predicts that the system's behaviour will be oscillatory. This is a good indication that the behaviour of Lotka's system cannot be realistically predicted without taking into account the random fluctuations that are inherent to the system [21, page 2351]. The reason why can be explained by estimating the expected value and variance of the number of molecules in Lotka's

system.

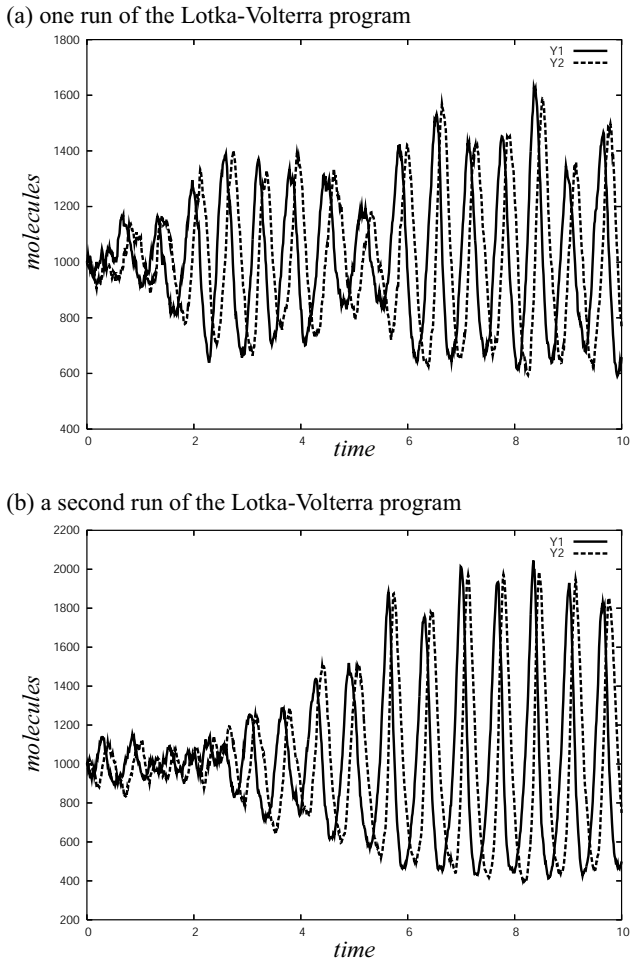


Figure 5.5: Stochastic simulation of the Lotka's chemical system: (a) A single run of the Lotka-Volterra program. (b) A second run of the exact same program but using a different sequence of random numbers.

Figure 5.6 shows an estimate for the expected value and variance of the number of molecules in Lotka's system over time t. The estimate of these values was taken over 10,000 runs of the program. The variable t was considered in small discrete time steps and the average state of the system was calculated in each step. In the

example shown in Figure 5.6 the time step is 0.01 units.

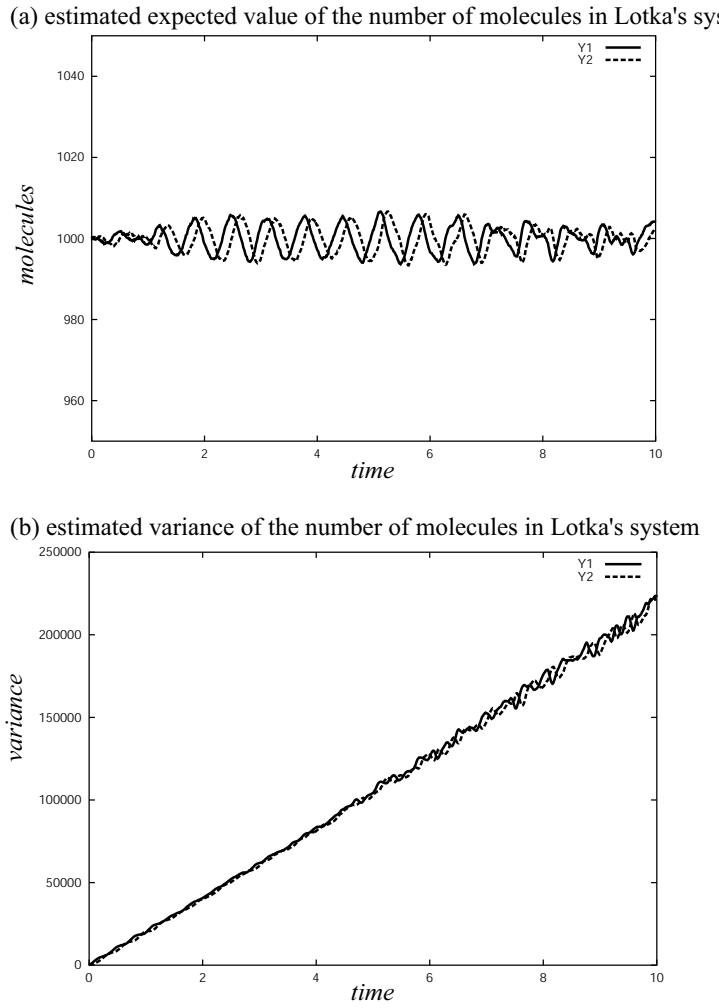


Figure 5.6: Expectation (a) and variance (b) of the number of molecules in Lotka's chemical system, estimated after 10,000 runs of the Lotka-Volterra program.

Figure 5.6a shows that the expected value of the number of Y_1, Y_2 molecules estimated by stochastically simulating Lotka's system with the initial conditions $Y_1 = c_3/c_2$ and $Y_2 = c_1X/c_2$ is close to the deterministic ODE model's solution. The number of molecules of Y_1 and Y_2 on average behave like the deterministic ODE

model predicts, and with a larger number of simulation runs that estimate would be even closer. Figure 5.6b, however, shows that the variance of the number of Y_1, Y_2 molecules is very large compared to the mean. This suggests that Lotka's reactions should not be modelled using deterministic ODEs because the third assumption of the approach cannot be satisfied (i.e. the fluctuations in concentration about the mean are not small). To further examine the affect of noise on Lotka's system, the phase plot in Figure 5.7 was generated using the stochastic simulation approach. It shows, that in contrast to the phase plot produced using the deterministic ODE approach (Figure 5.1), the random fluctuations inherent to the system cause the number of Y_1, Y_2 molecules to greatly change over the simulation time. Eventually, the number of molecules will end up being $Y_1 = \infty$ and $Y_2 = 0$ because if there are no molecules of species Y_2 left, the number of Y_1 molecules will increase indefinitely. This behaviour is not as easily found in the deterministic ODE model because the phase plot for that type of model suggests that the concentration of Y_1 and Y_2 never changes. Finding the final state, $\{Y_1 = \infty, Y_2 = 0\}$, of Lotka's system in the deterministic ODE approach requires additional mathematical analysis. It may be found by linearizing the ODEs using partial derivatives (i.e. finding the Jacobian matrix), and evaluating the partial derivatives at the steady states. In the stochastic simulation approach, however, we have it straight from the solution.

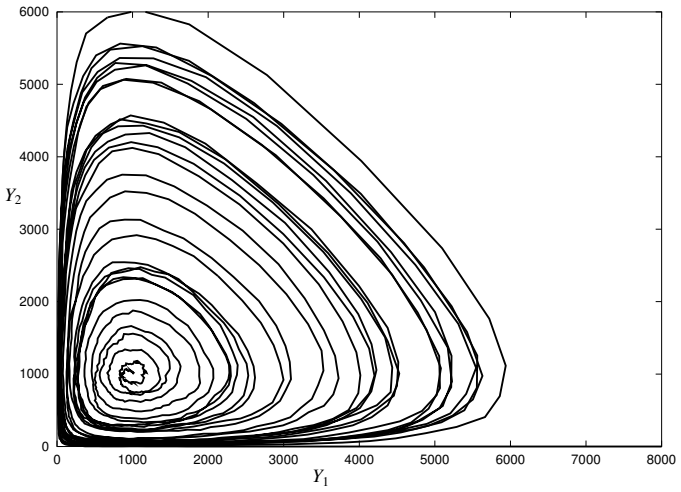


Figure 5.7: Phase plot of the number of molecules in Lotka's chemical system.

5.3 Quasi-steady State Enzyme Kinetics

A common type of process in many biological systems is based on the mechanism of enzyme catalysed reactions. The first term of the type of reaction-diffusion equations that will be considered in the next chapter are based on this mechanism. Thus, let us examine some enzymatic reactions and how the stochastic approach may be used to study them.

As a first example, consider the following enzymatic reactions studied by Michaelis and Menten [44],

$$\begin{aligned} S + E &\xrightarrow{k_1} Z \\ Z &\xrightarrow{k_2} S + E \\ Z &\xrightarrow{k_3} P + E, \end{aligned} \tag{5.18}$$

where E is the enzyme, S is the substrate, Z is an intermediate species, and P is

the product. The deterministic ODEs for these reactions are:

$$\begin{aligned}
\frac{ds}{dt} &= -k_1 se + k_2 z \\
\frac{de}{dt} &= -k_1 se + k_2 z + k_3 z \\
\frac{dz}{dt} &= k_1 se - k_2 z - k_3 z \\
\frac{dp}{dt} &= k_3 z, \qquad (5.19)
\end{aligned}$$

which may be simplified by letting e be the difference of the initial concentration of the enzyme and the intermediate species (i.e. $e = e_0 - z$) to reduce the number of equations. That is the simplified deterministic ODEs are:

$$\begin{aligned}
\frac{ds}{dt} &= -k_1 s(e_0 - z) + k_2 z \\
\frac{dz}{dt} &= k_1 s(e_0 - z) - k_2 z - k_3 z \\
\frac{dp}{dt} &= k_3 z. \qquad (5.20)
\end{aligned}$$

Michaelis and Menten made the assumption that the concentration of the intermediate species, z, does not change very much over time (i.e. $dz/dt \approx 0$, is in a quasi-steady state), so that

$$0 = k_1 s(e_0 - z) - k_2 z - k_3 z.$$

From this, we calculate z:

$$z = k_1 s(e_0)/(k_1 s + k_2 + k_3). \qquad (5.21)$$

Substituting Equation (5.21) into the equation for ds/dt, we obtain

$$\begin{aligned}
\frac{ds}{dt} &= -k_1 s e_0 + (k_1 s + k_2)\frac{k_1 s e_0}{k_1 s + k_2 + k_3} \\
&= -k_1 s e_0 \left[1 - \frac{k_1 s + k_2}{k_1 s + k_2 + k_3}\right] \\
&= -k_1 s e_0 \left[\frac{k_3}{k_1 s + k_2 + k_3}\right] \\
&= -\frac{k_3 s e_0}{\frac{k_2 + k_3}{k_1} + s}. \qquad (5.22)
\end{aligned}$$

If we let $V_{max} = k_3 e_0$ and $K_m = (k_2 + k_3)/k_1$, then we have reduced the Michaelis-Menten reactions to the following reaction,

$$S \longrightarrow P, \text{ with reaction rate } \frac{V_{max} s}{K_m + s}, \qquad (5.23)$$

where V_{max} is called the maximum velocity and K_m is called the Michaelis-Menten rate constant [47]. The assumption made by Michaelis and Menten is a good way of reducing the complexity of a model and the complexity of computations for studying the model [59, page 5000]. It can also be made in other reactions involving enzyme kinetics.

Let us now consider slightly different enzymatic reactions from the Michaelis-Menten ones. The reactions are given by,

$$\begin{aligned}
2S + E &\xrightarrow{k_1} Z \\
Z &\xrightarrow{k_2} 2S + E \\
Z &\xrightarrow{k_3} 2P + E, \qquad (5.24)
\end{aligned}$$

which are almost identical to the Michaelis-Menten reactions except for the factor of two associated with the substrate and product. The deterministic ODEs for these

reactions are given by,

$$\begin{aligned}
\frac{ds}{dt} &= -k_1 s^2(e_0 - z) + k_2 z \\
\frac{dz}{dt} &= k_1 s^2(e_0 - z) - k_2 z - k_3 z \\
\frac{dp}{dt} &= k_3 z. \qquad (5.25)
\end{aligned}$$

If we apply the same steady state assumption as in the Michaelis-Menten reactions, we obtain:

$$\begin{aligned}
\frac{ds}{dt} &= -k_1 s^2 e_0 + (k_1 s^2 + k_2)\frac{k_1 s^2 e_0}{k_1 s^2 + k_2 + k_3} \\
&= -k_1 s^2 e_0 \left[1 - \frac{k_1 s^2 + k_2}{k_1 s^2 + k_2 + k_3}\right] \\
&= -k_1 s^2 e_0 \left[\frac{k_3}{k_1 s^2 + k_2 + k_3}\right] \\
&= -\frac{k_3 s^2 e_0}{\frac{k_2 + k_3}{k_1} + s^2}. \qquad (5.26)
\end{aligned}$$

Based on this assumption, the three reactions (5.24) reduce to the following reaction,

$$S \longrightarrow P, \text{ with reaction rate } \frac{V_{max} s^2}{K_m + s^2}. \qquad (5.27)$$

The general form of this reaction is due to Hill [28], where the substrate is raised to any power η_H, called the Hill coefficient. Now, let us consider the stochastic approach to quasi-steady state enzyme kinetics.

5.3.1 Stochastic Approach to Enzyme Kinetics

Rao and Arkin derive the quasi-steady state solution for the stochastic formulation of the Michaelis-Menten reactions by separating the primary species from the intermediate species in the chemical master equation [59, page 5002]. This allows them to

make the same quasi-steady state assumption in the stochastic case that was made in the deterministic case. Using this assumption, they give the approximating chemical master equation for the Michaelis-Menten reactions (5.18) as the following,

$$\frac{dP(\{S,P\},t)}{dt} = -\frac{V_{max}S}{K_m+S}P(\{S,P\},t) + \frac{V_{max}(S+1)}{K_m+(S+1)}P(\{S+1,P-1\},t). \quad (5.28)$$

They derive this equation from the chemical master equation for the Michaelis-Menten reactions to show that it is valid in the stochastic case [59, page 5005]. In the case of the Hill reaction with a coefficient of two, the corresponding master equation has the form,

$$\begin{aligned}\frac{dP(\{S,P\},t)}{dt} &= -\frac{V_{max}S^2}{K_m+S^2}P(\{S,P\},t) \\ &\quad +\frac{V_{max}(S+1)^2}{K_m+(S+1)^2}P(\{S+1,P-1\},t). \end{aligned} \quad (5.29)$$

It is impractical to solve this equation for the purpose of studying quasi-steady state enzyme kinetics. Let us instead find the propensity of the reaction and use Gillespie's method to stochastically simulate the process. Our first task is to find the relationship between reaction rates and stochastic reaction constants.

The relationship between the reaction rates, k_1, k_2, and k_3 and the stochastic reaction constants, c_1, c_2, and c_3 for the Michaelis-Menten reactions (5.18) can be extracted from Table 2.1 and is given by,

$$\begin{aligned} c_1 &= k_1/V^2 \\ c_2 &= k_2 \\ c_3 &= k_3, \end{aligned} \quad (5.30)$$

where V is the volume of the chemical system. For the Hill reaction with a coefficient of two, the relationship for c_1 changes to $c_1 = 2k_1/V^2$. In the quasi-steady state, the

relationship between the constants is one-to-one because the reaction only involves one reactant. That is we use the same values of V_{max} and of K_m to find the propensity of a reaction in the stochastic simulation approach as for the reaction rate in the differential equation approach. Knowing this relationship, we can simulate quasi-steady state enzyme kinetics using Gillespie's method. Since the Hill reaction with a coefficient of two is used in the first term of the reaction-diffusion models considered in the next chapter, let us examine the stochastic simulation approach for this reaction.

5.3.2 Stochastic Simulation of the Hill Reaction

The propensity of the quasi-steady state Hill reaction is given by

$$\frac{s^2 V_{max}}{K_m + s^2} = \frac{\frac{S^2}{V^2} V_{max}}{K_m + \frac{S^2}{V^2}} = \frac{S^2 V_{max}}{K_m V^2 + S^2}. \tag{5.31}$$

A minor detail in this calculation is that the value of $V_{max} = k_3 e_0$ must be multiplied by the system's volume, V, because we are dealing with the number of initial molecules of the enzyme E and not the initial concentration (i.e. E_0 not e_0). Figure 5.8 shows the stochastic Petri-net for reaction (5.27). The propensity of one molecule of the product P being produced from the substrate S is written inside the transition box (instead of outside) for clarity. The lpfg program that simulates the process that this stochastic Petri-net describes is given in Program 3.

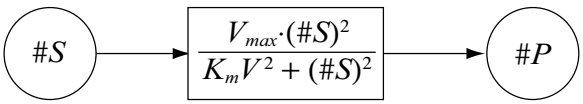


Figure 5.8: Stochastic Petri-net for the Hill reaction with a Hill coefficient of two, where $\#S$ and $\#P$ are the number of molecules of species S and P, respectively.

Program 3: A stochastic model of the Hill reaction implemented in lpfg using Gillespie groups

```
1  #include <lpfgall.h>
2
3  const float V = 1000.0; /* the volume of a cell */
4  const float e0 = 1.0;   /* initial concentration of E */
5  const float E0 = e0*V;  /* initial number of molecules of E */
6  const float k1 = 1.0;   /* the reaction rates from */
7  const float k2 = 1.0;   /* the continuous model */
8  const float k3 = 1.0;
9
10 Start: {UseGroup(1);}
11 derivation length: 1000;
12
13 /* a cell with parameters:
14    number of substrate & product molecules */
15 module Cell(int, int);
16
17 axiom: Cell(1 * V, 0 * V);
18
19 ggroup 1:
20 Cell(S, P) :
21 {
22   float Vmax = k3 * E0;
23   float Km = (k2 + k3) / k1;
24   propensity (Vmax * S * S / (Km * V * V + S * S))
25     produce Cell(S-1, P+1);
26 }
```

The number of molecules of the substrate, S, and the product, P, are initialized in line 17 as a deterministic concentration value multiplied by the cell's volume, `V`. In the Gillespie group, the propensity of one molecule of the product being produced from the substrate is calculated. Figure 5.9 compares a possible solution to the Hill reaction for the deterministic ODE approach and the stochastic simulation approach.

The models of Lotka's chemical system and quasi-steady state enzyme kinetics, which have been examined in this chapter, are good examples of chemical systems in a non-spatial structure. Let us now apply Gillespie's algorithm to a linear structure using L-systems, which will better show the power of combining the two concepts.

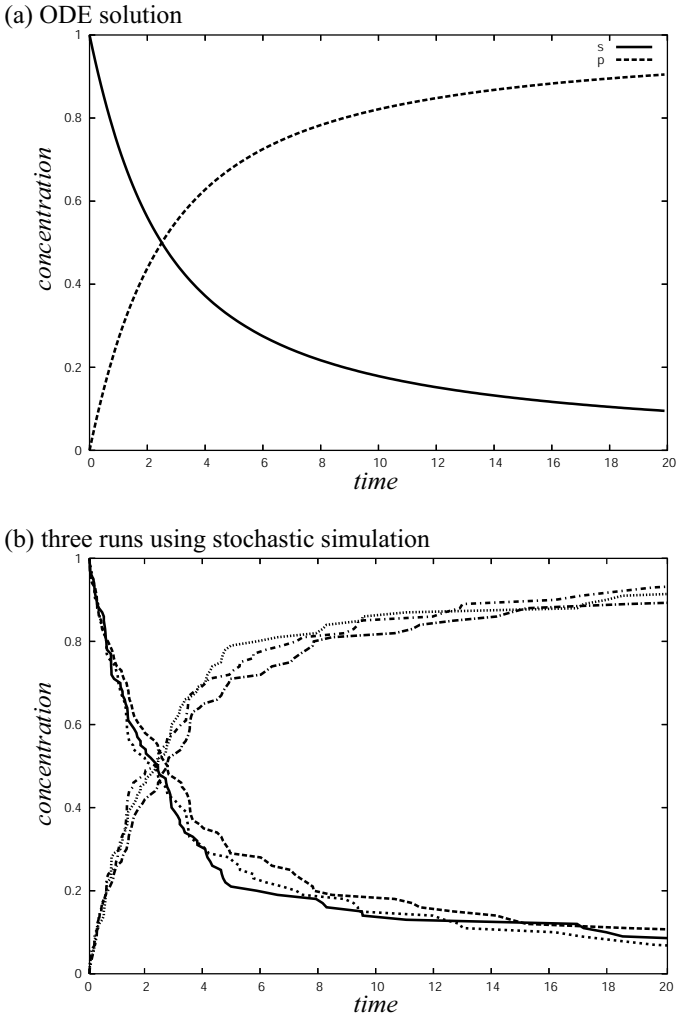


Figure 5.9: Comparison between the deterministic ODE solution and stochastic simulation solution for Hill's reaction.

Chapter 6

Gillespie's Method in a Linear Structure: Stochastic Simulation of Reaction-Diffusion

6.1 Introduction

The intention of Alan Turing's reaction-diffusion model was to explain morphogenesis [65]; the reason why patterns emerge from a homogeneous medium [52, 53]. His model included two substances, a and b, reacting and diffusing in an xy-plane, which he described using the following differential equations:

$$\begin{aligned} \frac{\partial a}{\partial t} &= f(a,b) + D_a \left(\frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} \right) \\ \frac{\partial b}{\partial t} &= g(a,b) + D_b \left(\frac{\partial^2 b}{\partial x^2} + \frac{\partial^2 b}{\partial y^2} \right). \end{aligned} \tag{6.1}$$

Turing named his model reaction-diffusion because the two substances are involved in reactions described by the functions, $f(a,b)$ and $g(a,b)$, and in diffusion described by the second term in both equations, where D_a and D_b are diffusion constants. He and several other researchers [66, 71, 73] used this type of model to generate patterns seen in nature. For instance, Meinhardt and Klinger [40, 41, 42] generated pigmentation patterns in sea shells using variants of the reaction-diffusion model proposed by Meinhardt and Gierer [19]. This technique of generating sea shell patterns was later used by Fowler, Meinhardt and Prusinkiewicz as a method for modelling sea shells suitable for image synthesis [14]. One of the variants is called the *activator-substrate*

model, and is described by the following equations,

$$\begin{aligned} \frac{\partial a}{\partial t} &= \rho s \left(\frac{a^2}{1+\kappa a^2} + \rho_0 \right) - \mu a + D_a \frac{\partial^2 a}{\partial x^2} \\ \frac{\partial s}{\partial t} &= \sigma - \rho s \left(\frac{a^2}{1+\kappa a^2} + \rho_0 \right) - \nu s + D_s \frac{\partial^2 s}{\partial x^2}, \end{aligned} \tag{6.2}$$

where the activator, a, and the substrate, s, react with each other and diffuse (separately) along a one-dimensional medium (the x axis) with rates D_a and D_s, respectively. The reactions that are a part of this model are the decay of the two substances captured by the terms, μa and νs, the production of the substrate s captured by the term, σ, and the production of the activator a, which is related to the depletion of s, captured by the remaining terms [14, page 382]. The production of a is an autocatalytic process because it is proportional to a^2, but it only occurs in the presence of s because s is a factor in that production. Similar equations to (6.2) were also used by Hammel and Prusinkiewicz in an L-system model of the bacteria *Anabaena catenula* to capture the "formation of equally spaced organs in a developing medium" [25]. Although these models have provided us with insight into the formation of patterns found in nature, they have not captured the noise that is inherent to the formation of such patterns. The purpose of this chapter is to describe a way of stochastically simulating reaction-diffusion using Gillespie groups in lpfg and to show the importance of capturing the noise that is part of this process. An important new element introduced in the stochastic simulation of reaction-diffusion, as opposed to the stochastic simulation of Lotka's chemical system, is the spatial character of the reaction-diffusion model. The importance of this element will become clear when a method for stochastically simulating the decay and diffusion process is examined.

6.2 The Decay-Diffusion Process

Before describing the stochastic simulation approach to the decay-diffusion process, let us examine the deterministic differential equation approach to this process. Consider the following partial differential equation [25, page 252]

$$\frac{\partial a}{\partial t} = -\mu a + D_a \frac{\partial^2 a}{\partial x^2}, \tag{6.3}$$

where a is the concentration of some chemical species that decays with rate μ and diffuses with rate D_a along a one-dimensional medium. This equation can be solved by using a *Forward Time Centred Space* differencing scheme, which is similar to the forward Euler differencing scheme except it uses a discretization in both space and time [51, page 847]. The update formula for the concentration of a_i in the space $i\Delta x$ at time $t + \Delta t$, given a_i at time t, is the following,

$$a_i(t + \Delta t) = a_i(t) + \left(-\mu a_i(t) + D_a \frac{a_{i-1}(t) - 2a_i(t) + a_{i+1}(t)}{(\Delta x)^2} \right) \Delta t, \tag{6.4}$$

where a_{i-1} and a_{i+1} are concentrations taken immediately to the left and right, respectively, of the current position in the one-dimensional medium. The easiest way to examine a solution to this equation is to produce a visual representation of it. Hammel and Prusinkiewicz used L-systems to simulate the decay-diffusion process and to produce a visual representation of the solution [25].

In a context-sensitive L-system model, the concentrations to the left and right of the current position in the medium are taken from the parameters of the left and right neighbouring modules in the string. This is a benefit of using L-systems because modelling the decay-diffusion process otherwise would require implementing an indexing scheme to access each subinterval in a discrete x axis, which is inconve-

nient and leads to less clear expressions when the model is of a growing structure. With that in mind, Hammel and Prusinkiewicz used the following L-system rule to solve Equation (6.3),

$$\begin{aligned} &Cell(a_l) < Cell(a) > Cell(a_r) \rightarrow \\ &Cell\left(a + \left(-\mu a + D_a \frac{a_l - 2a + a_r}{(\Delta x)^2}\right)\Delta t\right), \end{aligned} \tag{6.5}$$

where the rule computes Equation (6.4) for each module $Cell$ in the L-system's string during the derivation step. Using this rule, Hammel and Prusinkiewicz produced a visual representation of the solution to the decay-diffusion process [25, page 253]. Notice that the L-system rule does not take into account the first and last modules of the string because only those modules with a left and right neighbour are considered in the derivation step. The consequence is that there is an infinite supply of the chemical species diffusing from the first and last modules, which prevents the concentrations of the species in the other cells from decaying to zero. Now that the deterministic solution to the decay-diffusion process has been examined, let us find the stochastic solution.

6.2.1 Stochastic Decay & Diffusion

The stochastic approach to the decay-diffusion process in a one-dimensional medium requires finding the propensity of a molecule decaying or diffusing. The propensity for decay is already known (see Section 2.2.3) and Gillespie briefly describes an idea for calculating the propensity of diffusion between subvolumes in his original paper on the stochastic simulation algorithm [20, page 430]. To find the propensity of diffusion, let us first consider the decay-diffusion process using the chemical master

equation. We start by finding the transition probabilities of how the state of the process can change from time t to time $t + \Delta t$. These probabilities are given by

$$P(\{A_{i-1}, A_i, A_{i+1}\} \to \{A_{i-1}, A_i - 1, A_{i+1}\}) = \mu A_i \Delta t \tag{6.6}$$

$$P(\{A_{i-1}, A_i, A_{i+1}\} \to \{A_{i-1} + 1, A_i - 1, A_{i+1}\}) = \frac{D_a A_i}{(\Delta x)^2} \Delta t \tag{6.7}$$

$$P(\{A_{i-1}, A_i, A_{i+1}\} \to \{A_{i-1}, A_i - 1, A_{i+1} + 1\}) = \frac{D_a A_i}{(\Delta x)^2} \Delta t \tag{6.8}$$

$$P(\{A_{i-1}, A_i, A_{i+1}\} \to \{A_{i-1}, A_i, A_{i+1}\}) = 1 - \left[\mu A_i + \frac{2 D_a A_i}{(\Delta x)^2}\right] \Delta t, \tag{6.9}$$

where Equation (6.6) is the probability of one molecule decaying, Equation (6.7) is the probability of one molecule diffusing to the left, Equation (6.8) is the probability of one molecule diffusing to the right, and Equation (6.9) is the probability of the system's state not changing. Notice that, the equations deal with the number of molecules, A, instead of the concentration, a, as it would be the case for the deterministic ODE approach (Equation (6.4)). Now from these transition probabilities, the chemical master equation for the decay-diffusion process along a one-dimensional medium is given by

$$\begin{aligned} \frac{\partial P(\{A_{i-1}, A_i, A_{i+1}\}, t)}{\partial t} &= \mu(A_i + 1) P(\{A_{i-1}, A_i + 1, A_{i-1}\}, t) + \\ & D_a \frac{(A_i - 1)}{(\Delta x)^2} P(\{A_{i-1} + 1, A_i - 1, A_{i+1}\}, t) + \\ & D_a \frac{(A_i - 1)}{(\Delta x)^2} P(\{A_{i-1}, A_i - 1, A_{i+1} + 1\}, t) - \\ & \left[\mu A_i + D_a \frac{2 A_i}{(\Delta x)^2}\right] P(\{A_{i-1}, A_i, A_{i+1}\}, t). \end{aligned} \tag{6.10}$$

The general form of the chemical master equation for diffusion without decay in higher dimensional medium is given by Gardiner [15, page 307], and he finds the stationary solution of this equation to be a multivariate Poisson distribution. He does

not, however, find the solution to the chemical master equation for the reaction-diffusion process, and instead, investigates the behaviour of the process by approximating the solution using Langevin type stochastic differential equations. Alternatively, the stochastic simulation approach can be used to study the solution to the chemical master equation for this process, and the assumption that the number of molecules is large enough so that discrete changes can be approximated by continuous changes in concentration does not have to be made. Let us, therefore, use the propensities for decay and diffusion found in Equations (6.6)-(6.8) to stochastically simulate the decay-diffusion process.

6.2.2 Stochastic Simulation of Decay & Diffusion

Figure 6.1 shows a stochastic Petri net for the decay-diffusion process. It is an example of three cells separated by walls that undergo the decay and diffusion process along the x axis. The interpretation, according to this modularized stochastic Petri net, is that each cell has a place for the number of molecules in the cell, and a transition event for one molecule decaying. Each wall has two transition events: one for a molecule diffusing from a cell to its right neighbour, and the other for a molecule diffusing from a cell to its left neighbour. The stochastic reaction constant for each reaction is drawn along side the transition event for that reaction.

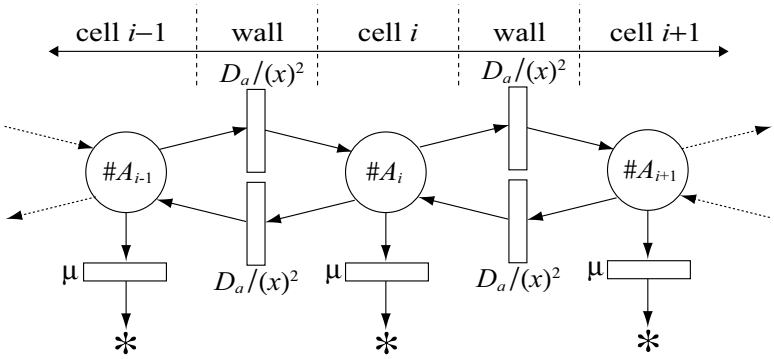


Figure 6.1: Stochastic Petri-net for decay & diffusion, where $\#A_i$ is the number of molecules of species A in cell i.

The decay and diffusion process that this stochastic Petri-net describes can be modelled by considering each cell and wall as a module in an L-system string, and using Gillespie groups in lpfg to handle the decay and diffusion events. The lpfg code that does this is given in Program 4.

Program 4: A stochastic model of the decay-diffusion process implemented in lpfg using Gillespie groups

```
1  #include <lpfgall.h>
2
3  const float MU = 0.01;      /* decay constant */
4  const float X = 1.0         /* discretized step of x-axis */
5  const float Da = 1.0/(X*X); /* diffusion constant */
6
7  /* diffusion direction through wall */
8  const int DIR_NONE = 0;
9  const int DIR_LEFT = 1;
10 const int DIR_RIGHT = -1;
11
12 int nextGroup = 1;
13 StartEach:{UseGroup(nextGroup);}
14 EndEach:{if (++nextGroup > 2) nextGroup = 1;}
15
```

```
16 derivation length: 1;
17
18 module Cell(int); /* a cell */
19 module Wall(int); /* a wall between two cells */
20
21 axiom: /* Create cells separated by walls */
22
23 /* find the propensity of diffusion, and decay */
24 ggroup 1:
25 Cell(Al) < Wall(dir) > Cell(Ar):
26 {
27   propensity (Da * Al) produce Wall(DIR_LEFT);
28   propensity (Da * Ar) produce Wall(DIR_RIGHT);
29 }
30 Wall(dirl) < Cell(A) > Wall(dirr):
31 {
32   propensity (A * MU) produce Cell(A-1);
33 }
34
35 /* diffuse a molecule from one cell to another */
36 group 2:
37 Wall(dirl) < Cell(A) > Wall(dirr):{produce Cell (A+dirl-dirr);}
38 Wall(dir): {produce Wall(DIR_NONE);}
```

This program requires two lpfg groups to function properly: `ggroup 1` to perform a step in Gillespie's algorithm and `group 2` to perform diffusion of a molecule between cells. Lines 12 to 14 ensure that, during a simulation run, lpfg alternates between the two groups. The Gillespie group has two production rules: one rule to calculate the propensity of diffusion from the left to the right (line 27) and from the right to the left (line 28), and another rule to calculate the propensity of a molecule decaying (line 31). Since Hammel and Prusinkiewicz did not consider the first and last modules in their deterministic solution to the decay-diffusion process (see L-system (6.5)), the rules for decay and diffusion do not apply to the first and last

modules in the string. This is done by not placing a `Wall` module at the beginning and end of the string. The purpose is to prevent all of the molecules from decaying in the entire cellular structure. Finally, the second group in the program ensures that in the case of a diffusion event one molecule is subtracted from the appropriate cell and is added to its neighbour. The appropriate cell is found by checking for the direction of diffusion through the `Wall` module, i.e, by checking the value of `dirl` and `dirr`. If the direction of diffusion in `dirl` is to the left, one molecule is added to the cell, otherwise one is subtracted. If the direction in `dirr` is to the right, one molecule is added to the cell, otherwise one is subtracted.

The benefit of using L-systems and Gillespie groups to model the decay-diffusion process stochastically should be clear from the simplicity of Program 3. Without Gillespie groups this program would have to compute the sum of all the propensities from all the cells, and find the cell where the next reaction takes place. Both of these steps in Gillespie's algorithm are done internally by lpfg in Gillespie groups. With this benefit in mind, we can now compare the deterministic and stochastic solutions to this process.

6.2.3 Results for the Deterministic & Stochastic Models of Decay & Diffusion

Visual representations of the deterministic and stochastic solutions to the decay-diffusion process over time are given in Figure 6.2. In each of the four images, a cell is visually interpreted as an unit-length line segment where the cell's colour depends on the concentration a of the diffusing substance in the cell. To obtain the concentration of the number of molecules A in the case of the stochastic solution,

the following equation is used: $a = A/V$, where V is the volume. In the first three images there are 32 cells per row and in the fourth image there are 64 cells per row, where each row shows the state of the cells at $\Delta t = 0.1$ time intervals.

The first image on the left was generated using ODEs while the other three images were generated stochastically using the lpfg program from Section 6.2.2. In all of the solutions the cells were initialized as follows: for the first cell, $a = 1$, for the last cell, $a = 1$, and for every other cell, $a = 0$. In the third solution the cells were initialized with 10 times as many molecules, which is the reason why the second stochastically generated image appears less noisy than the first. All of the solutions use the same values for the decay and diffusion constants: $\mu = 0.01$ and $D_a = 5$.

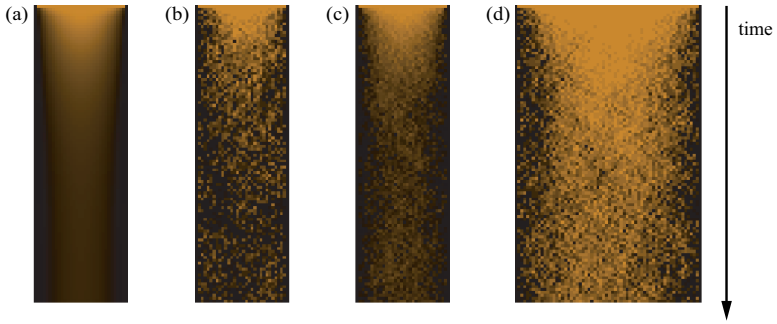


Figure 6.2: Visual representation of the deterministic and stochastic solutions to the decay-diffusion process: (a) deterministic solution produced by L-system (6.5), (b) stochastic solution produced by Program 4 with $V = 10$, (c) stochastic solution with $V = 100$, and (d) stochastic solution with twice as many cells as in (b).

By comparing the patterns generated in Figure 6.2, we can see that the stochastic solution to the decay-diffusion process (in a one dimensional medium) is similar to the deterministic solution. The advantage of using Gillespie's method is that it captures the inherent noise of the process, which is exemplified by the pattern labelled (b).

Notice that this noisy pattern is generated under the same parameters and initial boundary conditions as the solution to the deterministic differential equations, and only when we increase the number of molecules do we see a pattern that is less noisy. The disadvantage of using Gillespie's method is that a large amount of computing time is required to generate these patterns. While the deterministic pattern was generated in seconds, it took about 2 minutes to generate the first stochastic pattern, about 15 minutes to generate the second pattern, and about 4 minutes to generate the third. An investigation into how the number of cells and the rate of diffusion affects the simulation time results in a simple direct relationship between them. For example, the simulation time doubled between the pattern generated in Figure 6.2a and in Figure 6.2b. In the case of increasing the diffusion rate, the simulation time increases similarly, e.g, while the stochastic model with 32 cells and a diffusion rate of 5 took 2 minutes, a stochastic model with 32 cells and a diffusion rate of 50 took 18.3 minutes. Although it is important to explore ways of decreasing the computing time required to generate these patterns, for now let us complete stochastically modelling the reaction-diffusion equations.

6.3 Sea Shell Patterns

Let us review how sea shell patterns may be generated using the deterministic ODE approach. The pigmentation patterns generated by Fowler, Meinhardt and Prusinkiewicz [14] were based on an activator-substrate model, using Equations (6.2), similarly to Hammel and Prusinkiewicz [25]. The latter, however, used L-systems to generate these patterns because of the clear advantage of using the formalism for

modelling linear structures, and because of the simple expansion of their L-system rule for the decay-diffusion process to the activator-substrate process. Using their approach, deterministic sea shell patterns can be generated by the following L-system rule:

$$\begin{aligned} &Cell(a_l, s_l) < Cell(a, s) > Cell(a_r, s_r) \rightarrow \\ &Cell\left(a + \left(\rho s\left(\frac{a^2}{1+\kappa a^2} + \rho_0\right) - \mu a + D_a \frac{a_l - 2a + a_r}{(\Delta x)^2}\right)\Delta t,\right. \\ &\qquad \left. s + \left(\sigma - \rho s\left(\frac{a^2}{1+\kappa a^2} + \rho_0\right) - \nu s + D_s \frac{s_l - 2s + s_r}{(\Delta x)^2}\right)\Delta t\right). \end{aligned} \tag{6.11}$$

The meaning of the activator-substrate parameters used in this rule was given in Section 6.1. These parameters control the type of sea shell pattern that will emerge by solving the activator-substrate equations. For instance, a striped pattern that is stable in time and periodic in space can be generated by controlling the decay and diffusion rates of the substrate [14, page 384]. It is important, therefore, to find the relationship between the deterministic and stochastic values of these parameters. We already know how the parameters for decay and diffusion are related, and the relationship we have for V_{max} and K_m (from quasi-steady state enzyme kinetics) will help us with the parameters involved in the autocatalytic process. Let us start by finding the values of V_{max} and K_m from the activator-substrate equations (6.2). By comparing the form of the following expressions,

$$\frac{V_{max} a^2}{K_m + a^2} = \rho s\left(\frac{a^2}{1+\kappa a^2}\right), \tag{6.12}$$

we see they become identical when $V_{max} = \rho s/\kappa$ and $K_m = 1/\kappa$.

According to Section 5.3.2 on the stochastic model of the quasi-steady state Hill reaction, we find the propensity of the autocatalytic process in the activator-substrate

equations to be

$$\left(\frac{\rho V}{\kappa}\right)\left(\frac{S}{V}\right)\left(\frac{A^2}{\frac{V^2}{\kappa}+A^2}\right) = \rho S\left(\frac{A^2}{V^2+\kappa A^2}\right), \tag{6.13}$$

where the number of molecules, A and S, are used instead of the concentrations, a and s. The relationship between all the values in the activator-substrate equations are summarized in Table 6.1.

parameter	*deterministic value*	*stochastic value*
decay	μ, ν	μ, ν
diffusion	D_a, D_s	D_a, D_s
autocatalytic process	$\rho s\left(\frac{a^2}{1+\kappa a^2}\right)$	$\rho S\left(\frac{A^2}{V^2+\kappa A^2}\right)$
substrate production	σ	σV
activator production	ρ_0	$\rho_0 V$

Table 6.1: The deterministic and stochastic values of the parameters in the activator-substrate model

By combining the stochastic models of the decay-diffusion process and the autocatalytic process, we can build a stochastic activator-substrate model and use it to produce some sea shell patterns seen in nature.

6.3.1 Stochastic Activator-Substrate

The stochastic Petri net for the activator-substrate process is given in Figure 6.3. It shows all of the reactions that may occur within a single cell and the diffusion of the two substances to the left and right neighbours of the cell (recall we are modelling within a one-dimensional medium). The transitions that represent the autocatalytic process are drawn as boxes outlined in red, those that represent decay are boxes outlined in green, and those that represent diffusion are boxes outlined in blue. This

stochastic Petri net is obviously a combination of the one for the decay-diffusion process and for the Hill reaction, which suggests that the two lpfg programs can also be combined. Program 5 gives the lpfg code for stochastically simulating the activator-substrate process.

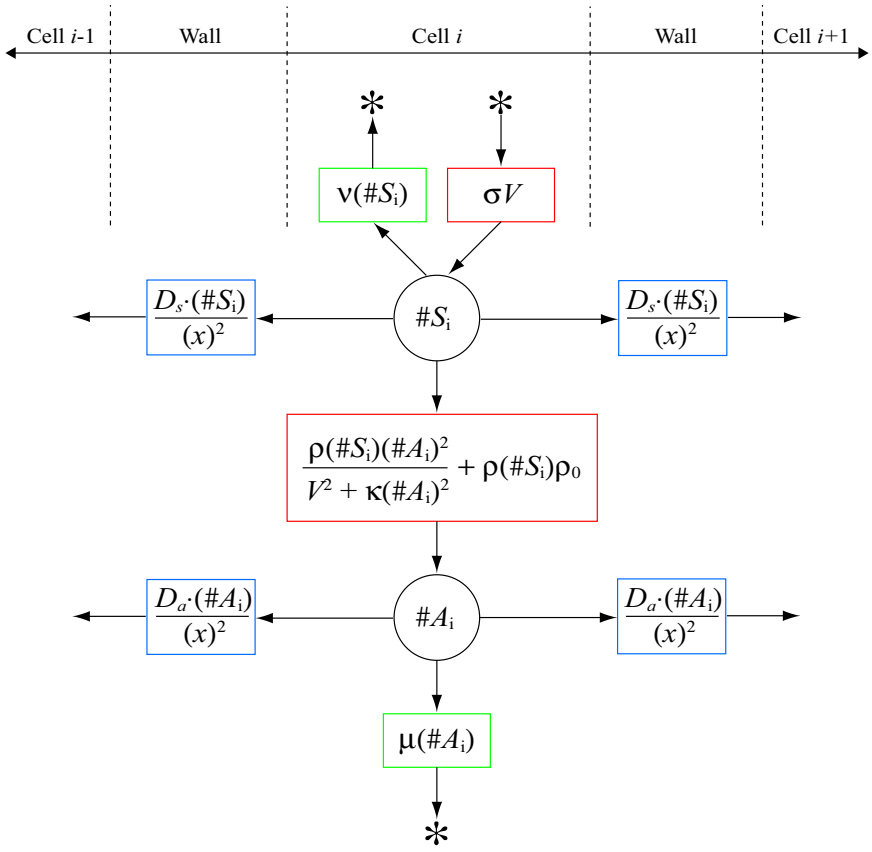


Figure 6.3: Stochastic Petri-net for the activator-substrate process.

Program 5: A stochastic model of the activator-substrate process implemented in lpfg using Gillespie groups

```
1  #include <lpfgall.h>
2
3  const float VOLUME = 100.0; /* volume of a cell */
```

```
4  const float D_A = 0.004;    /* diffusion rate of A */
5  const float D_S = 0.1;      /* diffusion rate of S */
6  const float MU = 0.1;       /* decay rate of A */
7  const float NU = 0.01;      /* decay rate of S */
8  const float SIGMA = 0.012;  /* production rate of S */
9  const float RHO = 0.1;      /* coefficient of proportionality */
10 const float KAPPA = 1.0;    /* activator saturation level */
11 const float RHO_0 = 0.005;  /* production rate of A */
12
13 int nextGroup = 1;
14 StartEach:{UseGroup(nextGroup);}
15 EndEach:{if (++nextGroup > 2) nextGroup = 1;}
16
17 derivation length: 1;
18
19 /* a cell with number of activator,substrate molecules */
20 module Cell(int,int);
21 /* a wall with diffusion direction of A,S */
22 module Wall(int,int);
23
24 axiom: /* Create cells separated by walls */
25
26 ggroup 1:
27 /* reactions for the autocatalytic process and decay */
28 Wall(Al,Sl) < Cell(A,S) > Wall(Ar,Sr):
29 {
30   float A_prod = (RHO*S*(A*A)) / (VOLUME*VOLUME+KAPPA*A*A)
31                  + RHO*S*RHO_0;
32   propensity (A_prod) produce Cell(A+1, S-1);
33   propensity (SIGMA*VOLUME) produce Cell(A,S+1);
34   propensity (MU*A) produce Cell(A-1,S);
35   propensity (NU*A) produce Cell(A,S-1);
36 }
37 /* likelihood for diffusion of A and S */
38 Cell(Al,Sl) < Wall(A,S) > Cell(Ar,Sr):
39 {
40   propensity (D_A*Al) produce Wall(1,0);
41   propensity (D_A*Ar) produce Wall(-1,0);
42
43   propensity (D_S*Sl) produce Wall(0,1);
```

```
44    propensity (D_S*Sr) produce Wall(0,-1);
45  }
46
47  group 2:
48  /* diffuse a molecule depending on its type, A or S */
49  Wall(Al,Sl) < Cell(A,S) > Wall(Ar,Sr):
50  {produce Cell (A+Al-Ar,S+Sl-Sr);}
51  Wall(dir): {produce Wall(0,0);}
```

This program is very similar to Program 4 except that there are additional propensity-produce statements defined in the first rule to handle the reactions involved in the autocatalytic process. The propensities match the ones in the stochastic Petri net exactly. The only other minor change is that there are two species that can diffuse instead of just one, and that means the second group in the program must adjust the number of molecules according to which species diffuses. Now we can compare the deterministic and stochastic solutions to this process.

6.3.2 Results

Figure 6.4 shows a theoretical sea shell pattern generated using the deterministic approach (shown in (a)), which is a reproduction of the pattern from the paper *Modelling seashells* [14, page 31], and the stochastic approach (shown in (b), (c), and (d)) which was produced by running Program 5. It is generated in exactly the same way as the visual representation of the solution to the decay-diffusion process (see Figure 6.2). The only differences are that the autocatalytic process was included in this model, and that different colours are used to show the concentration of a in each cell. In the deterministic solution, the cells are initialized with $a = 1$ and $s = 1$ concentrations. The stochastic solutions are initialized with the same values

as the concentrations but multiplied by a constant factor, the cell's volume, to get the number of molecules in each cell. The volume is $V = 1$, $V = 10$, and $V = 100$ for (b), (c), and (d) respectively. In all four of the solutions, the parameters were assigned the following values: $\rho = 0.01$, $\rho_0 = 0.001$, $\mu = 0.01$, $D_a = 0.002$, $\sigma = 0.015$, $\nu = 0$, $D_s = 0.4$, $\kappa = 0$. The ρ parameter, however, was subject to random fluctuations of 2.5% of its average value in the deterministic approach, which is necessary to start the pattern formation process. It is important to notice that this is not necessary in the stochastic simulation approach, and the pattern emerges without any extra parameter tweaking.

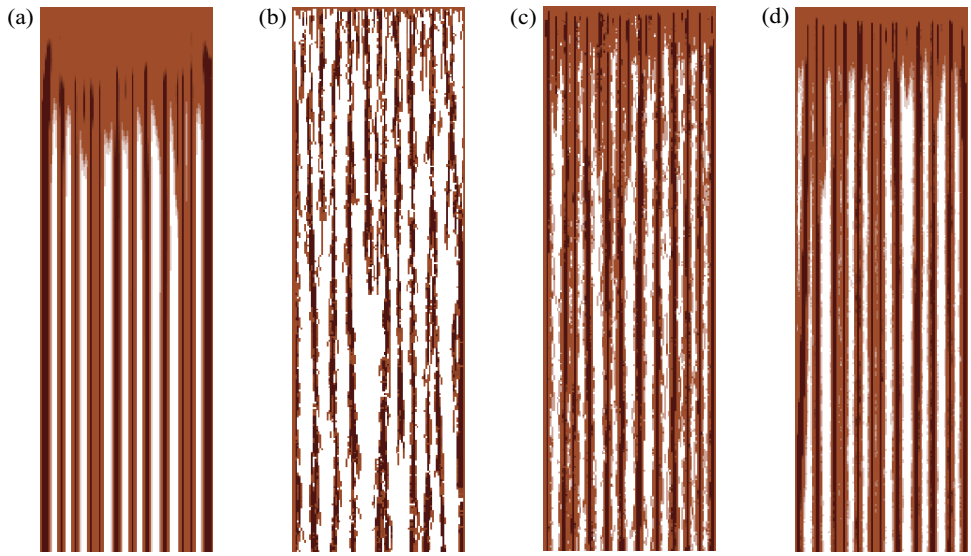


Figure 6.4: Several striped patterns generated using the deterministic and stochastic approaches to the activator-substrate model: (a) deterministic solution [14], (b) stochastic solution with the same initial conditions as the deterministic solution, (c) same as (b) but 10 times as many initial molecules, and (d) same as (b) but 100 times as many initial molecules.

Figure 6.5 shows that a realistic sea shells pattern, the one found on *Amoria un-*

dulata, can also be generated using the stochastic simulation approach. The deterministic solution shown in (a) is a reproduction from the paper *Modelling seashells* [14, page 32]. The stochastic solution was produced by running Program 5 with $V = 10$ for the pattern in (b), $V = 100$ for the pattern in (c), and $V = 1000$ for the pattern in (d). All four solutions used the following parameter values: $\rho = 0.1$, $\rho_0 = 0.005$, $\mu = 0.1$, $D_a = 0.004$, $\sigma_{min} = 0.02$, $\sigma_{max} = 0.032$, $\nu = 0$, $D_s = 0$, $\kappa = 1$. The σ parameter was modulated for each cell according to a sine function in order to generate lines of undulating shape [14, page 31]. That is, for each cell i, $\sigma = \sigma_{min} + sin(i)(\sigma_{max} - \sigma_{min})/2$. Furthermore, as in the striped pattern, the ρ parameter was subject to random fluctuations of 2.5% in the deterministic solution, but not in the stochastic solution.

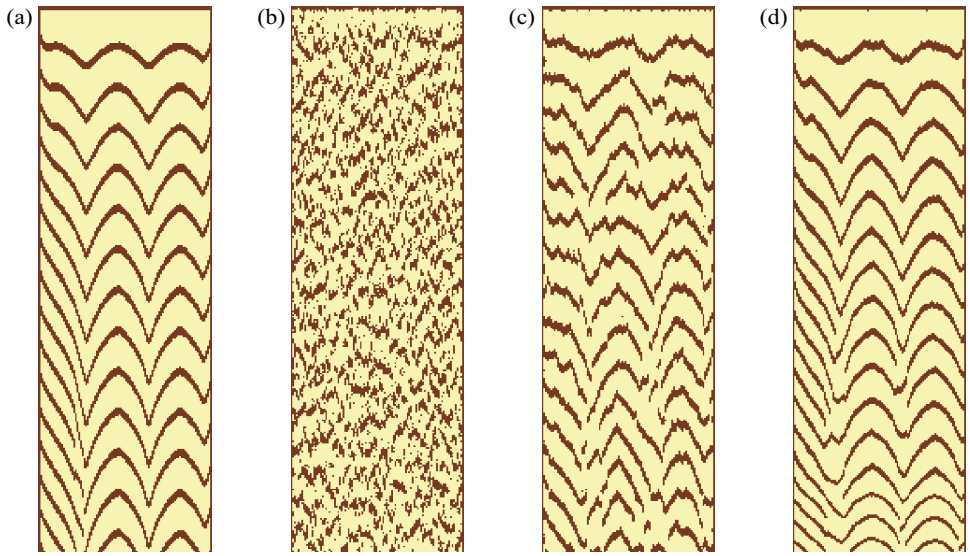


Figure 6.5: Sea shell pattern from *Amoria undulata* generated using the deterministic and stochastic approaches to the activator-substrate model: (a) deterministic solution [14], (b) stochastic solution initialized with about 10 molecules per cell, (c) about 100 molecules per cell, and (d) about 1000 molecules per cell.

The results for both the striped pattern and the *Amoria undulata* pattern show that the deterministic and stochastic solutions to the activator-substrate process are in agreement with each other. This is most easily observed in the case when the initial number of molecules is large (i.e. the volume of a cell, V, is high). In the other cases, the visualization was noisy and the pattern was not at all or less recognizable. This is to be expected, since large changes in a small number of molecules have a dramatic effect on the concentration of activator in the cell. For example, in a chemical system with $V = 10$ a change of one molecule will have a greater effect on the concentration than a similar change in a system with $V = 100$ (i.e. 10% versus 1%). The same type of reasoning can be applied to the visualization. If ten colours are used to represent the concentration of a species in a system with $V = 10$, the result will appear more noisy than using ten colours to represent the same concentration in a system with $V = 100$.

There is still the disadvantage of the computational cost associated with the stochastic approach. It takes only a few seconds to generate the deterministic solution, but it takes much longer to generate the stochastic solution for both the striped pattern and Amoria pattern. Specifically, the stochastic solutions with a large volume take the most time to generate. The striped pattern in Figure 6.4(b) took 6 minutes, in (c) took 36 minutes and in (d) took 4.2 hours. The Amoria pattern in Figure 6.5(b) took 9 minutes, in (c) took 1.1 hours, and in (d) took 9.2 hours to generate. These timings compared to the ones for the decay-diffusion process are quite large. The reason is most likely due to the large number of cells (i.e. 100 compared to 32) and the additional reactions. Now, let us continue our examination of the stochastic approach to reaction-diffusion by considering a model of a growing linear structure.

6.4 Anabaena catenula

In the previous section, we demonstrated that the reaction-diffusion process can be stochastically simulated using L-systems by generating sea shell patterns using the activator-substrate equations in a one-dimensional medium. According to Hammel and Prusinkiewicz, "the integration of reaction-diffusion processes and L-systems leads to a wider class of models" [25, page 254], which are models of reaction-diffusion in a growing medium. To demonstrate this, they presented a deterministic differential equation model of the bacterium, *Anabaena catenula*.

Anabaena is a cyanobacterium that consists of two types of cells: photosynthetic vegetative cells and anaerobic nitrogen-fixing heterocysts [27]. The cells are organized into a spaced pattern with sequences of vegetative cells separating heterocysts. The rules for growth and division of cells were studied by Mitchison and Wilcox [45]. They found that a vegetative cell either divides into two asymmetrical vegetative cells or differentiates into a heterocyst, which does not divide. The heterocysts appear about midway between two older heterocysts [45] and with about ten vegetative cells in between [27] (giving a spaced pattern). Two mathematical models have been proposed to capture the mechanisms that are responsible for the formation of this pattern in Anabaena.

Baker and Herman [3, 4] proposed a threshold model of Anabaena where the heterocysts produce a nitrogenous compound that diffuses throughout the organism and is decayed in the vegetative cells. If the compound drops below a certain threshold in a vegetative cell, the cell differentiates into a heterocyst [25, page 254]. This model was also implemented by Lindenmayer and de Koster using L-systems

[35, 8] (see also [56, page 43]). The resulting pattern of cells produced using this model closely resembles the real (spaced-heterocyst) pattern observed in Anabaena. A problem arises, though, with even slight manipulation of parameter values because heterocysts may sometimes appear simultaneously and close to each other [25, page 254]. For this reason, Hammel and Prusinkiewicz improved the model of Anabaena by introducing *proheterocysts* (presumptive heterocysts [45, page 110]) into the model. The proheterocysts compete with one another until one of them becomes a heterocyst and then suppresses its neighbours from also becoming heterocysts. To express this competition between proheterocysts, Hammel and Prusinkiewicz used a reaction-diffusion based model. The resulting pattern of evenly spaced heterocysts produced by their model also resembles the pattern seen in Anabaena, but is less sensitive to parameter values than Baker and Herman's model. Neither model, however, considers how noise may affect the formation of this pattern. To address this question, let us examine the stochastic simulation approach to these two models of Anabaena.

6.4.1 ODE Threshold Model of Anabaena

Lindenmayer and de Koster's implementation of the Anabaena threshold model uses deterministic differential equations to describe the production of a nitrogenous compound in heterocysts, and the decay and diffusion of that compound in vegetative cells. A slightly modified version of their implementation [12] gives the production of the compound, c, in a heterocyst as,

$$\frac{dc}{dt} = r_c(c_{max} - c), \tag{6.14}$$

where r_c is the rate of production and c_{max} is the maximum concentration of the compound. The decay and diffusion of the compound in vegetative cells is given by the familiar equations,

$$\frac{\partial c}{\partial t} = -\mu c + D_c \frac{\partial^2 c}{\partial x^2}, \tag{6.15}$$

where μ is the decay rate, D_c is the diffusion rate, and x is the cell length. The growth of heterocysts and vegetative cells is also described by a differential equation. If s is the size of a cell, where we presume size is equivalent to length, and r_s is the growth rate of a cell, the size changes according to the following equation for vegetative cells:

$$\frac{ds}{dt} = r_s s, \tag{6.16}$$

and the following equation for heterocysts:

$$\frac{ds}{dt} = r_s(s_{max} - s). \tag{6.17}$$

Here s_{max} is the maximum size of a cell. A heterocyst will never reach this maximum size, but when a vegetative cell reaches this maximum threshold it will divide according to the following rule: if the polarity of the cell is positive (or right), the right daughter cell will be smaller than the left daughter cell and if the polarity is negative (or left), the left daughter cell will be smaller than the right one [45]. Figure 6.6 is a visual representation of this rule for four cell divisions, where the polarity of a cell is represented by a right arrow for positive and a left arrow for negative. Finally, a vegetative cell differentiates into a heterocyst when the concentration of the nitrogenous compound falls below a certain threshold (i.e. when $c < c_{min}$), and the cell is short/young enough (i.e. when $s < s_{long}$).

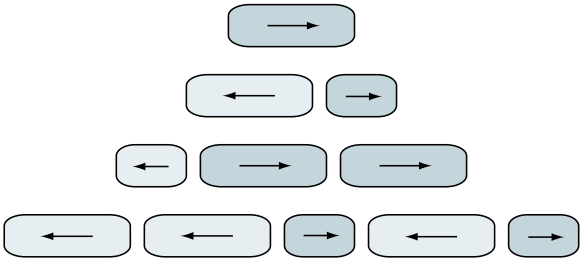


Figure 6.6: Anabaena cell division rules (redrawn from [56, page 5]): a right arrow represents positive polarity and a left arrow represents negative polarity.

A complete L-system for this Anabaena model, consisting of deterministic differential equations, is given in *The Algorithmic Beauty of Plants* [56, page 44], but here is a summary of the L-system's main parts. There are two types of modules that represent a vegetative cell or a heterocyst. The module, $Veg(c, s, p)$, represents vegetative cells and has three parameters: c is the concentration of the compound in the cell, s is the size of the cell, and p is the cell's polarity. The module, $Hetero(c, s)$, represents a heterocyst for which the parameters have the same meaning as for a vegetative cell. The L-system is initialized with one vegetative cell that is bounded by two heterocysts, using the following axiom:

$$Hetero(c(0), s(0))\ Veg(c(0), s(0), RIGHT)\ Hetero(c(0), s(0)). \tag{6.18}$$

Now, let us examine the L-system rules that govern the development of cells in this model. The rule for diffusion and decay of the nitrogenous compound in a vegetative cell and growth of the cell is given by,

$$\begin{aligned} &Veg(cl, sl, pl) < Veg(c, s, p) > Veg(cr, sr, pr) : c > c_{min} \wedge s > s_{long} \rightarrow \\ &Veg\left(c + \left(-\mu c + \frac{D_c(c_l - 2c + c_r)}{sw}\right)\Delta t, s + r_s s \Delta t, p\right), \end{aligned} \tag{6.19}$$

where w is the size of a wall separating two cells. This rule is applied under the condition that $c > c_{min}$ in the cell or that $s > s_{long}$, otherwise the rule for a vegetative cell differentiating into a heterocysts is applied. It is given by

$$Veg(c, s, p) : c \leq c_{min} \ \& \ s \leq s_{long} \rightarrow Hetero(c, s). \tag{6.20}$$

The rule for production of the compound in a heterocyst and growth of the heterocyst is given by

$$Hetero(c, s) \rightarrow Hetero(c + r_c(c_{max} - c)\Delta t, s + r_s(s_{max} - s)\Delta t). \tag{6.21}$$

The rule for vegetative cell division is a *decomposition* rule [55]. This type of rule is applied recursively to a module in the L-system's string, and can be used to divide a module into several modules. For this reason, an ending condition must be specified in each rule. The rules for cell division are given by

$$\begin{aligned} &Veg(c, s, p) : s = s_{max} \ \& \ p = LEFT \rightarrow \\ &Veg(c, k * s, LEFT)Veg(c, (1 - k) * s, RIGHT) \end{aligned} \tag{6.22}$$

$$\begin{aligned} &Veg(c, s, p) : s = s_{max} \ \& \ p = RIGHT \rightarrow \\ &Veg(c, k * s, RIGHT)Veg(c, (1 - k) * s, LEFT), \end{aligned} \tag{6.23}$$

where $k < 0.5$ is a constant that controls the lengths of the two daughter cells [25, page 255]. We can now consider corresponding rules using the stochastic simulation approach.

Stochastic Threshold Model of Anabaena

Of all the processes involved in the deterministic differential equation model of Anabaena, the only ones we have not encountered are for growth and division of cells.

To formulate these two processes using the stochastic simulation approach, however, is not difficult. The growth of a cell can be formulated similarly to decay, but instead of subtracting a molecule from the system one is added. This formulation of growth may be stretching Gillespie method a bit much because *cell size* is not equivalent to *number of molecules*. For that reason, we consider the chemical species S not as the size of the cell but rather a chemical that stimulates growth of the cell, which enables the use of Gillespie method to stochastically simulate growth. For division of vegetative cells there is only one change that is necessary to the deterministic differential equation formulation, the number of molecules in the mother cell must be divided between the two daughter cells. This division was not done in the deterministic Anabaena model because it dealt with concentrations, which do not change when a cell divides. The ideas of a growing and dividing cellular structure are captured in Figure 6.7. A cell that is growing according to the reaction, $S \longrightarrow 2S$ with rate r_s, will eventually reach a maximum value, S_{max}, and divide into two daughter cells. Each of the daughter cells will contain the same chemical reaction as the mother cell except that the number of molecules of S will be divided among them. In that case, the definition of a stochastic Petri-net has been extended to include the addition of new chemical reactions over time (i.e. a growing stochastic Petri-net). Let us now consider the stochastic formulation of all the reactions in the Anabaena threshold model.

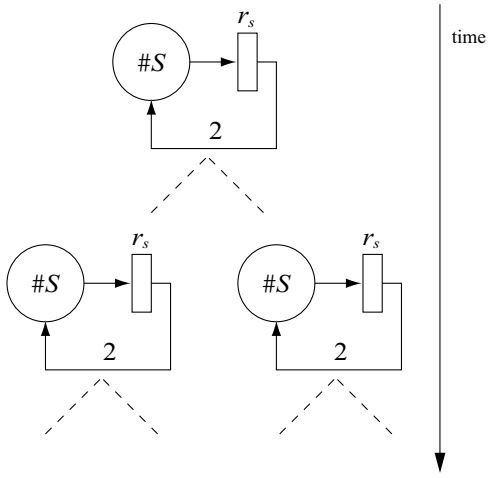


Figure 6.7: Example of a growing stochastic Petri-net: the growth is from top to bottom, where each reaction, $S \longrightarrow 2S$ with rate r_s, produces two new reactions of the same type.

In Figure 6.8 a stochastic Petri-net is given for the Anabaena threshold model, which shows the chemical reactions that may occur in vegetative cells or in heterocysts. The middle cell is a vegetative cell with four possible reactions: (1) decay of the compound, (2) diffusion from the cell to the left, (3) diffusion from the cell to the right, and (4) production of the chemical that stimulates growth. In the diffusion propensity calculation, the number of molecules of the species, S, must be divided by the volume of the cell because S does not represent the size of the cell. The species is a stimulate of growth in the cell and is given as the number of molecules. The vegetative cell is bounded by two heterocysts that have two possible reactions each. These reactions are for the production of the compound and for the stimulation of growth up to a maximum value (c_{max} and s_{max}, respectively). The stochastic Anabaena threshold model is given in lpfg Program 6.

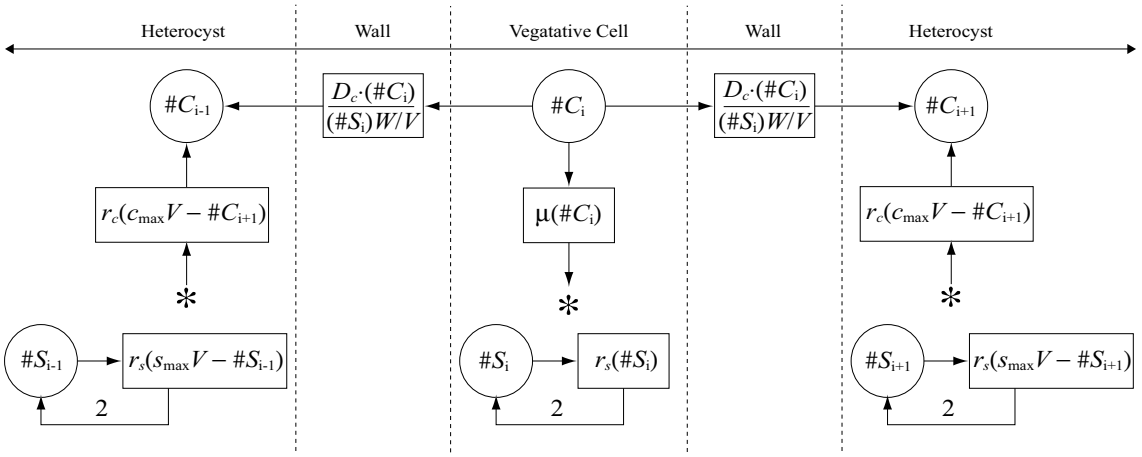


Figure 6.8: Stochastic Petri-net for the Anabaena threshold model.

Program 6: A stochastic threshold model of Anabaena implemented in lpfg using Gillespie groups

```
1  #include <lpfgall.h>
2
3  const float VOLUME = 10.0; /* volume of a cell */
4  const int RIGHT = 1;       /* polarity of vegetative cell */
5  const int LEFT = -1;
6  const float K = 0.38196;   /* minimum cell length */
7  const float W = 1.0;       /* wall length */
8  const float R_S = 0.01;    /* growth rate */
9  const float S_MAX = 1.0;   /* maximum size of cell */
10 const float S_LONG = 1-K;  /* size before differentiation */
11 const float MU = 0.2;      /* decay rate of C */
12 const float D_C = 1.0;     /* diffusion rate of C */
13 const float R_C = 10.0;    /* production of C in a heterocyst */
14 const float C_MIN = 0.05;  /* minimum concentration of C */
15 const float C_MAX = 0.64;  /* maximum concentration of C */
16
17 int nextGroup = 1;
18 StartEach:{UseGroup(nextGroup);}
19 EndEach:{if (++nextGroup > 3) nextGroup = 1;}
20 derivation length: 1;
21
```

```
22 module Veg (int, int, int); /* vegetative cell */
23 module Hetero (int, int);   /* heterocyst */
24 module Wall (int);          /* a wall between cells */
25
26 axiom: Hetero(C_MAX*VOLUME, K*VOLUME)
27        Veg(C_MAX*VOLUME, VOLUME, RIGHT)
28        Hetero(C_MAX*VOLUME, K*VOLUME);
29
30 ggroup 1:
31 /* reactions for decay and growth of vegetative cells */
32 Wall(dirl) < Veg(C,S,P) > Wall(dirr) :
33 {
34   propensity (MU*C) produce Veg(C-1,S,P);
35   propensity (R_S*S) produce Veg(C,S+1,P);
36 }
37 /* likelihood for diffusion between vegetative cells */
38 Veg(Cl,Sl,Pl) < Wall(dir) > Veg(Cr,Sr,Pr) :
39 {
40   propensity (D_C*VOLUME/(S*W)*Cl) produce Wall(1);
41   propensity (D_C*VOLUME/(S*W)*Cr) produce Wall(-1);
42 }
43 /* reaction for production of C and growth of heterocyst */
44 Hetero(C,S) :
45 {
46   propensity (R_C*(C_MAX*VOLUME-C)) produce Hetero(C+1,S);
47   propensity (R_S*(S_MAX-K*S)) produce Hetero(C,S+1);
48 }
49
50 group 2:
51 /* perform diffusion of C between vegetative cells */
52 Wall(dirl) < Veg(C,S,P) > Wall(dirr):{produce Veg(C+dirl-dirr);}
53 Wall(dir): {produce Wall(0);}
54
55 group 3:
56 Veg(C,S,P) :
57 {
58   /* check if this cell should differentiate into a heterocyst */
59   if (C <= C_MIN*VOLUME && S <= S_LONG*VOLUME)
60     produce Hetero(C,S);
61 }
```

```
62 decomposition:
63 Veg(C,S,P) :
64 {
65   /* check if this cell should divide */
66   if (K*S >= S_MAX*VOLUME)
67   {
68     /* divide according to cell's polarity */
69     if (P == RIGHT)
70       produce Veg(C*(1-K), S*(1-K), LEFT)
71               Veg(C-C*(1-K), S-S*(1-K), RIGHT);
72     else if (P == LEFT)
73       produce Veg(C*K, S*K, LEFT)
74               Veg(C-C*K, S-S*K, RIGHT);
75   }
76 }
```

This lpfg program is very similar to Program 4 for the decay-diffusion process, and the `propensity...produce...` statements are an exact implementation of the reactions described by the stochastic Petri-net in Figure 6.8. The novel part of this program is the use of Gillespie's method to model growth of cells, but it turns out this process is captured by a simple reaction (i.e. $S \rightarrow 2S$ with rate r_s). The third group where differentiation and division of vegetative cells is performed is done by using conditional statements. For example, if the number of molecules of the compound falls below a threshold, `C_MIN*VOLUME`, and the cell size is less than a differentiation threshold, `S_LONG*VOLUME`, the vegetative cell differentiates into a heterocyst. The thresholds are multiplied by `VOLUME` since they are specified in terms of concentrations. Let us now compare the deterministic and stochastic versions of this Anabaena model.

Results for the Anabaena Threshold Model

Figure 6.9 shows a visualization of the Anabaena threshold model for (a) the deterministic differential equation approach, and for (b) and (c) the stochastic simulation approach with $V = 100$ and $V = 1000$, respectively. The entire model is visualized as a row of cells where each cell is drawn as a rounded rectangle with the concentration of the compound, c, drawn as a line above the cell. The vegetative cells are coloured green, the heterocysts are coloured red, and the bars showing the concentration are coloured grey. Each simulation was started with a vegetative cell bounded by two heterocysts, and was stopped when the simulation time, t, reached 400 units. The parameters were assigned the following values: $k = 0.38196$, $r_s = 0.01$, $s_{max} = 1$, $\mu = 0.2$, $D_c = 1$, $r_c = 10$, $c_{min} = 0.05$, $c_{max} = 0.64$. The initial conditions were $c(0) = c_{max}$ and $s(0) = k$ for all three cells, and the polarity of the vegetative cell was set to positive (or right).

We can see from the visualization that the deterministic and stochastic approaches to the Anabaena threshold model are in agreement with each other. The stochastic simulation approach, however, shows that with a small number of molecules the heterocysts are not as evenly distributed among the vegetative cells as the deterministic model predicts (compare (a) to (b)). Only when the initial number of molecules is increased by 1,000 do the heterocysts appear more evenly distributed (compare (a) to (c)). The difference between the two stochastically generated patterns in (b) and (c) is most likely due to the model's use of a threshold for determining cell differentiation. Since a small number of molecules in a cell causes the noise to be large near the threshold, two vegetative cells may differentiate into heterocysts even though the cells are close to each other. This problem is not as

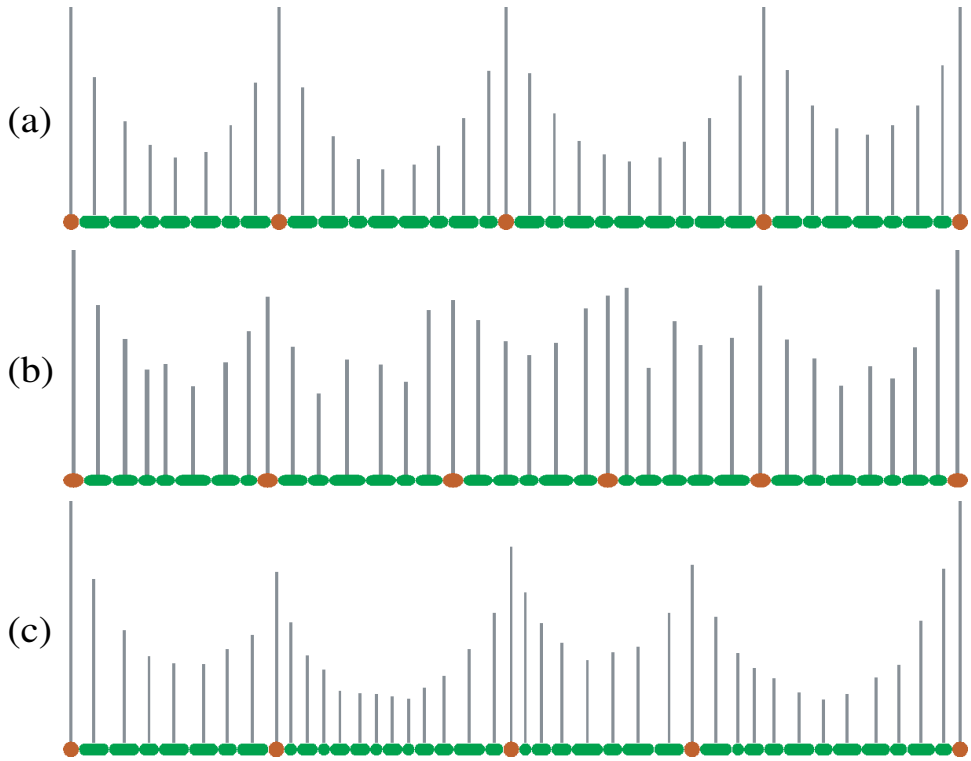


Figure 6.9: Visualization of the Anabaena threshold model: (a) the deterministic differential equation approach, (b) and (c) the stochastic simulation approach with $V = 100$ and $V = 1000$ respectively.

evident when there are a large number of molecules. To further explore the issue of inter-heterocyst spacing, a measure of the expected value and variance of the number of vegetative cells between heterocysts could be made. A similar measure of inter-heterocyst spacing was used by Yoon and Golden [72], where the frequency of the number of vegetative cells separating heterocysts was made for a 1,000-cell filament. This measurement, however, is not possible with the current stochastic simulation approach because of the large simulation time required to produce such a long filament. It took about 1.5 minutes to generate the pattern in (b) and about 20

minutes to generate the pattern in (c), where the number of cells was only around 40. Let us now consider an Anabaena model proposed by Hammel and Prusinkiewicz to reduce the sensitivity of the parameter values.

6.4.2 ODE Activator-Inhibitor Model of Anabaena

The deterministic differential equation model of Anabaena proposed by Hammel and Prusinkiewicz uses a reaction-diffusion process of activator-inhibitor type [39] to capture competition between proheterocysts, which later form heterocysts. This reaction-diffusion process is given by the following equations,

$$\begin{aligned} \frac{\partial a}{\partial t} &= \frac{\rho}{h}\left(\frac{a^2}{1+\kappa a^2} + a_0\right) - \mu a + D_a \frac{\partial^2 a}{\partial x^2} \\ \frac{\partial h}{\partial t} &= \rho\left(\frac{a^2}{1+\kappa a^2} + h_0\right) - \nu h + D_h \frac{\partial^2 h}{\partial x^2}, \end{aligned} \tag{6.24}$$

which describe the change in concentration of an activator species, a, and an inhibitor species, h, over time. The first term in both equations, the autocatalytic process, appears to be similar to the first term in the activator-substrate equations. This is misleading, however, because the production of a in the activator-substrate equations is proportional to s, but in the activator-inhibitor equations the production of a is inversely proportional to h.

In the implementation of the Anabaena activator-inhibitor model by Hammel and Prusinkiewicz, each cell solves the activator-inhibitor equations using the forward Euler method. The major difference between this Anabaena model and the threshold model is that only one module is used to represent both the vegetative cells and heterocysts instead of two. Consequently, the difference between heterocysts and vegetative cells becomes quantitative, rather than qualitative, in nature. This makes

it possible for nearby proheterocysts to compete with each other, preventing both of them from becoming heterocysts. For this reason the activator-inhibitor model is not as sensitive to parameter changes as the previous Anabaena model. Let us now use the stochastic simulation approach to better examine the activator-inhibitor model.

In the stochastic simulation approach to the activator-inhibitor process, the propensities of the reactions involved in the production of activator and inhibitor can be found similarly as for the activator and substrate production. The first step is to find the values of V_{max} and K_m from the activator-inhibitor equations (6.24). By comparing the form of the expression for the quasi-steady state Hill reaction and the activator-inhibitor equations, we have $V_{max} = \rho/\kappa$ and $K_m = 1/\kappa$. There is a problem, however, with the dependence of activator production on the inverse concentration of the inhibitor (i.e. the term ρ/h) because there is no chemical kinetic basis for this division (the law of mass action does not apply). The fundamental solution would be to find the chemical reactions that underlie this process and adjust the activator-inhibitor model accordingly. A simpler solution is to divide the propensity for activator production by the concentration of the inhibitor, formally treating the entire fraction ρ/h as a coefficient. Hence, the propensity of activator production is given by the expression

$$\left(\frac{\rho V}{\kappa}\right)\left(\frac{1}{H/V}\right)\left(\frac{A^2}{\frac{V^2}{\kappa}+A^2}+A_0\right)=\frac{\rho V^2}{H}\left(\frac{A^2}{V^2+\kappa A^2}+A_0\right), \tag{6.25}$$

where the numbers of molecules, H and A, are divided by the volume, V, to obtain concentrations. Similarly, the propensity for inhibitor production is given by the

expression

$$\left(\frac{\rho V}{\kappa}\right)\left(\frac{A^2}{\frac{V^2}{\kappa}+A^2}+H_0\right)=\rho V\left(\frac{A^2}{V^2+\kappa A^2}+H_0\right). \tag{6.26}$$

Now that the expressions to calculate the propensities for the reactions involved in the activator-inhibitor process are known, let us build a model of Anabaena based on this process.

Stochastic Activator-Inhibitor Model of Anabaena

Figure 6.10 gives the stochastic Petri net for one cell (a vegetative cell or a heterocyst) in a linear structure according to the activator-inhibitor model. The stochastic Petri net is similar to the one given for the Anabaena threshold model except for the additional reactions involved in the autocatalytic process. The transitions that represent the autocatalytic process are drawn as boxes outlined in red, those that represent decay are boxes outlined in green, and those that represent diffusion are boxes outlined in blue. Diffusion is only shown for the inhibitor because, usually, in real Anabaena the activator does not diffuse (it is to large a protein). The lpfg Program 7 models this process.

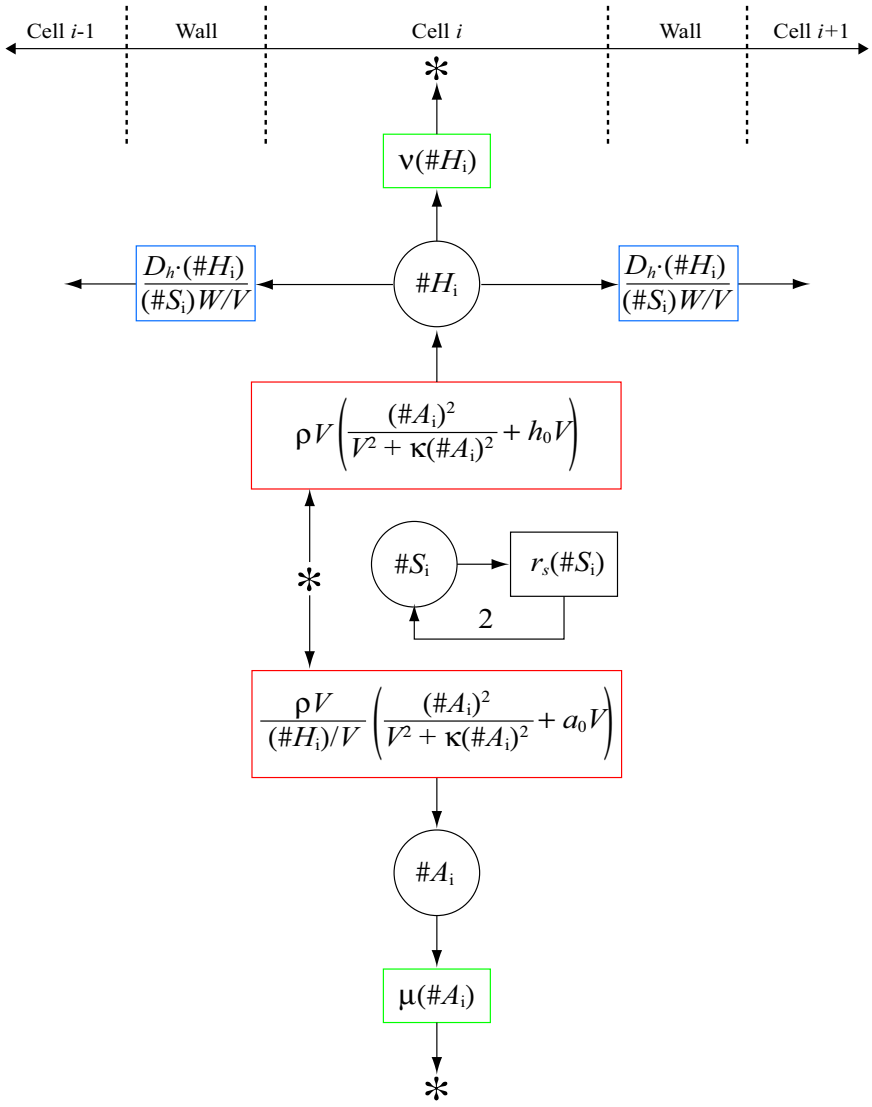


Figure 6.10: Stochastic Petri-net for the Anabaena activator-inhibitor model.

Program 7: A stochastic activator-inhibitor model of Anabaena implemented in lpfg using Gillespie groups

```
1  #include <lpfgall.h>
2
3  const float V = 100.0;   /* volume of a cell */
```

```
4  const int RIGHT = 1;     /* polarity of a cell */
5  const int LEFT = -1;
6  const float K = 0.45;    /* minimum cell length */
7  const float WS = 0.01;   /* wall size */
8  const float R_S = 0.002; /* growth rate */
9  const float S_MAX = 1.0; /* maximum size of cell */
10
11 /* activator-inhibitor constants */
12 const float RHO = 3.0;     /* coefficient of proportionality */
13 const float KAPPA = 0.001; /* activator saturation level */
14 const float A0 = 0.01;     /* production rate of A */
15 const float H0 = 1.0;      /* production rate of H */
16 const float MU = 0.1;      /* decay rate of A */
17 const float NU = 0.45;     /* decay rate of H */
18 const float D_H = 1.0;     /* diffusion rate of H */
19
20 /* concentration of A that triggers heterocyst formation */
21 const float A_MAX = 1.0;
22
23 int nextGroup = 1;
24 StartEach:{UseGroup(nextGroup);}
25 EndEach:{if (++nextGroup > 3) nextGroup = 1;}
26 derivation length: 1;
27
28 /* cell with parameters: activator, inhibitor, size, polarity */
29 module Cell (int, int, int, int);
30 /* wall with parameter: direction of diffusion */
31 module Wall (int);
32
33 axiom: Wall(0) Cell(0.1*V,1.0*V,V*(1-K),RIGHT)
34        Wall(0) Cell(0.1*V,1.0*V,V*K,RIGHT) Wall(0);
35
36 ggroup 1:
37 Wall(Al,Hl) < Cell(A,H,S,P) > Wall(Ar,Hr) :
38 {
39   float hill = A*A/(V*V + KAPPA*A*A);
40   /* check for a division by zero in activator-inhibitor eqns */
41   if (H > 0)
42     prod_A = (RHO*V*V/H) * hill + (RHO*V/H)*(A0*V);
43   else
```

```
44      prod_A = (RHO*V/0.001) * hill + (RHO/0.001)*(A0*V);
45    float prod_H = (RHO*V) * hill + H0*V;
46
47    /* autocatalytic reactions for production of A and H */
48    propensity (prod_A/S) produce Cell(A+1,H,S,P);
49    propensity (prod_H/S) produce Cell(A,H+1,S,P);
50
51    /* the decay of A and H */
52    propensity (MU*A) produce Cell(A-1,H,S,P);
53    propensity (NU*A) produce Cell(A,H-1,S,P);
54
55    /* the growth of a cell, upto a maximum value */
56    if (size < S_MAX*V)
57      propensity (R_S*S) produce Cell(A,H,S+1,P);
58  }
59  /* likelihood for diffusion between cells */
60  Cell(Al,Hl,Sl,Pl) < Wall(dir) > Cell(Ar,Hr,Sr,Pr) :
61  {
62    propensity (D_H*V*Hl/(Sl*WS)) produce Wall(1);
63    propensity (D_H*V*Hr/(Sr*WS)) produce Wall(-1);
64  }
65
66  group 2:
67  /* rules for diffusion of H between cells */
68  Wall(dirl) < Cell(A,H,S,P) > Wall(dirr):
69  {produce Cell(A,H+dirl-dirr,S,P);}
70  Wall(dir): {produce Wall(0);}
71
72  group 3:
73  decomposition:
74  Cell(A,H,S,P) :
75  {
76    /* check if this cell is a heterocyst */
77    if (S < S_MAX*V || A > A_MAX*V)
78      produce Cell(A,H,S,P)
79    else /* this cell should divide */
80    {
81      /* divide according to cell's polarity */
82      if (P == RIGHT)
83        produce Cell(A*(1-K), H*(1-K), S*(1-K), LEFT)
```

```
84                Wall(0)
85                Cell(A-A*(1-K), H-H*(1-K), S-S*(1-K), RIGHT);
86      else if (P == LEFT)
87        produce Cell(A*K, H*K, S*K, LEFT)
88                Wall(0)
89                Cell(A-A*K, H-H*K, S-S*K, RIGHT);
90    }
91  }
```

The three groups in this lpfg program for the Anabaena activator-inhibitor model have a similar purpose to the ones in Program 6 for the Anabaena threshold model. The first group, `ggroup 1`, is a Gillespie group that defines the propensities and the changes in the number of molecules involved in the activator-inhibitor process. The second group performs diffusion of the inhibitor molecule H between cells when it is necessary, and the last group performs cell division of vegetative cells (but not of heterocysts, as they do not divide). If the size of a cell has reached its maximum value, `S_MAX`, and the number of molecules of the activator species A is less than the heterocyst triggering value, `A_MAX*V`, the cell divides. In all other cases, the cell does not change.

Results for the Activator-Inhibitor Model of Anabaena

Figure 6.11 shows the resulting pattern which is obtained when the Anabaena activator-inhibitor model is simulated using lpfg until the simulation time, t, reaches 4000 units. The pattern labelled (a) was generated using the deterministic ODE approach (based on the model by Hammel and Prusinkiewicz [25]) and the patterns labelled (b) and (c) were generated using the stochastic simulation approach (Program 7). In all three patterns, vegetative cells are visualized as rounded rectangles and heterocysts are visualized as circles. The vertical bars above the rounded rectangles represent

the natural log of the concentration of the activator, and the vertical bars below represent the natural log of the concentration of the inhibitor. Similarly to the visualization of the Anabaena threshold model, the colour of a vegetative cell is green, the colour of a heterocyst is red, and the vertical bars are coloured grey.

To generate the patterns shown in Figure 6.11, the initial structure of the L-system for the Anabaena activator-inhibitor model was set up using two vegetative cells with walls on both sides of the cells. The cells were of different size to ensure that only one of them would become a heterocyst. The initial concentration of the activator was set to 0.1 and of the inhibitor to 1.0. For the stochastic simulation, the initial number of molecules for each species was set to the concentration multiplied by the cell's volume. Finally, the parameters for the activator-inhibitor equations were set to: $\rho = 3$, $\kappa = 0.001$, $a_0 = 0.01$, $h_0 = 1$, $\mu = 0.1$, $\nu = 0.45$, and $D_h = 0.01$.

The patterns produced by the deterministic and stochastic models in Figure 6.11 are similar, and once again the stochastic model which has a high number of initial molecules ($V = 100$) resembles the deterministic model closely. In the stochastic model with a lower volume, the heterocysts are fairly evenly spaced apart but one too many vegetative cells differentiate. This is most likely caused by the lack of activator in most of the cells (in the visualization, the bars above the vegetative cells show very little activator concentration), which suggest there is almost no competition between cells for differentiating into heterocysts. A measure of the expected value and variance of inter-heterocysts spacing would be useful, but once again the simulation time is too large. The amount of CPU time required to deterministically generate the heterocyst pattern was about 12 seconds, and to stochastically generate the two patterns was about 1 hour for $V = 10$ and 2 hours for $V = 100$.

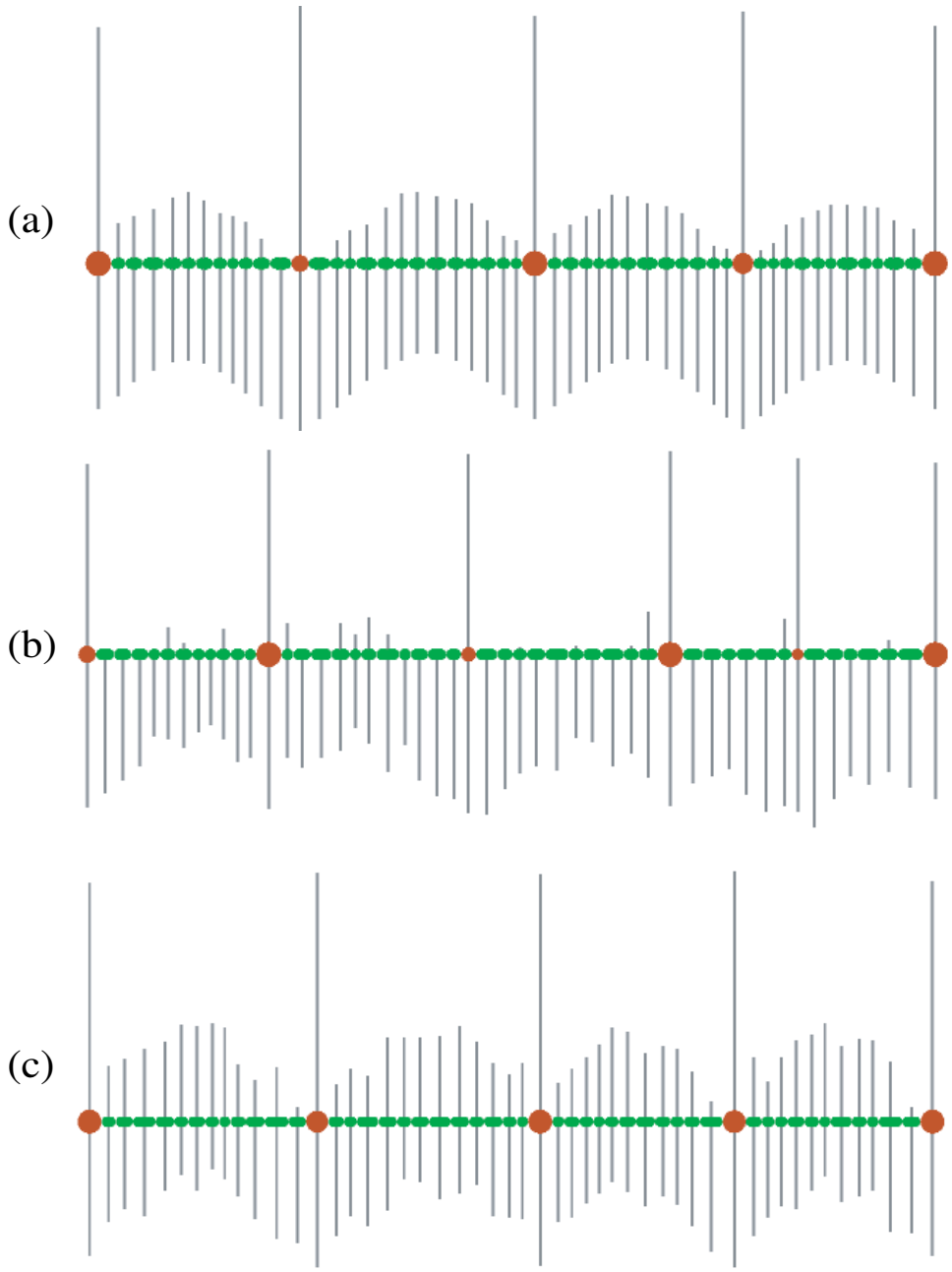


Figure 6.11: Visualization of the activator-inhibitor model of Anabaena: (a) the deterministic differential equation approach, (b) the stochastic simulation approach with $V = 10$, and (c) the stochastic simulation approach with $V = 100$.

Chapter 7

Conclusions

The main contribution of this work is the incorporation of Gillespie's algorithm for stochastically simulating chemical reactions into the formalism of L-systems. On the basis of theoretical considerations, the L+C modelling language and the simulation program lpfg were extended to include a new stochastic rewriting strategy based on Gillespie's algorithm. The resulting software provides a convenient means for simulating stochastic processes in both well-mixed systems and spatial (linear or branching) structures, and is particularly useful in the case of growing structures. Examples of spatially-explicit processes considered in this thesis include the formation of shell pigmentation patterns and heterocyst differentiation in growing filaments of blue-green bacterium Anabaena. To simulate these processes, we derived stochastic counterparts of the activator-substrate and activator-inhibitor reaction-diffusion models from the first principles, on the basis of their standard definition in terms of differential equations. The derivation of the autocatalytic component of these models was based on the assumption of quasi-steady-state enzyme kinetics.

The motivation for stochastic simulations is that they inherently capture stochastic phenomena (noise) that may occur in the modelled systems. In the thesis, this point was illustrated by reproducing Gillespie's comparison of deterministic and stochastic solutions to Lotka's chemical system [21]. The deterministic solution (obtained as a numerical solution to a pair of coupled differential equations) incorrectly predicts that if Lotka's system is in an equilibrium, it will remain in that state for-

ever. Also, the deterministic solution predicts that oscillations will be repeated with a constant period forever. In contrast, the stochastic model shows that the solution is oscillatory with a varying period. In the case of spatially explicit processes, stochastic simulations reveal the noise present in reaction-diffusion patterns when the numbers of molecules are small. When these numbers are increased, the patterns generated using Gillespie's and deterministic methods converge.

One objective of including Gillespie's stochastic rewriting strategy into the L+C modelling language and the related simulator lpfg was to simplify the task of specifying stochastic models. To illustrate the results, we compared two stochastic implementations of Lotka's system, one written in plain L+C and another written in L+C with the Gillespie extension. In the absence of language support for Gillespie's algorithm (Program 1), half of the model code was devoted to a generic specification of Gillespie's algorithm and had little to do with the Lotka's system itself. With Gillespie's method incorporated into L+C and lpfg, the specification of stochastic Lotka's system became much shorter (from 50 lines to 22 lines of code) and more readable (Program 2).

In general, L+C models using the Gillespie extension appear to be easy to write, modify, and experiment with. For example, the stochastic model of the activator-substrate process (Program 5) was obtained by extending a model of decay and diffusion (Program 4). This extension required the addition of only a few statements, which characterized the production of the activator and substrate.

A drawback of using stochastic simulations, as opposed to solving differential equations, is that large computation times are required. This is particularly noticeable in the case of spatially explicit models. Table 7.1 collects the CPU times

required to generate patterns presented in this thesis.

Pattern	**Model**	**CPU Time**
Figure 6.2,	a) deterministic, Cells = 32	2 sec
decay and	b) stochastic, $V = 10$, Cells = 32	89 sec
diffusion	c) stochastic, $V = 100$, Cells = 32	855 sec
	d) stochastic, $V = 10$, Cells = 64	209 sec
Figure 6.4,	a) deterministic	2 sec
striped	b) stochastic, $V = 1$	325 sec
sea shell	c) stochastic, $V = 10$	2154 sec
	d) stochastic, $V = 100$	15097 sec
Figure 6.5,	a) deterministic	2 sec
Amoria	b) stochastic, $V = 10$	545 sec
sea shell	c) stochastic, $V = 100$	4035 sec
	d) stochastic, $V = 1000$	33374 sec
Figure 6.9,	a) deterministic	5 sec
Anabaena	b) stochastic, $V = 100$	92 sec
threshold	c) stochastic, $V = 1000$	1148 sec
model		
Figure 6.4,	a) deterministic	12 sec
Anabaena	b) stochastic, $V = 10$	4330 sec
activator-inhibitor	c) stochastic, $V = 100$	6707 sec
model		

Table 7.1: The amount of CPU time required to generate the reaction-diffusion patterns presented in this thesis.

In the stochastic simulation of a reaction-diffusion system, the following factors have a significant impact on the computation time:

1. the total number of molecules,
2. the total number of reactions,
3. the rates of reactions,
4. the total number of cells.

Any of these factors may cause the sum of the propensities for some or all reactions to be large. The inter-reaction time τ is then small, leading to a large number of iterations required to reach a final simulation time T. In the thesis, this effect was demonstrated using reaction-diffusion as an example. A tenfold increase in the volume of a cell, and thus in the number of molecules involved, resulted in about a tenfold increase in the simulation time. Also, a twofold increase of the number of cells in the case of the decay-diffusion pattern of Figure 6.2, resulted in about a twofold increase of the simulation time. This investigation into the impact of the factors which have an effect on computation time shows a direct relationship between those factors and simulation time.

As stochastic simulations of reaction-diffusion models are often too slow for practical applications, the question of how to accelerate these simulations arises. An optimization of Gillespie's method, proposed by Gibson and Bruck in 2000 [17], is the *next reaction method* for performing the Monte Carlo step (i.e. for choosing which reaction takes place next and the time of that reaction). Gibson and Bruck introduced a dependency graph to recalculate only the propensities that change after the occurrence of a reaction, and an indexed priority queue to store the inter-reaction times for each reaction. By reusing the inter-reaction times for reactions that are not affected by the last reaction, they reduced the amount of random numbers required per simulation step from M (the number of reactions) to 1. The next reaction method changes the time complexity of Gillespie's algorithm from $O(M)$ to $O(1)$, and would thus be advantageous as an alternative to, or a replacement of, the direct Gillespie's algorithm as the stochastic simulation strategy in L+C.

While the next reaction method is as exact as Gillespie's stochastic simulation

algorithm, the computation time can be further improved if some loss in the accuracy of a solution is acceptable. For the case when the exact timings of reactions are not necessary, Gillespie proposed the τ-leap method [23]. This method advances simulation time by intervals of duration τ, and generates the number of reactions that will occur in each interval using a Poisson process. The difficulty in applying τ-leaping is the selection of the size of the time interval τ, which must be large enough to advance simulation efficiently, yet small enough to ensure that the propensities of reactions remain approximately constant in each interval. If this condition is not met, the accuracy of the solution may be compromised (e.g., oscillations may be missed in the solution). Several techniques for selecting τ exist, including less stable, but easier to implement, explicit schemes [23], and more stable implicit schemes [60]. Also, a technique that uses binomial random variables instead of Poisson random variables has been proposed to improve the efficiency of the τ-leap method [64].

If the acceleration achieved using the τ-leap method is still insufficient, a hybrid simulation method may provide a solution. In that case, Gillespie's method may be used to simulate slow reactions (with small propensities) while the Langevin-type stochastic differential equations may be used to simulate fast reactions (with large propensities). The feasibility of such a hybrid approach in the case of non-spatial models has been investigated by Puchalka and Kierzek [58] (for further review see [43]). The hybrid approach raises, however, some questions, such as: (1) what criteria should be used to classify a reaction as slow or fast, (2) how to maintain consistency between the parameters used in the stochastic and continuous components of a hybrid model, and (3) how to handle the situation when the propensity of a reaction significantly changes in the course of a simulation. In this latter case, is it possible

and useful to dynamically transfer a reaction between the slow and fast categories?

The extension of stochastic simulation algorithm to spatial structures, and the software environment presented in this thesis, are adequate for individual simulations of spatially explicit biochemical processes. With future optimizations and extensions of the algorithm, it may also become possible to execute long simulation runs, potentially involving hundreds or thousands of individual simulations. The resulting method and software may then become useful in the studies that require estimating mean values and variances of variables involved in the simulations. Such studies occur, for example, in the analysis of robustness of natural processes. One example would be a detailed exploration of the effect of noise on the robustness of Anabaena heterocyst spacing.

Glossary

chemical master equation: A first-order differential equation which describes the time-evolution of the grand probability function [page 33].

cumulative (probability) distribution function: A function, $F(x)$, which gives the probability that the random variable, X, is less than or equal to x.

expectation: Average value of a random variable, which is weighted by probabilities [page 26].

grand probability function: A function which gives the probability of a chemical system being in state $X = (X_1, X_2, \ldots, X_N)$ at time t [page 31].

law of mass action: The rate of a reaction is directly proportional to the product of the concentration of the reactants.

probability density function: A function, $f(x)$, which is such that $f(x)dx$ is the probability that a random variable, X, lies within $(x, x + dx)$.

propensity: The value assigned to the propensity function $a_j(X)$ (not necessarily between 0 and 1).

propensity function: A function, $a_j(X)$, which is such that $a_j(X)dt$ is the probability reaction R_j occurs within the infinitesimal time interval $[t, t + dt)$ [page 14].

random variable: An object X defined by: (a) a discrete or continuous set of possible values, and (b) a probability distribution over that set.

reaction rate constant, k_μ: Coefficient of proportionality in the law of mass action.

stochastic process: A function of two variables: time, t, and a random variable X, which gives a collection of values of X over t.

stochastic reaction constant, c_μ: The value such that $c_\mu dt$ gives the average probability that a specific combination of molecules will react in $(t, t + dt)$.

variance: The spread or dispersion of the possible values of a random variable around its expected value [page 27].

Bibliography

[1] M. Ander, P. Beltrao, B. Di Ventura, J. Ferkinghoff-Borg, M. Foglierini, A. Kaplan, C. Lemerle, I. Tomas-Oliverira, and L. Serrano. SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: analysis of simple networks. *Systems Biology*, 1(1):129–138, 2004.

[2] A. Arkin, J. Ross, and H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 4:1633–1648, 1998.

[3] R. Baker and G. Herman. CELIA - a cellular linear iterative array simulator. In *Proceedings of the Fourth Conference on Applications of Simulation*, pages 64–73, December 1970.

[4] R. Baker and G. Herman. Simulation of organisms using a developmental model, parts I and II. *International Journal of Bio-Medical Computing*, 3:201–215 and 251–267, 1972.

[5] A. Bartholomay. Stochastic models for chemical reactions. I. theory of the unimolecular reaction process. *Bull. Math. Biophys*, 20:175–190, 1958.

[6] O. Bastiansen. *The Law of Mass Action*. Det Norske Videnskaps-Akademi i Oslo Universitetsforlaget, Oslo, 1964.

[7] A. Bharycha-Reid. *Elements of the Theory of Markov Processes and Their Applications*. McGraw-Hill, Toronto, 1960.

[8] C. de Koster and A. Lindenmayer. Discrete and continuous models for heterocyst differentiation in growing filaments of blue-green bacteria. *Acta Biotheoretica*, 36:249–273, 1987.

[9] P. Dhar, T. Meng, S. Somani, L. Ye, K. Sakharkar, A. Krishnan, A. Ridwan, S. Wah, M. Chitre, and Z. Hao. Grid Cellware: the first grid-enabled tool for modelling and simulating cellular processes. *Bioinformatics*, 21(7):1284–1287, 2005.

[10] J. Elf, A. Donic, and M. Ehrenberg. Mesoscopic reaction-diffusion in intracellular signaling. *Proceedings of the SPIE first International Symposium on Fluctuations and Noise*, 2003.

[11] B. Ermentrout. *Simulating, Analyzing, and Animating Dynamical Systems: a guide to XPPAUT for researchers and students*. Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[12] P. Federl and P. Prusinkiewicz. Solving differential equations in developmental models of multicellular structures expressed using L-systems. In M. Bubak, G. van Albada, P. Sloot, and J. Dongarra, editors, *Proceedings of Computational Science. ICCS 2004 (Krakow, Poland, June 6-9, 2004), Part II, Lecture Notes in Computer Science 3037*, pages 65–72. Spinger-Verlag, Berlin, 2004.

[13] N. Federoff and W. Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 2002.

[14] D. Fowler, H. Meinhardt, and P. Prusinkiewicz. Modeling seashells. *Computer Graphics*, 26(2):379–387, 1992.

[15] C. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences*. Springer-Verlag, New York, third edition, 2004.

[16] J. Giavitto and O. Michel. Modeling the topological organization of cellular processes. *BioSystems*, 70:149–163, 2003.

[17] M. Gibson and J. Bruck. Efficient stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry*, 104:1876–1889, 2000.

[18] M. Gibson and E. Mjolsness. Modeling the activity of single genes. In J. Bower and H. Bolouri, editors, *Computional Methods for Molecular and Cellular Biology*, chapter 1. MIT Press, 2004.

[19] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972.

[20] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22:403–434, 1976.

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

[22] D. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.

[23] D. Gillespie. Approximate accelerated stochastim simulation of chemically reacting systems. *Journal of Chemical Physics*, 115:1716–1733, 2001.

[24] P. Goss and J. Pecaud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. *PNAS*, 95:6759–6755, 1998.

[25] M. Hammel and P. Prusinkiewicz. Visualization of developmental processes by extrusion in space-time. *Graphics Interface 96*, pages 246–258, 1996.

[26] J. Hanan. *Parameteric L-systems and their application to the modelling and visualization of plants*. PhD thesis, University of Regina, 1992.

[27] R. Haselkorn. How cyanobacteria count to 10. *Science*, 282:891–892, 1998.

[28] A. Hill. The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *J. Physiol. (London)*, 40:iv–vii, 1910.

[29] C. Huffaker. Experimental studies on predation: dispersion factors and predator-prey oscillations. *Hilgardia*, 27(14):343–383, 1958.

[30] E. Leigh Jr. The ecological role of volterra's equations. In *Some Mathematical Problems in Biology*, pages 1–61, 1968.

[31] R. Karwowski. *Improving the process of plant modeling: The L+C modeling language*. PhD thesis, University of Calgary, 2002.

[32] R. Karwowski and P. Prusinkiewicz. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science 86.2*, 2003.

[33] M. Kerszberg. Noise, delays, robustness, canalization and all that. *Current Opinion in Genetics and Development*, 14:440–445, 2004.

[34] A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.

[35] A. Lindenmayer. Adding continuous components to L-systems. In G. Rozenberg and A. Salomaa, editors, *L-Systems, Lecture Notes in Computer Science 15*, pages 53–68. Spinger-Verlag, Berlin, 1974.

[36] A. Lotka. Undamped oscillations derived from the law of mass action. *J. Amer. Chem. Soc.*, 42:1595–1599, 1920.

[37] H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Science*, 94:814–819, 1997.

[38] D. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4:413, 1967.

[39] H. Meinhardt. *Models of Biological Pattern Formation*. Academic Press, Toronto, 1982.

[40] H. Meinhardt. Models for positonal signalling, the threefold subdivision of segments and the pigmentations pattern of molluscs. *Journal of Embryology and Experimental Morphology*, 83:289–311, 1984.

[41] H. Meinhardt and M. Klinger. A model for pattern formation on the shells of molluscs. *Journal of Theoretical Biology*, 126:63–89, 1987.

[42] H. Meinhardt and M. Klinger. Pattern formation by coupled oscillations: The pigmentation patterns on the shells of molluscs. In *Lecture Notes in Biomathematics*, volume 71, pages 184–198, Berlin, 1987. Springer-Verlag.

[43] T. Meng, S. Somani, and P. Dhar. Modeling and simulation of biological systems with stochasticity. *In Silico Biology*, 4(3):293–309, 2004.

[44] L. Michaelis and M. Menten. Die kinetik der invertinwirkung. *Biochem. Z.*, 49:333–369, 1913.

[45] G. Mitchison and M. Wilcox. Rules governing cell division in anabaena. *Nature*, 239:110–111, 1972.

[46] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, volume 77, pages 541–580, 1989.

[47] J. Murray. *Mathematical Biology: I. An Introduction.* Springer, New York, third edition, 2002.

[48] J. Murray. *Mathematical Biology: II. Spatial Models and Biomedical Applications.* Springer, New York, third edition, 2002.

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

[50] Y. Pinchover and J. Rubinstein. *An Introduction to Partial Differential Equations.* Cambridge University Press, New York, first edition, 2005.

[51] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: the art of scientific computing.* Cambridge University Press, New York, second edition, 1992.

[52] P. Prusinkiewicz. Modeling and visualization of biological structures. In *Proceedings of Graphics Interface '93*, pages 128–137, 1993.

[53] P. Prusinkiewicz. Visual models of morphogenesis. *In Artificial Life*, 1(1/2):67–74, 1994.

[54] P. Prusinkiewicz. Art and science for life: Designing and growing virtual plants with L-systems. In C. Davidson and T. Fernandez, editors, *Nursery Crops: Development, Evaluation, Production and Use: Proceedings of the XXVI International Horticultural Congress. Acta Horticulturae*, volume 630, pages 15–28. 2004.

[55] P. Prusinkiewicz, J. Hanan, and R. Mech. An L-system-based plant modeling language. In M. Nagl, A. Schuerr, and M. Muench, editors, *Applications of graph transformations with industrial relevance. Proceedings of the International workshop AGTIVE'99, Kerkrade, The Netherlands, September 1999. Lecture Notes in Computer Science*, volume 1779, pages 395–410. Spinger, Berlin, 2000.

[56] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[57] P. Prusinkiewicz and W. Remphrey. Characterization of architectural tree models using L-systems and petri nets. In *M. Labrecque (Ed.): L'arbre - The Tree*, pages 177–186, 2000.

[58] J. Puchałka and A. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86:1357–1372, 2004.

[59] C. Rao and A. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *Journal of Chemical Physics*,

118(11):4999–5010, 2003.

[60] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie. Stiffness in stochastic chemically reacting systems: The implict tau-leap method. *Journal of Chemical Physics*, 119(24).

[61] A. Rényi. Betrachtung chemischer reaktionen mit hilfe der theorie der stochastischen prozesse (in Hungarian). *Magyar Tud. Akad. Alkalm. Mat. Int. Közl*, 2:93–101, 1953.

[62] C. Smith, P. Prusinkiewiz, and F. Samavati. Local specification of surface subdivision algorithms. In *Lecture Notes in Computer Science*, volume 3062, pages 313–327, Berlin, 2003. Spinger.

[63] A. Stundzia and C. Lumsden. Stochastic simulation of coupled reaction-diffusion processes. *Journal of Computational Physics*, 127:196–207, 1996.

[64] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *Journal of Chemical Physics*, 121(21).

[65] A. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237:37–72, 1952.

[66] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics*, 25(4):289–298, 1991.

[67] N. van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, New York, second edition, 1981.

[68] V. Volterra. Variazionie fluttuazioni del numero d'individui in specie animali conviventi. *Mem. Acad. Lincei*, 2:31–113, 1926. Variations and Fluctuations of a number of individuals in animal species living together. Translation by R. Chapman. In: Animal Ecology. pp. 409-448. McGraw Hill, New York, 1931.

[69] D. Wackerly, W. Mendenhall, and R. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Press, Toronto, fifth edition, 1996.

[70] A. Wagner. *Robustness and Evolvability in Living Systems*. Princeton University Press, New Jersey, 2005.

[71] A. Witkin and M. Kass. Reaction-diffusion textures. *Computer Graphics*, 25(4):299–308, 1991.

[72] H. Yoon and J. Golden. Heterocyst pattern formation controlled by a diffusible peptide. *Science*, 282:935–938, 1998.

[73] D. Young. A local activator-inhibitor model of vertebrate skin patterns. *Math. Biosciences*, 72:51–58, 1984.