Comparing stochastic discrete and deterministic continuum models of cell migration

Christian Yates

St. Catherine's College

University of Oxford

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Oxford

Trinity term 2011

Supervisors:

Professor Philip K. Maini, Centre for Mathematical Biology, Mathematical Institute, University of Oxford

Dr Ruth E. Baker, Centre for Mathematical Biology, Mathematical Institute, University of Oxford

Dr Radek Erban, Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, University of Oxford

Acknowledgements

I would like to thank my supervisors, Professor Philip Maini, Dr Ruth Baker and Dr Radek Erban, for all the help, support and patience they have shown me over the three (or so) years they have been supervising me. Their advice, both about this thesis and about (academic) life in general, has been invaluable. It has been a pleasure working with them on the various papers we have produced and I hope we will continue to work together in the future.

I would like to thank all those at the Doctoral Training Centre who have worked so hard behind the scenes to secure funding for the Systems Biology course on which I am one of the inaugural students. I would also like to show my appreciation for the enjoyable introduction to systems biology presented by the teaching staff of the DTC and, of course, to my DTC 2007 cohort who have helped to make my degree so thoroughly enjoyable. Through the DTC I, of course, extend my gratitude to the Engineering and Physical Sciences Research Council for funding this study.

Thanks must also go to all the staff and students of the Centre for Mathematical Biology. In particular, I owe a debt of gratitude to Aaron Smith for reading and commenting on my thesis in its entirety, whilst under the pressure of having to write his own. Thanks also to Gabriel Rosser and Trevor Wood for lending a mathematical ear to my problems and for frequently providing a welcome distraction with our 'extracurricular' cell-tracking, Biochemistry collaboration with David Wilkinson. Kevin Burrage has also been open in our discussions of stochastic simulation algorithms and a willing collaborator so I offer him my gratitude.

I am especially grateful to my family and friends for providing love, support and

encouragement along the way. In particular I would like to thank my parents, Tim and Mary, for proof-reading my entire thesis and many of my papers despite not knowing the language. I would especially like to thank my Dad for providing inspiration of a non-mathematical kind and for ensuring that I always have a sharp pencil at hand.

Finally, I would like to thank Caroline for always being there at the end of the day, through highs and lows.

Abstract

Multiscale mathematical modelling is one of the major driving forces behind the systems biology revolution. The inherently interdisciplinary nature of its study and the multiple spatial and temporal scales which characterise its dynamics make cell migration an ideal candidate for a systems biology approach. Due to its ease of analysis and its compatibility with the type of data available, phenomenological continuum modelling has long been the default framework adopted by the cell migration modelling community. However, in recent years, with increased computational power, complex, discrete, cell-level models, able to capture the detailed dynamics of experimental systems, have become more prevalent. These two modelling paradigms have complementary advantages and disadvantages. The challenge now is to combine these two seemingly disparate modelling regimes in order to exploit the benefits offered by each in a comprehensive, multiscale equivalence framework for modelling cell migration.

The main aim of this thesis is to begin with an on-lattice, individual-based model and derive a continuum, population-based model which is equivalent to it in certain limits. For simple models this is relatively easy to achieve: beginning with a one-dimensional, discrete model of cell migration on a regular lattice we derive a partial differential equation for the evolution of cell density on the same domain. We are also able to simply incorporate various signal sensing dynamics into our fledgling equivalence framework. However, as we begin to incorporate more complex model attributes such as cell proliferation/death, signalling dynamics and domain growth we find that deriving an equivalent continuum model requires some innovative mathematics. The same is true when considering a non-uniform domain discretisation in the one-dimensional model

and when determining appropriate domain discretisations in higher dimensions.

Higher-dimensional simulations of individual-based models bring with them their own computational challenges. Increased lattice sites in order to maintain spatial resolution and increased cell numbers in order to maintain consistent densities lead to dramatic reductions in simulation speeds. We consider a variety of methods to increase the efficiency of our simulations and derive novel acceleration techniques which can be applied to general reaction systems but are especially useful for our spatially extended cell migration algorithms.

The incorporation of domain growth in higher dimensions is the final hurdle we clear on our way to constructing a complex discrete-continuum modelling framework capable of representing signal-mediated cell migration on growing (possibly non-standard) domains in multiple dimensions.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AMD ..... Advanced Micro Devices.

ASSA ..... Accelerated Stochastic Simulation Algorithm.

BTLM ..... Binomial τ-Leap Method.

CA ..... Cellular Automaton.

cAMP ..... cyclic Adenosine MonoPhosphate.

CB ..... Confidence-Based.

CGP ..... Cao-Gillespie-Petzold.

CLE ..... Chemical Langevin Equation.

CMF ..... Cumulative Mass Function.

DLA ..... Diffusion Limited Aggregation.

DM ..... Direct Method.

ESSA ..... Exact Stochastic Simulation Algorithm.

EW ..... Edwards-Wilkinson.

FEM ..... Finite Element Method.

FPE ..... Fokker-Planck Equation.

FRM ..... First Reaction Method.

FVM ..... Finite Volume Method.

HDE ..... Histogram Distance Error.

IDE ..... Integro-Differential Equation.

KPZ ..... Kardar-Parisi-Zhang.

LHS ..... Left-Hand Side.

NAG ..... Numerical Analysis Group.

NRM ..... Next Reaction Method.

NSM ..... Next Sub-volume Method.

ODE ..... Ordinary Differential Equation.

ODM ..... Optimised Direct Method.

PDE ..... Partial Differential Equation.

PDF ..... Probability Density Function.

PGF ..... Probability Generating Function.

PMF ..... Probability Mass Function.

RDME ..... Reaction-Diffusion Master Equation.

RHS ..... Right-Hand Side.

SDE ..... Stochastic Differential Equation.

SDM ..... Sorting Direct Method.

SPDE ..... Stochastic Partial Differential Equation.

SSA ..... Stochastic Simulation Algorithm.

TLM ..... τ-Leap Method.

UC ..... Universality Class.

VEGF ..... Vascular Endothelial Growth Factor.

Chapter 1

Introduction

In humans, cell migration is an integral feature of many developmental and homeostatic mechanisms, including embryo formation (Keller, 2005), wound healing (Deng et al., 2006) and immune response (Madri and Graesser, 2000). In addition, cell migration is critical for the development and progression of pathogeneses such as cancer (Hanahan, 2000), vascular disease (e.g. atherosclerosis, Raines 2000) and chronic inflammatory diseases (e.g. arthritis, Edwards et al. 1999). Cell migration is vital, not only to animals like ourselves, but also to plants and single-celled organisms such as the amoeba *Dictyostelium discoideum* (*Dictyostelium*).

Often cells migrate towards, or more generally in response to, external signals which regulate and guide their motility. In chemotaxis, for example, these external signals can often take the form of chemical gradients in the medium which direct cell taxis. Examples of cell migration in response to external signals are extensive and have a wide range of biological connotations. These include the inflammation response of neutrophils (Bajaj et al., 1992), the directed positioning of neurons (Ward et al., 2003), the accurate dispersal of neural crest cells (Keynes and Cook, 1992) and the aggregation, in response to starvation, of *Dictyostelium* (Wessels et al., 1992).

Signalling molecules promoting cell migration have implications for the treatment of neurological birth defects (caused by deficiencies of cell migration) and the promotion of wound healing and immune responses. On the other hand, signals reducing cell motility may be beneficial to the treatment of autoimmune diseases such as rheumatoid arthritis

and in cancer therapy (Cai et al., 2006).

1.1 Why use mathematical modelling?

Although great strides have been made in furthering our understanding of the detailed mechanisms of cell migration there are areas which remain impenetrable to experimentalists. For example, carrying out comprehensive controlled experiments *in vivo* or mimicking a realistic environment *in vitro* is often extremely difficult or simply not possible (Staton et al., 2004). In addition, the large number of variables in many cell migration experiments, coupled with their complexity and inherent time constraints, means that carrying out fully factorial sets of experiments to test every alternative outcome is an impossibility. Even in those experiments which are short and simple enough to be implemented, the accurate interpretation of the results is often hampered by incomplete knowledge of all the variables in the system at the desired time points (Westermann et al., 2003).

In recent years, in part due to the advent of systems biology, increasing emphasis has been placed on a partnership between mathematical modelling and experimental exploration. Mathematical models were originally introduced to validate experimentally generated hypotheses. However, with the advances in molecular biology, genomics and gene manipulation, detailed and accurately parametrised mathematical models, capable of generating experimentally testable hypotheses or investigating questions beyond the reach of current experimental techniques, are being generated (Tomlin and Axelrod, 2007). For example, with a mathematical model it is possible to simulate multiple, parallel repeats of a system with the same or randomised initial conditions. This is especially useful for situations in which randomness has a significant effect and where viewing only a single instance of the evolution of a system can be inconclusive (Lee and Wolgemuth, 2011). Equally, repeats with slightly altered initial conditions are ideal for sensitive systems in which a small fluctuation in the initial state of the system can have a significant effect on the outcome of the experiment. Mathematical models also allow for the concise investigation of the outcome of an experiment since all the variables of the system can be accessed easily (Flaherty et al., 2007).

In the context of *in vivo* studies of T-cell migration, Westermann et al. (2003) note a particular difficulty in differentiating labelled, injected T-cells from host T-cells. Techniques for labelling injected cells are often hampered by side-effects which alter the behaviour of the cells or even contribute to their premature death. Even when cells can be correctly identified it is difficult to take into account possible cell proliferation and death which may alter the expected number of cells harvested (Hall, 1985). In addition, in order to gain an insight into the dynamics of the migration process, cells must be harvested at several time-points, which requires the use of an equivalent number of host organisms per time course, as well as those organisms required for the necessary repeats and controls (Westermann et al., 2003).

In contrast, in an individual-based, *in silico* simulation each cell may be labelled individually, cell death and proliferation incorporated accurately and the progression of cells (and their possible progeny) tracked for the duration of the simulation (Anderson and Quaranta, 2008). As many repeats and controls as necessary can be implemented at minimal additional expense (Flaherty et al., 2007).

1.2 Continuum *vs.* discrete modelling

An early mathematical model of cell migration in response to a signal profile which has been extensively studied over the last 40 years was presented by Keller and Segel (1970). This macroscopic approach considers a population-level characteristic: cell density, $u(\boldsymbol{x}, t)$, at a specific position, $\boldsymbol{x}$, and time, t. This approach leads to a continuous reaction-diffusion model where the diffusion of cells can be modelled by analogy to Fick's law (solute flux is directly proportional to the gradient of the solute concentration) and in which the reactions are viewed as functions of the population density and some external signal or control substance, $s(\boldsymbol{x}, t)$. Keller and Segel (1970) postulated that the cell density, $u(\boldsymbol{x}, t)$, and signal,

$s(\boldsymbol{x}, t)$, evolve on the domain, Ω, according to their, now classic, chemotaxis equations[1]

$$\frac{\partial u}{\partial t} = \nabla \cdot (D\nabla u - \chi u \nabla s), \tag{1.2.1}$$

$$\frac{\partial s}{\partial t} = \nabla^2 s + u - s, \qquad (\boldsymbol{x}, t) \in \Omega \times [0, \infty), \tag{1.2.2}$$

where D is the nondimensionalised diffusion coefficient and χ is known as the chemotactic sensitivity which can depend explicitly on u, s, $\boldsymbol{x}$ and t. Throughout this thesis, unless stated otherwise, we will assume that the signal profile remains constant and does not evolve according to equation (1.2.2).

Keller and Segel (1970) originally formulated their model to represent the crawling behaviour of the cellular slime mould *Dictyostelium*. Keller and Segel (1971) and later Brenner et al. (1998) were also able to use these classical chemotaxis equations for macroscopic modelling of the swimming patterns of the bacterium *Escherichia coli* (*E. coli*), despite significant differences in the microscopic (individual-based) behaviours of these two organisms (*Dictyostelium* and *E. coli*).

The original Keller-Segel model has been extended and altered to incorporate signal dynamics, signal- and density- dependent chemotactic sensitivity, non-linear diffusion, non-local sensing and non-linear signal and/or cell density kinetics (Hillen and Painter, 2009). The Keller-Segel model, or adaptations thereof, has been used to model vertebrate limb development, neural crest cell migration (Landman et al., 2003), the inflammatory response of leukocytes (Lauffenburger and Kennedy, 1983) and tumour angiogenesis (Chaplain, 2000).

Keller-Segel-type models are particularly successful when cell populations are large and can be approximated as a continuum. Inevitably, by making this assumption, the discrete nature of individual cells is lost[2] and, as a result, explicitly linking the model parameters and processes to the behaviour of individual cells is difficult. In addition, there is no rigorous way to incorporate into these models the huge amounts of cellular and molecular data currently being generated as a result of advances in our experimental understanding

[1] In their original formulation Keller and Segel allowed for more general cell and signal kinetics. This reduced formulation, however, will be sufficient for the purposes of this thesis.

[2] Hence two organisms (*Dictyostelium* and *E. coli*) which move in very different manners at the microscopic-level can be represented by the same macroscopic model.

of cell migration.

Individual-based models, however, treat each cell as a discrete entity which can be endowed with its own rules for movement and response to external signals taken directly from experimental observation. As such, individual-based models present a natural environment in which to incorporate the available experimental data. Although it is not always the case, individual-level models of cell migration are often probabilistic/stochastic as opposed to population-based models which are more frequently deterministic.

Individual-level models have also been used successfully, although less ubiquitously than population-level models, to represent cell migration. These models come in two basic varieties: off-lattice and on-lattice.

Off-lattice models themselves come in several forms: off-lattice velocity-jump processes have been used to model bacterial chemotaxis (Erban and Othmer, 2005; Othmer et al., 1988); and off-lattice cell-based models (such as the cell-centre model and the vertex-based model) have been successfully incorporated into a general purpose simulation package, CHASTE, aimed at multi-scale modelling of tissue homeostasis and carcinogenesis (Osborne et al., 2010).

Broadly speaking, on-lattice models can be classified as cellular automata (first introduced by von Neumann) in which each lattice site can take a finite number of states. On-lattice models also come in multiple forms: cellular Potts models have been used by used by Turner and Sherratt (2002) and Turner et al. (2004) in the context of cancer modelling and by Graner and Glazier (1992) in the context of cell sorting via differential adhesion; the on-lattice position-jump model has been used by Othmer and Stevens (1997) to model the aggregation of *Dictyostelium* and by Painter and Hillen (2002) to model a general chemotaxis process with volume filling constraints and a quorum sensing mechanism for cell density sensing. These on-lattice models have the advantages that they are conceptually easy to construct and are usually far less computationally intensive than their off-lattice, individual-based counterparts. However, rules for such models are often, necessarily, chosen in a phenomenological manner since biological cells can rarely be considered to move on a discrete lattice. On-lattice models often fail to engender the opportunity to impart rigorous physics-based rules on the model cells, an opportunity that is afforded naturally by their

off-lattice counterparts. Despite this, it may be possible to parametrise transition rates for an on-lattice position-jump model by simply overlaying a grid on experimental data of cell migration.

In general, individual stochastic models have some downfalls which, in part, have contributed to the relative scarcity of their use in comparison to continuum population-based models. Typically, simulation times scale with cell numbers which makes simulating large numbers of cells computationally intractable. It is only since the recent advent of widely available, powerful computing resources that even modest cell numbers have been simulated (Anderson and Quaranta, 2008). In addition, it is often difficult to link simulation results to specific model parameters due to the complex emergent nature of the outcomes of many individual-based systems (Anderson and Quaranta, 2008).

1.3 Discrete-continuum equivalence frameworks

Depending on the biological questions of relevance and available experimental data, either the deterministic-continuum or stochastic-discrete models (or both) may be appropriate. It is evident from Tables 1.1 and 1.2 that the two modelling frameworks have complementary strengths and weaknesses: discrete models are generally intuitive to formulate and, as such, can be more easily related to experimental data, whereas the formulation of continuum models often necessitates phenomenological assumptions. Continuum models for large cell populations can generally be solved quickly and their solutions may be expressed directly in terms of the model parameters, whereas individual-level simulations typically require multiple, slow, computationally intensive (especially so for large numbers of cells) repeats in order to gain insight into the population-level behaviour. Ideally we would combine an individual-level, discrete model and a population-level, continuum model and exploit their complementary advantages. We must be careful, however, when we choose models from the two regimes to describe the same phenomenon, that they agree with each other.

Indeed, it is possible to take certain individual-based models and derive, from them, continuum models for cell density that accurately capture cell-level detail. Bodnar and Velazquez (2005) and Murray et al. (2009) derive continuum versions of off-lattice individual-

Population-level, deterministic, continuum models

Advantages	Disadvantages
Fast to simulate	Cannot capture fine structural detail
Amenable to mathematical analysis	Unable to capture important discrete events
Consistent results	Ignore possible stochastic effects
Amenable to large cell numbers	Difficult to link directly to experimental data

Table 1.1: *A comparison of the pros and cons of population-level models. Quick solution times for population-level models come at the cost of individual-level detail. Deterministic models also risk ignoring potentially important stochastic effects, but only require a single numerical simulation (or mathematical analysis) to provide a consistent result due to their deterministic nature.*

Individual-level, stochastic, discrete models

Advantages	Disadvantages
Detailed structure and cellular identity	Slow to simulate with large cell numbers
Intuitive to formulate	Insusceptible to mathematical analysis
Incorporate stochastic effects	Different results each time
Link to experimental data	Require multiple simulations

Table 1.2: *A comparison of the pros and cons of individual-level models. The costs of detailed individual-level simulations include analytical intractability and computational complexity and the cost of building in stochasticity is the necessity for multiple repeated simulations to generate sufficiently reliable statistics.*

based models in one dimension. Bodnar and Velazquez (2005) assume that cell-cell interactions can be modelled by attractive and/or repulsive potential functions and derive different continuum equations depending on whether the interactions are assumed to be long-range (i.e. interacting with many, possibly non-neighbouring, cells) or short-range (i.e. interacting only with local cells). Murray et al. (2009) consider a model in which cells are linked by linear springs and the system relaxes to an equilibrium position. The continuum equations derived for cell density take the form of a non-linear diffusion equation. Murray et al. (2009) are also able to derive continuum models which correspond to individual-level models incorporating mitosis and incompressible cells. Extending these types of models into higher dimensions is inherently difficult and attempts are still at an early stage.

Considering on-lattice, individual-based models, it is often possible to write down a probability master equation describing the evolution of the probability that a system is in a certain state. Turner et al. (2004) use the probability master equation on a one-dimensional cellular Potts model incorporating cell-cell adhesion in order to derive a linear diffusion

equation whose diffusion coefficient can be linked explicitly to the model parameters. Alber et al. (2007) extend this work into two dimensions, simultaneously introducing a long-range chemotactic response. The equations they derive reduce to the classic Keller-Segel model if cell-cell interactions are ignored.

The individual-level models which allow for the most straightforward derivation of a continuum limit are on-lattice, position-jump-type models. Cells are assumed to be point particles (as opposed to having finite size as in the cellular Potts model) and are restricted to (local) movement to one of their neighbouring lattice sites with prescribed, possibly signal mediated, transition rates. Elementary examples of such discrete-continuum equivalence frameworks based on simple random walk models can be found in the influential texts of Berg (1993) and Lin and Segel (1988).

Othmer et al. (1988) introduce a more general off-lattice position-jump (or 'kangaroo' as they refer to it) model. They derive a continuum limit by considering an appropriate probability master equation and show, with a special choice of the jump kernel (so that only unbiased, discrete jumps of size Δ are allowed), that the continuum equation reduces to the diffusion equation.

Othmer and Stevens (1997) model the gliding and aggregation behaviour of *Dictyostelium* as a biased random walk on a square domain with periodic boundary conditions. Chemotaxis, in response to a diffusible chemical secreted by the amoebae themselves, is also included in the individual model. They derive a Keller-Segel-type equation describing the cell density and are able to link the functional form of the chemotactic coefficient to the model parameters and the specific mechanism in which the cells are allowed to sense the signal in the individual-based model.

Painter and Hillen (2002) incorporate volume filling (to mimic finite cell size) and quorum sensing (as a method of cell density detection) into the transition rates for cell movement in an on-lattice, individual-based model. Upon taking the continuum limit, in a similar manner to Othmer and Stevens (1997), they arrive at a system of Keller-Segel-like equations. The solutions of these equations avoid the unrealistic phenomenon of 'blow-up', in which cell density becomes infinite, which had been a problem for several phenomenologically-derived continuum models of chemotaxis (Hillen and Painter, 2001).

Grima and Newman (2004) derive a numerical scheme for advection-diffusion equations which only uses discrete approximations to the continuous functions at the current and nearest neighbour grid points (i.e. u_{i-1}, u_i and u_{i+1}) to approximate spatial derivatives. In so doing they are able to derive transition rates, of the Arrhenius form, for an individual-based position-jump model of the same process. They note that their discretisation could be used to determine the correct transition rates for the Keller-Segel equations, where the advection term in the cell density evolution equation is due to the chemoattractant concentration field.

Anderson and Chaplain (1998) arrive at a continuum-discrete modelling framework when modelling endothelial cell migration in the context of angiogenesis[3]. They arrive at this framework by considering a finite difference approximation to a set of three phenomenologically-derived, continuum 'angiogenesis equations'. The on-lattice discretisation allows them to derive transition rates for a probabilistic individual-based model which can be shown, in a manner similar to Othmer and Stevens (1997), to recapitulate the constituent continuum equations. Their approach serves to highlight the benefits of such equivalence frameworks in which cells can be represented either as a continuum or as discrete entities depending on the nature of the biological questions of interest: using the continuum model they are able to demonstrate the importance of haptotaxis (movement in response to a gradient of adhesion), rather than increased random motility (as suggested by Stokes and Lauffenburger, 1991), for the lateral movement of endothelial cells. This is in agreement with experimental results that demonstrate that lateral movement contributes to anastomosis (the recombination of divergent blood vessels) but that that there is very little random motility of endothelial cells (Paku and Paweletz, 1991; Paweletz and Knierim, 1989); using the individual-based model, Anderson and Chaplain (1998) are able to simulate the structure and morphology of capillary networks, including detailed examples of both capillary branching and anastomosis. The individual-level model permits a quantitative comparison of realistic capillary networks with *in vivo* data, whereas before, with the continuum model, only qualitative comparisons were possible.

[3]Angiogenesis is the directional growth and formation of new blood vessels, contributing to tumour formation and sustenance. The tumour induces the directed growth of blood vessels by biasing the migration of endothelial cells using signalling factors including vascular endothelial growth factor (VEGF).

The development of continuum-discrete modelling frameworks is clearly of great importance both from a mathematical and a biological viewpoint. Mathematically, it is important that the two model descriptions are consistent in order for us to exploit their complementary advantages in different parameter regimes. Biologically, these frameworks allow us to understand both the macroscopic and microscopic properties of a system. Detailed cell-level information can be incorporated into the individual-level description and wide-ranging population-level information into the continuum description in order to create models which are capable of making testable experimental predictions at both the micro- and macroscale.

It is for these reasons that, for the majority of this thesis, we focus on extending the discrete-continuum frameworks presented above. In particular we choose the on-lattice position-jump model as our discrete representation of cell migration since the equivalence framework is well established (Anderson and Chaplain, 1998; Othmer et al., 1988; Othmer and Stevens, 1997; Painter and Hillen, 2002). Despite the fact that this framework is well studied there remain multiple biologically motivated areas into which it can be extended. Some of these areas, on which we focus in this thesis, are detailed below.

1.3.1 Domain growth

Growth is essential to the development of all organisms. Mammalian embryos, for example, can change in size by several orders of magnitude during morphogenesis. At the same time as the embryo is growing the organisation of complex biological superstructures (such as the limbs) occurs and it is vital that these two processes cooperate with each other in order to achieve the correct results (Chevallier et al., 1977). Large numbers of cells migrate accurately at the same time as the organism is undergoing complex growth. Therefore, the integration of domain growth into models of cell migration is a vital component of any realistic description of development.

Using standard conservation of matter arguments, as in the case of the non-growing domain, in combination with the Reynolds transport theorem, we can alter the classic chemotaxis equation (1.2.1) to account for the time-dependent growing domain, $\Omega(t)$. We

arrive at the following partial differential equation (PDE) for the cell density, $u(\boldsymbol{x}, t)$:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u) = \nabla \cdot (D\nabla u - \chi u \nabla s), \qquad (\boldsymbol{x}, t) \in \Omega(t) \times [0, \infty), \tag{1.3.1}$$

where $\boldsymbol{v}(\boldsymbol{x}, t) = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$ is the velocity field generated by domain growth.

Domain growth has been successfully incorporated into models of cell migration and such models have been used to demonstrate peak insertion and splitting in fish skin patterning (Kondo and Asai, 1995; Painter et al., 1999). Adapted chemotaxis models with volume filling have also been used to show that peak insertion or splitting is not a necessary result of pattern formation on a growing domain: fixed wavenumber patterns have been demonstrated to be robust in the face of domain growth (Painter and Hillen, 2002). Despite these successes in a continuum setting it is not immediately clear how domain growth should be realistically incorporated into on-lattice, individual-level models of cell migration. Neither is it obvious how to couple such discrete models, incorporating domain growth, to equivalent continuum models.

1.3.2 Higher dimensions

Although the majority of the detailed derivations are carried out in a single dimension, continuum-discrete equivalence frameworks have been demonstrated to hold in higher dimensions (Othmer et al., 1988). Typically the detailed derivations and the effects of geometry in higher dimensions are not considered explicitly. Domains for individual-based models are typically square, tessellated with a square mesh. However, it may be of benefit to use a hexagonal domain discretisation, for example, since it is the least anisotropic of all the regular tessellations of the domain (Deroulers et al., 2009). In some cases it may even be necessary to consider an unstructured mesh in order to completely avoid anisotropy (Block et al., 2007) and to represent domain geometries accurately (Engblom et al., 2009; Kieri, 2011).

By formulating a general approach in two dimensions we investigate the effects of different domain tessellations and geometries on the population-level model derived. We also consider methods for incorporating domain growth in higher dimensions into our discrete-

continuum modelling framework.

1.3.3 Stochastic simulation

When simulating stochastic, individual-based models of cell migration we use stochastic simulation algorithms (SSAs), which were first introduced by Gillespie (1976) in two forms: the direct method (DM) and the first reaction method (FRM). The use of these algorithms (or refined versions thereof), originally designed to simulate simple chemical systems with statistical exactness, has now become widespread. The success of such SSAs in recent years has led to them being applied to much larger systems than was originally anticipated (including the systems presented here for modelling cell migration). For example, Arkin et al. (1998) used the SSA to simulate a stochastic version of the classic model of Shea and Ackers (1985) describing the processes which control the switch from lysogenic to lytic modes of growth in a virus, lambda phage, which is now a common experimental model system. The stochastic model contains 75 equations in 57 chemical species (Gibson and Bruck, 2000).

In spatially extended stochastic models, such as the ones with which we deal for the individual-level simulations in one dimension, species numbers can be of the order of hundreds and reaction channels of a similar order. Incorporating domain growth and extending these models into higher dimensions can only increase the computational complexity of our algorithms and, in turn, decrease their speed. It is, therefore, to our advantage to make the SSAs we are using as efficient as possible. Unfortunately, with the extension of our framework into higher dimensions, we may find that even the most efficient exact SSA is not sufficiently fast. In this case it may be necessary to consider sacrificing the accuracy of the SSA and opting for an accelerated stochastic simulation algorithm (ASSA). ASSAs were first introduced by Gillespie (2001) in the form of the 'τ-leaping' method (TLM). Such simulation algorithms have been demonstrated to increase the speed of stochastic simulations by orders of magnitude in a wide variety of situations (Gillespie, 2001; Gillespie and Petzold, 2003).

However, ASSAs are only an approximation to the exact simulation dynamics, and it is possible that the approximated species numbers can become negative, a non-physical

situation which can cause simulations to break down. This problem is potentially a serious challenge to the credibility of the ASSA, especially in the context of low copy numbers of one or more of the reactant species. Although the problem has since been corrected by a number of authors (Cao et al., 2005; Chatterjee et al., 2004; Tian and Burrage, 2004), it remains unclear that it has been done in the most efficient manner, especially for spatially extended systems of reactions akin to the ones we consider for our individual-based models.

1.4 Thesis outline

The broad remit of this thesis is to create a comprehensive discrete-continuum modelling framework for cell migration, incorporating a variety of biologically motivated features, such as those outlined above. In order to arrive at a general model of cell migration we must first abstract from the explicit detail of experimental systems and begin with a basic representation. Such a model can subsequently be improved by adding biologically motivated facets incrementally, in order to represent increasingly realistic scenarios. This is the task we undertake in this thesis.

In Chapter 2 we begin by replicating and generalising some of the results of Othmer and Stevens (1997) for cells moving in response to a fixed signalling profile sensed via a variety of different mechanisms. Into this basic, discrete-continuum equivalence framework we are able to incorporate time-varying signal profiles, cell proliferation and death, and density-dependent cell migration. In the second part of Chapter 2 we extend this framework by introducing a simple method for domain growth in a discrete setting. We are able to demonstrate the equivalence of this discrete model of cell migration on a growing domain and a continuum counterpart. The 'interval-splitting' domain growth method we incorporate is sufficiently versatile for us to induce linear, logistic or even density-dependent domain growth in the discrete model and still find a continuum equivalent. Our findings are corroborated with numerical simulations and comparisons of the discrete and continuum models.

Although this initial method for incorporating domain growth is versatile and robust we struggle to justify it biologically due to its overtly discrete nature. In Chapter 3 we establish

a method of domain growth which can, by tuning a parameter, be made arbitrarily similar to a continuum approach, whilst maintaining its discrete nature. Domain elements are allowed to grow incrementally and independently. As a result, the lattice upon which the cells move is allowed to become non-uniform. This causes us to consider more carefully the appropriate transition rates between intervals of different sizes and precisely what we mean by 'cell density' in the discrete version of the model. We do this initially for the non-growing domain, considering the incorporation of domain growth only when the correct transition rates have been determined. The non-uniform domain partition also engenders two different domain division paradigms, both of which we investigate when we reincorporate domain growth. Through a novel consideration of the probability master equation we are able to demonstrate the equivalence of our revised, discrete domain growth model and a continuum counterpart. We again corroborate our findings using numerical simulations and appropriate qualitative and quantitative comparisons.

As we add functionality to our discrete models and consider venturing into higher dimensions, the computational complexity of simulating these models increases. With this in mind, in Chapter 4, we review several methods of increasing the efficiency (and hence the speed) of the stochastic simulations of the individual-based models. However, as previously noted, the increased complexity of higher dimensional simulations may necessitate the sacrifice of statistical accuracy in order to restore reasonable simulation times. One way to achieve this goal is to use ASSAs. We review the current state-of-the-art method which produces accelerated simulations whilst avoiding negative species populations (Cao et al., 2005). We note, however, that the method of avoiding negative species populations is of less than optimal efficiency. We therefore introduce a confidence-based method which serves (as we demonstrate through numerical comparisons) to accelerate the majority of simulations (i.e. for general reaction systems) in comparison to the previous method, whilst continuing to avoid negative species populations. The acceleration is especially evident in our discrete simulations of cell migration.

Moving into higher dimensions our equivalence framework is presented with a number of additional hurdles to clear. In Chapter 5 we demonstrate that the continuum model we derive is independent of the method used to discretise a two-dimensional domain. We

also demonstrate that it is possible to compare the discrete and continuum models on non-square domain geometries. The majority of the chapter, however, is dedicated to implementing domain growth in two dimensions. By considering cluster aggregation theory we review a series of models with the potential for producing a radially growing domain through elemental division (the two-dimensional analogue of the domain growth method we introduce in Chapter 2). After rigorous investigation we eventually determine that no such model exists in the literature and are forced to invent our own. We demonstrate that our model produces isotropically growing domains via cell division and is therefore suitable as an underlying representation of domain growth. We complete this chapter by demonstrating excellent agreement between the densities of our discrete and continuum models on a radially growing domain[4].

We conclude this work in Chapter 6 with a short discussion. We summarise the major results in this thesis and place them in context. We also consider a variety of possible directions in which the work presented herein might be extended.

[4]Movies comparing the evolution of cell density for the individual-based and population-based models to accompany all the simulations mentioned in this thesis can be found at http://people.maths.ox.ac.uk/yatesc/Thesis_movies/Chapter_X, where X denotes the relevant chapter number. For long term storage all these movies can also been found at the youtube channel http://www.youtube.com/user/kityates.

Chapter 2

A comparison of an individual-level model and a population-level model in one dimension

Overview

Over the course of this thesis we will build a discrete-continuum infrastructure for modelling cell migration in a variety of potentially realistic scenarios. It is important, therefore, to lay a solid foundation which will allow us to create this infrastructure. In this chapter we undertake the groundwork for this foundation.

In Section 2.1 we consider a general microscopic random walker model describing the stochastic motion of individual cells in response to a signalling profile on a stationary[1] domain (as in Othmer and Stevens 1997). We introduce the reaction-diffusion master equation (RDME) formalism in order to demonstrate the equivalence, in certain limits, of this individual-based model to a PDE. That is to say that the discrete and continuum models can often be shown to share inherent properties which mean that they give quantitatively similar behaviour for the evolution of cell density across the domain. In Section 2.2 this equivalence is demonstrated qualitatively (by comparing plots of the densities of the two

[1]The phrases 'stationary', 'time-independent' and 'non-growing' will all be taken to be interchangeable descriptions of a domain whose size does not change throughout the course of the simulation.

models) and quantitatively (using an appropriate comparison metric) for some special cases of the general model.

We continue to add complexity to this quintessential model, in Section 2.3, in the form of time-varying signal profiles, which cells can sense and respond to, and by allowing cells to proliferate and die, as well as migrate in a manner which depends on the density of cells around them.

In Sections 2.4-2.5 we naïvely extend the random walker model to a lattice which grows via interval division and demonstrate its equivalence to a PDE model of the same process, derived from continuum considerations following Crampin (2000). In Section 2.6 we aim to make the growth processes that our RDME-PDE equivalence framework can deal with as general as possible. As special cases we explore linear and logistic domain growth in Section 2.7 and a variety of density-dependent growth mechanisms in Section 2.8. For each of the different features of our framework we draw quantitative parallels with continuum models and we corroborate our findings through numerical simulations of both models.

The approach taken in this chapter allows us, for each computationally intensive stochastic model, to derive a fast, easy-to-simulate, population-based PDE counterpart which has a plausible microscopic basis. The development of this stochastic-deterministic equivalence framework will become increasingly important as we extend our models into higher spatial dimensions (see Chapter 5) in which stochastic simulations can become prohibitively (computationally) expensive.

Movies corresponding to the simulations presented in this chapter can be found at http://people.maths.ox.ac.uk/yatesc/Thesis_movies/Chapter_2.

The majority of the work in this chapter appears in the paper: R.E. Baker, C.A. Yates and R. Erban. From microscopic to macroscopic descriptions of cell migration on growing domains. *Bull. Math. Biol.*, 72(3):719–762, 2010, and is reproduced with permission from Springer Science+Business Media.

2.1 A reaction-diffusion master equation approach for a general signal sensing model.

All the individual-cell-based stochastic models outlined in this chapter and in our paper (Baker et al., 2010) consider some fixed (unless otherwise stated) population of $\mathcal{N}$ motile cells restricted to move on a finite domain. The domain is not, for the moment, considered to be a biological tissue. Instead we consider a general framework which will allow us to subsequently incorporate biologically motivated features. Nevertheless we still refer to an individual section of the domain as a tissue element, interval or box throughout the rest of this work. The cells are allowed to move according to either random diffusion or in response to signals from the external environment. The external cues will influence the rate and direction in which the cells move. We assume, unless explicitly stated, that there is no cell division or death on the time-scale of interest and that cells may not enter or leave the domain. These conditions render the total population constant in time. The first model presented for this process is on a discrete, cell-level basis in which all the cells are modelled as identical individuals moving between intervals subject to specific rules.

Consider the stationary domain $x \in [0, L]$ discretised into k intervals each of length $\Delta x = L/k$, where L can be chosen to correspond to the particular experimental domain size[2]. In what follows we nondimensionalise the domain so that $x \in [0, 1]$ and $\Delta x = 1/k$ (see Fig. 2.1). A cell in the domain can move to one of its neighbouring intervals during each time-step, δt, but no further. The random variable denoting the number of cells in interval i is N_i, where the evolution of cell density over the whole domain can be denoted by the vector $\boldsymbol{N}(t) = [N_1(t), N_2(t), \ldots, N_i(t), \ldots, N_k(t)]$. The vector of random variables representing cell numbers in each of the intervals of the domain, $\boldsymbol{N}$, can be thought of as a density per unit length across the domain. This idea becomes useful when considering the deterministic population-level PDE model (Baker et al., 2010), but must be reconsidered carefully when employing non-uniform domain partitions, as in Chapter 3.

[2]For example, when modelling somite formation the length of the presomitic mesoderm over which cells migrate has been taken to be $\sim$ 1mm (Goldbeter et al., 2007). *In vitro* investigations of neutrophil migration in response to the chemoattractant interleukin 8 have been undertaken on domains of size 3.5 mm (Lin et al., 2004).

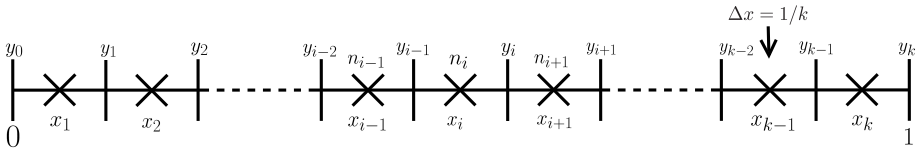


Figure 2.1: *The spatial grid upon which particles move. Each interval is of length* $\Delta x = 1/k$*, the edges of interval* i *are positioned at* $y_{i-1} = (i-1)\Delta x$ *and* $y_i = i\Delta x$ *and the centre at* $x_i = (i-1/2)\Delta x$*.* n_i *cells reside in interval* i *and the normalised domain runs from 0 to 1. Intervals* $2, \ldots, k-1$ *will be known as the 'interior' intervals and intervals* 1 *and* k *as the 'end' intervals.*

In this stochastic position-jump model we assume that the cells are restricted to move on the spatial grid with grid points defined to be at the centre of each interval, so that the i^{th} interval is $[(i-1)/k, i/k)$ and a cell in the i^{th} interval sits at $(2i-1)/2k$. We denote the transition rates (i.e. the rates per unit time of moving from one interval to another) for a cell to move left as T_i^- and right as T_i^+ so that the rate per unit time of moving left or right from interval i, $T_i^\pm = T_i^\pm(\boldsymbol{N}, \boldsymbol{s}, t)$ is some function of the cell numbers, $\boldsymbol{N}$, external signalling factors, $\boldsymbol{s}$, and time, t. To complete the formulation of the model we need to specify boundary and initial conditions. In order to model conservation of cells on the domain, we assume that at the left- and right-hand ends of the domain there exist transition rates, T_1^- and T_k^+, for the cells to move left and right, respectively, but that instead of the cell leaving the domain it remains stationary (or equivalently is reflected back from the boundary into the same interval) with the appropriate probability. T_1^- and T_k^+ are considered to be *status quo* probabilities rather than transition probabilities[3]. This is the analogue of 'reflecting' or zero flux boundary conditions in a continuum model. Other types of boundary conditions could be implemented, although, as noted by Erban and Chapman (2007), we must take care when implementing reactive boundary conditions. Two types of initial condition will be considered throughout this chapter. One such initial condition assumes that all $\mathcal{N}$ cells are released from the centre of the left-most interval at time $t = 0$. Another has the initial density of cells decaying exponentially across the domain. It will be made clear which of these initial conditions is to be used in each simulation at the appropriate juncture.

We would like to construct an RDME describing the evolution of cell density, $\boldsymbol{N}$ (assuming, at first, a constant signalling profile). Let $\Pr(\boldsymbol{n}, \boldsymbol{s}, t|\boldsymbol{n_0}, \boldsymbol{s_0})$ be the joint probability

[3]This method of implementing zero flux boundary conditions is equivalent to that where we set $T_1^- = T_k^+ = 0$, but is easier to implement computationally.

that $\boldsymbol{N} = \boldsymbol{n}$ at time, t, under constant external conditions, $\boldsymbol{s}$, with $\boldsymbol{n} = [n_1, n_2, \ldots, n_k]$ and $\boldsymbol{s} = [s_1, s_2, \ldots, s_k]$, predicated on the initial conditions $\boldsymbol{n_0}$ and $\boldsymbol{s_0}$ for the cells and the signalling molecules, respectively. For ease of notation we will denote this using the shorthand $\Pr(\boldsymbol{n}, \boldsymbol{s}, t)$. Define the operators $J_i^+ : \mathbb{R}^k \to \mathbb{R}^k$, for $i = 1, 2, \ldots, k-1$ and $J_i^- : \mathbb{R}^k \to \mathbb{R}^k$, for $i = 2, \ldots, k$, by

$$\begin{aligned} J_i^+ : [n_1, \ldots, n_i, \ldots, n_k] &\to [n_1, \ldots, n_{i-2}, n_{i-1}, n_i + 1, n_{i+1} - 1, n_{i+2} \ldots, n_k], \\ J_i^- : [n_1, \ldots, n_i, \ldots, n_k] &\to [n_1, \ldots, n_{i-2}, n_{i-1} - 1, n_i + 1, n_{i+1}, n_{i+2} \ldots, n_k]. \end{aligned}$$

The operator J_i^+ moves a cell from interval $i+1$ to interval i and J_i^- moves a cell from interval $i-1$ to interval i.

By considering all possible cell movements we can write down an equation for the probability distribution $\Pr(\boldsymbol{n}, \boldsymbol{s}, t + \delta t)$ a short time δt after the current time t, where δt is sufficiently small that the probability of more than one cell movement taking place in the interval $[t, t + \delta t)$ is $o(\delta t)$.

$$\begin{aligned} \Pr(\boldsymbol{n}, \boldsymbol{s}, t + \delta t) &= \sum_{i=1}^{k-1} T_i^+ (n_i + 1) \Pr(J_i^+ \boldsymbol{n}, \boldsymbol{s}, t) \delta t \\ &+ \sum_{i=2}^{k} T_i^- (n_i + 1) \Pr(J_i^- \boldsymbol{n}, \boldsymbol{s}, t) \delta t \\ &+ \Pr(\boldsymbol{n}, \boldsymbol{s}, t) - \sum_{i=1}^{k-1} T_i^+ n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \delta t - \sum_{i=2}^{k} T_i^- n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \delta t. \end{aligned} \tag{2.1.1}$$

The right-hand side (RHS) of the above equation summarises mathematically the probability of finding a cell distribution $\boldsymbol{N} = \boldsymbol{n}$ at time $t + \delta t$ by considering the probability of all possible single cell movements in time δt.

In equation (2.1.1), if we take $\Pr(\boldsymbol{n}, \boldsymbol{s}, t)$ to the left-hand side (LHS) , divide through by δt, and let $\delta t \to 0$, then in the limit we have the RDME:

$$\begin{aligned} \frac{\partial \Pr(\boldsymbol{n}, \boldsymbol{s}, t)}{\partial t} &= \sum_{i=1}^{k-1} T_i^+ \left\{ (n_i + 1) \Pr(J_i^+ \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \right\} \\ &+ \sum_{i=2}^{k} T_i^- \left\{ (n_i + 1) \Pr(J_i^- \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \right\}. \end{aligned} \tag{2.1.2}$$

Consider the vector of means, defined as

$$\boldsymbol{M}(t) = [M_1(t), \ldots, M_k(t)] = \sum_{\boldsymbol{n}} \boldsymbol{n} \Pr(\boldsymbol{n}, t) \equiv \sum_{n_1=0}^{\mathcal{N}} \sum_{n_2=0}^{\mathcal{N}} \ldots \sum_{n_k=0}^{\mathcal{N}} \boldsymbol{n} \Pr(\boldsymbol{n}, t), \qquad (2.1.3)$$

where, as before, $\mathcal{N}$ is the number of motile cells populating the domain. By multiplying equation (2.1.2) by n_j, summing over all possible values that the vector of cell densities $\boldsymbol{n}$ can take and repeating for $j = 1, \ldots, k$, it can be shown that these means satisfy the following k equations (see Appendix A.1):

$$\frac{\partial M_1}{\partial t} = T_2^- M_2 - T_1^+ M_1, \qquad (2.1.4)$$

$$\frac{\partial M_i}{\partial t} = T_{i-1}^+ M_{i-1} - \left(T_i^+ + T_i^-\right) M_i + T_{i+1}^- M_{i+1}, \qquad (2.1.5)$$

$$\frac{\partial M_k}{\partial t} = T_{k-1}^+ M_{k-1} - T_k^- M_k, \qquad (2.1.6)$$

where equation (2.1.5) holds for $i = 2, \ldots, k-1$. This result is not new. Indeed equation (2.1.5) is the equation for the evolution of the mean cell numbers for a random walk on a lattice as studied by Othmer and Stevens (1997) and Painter and Hillen (2002).

In many scenarios where cell migration is important, active cell movement is regulated by external chemical signalling profiles. These signalling profiles cause cells to migrate at varying rates, depending on where they are in the domain. The precise mechanisms by which different cell types sense and interpret the signalling profile is unclear, but Berg and Purcell (1977), Parent and Devreotes (1999) and Levine et al. (2006) suggest several distinct types of active signal-sensing. We assume that the transition rates are some functions of a signalling profile, $s(x)$, at the current and neighbouring sites: $T_i^{\pm} = f^l(s_i) + f^n(s_{i\pm1})$. Signal transduction may amplify or damp the signal profile, making the transition rate profile different to that of the signal molecule (Erban and Othmer, 2005). Since, in many biological contexts, the precise intracellular signal transduction mechanisms are unknown (Tranquillo et al., 1988), and in order to keep our models as general as possible, we will not attempt to specify the detailed dynamics of intracellular signalling mechanisms. Instead the functions f^l and f^n take account of the effects of the signal transduction contributions from the locally and non-locally sensed signalling profiles (respectively). In this work we will

generally assume a fixed negative exponential profile[4] for the signalling molecule (unless otherwise stated). Exponential signalling profiles, supposedly the result of the diffusion of morphogen molecules from a point source together with decay, have been documented in a variety of experimental contexts (Axelrod et al., 1976; Kicheva et al., 2007). We ignore the effects of signal amplification. This means that f^l and f^n are simple scalar multiplication functions. From now on, for brevity, we write $f^{l,n}(s(x_i)) = d^{l,n}s(x_i)$ and hence $T_i^{\pm} = d^l s(x_i) + d^n s(x_{i\pm1})$.

In order to draw a rigorous correspondence between equations (2.1.4)-(2.1.6) and a population-level description of the variation of cell density with time, we expand terms $M(x_{i\pm1})$ and $s(x_{i\pm1})$ about x_i. Upon taking the limit $\Delta x \to 0$ such that $\lim_{\Delta x\to 0} d^l(\Delta x)^2 = D_l$ and $\lim_{\Delta x\to 0} d^n(\Delta x)^2 = D_n$, as in Lin and Segel (1988), we arrive at the following PDE:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left[\{D_l s(x) + D_n s(x)\}\frac{\partial u}{\partial x} + u\frac{\partial}{\partial x}\{D_l s(x) - D_n s(x)\}\right], \quad (x,t) \in [0,1]\times[0,\infty),\\ &= \frac{\partial\left[\mathcal{J}(u(x,t),s(x))\right]}{\partial x}, \qquad (x,t)\in[0,1]\times[0,\infty), \end{aligned} \tag{2.1.7}$$

where, for notational convenience, we have replaced the term in square brackets in the first line of the above equation with the flux operator[5], $\mathcal{J}$. As part of the formulation of the stochastic model we chose to conserve the number of cells on the domain. Boundary conditions consistent with this assumption and the derived PDE (2.1.7) can be found by using Appendix A.2:

$$\mathcal{J}(u(x,t),s(x))\Big|_{x=0,1} = 0. \tag{2.1.8}$$

Upon substituting $\partial u/\partial t = 0$ into equation (2.1.7), in order to find the steady state density, $u^{st}(x)$, we can integrate with respect to x and use the above boundary condition to write

$$\frac{du^{st}}{dx} = \frac{(D_n - D_l)}{(D_l + D_n)}\frac{u^{st}}{s(x)}\frac{ds(x)}{dx}, \tag{2.1.9}$$

[4]Other signalling profiles are simple to implement. We should note that the individual-level model is unable to resolve differences in the signalling profile on length scales less than the interval spacing, Δx.

[5]Note that the flux operator is the negative of the actual cell flux.

which can be solved to give

$$u^{st}(x) = Bs(x)^{\frac{(D_n - D_l)}{(D_l + D_n)}}, \tag{2.1.10}$$

where B can be found by considering conservation of u across the domain.

2.2 Specific types of signal-sensing

Consider the simplest possible signalling scenario. Either the signalling profile is constant ($s(x) = s \ \forall \ x \in [0,1]$) or, as a special case of this, its effects on the cell population are absent ($s = 1$)[6]. In this case the general transition rates reduce to $T_i^{\pm} = (d^l + d^n)s = d$. In this simple case the equation for evolution of cell density (2.1.7) collapses to

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}, \qquad (x,t) \in [0,1] \times [0,\infty), \tag{2.2.1}$$

where $D = (D_l + D_n)s$, and the corresponding boundary conditions are

$$\left.\frac{\partial u}{\partial x}\right|_{x=0,1} = 0. \tag{2.2.2}$$

We have recapitulated the diffusion equation, as might reasonably have been expected given the lack of bias in the transition rates. The expectation that, in the long-time limit, cell density tends to a homogeneous steady state is confirmed by considering equation (2.1.10) with constant signalling profile, $s(x) = s$.

In the next simplest signal sensing scenario, transition rates are assumed to be a function of a non-constant signalling profile, sensed at the current site only (i.e. $T_i^{\pm} = d^l s(x_i)$). We call this *local* sensing, hence the superscript, l, in the transition rates and subscript on the coefficient D_l of the continuum equation:

$$\frac{\partial u}{\partial t} = D_l \frac{\partial}{\partial x}\left[s(x)\frac{\partial u}{\partial x} + u\frac{\partial s(x)}{\partial x}\right], \qquad (x,t) \in [0,1] \times [0,\infty). \tag{2.2.3}$$

[6] In the absence of a signalling profile cells have been found to undergo random diffusion. Cells in the presence of a uniform concentration of a chemoattractant also exhibit diffusion but with a signal-dependent diffusion coefficient (Lin et al., 2004).

The corresponding boundary conditions are:

$$\left.\frac{\partial(su)}{\partial x}\right|_{x=0,1} = 0. \tag{2.2.4}$$

Equation (2.2.3) permits a non-homogeneous steady state (plotted as a green curve on Fig. 2.2 (e)),

$$u = \frac{B}{s(x)}, \tag{2.2.5}$$

indicating that cells perform a (negative) taxis in response to the signalling profile.

Fig. 2.2 displays a comparison between stochastic simulations averaged over 40 realisations (using the exact direct method (DM)[7] SSA of Gillespie, 1977) and a numerical approximation to the solution of the local sensing equation (2.2.3) at several time points[8]. The similarity between the two is evident and corroborates the theory. To quantify this similarity we calculate the histogram distance error (HDE) metric. The HDE between two curves (defined at discrete points), having normalized frequencies a_i and b_i at point i (i.e. $\sum a_i = \sum b_i = 1$), is given by

$$HDE = \frac{1}{2}\sum_{i=1}^{k} |a_i - b_i|, \tag{2.2.6}$$

where the sum is over all i such that either $a_i \neq 0$ or $b_i \neq 0$ (Cao and Petzold, 2005).

The HDE for local sensing (see Fig. 2.2 (f)) saturates at a low value (as it does for all other types of signal sensing considered in this section (data not shown)), indicating that the PDE and individual-based models are in good agreement even when averaging over as few as 40 stochastic simulations. The initial peak in the HDE is due to the unavoidable disparity between the initial condition of the PDE and the stochastic simulation.

If cells are allowed to sense the signalling gradient in their neighbouring intervals, but not in their current interval, we call this *non-local* sensing and the corresponding transition

[7]The DM, first outlined by Gillespie (1977), is implemented by choosing a statistically accurate time for the next reaction to occur and then deciding randomly, but with the correct probability, which reaction will occur next. This algorithm is trivial to implement (although harder to implement efficiently, as we shall see in Chapter 4) for all the different types of signal-sensing described in this section.

[8]All PDEs defined on the non-growing domain are solved using `Matlab`'s parabolic/elliptic PDE solver `PDEPE`. This solver bounds the relative error between the numerical solution and the actual solution of the PDE at 0.001 by default (i.e. 0.1%), although the tolerance can also be user defined. This error in the PDE solution is small in comparison to the magnitude of the difference between the PDE solution and the stochastic simulations (see Fig. 2.2 (f), for example).

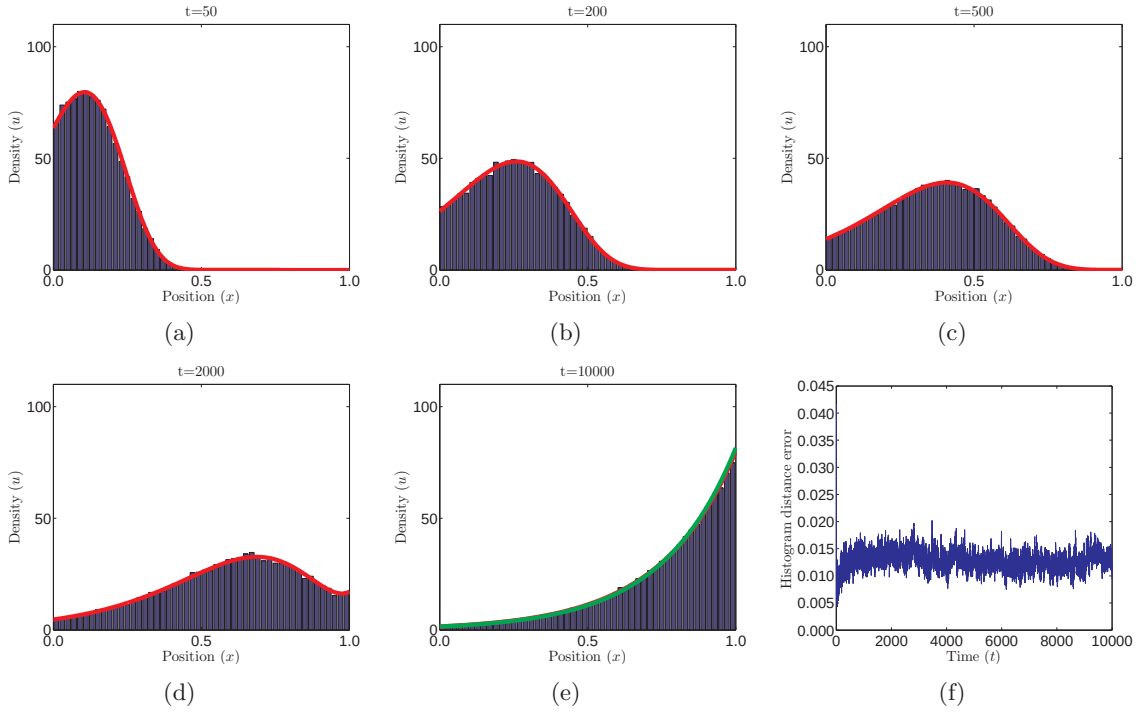


Figure 2.2: *(a)-(e) Time course of cells moving in response to a locally sensed signal profile, $s(x)$. The signalling function is $s(x) = \exp(-ax)$, where $a = 4$. The histograms represent an average of 40 stochastic realisations of the system. The red curves represent the solution of PDE* (2.2.3) *and the green curve in (e) represents the steady state solution of the PDE found analytically. (f) The evolution of the HDE. All $\mathcal{N} = 1000$ cells are released from the centre of the first interval (of the $k = 50$ that the domain is divided into), i.e. $x = 0.01$. The parameter values are $d^l = 1.0$ and $D_l = d^l(\Delta x)^2$ where $\Delta x = 1/k = 0.02$.*

rates are $T_i^{\pm} = d^n s(x_{i\pm1})$. These transition rates give rise to a model whose cell densities evolve according to

$$\frac{\partial u}{\partial t} = D_n \frac{\partial}{\partial x}\left[s(x)\frac{\partial u}{\partial x} - u\frac{\partial s(x)}{\partial x}\right], \qquad (x,t) \in [0,1] \times [0,\infty), \tag{2.2.7}$$

in the continuum limit, with the corresponding boundary conditions,

$$s(x)\frac{\partial u}{\partial x} - u\frac{\partial s(x)}{\partial x}\bigg|_{x=0,1} = 0. \tag{2.2.8}$$

Equation (2.2.7) also permits a non-homogeneous steady state, although, this time, the corresponding taxis is positive as the cells climb the signal gradient.

In the full general model, which we refer to as *average* sensing, providing $D_l \neq D_n$ we can achieve either a positive or a negative taxis in response to the signalling profile. The

sign of the taxis will depend on the relative magnitudes of D_l and D_n which, in turn, are related to the strengths of the local and non-local signal sensing mechanisms.

2.3 Extensions to the RDME approach

In order to make the RDME-based comparison of these two models more versatile we can add more biologically motivated properties. For example, cells undergoing directed migration can sense and respond in different ways to more than one signalling profile (Lin et al., 2005). We could introduce further complexity to the individual-based model by incorporating two or more separate signalling profiles. This extension is trivial to implement and, as such, will not be discussed further. However, in the subsequent sections we will implement the following extensions to the model: (i) allow cells to proliferate and die, (ii) allow the movement of cells to depend on the density of the surrounding cells and (iii) allow cells to alter the environment in which they find themselves (by altering the signalling profile) and to amplify the signal, replicating the effects of inter-cellular signal transduction. We will discuss each of these in the subsections that follow.

2.3.1 Cell flux and decay

We can consider cell flux (possibly due to cell division (birth), migration into and out of the domain or cell death) by assuming that a cell in interval i dies at rate p_i^a and that a new cell is placed in interval i at constant rate p_i^c per unit time. p_i^c can be chosen proportional to n_i in order to reflect cell proliferation if necessary. In order to derive an RDME and hence a corresponding PDE we will neglect active cell movement as the derivation of the terms in the PDE relating to active cell movement is the same as in the above cases and can be reintroduced easily.

In order to succinctly incorporate proliferation and decay into the RDME we introduce the creation and annihilation (resp.) operators A_i^c: $\mathbb{R}^k \to \mathbb{R}^k$ and A_i^a: $\mathbb{R}^k \to \mathbb{R}^k$ (resp.)

defined by

$$A_i^c : [n_1, \ldots, n_i, \ldots, n_k] \to [n_1, \ldots, n_i + 1, \ldots, n_k],$$
$$A_i^a : [n_1, \ldots, n_i, \ldots, n_k] \to [n_1, \ldots, n_i - 1, \ldots, n_k],$$

for $i = 1, \ldots, k$. Incorporating these effects into the RDME gives

$$\begin{aligned} \frac{\partial \Pr(\boldsymbol{n}, \boldsymbol{s}, t)}{\partial t} &= \sum_{i=1}^{k} p_i^a \left\{ (n_i + 1) \Pr(A_i^c \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \right\} \\ &+ \sum_{i=1}^{k} p_i^c \left\{ \Pr(A_i^a \boldsymbol{n}, \boldsymbol{s}, t) - \Pr(\boldsymbol{n}, \boldsymbol{s}, t) \right\}. \end{aligned} \tag{2.3.1}$$

Considering the means we find the relationship,

$$\frac{\partial M_i}{\partial t} = p_i^c - p_i^a M_i, \qquad \text{for } i = 1, \ldots, k, \tag{2.3.2}$$

as, perhaps, we might intuitively have expected. Therefore, in the limit as interval size tends to zero, the corresponding PDE is

$$\frac{\partial u}{\partial t} = p^c(x) - p^a(x) u, \qquad (x, t) \in [0, 1] \times [0, \infty). \tag{2.3.3}$$

Re-incorporating the terms due to general cell movement, the PDE becomes

$$\frac{\partial u}{\partial t} = \frac{\partial \left(\mathcal{J}(u(x,t), s(x)) \right)}{\partial x} + p^c(x) - p^a(x) u, \qquad (x, t) \in [0, 1] \times [0, \infty), \tag{2.3.4}$$

with the corresponding zero flux boundary conditions,

$$\mathcal{J}(u(x,t), s(x)) \Big|_{x=0,1} = 0. \tag{2.3.5}$$

Fig. 2.3 shows the comparison between the derived PDE and the stochastic simulations when cell flux and decay are incorporated into the model. There is an influx of cells into all intervals of the domain at a constant rate, p^c, and all cells decay at rate p^a. Good agreement

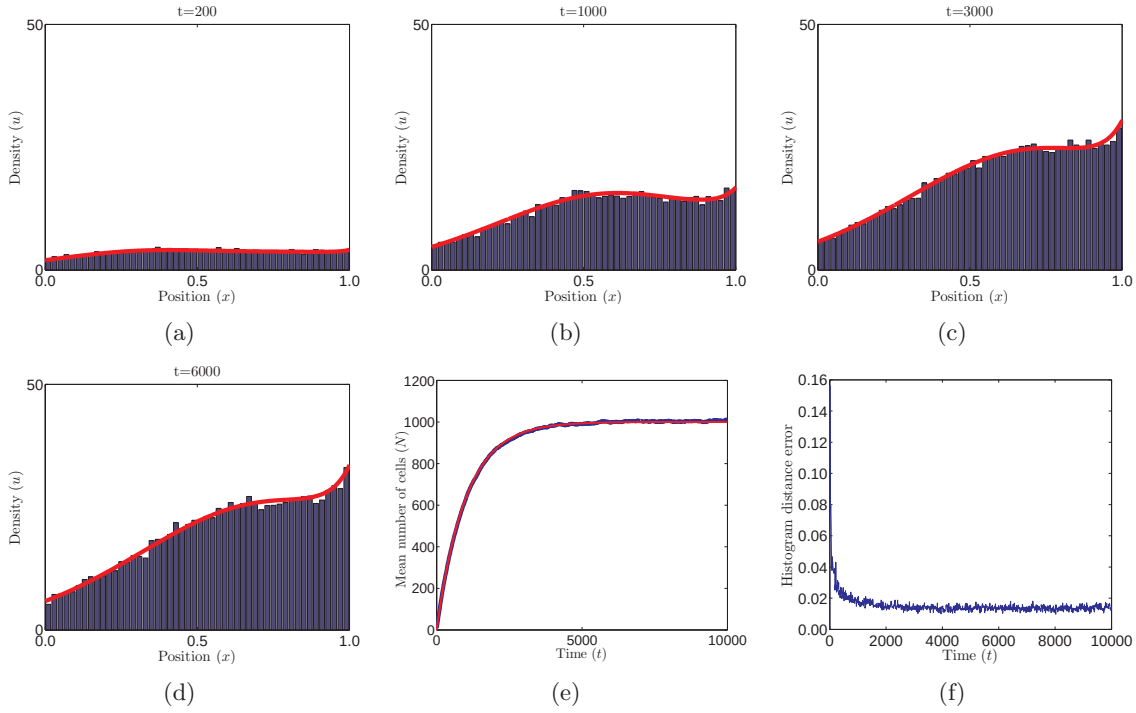


Figure 2.3: *(a)-(d) Time course of cells moving in response to a locally sensed signal profile,* $s(x) = \exp(-ax)$*, with cell flux and decay. The histograms represent an average of 40 realisations of the SSA with transition probabilities* $T^{\pm} = d^l s(x_i)$*, creation rate,* p_i^c*, and annihilation rate,* p_i^a*. The red curves represent the solution of the PDE (2.3.4) found numerically using* `Matlab`*'s* `pdepe.m` *PDE solver. (e) Comparison of the evolution of the number of cells populating the domain for the stochastic (blue) and continuum (red) models. An approximate steady state in cell numbers is reached by* $t \approx 6000$*. (f) The evolution of the HDE. Initially all intervals of the domain are empty and the boundaries are assumed to be reflecting. The local sensing mechanism is the same as for Fig. 2.2. The parameter values are* $a = 4$*,* $p_i^c = 0.02$ *for* $i = 1, \ldots, k$*,* $p_i^a = 0.001$ *for* $i = 1, \ldots, k$*,* $d^l = 1$ *and* $D_l = d^l(\Delta x)^2$ *where* $\Delta x = 1/k = 0.02$*.*

between the stochastic simulation and the solution of the PDE through time can be seen. With low cell numbers initially the HDE takes a relatively large value, but as cell numbers increase the HDE decreases indicating a closer correspondence of the stochastic model to the continuum (see Fig 2.3 (f)).

Consider the simple deterministic differential equation for the number of cells in the domain at time t:

$$\frac{dN}{dt} = \sum_{i=1}^{k} \left[p_i^c - p_i^a N_i(t) \right], \tag{2.3.6}$$

which, if $p_i^c = p_1^c$ and $p_i^a = p_1^a$ for all $i = 1, \ldots, k$ (as in Fig. 2.3), simplifies to

$$\frac{dN}{dt} = p_1^c k - p_1^a N. \tag{2.3.7}$$

We can solve equation (2.3.7) to give

$$N(t) = \frac{p_1^c k}{p_1^a} \left(1 - \exp(-p_1^a t)\right), \tag{2.3.8}$$

given an initially empty domain. Clearly a steady state is reached as $t \to \infty$. Using the parameter values given in Fig. 2.3, the steady state number of cells can be found to be $N_{st} = 1000$. By time $t \approx 6000$ the process is stationary with cell densities in the stochastic regime approximated well by those suggested by the PDE solution.

2.3.2 Density-dependent diffusion

We now assume that transition rates between intervals depend on the density of cells in the current or neighbouring intervals. We present two possible ways to formulate this mathematically:

$$T_i^{\pm} = d^l f(n_i) + d^n g(n_{i\pm1}) \text{ or } T_i^{\pm} = df(n_i)g(n_{i\pm1}), \tag{2.3.9}$$

depending on the sort of density-dependence we are interested in modelling. Transition rates of the first type will be referred to as modelling 'additive' density-dependence and the second type will be referred to as modelling 'multiplicative' density-dependence. Several forms of density-dependence are investigated by Painter and Hillen (2002) in the context of quorum sensing and volume filling. Anguige and Schmeiser (2009) also incorporate volume filling and cell-cell adhesion in a discrete-continuum equivalence framework. Here we present a more general approach to incorporating density-dependence into models of cell migration, which can be reduced to the methods of Painter and Hillen (2002) by the appropriate choice of f and g.

We can still write the RDME in the usual form of equation (2.1.2), but note that including density-dependence in the model introduces non-linearities into the calculations of the mean cell numbers of the underlying stochastic process. In the case of additive

density-dependence the RDME reads as follows:

$$\begin{aligned}\frac{\partial M_i}{\partial t} &= \left\langle \left[n_{i-1} \left(d^l f(n_{i-1}) + d^n g(n_i) \right) \right] \right\rangle + \left\langle \left[n_{i+1} \left(d^l f(n_{i+1}) + d^n g(n_i) \right) \right] \right\rangle \\ &- \left\langle \left[n_i \left(d^l f(n_i) + d^n g(n_{i-1}) \right) \right] \right\rangle - \left\langle \left[n_i \left(d^l f(n_i) + d^n g(n_{i+1}) \right) \right] \right\rangle, \end{aligned} \tag{2.3.10}$$

and in the case of multiplicative density-dependence it reads:

$$\begin{aligned}\frac{\partial M_i}{\partial t} &= \langle [d n_{i-1} f(n_{i-1}) g(n_i)] \rangle + \langle [d n_{i+1} f(n_{i+1}) g(n_i)] \rangle \\ &- \langle [d n_i f(n_i) g(n_{i-1})] \rangle - \langle [d n_i f(n_i) g(n_{i+1})] \rangle, \end{aligned} \tag{2.3.11}$$

for $i = 2, \ldots, k-1$, where, as before, for a general function of the cell density $f(n_j)$, $\langle f(n_j) \rangle = \sum_{\boldsymbol{n}} f(n_j) \Pr(\boldsymbol{n}, \boldsymbol{s}, t)$. Similar equations hold for $i = 1, k$. In order to close render the above equations tractable we make the moment closure approximations that the mean of a function of cell density is approximately equal to the function of the mean cell density and that the mean of a product is approximately equal to the product of the means (e.g. $\langle n_i f(n_i) \rangle = \langle n_i \rangle f(\langle n_i \rangle)$). We then Taylor expand the resulting terms about x_i. Upon taking the usual limit as $\Delta x \to 0$ such that $\lim_{\Delta x \to 0} d_l (\Delta x)^2 = D_l$, $\lim_{\Delta x \to 0} d_n (\Delta x)^2 = D_n$ and $\lim_{\Delta x \to 0} d (\Delta x)^2 = D$, respectively, we arrive at the following limiting cases:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left\{ \frac{\partial u}{\partial x} \left[D_l f(u) + D_n g(u) \right] + u \left[D^l \frac{\partial f(u)}{\partial x} - D^n \frac{\partial g(u)}{\partial x} \right] \right\}, \qquad (x,t) \in [0,1] \times [0,\infty), \tag{2.3.12}$$

and

$$\frac{\partial u}{\partial t} = D \frac{\partial}{\partial x} \left\{ g(u) \frac{\partial (u f(u))}{\partial x} - u f(u) \frac{\partial g(u)}{\partial x} \right\}, \qquad (x,t) \in [0,1] \times [0,\infty), \tag{2.3.13}$$

for additive and multiplicative density-dependence respectively. Again zero flux boundary conditions are consistent with the restrictions on cell movement in the individual-based model at $x = 0$ and $x = 1$.

For certain choices of the density-dependence functions f and g it is possible for equations (2.3.12) and (2.3.13) to have diffusivities which are negative making the corresponding PDEs ill-posed. Ill-posedness implies that there is no longer continuous dependence on ini-

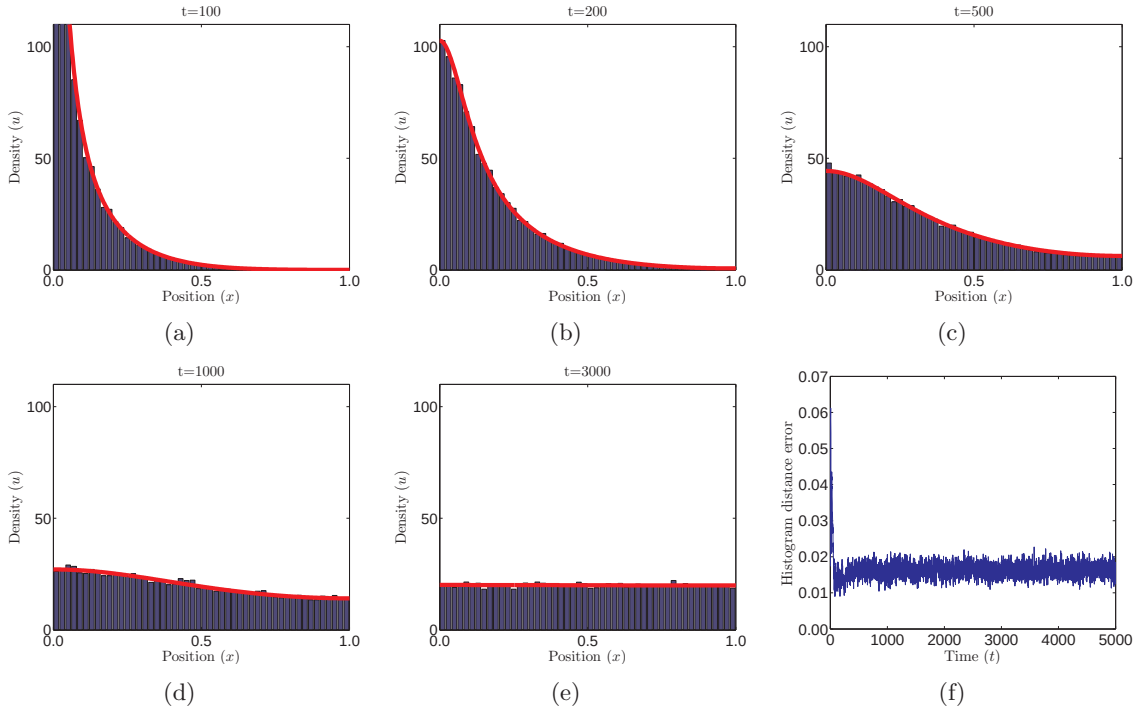


Figure 2.4: *(a)-(e) Time course of cells moving due to density-dependent diffusion. Given the transition rates as a function of local cell numbers:* $T^{\pm} = d^l A/(A + n_i)$*, the continuum solution can be found from PDE* (2.3.10)*. Figure descriptions and initial and boundary conditions are as in Fig. 2.2. The parameter values are* $A = 50$*,* $d^l = 1$ *and* $D_l = d^l(\Delta x)^2$*, where* $\Delta x = 0.02$*.*

tial data or equivalently that an infinitesimal perturbation of boundary data can grow in an unbounded manner away from the unperturbed solution. A canonical example of an ill-posed parabolic PDE is the backwards heat/diffusion equation:

$$\frac{\partial u}{\partial t} = -D\frac{\partial^2 u}{\partial x^2}. \tag{2.3.14}$$

Anguige and Schmeiser (2009) study a model of cell migration which incorporates cell-cell adhesion and volume filling. They find that in certain parameter regimes (high cell-cell adhesion) the non-linear PDE they derive from an individual-based position-jump model can have negative diffusivity. In these cases the PDE is ill-posed and its numerical solution becomes impossible. In this case, in order to gain insight into the behaviour of the cells, it is necessary to return to the individual-based model.

Fig. 2.4 compares the stochastic simulation of additive density-dependent diffusion with purely local dependence on cell density to the PDE approximation from equation (2.3.12).

As predicted, with large cell numbers the agreement is good. Cell density eventually progresses towards a homogeneous steady state in both models.

2.3.3 Stochastic modelling of the environment

As noted by Othmer and Stevens (1997), an important aspect of taxis is whether the individual merely detects the signal or alters it as well by, for example, producing or degrading the signalling molecules. When the individual alters the signal in this manner the local density of the individuals and the intensity of the signal become inter-dependent. In a biological context this can happen when individuals aggregate in response to signals from so-called 'organiser' individuals, and then relay the signal themselves. Specific examples include the myx-amoeba *Dictyostelium*. *Dictyostelium* amoebae can modify the extracellular chemoattractant cyclic adenosine monophosphate (cAMP) concentration in their environment both by secretion of additional cAMP and by phosphodiesterase destruction of cAMP (Othmer and Stevens, 1997). On a larger scale ants can follow and contribute to pheromone trails left by other members of their cohort in order to share information about food sources (Sumpter and Beekman, 2003).

We can extend our stochastic-deterministic equivalence framework to include cases in which the migrating cells control the signal density themselves. For simplicity we consider signalling molecules which are allowed to diffuse with transition rates $T^{\pm}_{s,i} = d^s$ for $i = 1, \ldots, k$, and cells which move according to the signalling molecule concentration in a local manner: $T^{\pm}_{c,i} = d^c f(s_i)$, where s_i now represents the *varying* number of signalling molecules in interval i and f is, as before, a function which allows cells to amplify the signal in the model if necessary. The subscripts s and c on the transition rates refer to the species they act on: signal molecules and cells, respectively.

In order to demonstrate the coherence of the modular way in which we have extended our model in Sections 2.3.1-2.3.3 we combine aspects of a variety of features from these sections as follows. We assume (i) a production of cells in the first interval of the domain with rate p^c, (ii) that cells themselves produce the signalling molecule at rate p^s and (iii) that both the cells and the signalling molecules decay with rates λ_c and λ_s, respectively. Neither cells nor signalling molecules will be allowed to leave the domain.

Let $\Pr(\boldsymbol{n}, \boldsymbol{s}, t)$ be the joint probability that $N_i = n_i$ and $S_i = s_i$ for $i = 1, \ldots, k$, where, as before, $\boldsymbol{n} = [n_1, \ldots, n_k]$ and $\boldsymbol{s} = [s_1, \ldots, s_k]$. It is possible (although time-consuming) to show that the RDME can be written as

$$\begin{aligned}
\frac{\partial}{\partial t}\Pr(\boldsymbol{n}, \boldsymbol{s}, t) &= \sum_{i=1}^{k-1} d^s \left\{(s_i + 1)\Pr(\boldsymbol{n}, J_i^+ \boldsymbol{s}, t) - s_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=2}^{k} d^s \left\{(s_i + 1)\Pr(\boldsymbol{n}, J_i^- \boldsymbol{s}, t) - s_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=1}^{k} p^s \left\{n_i \Pr(\boldsymbol{n}, A_i^a \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=1}^{k} \lambda_s \left\{(s_i + 1)\Pr(\boldsymbol{n}, A_i^c \boldsymbol{s}, t) - s_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=1}^{k-1} d^c f(s_i) \left\{(n_i + 1)\Pr(J_i^+ \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=2}^{k} d^c f(s_i) \left\{(n_i + 1)\Pr(J_i^- \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ \sum_{i=1}^{k} \lambda_c \left\{(n_i + 1)\Pr(A_i^c \boldsymbol{n}, \boldsymbol{s}, t) - n_i \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\} \\
&+ p^c \left\{\Pr(A_1^a \boldsymbol{n}, \boldsymbol{s}, t) - \Pr(\boldsymbol{n}, \boldsymbol{s}, t)\right\}, \qquad (2.3.15)
\end{aligned}$$

where $J_i^{\pm}$ are the familiar state change operators and A_i^c and A_i^a the creation and annihilation operators, respectively, of Section 2.3.1. The mean signalling molecule numbers evolve according to

$$\begin{aligned}
\frac{\partial \langle s_1 \rangle}{\partial t} &= d^s \left(\langle s_2 \rangle - \langle s_1 \rangle\right) + p^s \langle n_1 \rangle - \lambda_s \langle s_1 \rangle, \qquad (2.3.16) \\
\frac{\partial \langle s_i \rangle}{\partial t} &= d^s \left(\langle s_{i+1} \rangle - 2\langle s_i \rangle + \langle s_{i-1} \rangle\right) + p^s \langle n_i \rangle - \lambda_s \langle s_i \rangle, \qquad (2.3.17) \\
\frac{\partial \langle s_k \rangle}{\partial t} &= d^s \left(\langle s_{k-1} \rangle - \langle s_k \rangle\right) + p^s \langle n_k \rangle - \lambda_s \langle s_k \rangle, \qquad (2.3.18)
\end{aligned}$$

where equation (2.3.17) holds for $i = 2, \ldots, k-1$. Those for the cell numbers evolve

according to

$$\frac{\partial\langle n_1\rangle}{\partial t} = d^c\langle n_2 f(s_2)\rangle - d^c\langle n_1 f(s_1)\rangle + p^c - \lambda_c\langle n_1\rangle, \tag{2.3.19}$$

$$\frac{\partial\langle n_i\rangle}{\partial t} = d^c\langle n_{i-1} f(s_{i-1})\rangle - 2d^c\langle n_i f(s_i)\rangle + d^c\langle n_{i+1} f(s_{i+1})\rangle - \lambda_c\langle n_i\rangle, \tag{2.3.20}$$

$$\frac{\partial\langle n_k\rangle}{\partial t} = d^c\langle n_{k-1} f(s_{k-1})\rangle - d^c\langle n_k f(s_k)\rangle - \lambda_c\langle n_k\rangle, \tag{2.3.21}$$

where, similarly, equation (2.3.20) holds for $i = 2, \ldots, k-1$. Now that the signalling profile changes dynamically, non-linearities have been introduced into the equations for the mean cell numbers, whereas, since it is only the *production* of signalling molecule that depends directly on cell numbers, rather than their movement, the equations for the means of the signalling molecules remain linear in s and n. Since equations (2.3.19)-(2.3.21) depend on higher order moments than means, we make the moment closure approximation $\langle n_i f(s_j)\rangle = \langle n_i\rangle f(\langle s_j\rangle)$. This enables us to draw correspondence with the following system of PDEs:

$$\frac{\partial s}{\partial t} = D_s\frac{\partial^2 s}{\partial x^2} + p^s n - \lambda_s s, \tag{2.3.22}$$

$$\frac{\partial u}{\partial t} = D_c\frac{\partial}{\partial x}\left[f(s)\frac{\partial u}{\partial x} + u\frac{\partial f(s)}{\partial x}\right] - \lambda_c u. \tag{2.3.23}$$

For this system $x \in [0,1]$ and $t \in [0,\infty)$. The variables s and u represent signalling molecule and cell density, respectively, and, as usual, D_s and D_c result from taking limits as $\Delta x \to 0$: $D_s = \lim_{\Delta x\to 0} d^s(\Delta x)^2$ and $D_c = \lim_{\Delta x\to 0} d^c(\Delta x)^2$. Boundary conditions for the 'cell number' PDE (2.3.23) are specified for consistency with equations (2.3.19) and (2.3.21), namely a flux at rate p^c into the domain at $x = 0$ and zero flux at $x = 1$. Zero flux conditions are imposed at both boundaries for equation (2.3.22). Following the same procedure we can achieve similar results for non-local and average sensing of the signalling molecule (not shown).

In Fig. 2.5 we see good agreement between the average of the individual-based stochastic simulations and the PDEs derived from the master equations describing the process for both the stochastic signal and cell densities.

Consider the simple system of deterministic differential equations for the number of cells

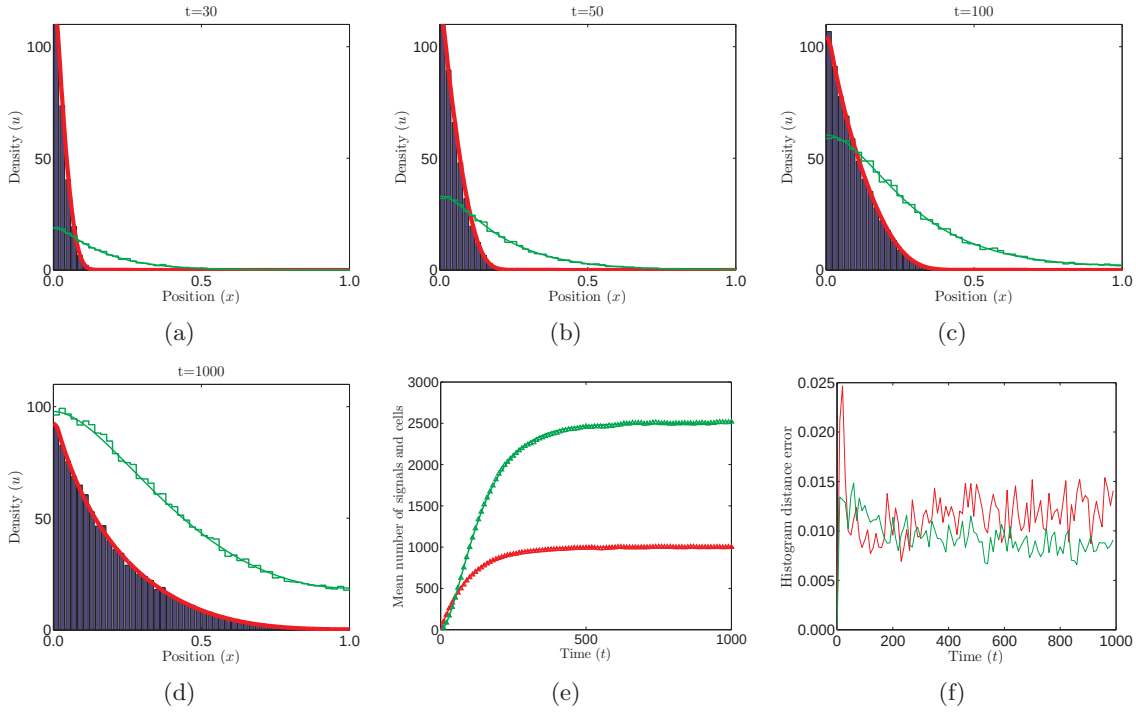


Figure 2.5: *Cells moving according to a local signalling mechanism with inter-dependence of signal molecules and cells. (a)-(d) The histograms and piecewise continuous green curves (resp.) represent an average of 40 realisations of the SSA for cell numbers and signal molecule numbers (resp.). The solid, continuously differentiable lines show the results of the numerical simulation of the equivalent continuum model given by equations* (2.3.22) *(green, for signal molecules) and* (2.3.23) *(red, for cells). Transition rates for the cells are given by* $T^{\pm}_{c,i} = d^c s_i$ *and those for the signalling molecules are given by* $T^{\pm}_{s,i} = d^s$*. (e) Comparison of the evolution of the number of cells populating the domain for the stochastic (red triangles,* $\triangle$*) and continuum (red line) models and the number of signalling molecules for the stochastic (green triangles,* $\triangle$*) and continuum (green line) models. An approximate steady state in cell numbers and signalling molecules is reached by* $t \approx 700$*. (f) Evolution of the HDE for cells (red) and signal (green). Initially the domain is assumed to be empty of both cells and signalling molecules. The parameter values are* $d^c = 0.01$*,* $d^s = 5$*,* $p^c = 10$*,* $p^s = 0.05$*,* $\lambda_c = 0.01$*,* $\lambda_s = 0.02$ *and* $\Delta x = 0.02$*.*

and signalling molecules in the domain at time t:

$$\frac{dN}{dt} = p^c - \lambda_c N, \tag{2.3.24}$$

$$\frac{dS}{dt} = p^s N - \lambda_s S, \tag{2.3.25}$$

where the domain is initially empty of both cells and signal molecules. We can solve the

system (2.3.24)-(2.3.25) for the evolution of $N(t)$ and $S(t)$:

$$N(t) = \frac{p^c}{\lambda_c}\left(1-\exp(-\lambda_c t)\right), \tag{2.3.26}$$

$$S(t) = \frac{p^s p^c}{\lambda_c}\left(\frac{1}{\lambda_s}+\frac{\exp(-\lambda_c t)}{\lambda_c-\lambda_s}-\frac{\lambda_c\exp(-\lambda_s t)}{\lambda_s(\lambda_c-\lambda_s)}\right). \tag{2.3.27}$$

Clearly a steady state in both cell and signal molecule numbers is achieved as $t\to\infty$. Using the parameter values given in Fig. 2.5, the steady state number of cells can be found to be $N_{st}=1000$ and the steady state number of signalling molecules is $S_{st}=2500$. By time $t\approx 700$ a approximate steady state has been reached. Cell densities in the stochastic regime are similar to those suggested by the PDE solution.

2.4 Incorporating domain growth

Considering cell motility on a purely stationary domain may often be unrealistic for biological processes: cell growth, division and movement itself will cause the size of the tissue to change dynamically in many biologically plausible situations. Specifically, we are interested in domain growth (rather than shrinkage) as this is the more biologically important of the two processes. For example, domain growth occurs in developmental processes such as somitogenesis (Tam, 1981), *Drosophila* wing-disc formation (Martín et al., 2004; Rogulja and Irvine, 2005) and various other processes involved in embryogenesis and, more generally, in development (Wolpert et al., 2006).

As previously noted, from a biological point of view, it is important to be able to incorporate domain growth into our models of cell migration. In this section we present the diffusion equation on an exponentially growing domain derived from purely continuum considerations. We also introduce a simple form of domain growth in the on-lattice stochastic model (a constant rate of interval-splitting) and are able to show that the stochastic and continuum models are equivalent in the same manner as in the previous sections. The derivation is, however, not as clear-cut and requires the use of less intuitive moment closure approximations, which we attempt to justify physically. We extend our equivalence framework on the growing domain to incorporate other, more complex and biologically plausible

forms of domain growth. These extensions will make our stochastic and deterministic representations more applicable to experimental situations, especially in the context of modelling developmental processes where domain growth is inherent.

2.4.1 Kinematic derivation of the general cell movement equations for a growing domain in the population-based model

In deriving the equation for expected cell density, $u(\boldsymbol{x}, t)$, at time t and position $\boldsymbol{x}$ on the growing domain, $\Omega(t)$ (with boundary $\partial\Omega(t)$), we follow Crampin (2000) by considering conservation of matter arguments in a generalised approach for a single chemical species. Consider the rate of production of cells, the reaction kinetics and the flux of cells in and through an arbitrary elemental volume, $V(t)$, moving with the flow caused by domain growth. Using conservation of matter arguments and Reynolds transport theorem we can show

$$\int_{V(t)} \left[\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u)\right] \mathrm{d}\boldsymbol{x} = \int_{V(t)} \left[\nabla \cdot \boldsymbol{\mathcal{J}}(u, s) + R(u)\right] \mathrm{d}\boldsymbol{x}, \tag{2.4.1}$$

where $\boldsymbol{v}$ is the velocity field of the flow generated by the growth of the domain:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}(\boldsymbol{x}, t). \tag{2.4.2}$$

$R(u)$ represents any reactions taking place, which in the case of cell movement is simply the expected net rate of cell creation/degradation, and $\boldsymbol{\mathcal{J}}(u, s)$ is the higher-dimensional analogue of the one-dimensional flux operator, $\mathcal{J}(u, s)$, defined in equation (2.1.7) for general signal sensing on the stationary domain. Applying the usual argument of the arbitrary choice of $V(t)$ implies that equation (2.4.1) holds everywhere on the growing domain. Therefore

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u) &= \nabla \cdot \boldsymbol{\mathcal{J}}(u, s) + R(u), \\ \boldsymbol{x} \in \Omega(t), &\qquad t \in [0, \infty). \end{aligned} \tag{2.4.3}$$

The two new terms introduced to the reaction-diffusion equation (2.4.3), due to domain growth, are $\boldsymbol{v} \cdot \nabla u$, which represents material transported around the domain by the flow induced by domain growth and $u \nabla \cdot \boldsymbol{v}$ which represents dilution due to local volume increase (Crampin, 2000). To close the problem we have to specify the appropriate initial and boundary conditions. For simplicity we will generally use zero flux boundary conditions, whose implementation is discussed in Appendix A.3. We will consider the same initial conditions as in previous simulations.

2.4.2 Local specification of domain growth in the population-based model

So far we have worked in complete generality for a reaction-diffusion system on a growing n-dimensional domain, but in this section we consider a system without reactions (only signal sensing) on a one-dimensional domain. Our system (2.4.3) reduces to:

$$\frac{\partial u}{\partial t} + \frac{\partial (vu)}{\partial x} = \frac{\partial \left(\mathcal{J}\left(u, s\right)\right)}{\partial x}, \quad x \in [0, L(t)] \quad t \in [0, \infty), \tag{2.4.4}$$

where $\mathcal{J}(u, s)$ is the familiar one-dimensional flux operator.

In order to deal with this free boundary problem we convert to Lagrangian coordinates X and τ (i.e. coordinates that move with the flow). General detailed descriptions of how to do this can be found in Crampin (2000). For our purposes an abridged one-dimensional demonstration will suffice.

The Lagrangian coordinate system takes advantage of the premise that, given the initial position of a tissue element and the strain, $S(x, t, u(x, t))$, acting to deform the domain, we will be able to determine the position of that element at all times, giving the element trajectory or pathline, $\Gamma(X, \tau)$. Choosing a reference point (the origin) which remains fixed we complete the definition of the coordinate system and can specify the initial condition $\Gamma(X, 0) = X$ and the boundary condition $\Gamma(0, \tau) = 0$.

We assume that growth is specified locally by the strain, $S(x, t, u(x, t))$, which can be related to the velocity, v, of the domain in Eulerian coordinates (see page 51):

$$v = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{d}\Gamma}{\mathrm{d}t} = \int_0^{x(t)} S(\bar{x}, t, u(\bar{x}, t)) \mathrm{d}\bar{x}. \tag{2.4.5}$$

In addition to enabling us to work out the flow velocity from the strain, we can use this relationship to find a PDE which determines the tissue trajectories:

$$\frac{\partial^2 \Gamma}{\partial \tau \partial X} = \frac{\partial \Gamma}{\partial X} S, \quad X \in [0,1], \quad \tau \in [0,\infty). \tag{2.4.6}$$

We can also convert equation (2.4.4) into Lagrangian coordinates using the following expressions for the partial derivatives:

$$\frac{\partial}{\partial t} = -\frac{\Gamma_\tau}{\Gamma_X}\frac{\partial}{\partial X} + \frac{\partial}{\partial \tau}, \tag{2.4.7}$$

$$\frac{\partial}{\partial x} = \frac{1}{\Gamma_X}\frac{\partial}{\partial X}, \tag{2.4.8}$$

$$\frac{\partial^2}{\partial x^2} = -\frac{\Gamma_{XX}}{\Gamma_X^3}\frac{\partial}{\partial X} + \frac{1}{\Gamma_X^2}\frac{\partial^2}{\partial X^2}. \tag{2.4.9}$$

In Lagrangian coordinates equation (2.4.4) for the evolution of cell density becomes:

$$\frac{\partial u}{\partial \tau} = \frac{1}{\Gamma_X}\frac{\partial J(u(X,\tau), s(X,\tau))}{\partial X} - Su. \tag{2.4.10}$$

For example, in the specific case when $\mathcal{J}(u,s) = D\partial u/\partial x$ in Eulerian coordinates (for the diffusive process), the equation for the evolution of cell density in Lagrangian coordinates reduces to

$$\frac{\partial u}{\partial \tau} = D\left(\frac{1}{\Gamma_X^2}\frac{\partial^2 u}{\partial X^2} - \frac{\Gamma_{XX}}{\Gamma_X^3}\frac{\partial u}{\partial X}\right) - Su. \tag{2.4.11}$$

Once we specify the strain we have all the information we need to solve the above PDEs for the evolution of the domain (2.4.6) and the cell density (2.4.10).

If we assume, for example, that the strain, $S(x,t,u(x,t)) = r$, is constant. We can find the flow, $v = rx$, by considering equation (2.4.5). In this simple case, we can also use equation (2.4.5) to determine the evolution of the pathlines[9]:

$$x = \Gamma(X,\tau) = X\exp(r\tau), \tag{2.4.12}$$

where we have made use of the initial condition $\Gamma(X,0) = X$. We are now in a position to

[9]We note that for a more general strain we will usually be unable to determine an explicit formula for the pathlines and, as such, we will have to solve equations (2.4.6) and (2.4.10) numerically

substitute our expression for $\Gamma(X,\tau)$ into equation (2.4.10):

$$\frac{\partial u}{\partial \tau} = \frac{1}{\exp(rt)} \frac{\partial J(u(X,\tau), s(X,\tau))}{\partial X} - ru, \quad X \in [0,1], \quad \tau \in [0,\infty), \tag{2.4.13}$$

which simplifies to

$$\frac{\partial u}{\partial \tau} = \frac{1}{\exp(2r\tau)} D \left(\frac{\partial^2 u}{\partial X^2} \right), \quad X \in [0,1], \quad \tau \in [0,\infty). \tag{2.4.14}$$

in the case of a diffusive flux.

The problem is closed by conversion of the zero flux boundary conditions to Lagrangian coordinates and the specification of suitable initial conditions. For a discussion of the derivation of the appropriate boundary conditions see Appendix A.3.

2.4.3 Incorporation of domain growth 'reactions' into the individual-based model

In order to include domain growth in the individual-based model in a manner consistent with our stochastic formalism, we now introduce a growth 'reaction' which will allow us to simulate constant, time- or density-dependent domain growth.

As in the previous sections we split the domain, initially of length L_0[10], into k_0 finite intervals, $x_1, \ldots, x_{k_0}$, each of length $\Delta x = L_0/k_0$. One of the k_0 original intervals is chosen, with equal probability, to split into two new intervals (at rate r) via

$$x_i \xrightarrow{r} x_i, x_{i+1}. \tag{2.4.15}$$

Consequently the other intervals are relabelled as follows (see Fig. 2.6):

$$x_j \to \begin{cases} x_j & \text{for} \quad j < i, \\ x_{j+1} & \text{for} \quad j > i. \end{cases}$$

Each 'interval replication' is a discrete instantaneous 'reaction' in which one 'parent' interval becomes two 'daughter' intervals, with each daughter interval having the same length, Δx,

[10] As in previous sections we normalise the initial domain so that $L_0 = 1$.

as the parent interval (see Fig. 2.6) thus lengthening the domain by Δx. The cells that were in the parent interval before splitting are then redistributed to the daughter intervals, using some *symmetric* probability distribution, π, in order to avoid biases in cell movement caused by cell redistribution (see Fig. 2.6). Whether this is a biologically realistic method of cell redistribution (or whether indeed such a redistribution or interval-splitting process occurs biologically at all) seems unlikely and will be discussed further in Chapter 3. However, mathematically, at least, this method is robust. It ensures that no artificial drift of cells is introduced due to the interval-splitting process and can be naturally extended to produce growth in higher dimensions (see Chapter 5).

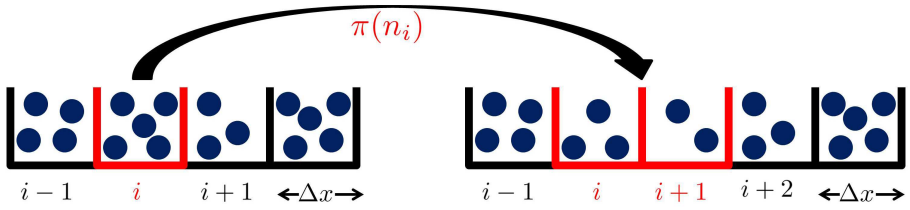


Figure 2.6: *A visual illustration of interval replication. The i^{th} interval (red) splits and the cells (blue circles, •) are redistributed into the two daughter intervals using a symmetric probability distribution ($\pi(n_i)$). The daughter intervals are re-numbered i and $i+1$ and all subsequent intervals have their index incremented by one. All intervals remain the same size, Δx.*

2.5 From the individual-level model to the population-level model on the growing domain

We now return to considering the RDME approach to describe individual-level behaviour, in order to draw a comparison with the PDE (2.4.4) describing the diffusion of cells on an exponentially growing domain. In order to write the RDME for domain growth succinctly we introduce the 'inverse growth' operators G_i: $\mathbb{R}^{k(t)} \to \mathbb{R}^{k(t)-1}$, for each of the intervals, i, that were in existence before the last interval-splitting event, $i = 1, 2, \ldots, k(t)-1$:

$$G_i : [n_1, \ldots, n_{i-1}, n_i, n_{i+1}, n_{i+2}, \ldots, n_k] \to [n_1, \ldots, n_{i-1}, n_i + n_{i+1}, n_{i+2}, \ldots, n_k]. \quad (2.5.1)$$

In effect G_i concatenates intervals i and $i+1$, adding their cell numbers together; the 'inverse' of interval-splitting. For the sake of simplicity, we temporarily ignore the terms of

the RDME describing active cell movement, to arrive at:

$$\frac{\partial P^{k(t)}(\boldsymbol{n},t)}{\partial t} = r\sum_{i=1}^{k(t)-1}\pi(n_i,n_{i+1}|n_i+n_{i+1})\times P^{k(t)-1}(G_i\boldsymbol{n},t) - r\sum_{i=1}^{k(t)}P^{k(t)}(\boldsymbol{n},t), \quad (2.5.2)$$

where the superscript $k(t)$ indicates the number of intervals at time t, as before. The second sum in this expression is derived from the probability that we already had k intervals at time t and that none of them split, where r, as before, is the rate at which intervals split. The first sum is the probability that we had $k(t)-1$ intervals at time t, with cells specifically distributed as $G_i\boldsymbol{n}$, multiplied by the probability that an interval splits and by the probability that the cells in that interval are redistributed to the two daughter intervals in the correct manner, $\pi(n_i,n_{i+1}|n_i+n_{i+1})$, summed over all the $k(t)-1$ intervals in existence at time t. $\pi(n_i,n_{i+1}|n_i+n_{i+1})$ is the probability that, given interval i contained n_i+n_{i+1} cells before splitting, n_i cells end up in interval i and n_{i+1} in interval $i+1$ after the splitting of the i^{th} interval. We make a simple but crucial assumption about π; that it is symmetric so that

$$\pi(n_i,n_{i+1}|n_i+n_{i+1}) = \pi(n_{i+1},n_i|n_i+n_{i+1}). \quad (2.5.3)$$

This means that there is no bias inflicted on the cells' destinations during interval-splitting: they are no more inclined to stay in the original interval i than they are to move to interval $i+1$.

Since π is a probability distribution we note that it necessarily satisfies the normalisation condition:

$$\sum_{n_i=0}^{n_i+n_{i+1}}\pi(n_i,n_{i+1}|n_i+n_{i+1}) = 1. \quad (2.5.4)$$

Throughout the rest of this chapter we will consider interval-splitting to be implemented using a symmetric binomial distribution such that

$$\pi(n_i,n_{i+1}|n_i+n_{i+1}) = \frac{1}{2^{(n_i+n_{i+1})}}\binom{n_i+n_{i+1}}{n_i}. \quad (2.5.5)$$

It is also possible to implement the splitting algorithms using an 'even-splitting' method

whereby $\lfloor (n_i + n_{i+1})/2 \rfloor$ cells are distributed to each of the resulting intervals (where $\lfloor x \rfloor$ represents the largest integer smaller than x) with the extra cell being assigned randomly to one of the two intervals in the case of $n_i + n_{i+1}$ being odd. Another possibility is uniform splitting where $\pi(n_i, n_{i+1}|n_i + n_{i+1}) = (n_i + n_{i+1} + 1)^{-1}$. In the presence of cell migration we found the specific cell redistribution method not to have a significant effect on the model result (results not shown). We therefore chose to implement interval-splitting using the binomial distribution and will not discuss the latter two interval-splitting distributions further.

We now redefine the mean cell numbers of the underlying stochastic process in a similar manner to the case of the non-growing domain (see equation (2.1.3)) with the notable difference of addition of the superscript $k(t)$ to indicate the number of intervals at time t:

$$\boldsymbol{M}^{k(t)}(t) = [M_1^{k(t)}(t), \ldots, M_k^{k(t)}(t)] = \sum_{\boldsymbol{n}} \boldsymbol{n} P^{k(t)}(\boldsymbol{n}, t). \tag{2.5.6}$$

Multiplying the master equation (2.5.2) by n_j and summing over $\boldsymbol{n}$ we have,

$$\frac{\partial M_j^{k(t)}}{\partial t} = r \sum_{i=1}^{k(t)-1} \sum_{\boldsymbol{n}} n_j \pi(n_i, n_{i+1}|n_i + n_{i+1}) P^{k(t)-1}(G_i \boldsymbol{n}, t) - r \sum_{i=1}^{k(t)} M_j^{k(t)}(t), \tag{2.5.7}$$

which holds for $j = 1, \ldots, k(t) - 1$. We will concern ourselves with the splitting of a general interval i. For $i \neq j, j-1$ the evaluation of the inner sum in equation (2.5.7),

$$\sum_{\boldsymbol{n}} n_j \pi(n_i, n_{i+1}|n_i + n_{i+1}) P^{k(t)-1}(G_i \boldsymbol{n}, t), \tag{2.5.8}$$

is straightforward. Rewrite the sum over $\boldsymbol{n}$ as the sum over the vector of cell values before the splitting of interval i (i.e. $G_i \boldsymbol{n}$) and then sum over all the possible ways the cells could be redistributed once the split occurs i.e.

$$\sum_{G_i \boldsymbol{n}} \left[\sum_{\bar{n}_i = 0}^{n_i + n_{i+1}} \pi(\bar{n}_i, n_i + n_{i+1} - \bar{n}_i | n_i + n_{i+1}) \right] n_j P^{k(t)-1}(G_i \boldsymbol{n}, t). \tag{2.5.9}$$

The term inside the square brackets is the sum of all the possible values of a discrete

probability distribution and, as noted in equation (2.5.4), is equal to unity. We are left with

$$\sum_{G_i\boldsymbol{n}} n_j P^{k-1}(G_i\boldsymbol{n},t) = \begin{cases} M_{j-1}^{k(t)-1} & \text{for} \quad i < j-1, \\ M_j^{k(t)-1} & \text{for} \quad i > j. \end{cases} \tag{2.5.10}$$

In order to tackle terms involving $i = j-1, j$ we make use of the symmetry assumption detailed in (2.5.3). For example, if we first take $i = j$ then we can consider the inner sum in equation (2.5.7),

$$\sum_{\boldsymbol{n}} \pi(n_j, n_{j+1}|n_j+n_{j+1}) n_j P^{k(t)-1}(G_j\boldsymbol{n},t), \tag{2.5.11}$$

to be equivalent to

$$\sum_{\boldsymbol{n}} \pi(n_j, n_{j+1}|n_j+n_{j+1}) n_{j+1} P^{k(t)-1}(G_j\boldsymbol{n},t), \tag{2.5.12}$$

by simply swapping the indices j and $j+1$ and using the symmetry assumption. Adding ((2.5.11) and (2.5.12)) gives

$$\begin{aligned} &2 \times \sum_{\boldsymbol{n}} \pi(n_j, n_{j+1}|n_j+n_{j+1}) n_j P^{k(t)-1}(G_j\boldsymbol{n},t), \\ &= \sum_{\boldsymbol{n}} \pi(n_j, n_{j+1}|n_j+n_{j+1}) (n_j+n_{j+1}) P^{k(t)-1}(G_j\boldsymbol{n},t), \\ &= \sum_{G_j\boldsymbol{n}} \left[\sum_{\bar{n}_j=0}^{n_j+n_{j+1}} \pi(\bar{n}_j, n_j+n_{j+1}-\bar{n}_j|n_j+n_{j+1}) \right] (n_j+n_{j+1}) P^{k(t)-1}(G_j\boldsymbol{n},t), \\ &= M_j^{k(t)-1}, \end{aligned} \tag{2.5.13}$$

with the final equality holding since the term in square brackets is again equal to unity. Clearly this can be rewritten as

$$\sum_{\boldsymbol{n}} \pi(n_j, n_{j+1}|n_j+n_{j+1}) n_j P^{k(t)-1}(G_j\boldsymbol{n},t) = \frac{1}{2} M_j^{k(t)-1}, \tag{2.5.14}$$

and in an analogous way, for $i = j-1$, we have

$$\sum_{\boldsymbol{n}} \pi(n_{j-1}, n_j|n_{j-1}+n_j) n_j P^{k(t)-1}(G_{j-1}\boldsymbol{n},t) = \frac{1}{2} M_{j-1}^{k(t)-1}. \tag{2.5.15}$$

Substituting the equations (2.5.10), (2.5.14) and (2.5.15) into equation (2.5.7) gives

$$\frac{\partial M_j^{k(t)}}{\partial t} = r\sum_{i=1}^{j-2} M_{j-1}^{k(t)-1} + r\sum_{i=j+1}^{k(t)-1} M_j^{k(t)-1} + \frac{r}{2}M_j^{k(t)-1} + \frac{r}{2}M_{j-1}^{k(t)-1} - r\sum_{i=1}^{k(t)} M_j^{k(t)}, \quad (2.5.16)$$

for $j = 1, \ldots, k(t)-1$ (note that in the case of $j = k(t)-1$ we evaluate the second summation in equation (2.5.16) as zero). Since all the terms in the summations are independent of i this can simply be rewritten as

$$\frac{\partial M_j^{k(t)}}{\partial t} = r(j-3/2)M_{j-1}^{k(t)-1} + r(k(t)-j-1/2)M_j^{k(t)-1} - rk(t)M_j^{k(t)}. \quad (2.5.17)$$

To complete the extension of our RDME-PDE equivalence framework on the growing domain we would like to derive a correspondence between this simplified RDME and a PDE. In order to do this we make a heuristic approximation relating the mean cell numbers when the domain has $k(t)$ intervals, $M_j^{k(t)}$, to the mean cell numbers when the domain has $k(t)-1$ intervals, $M_j^{k(t)-1}$:

$$(k(t)-1)M_j^{k(t)-1} = k(t)M_j^{k(t)}. \quad (2.5.18)$$

Assuming diffusion occurs on a faster time scale than growth, this heuristic approximation implies that cell density dilutes in a natural way. It is also intrinsically linked with the idea of conservation of cells: on average the mean number of cells in each interval when we have $k(t)$ intervals should be smaller by a factor of $(k(t)-1)/k(t)$ than the mean number of cells in each interval when there are $k(t)-1$ intervals. For a further discussion of the validity of this heuristic see Baker et al. (2010).

Applying the heuristic approximation to equation (2.5.17) yields

$$\frac{k(t)-1}{k(t)}\frac{\partial M_j^{k(t)-1}}{\partial t} = r(j-3/2)M_{j-1}^{k(t)-1} + r(k(t)-j-1/2)M_j^{k(t)-1} - r(k(t)-1)M_j^{k(t)-1}, \quad (2.5.19)$$

into which we can substitute the usual Taylor expansion to give

$$\begin{aligned}\frac{(k(t)-1)}{k(t)}\frac{\partial M^{k(t)-1}}{\partial t}(x_j,t) &= r(j-3/2)\left[M^{k(t)-1}(x_j,t)-\Delta x\frac{\partial M^{k(t)-1}}{\partial x}(x_j,t)+\ldots\right]\\ &+ r(k(t)-j-1/2)M^{k(t)-1}(x_j,t)-r(k(t)-1)M^{k(t)-1}(x_j,t),\\ &= -r\left[j\Delta x\frac{\partial M^{k(t)-1}}{\partial x}(x_j,t)+M^{k(t)-1}(x_j,t)\right]+o(\Delta x),\\ &= -r\left[x_j\frac{\partial M^{k(t)-1}}{\partial x}(x_j,t)+M^{k(t)-1}(x_j,t)\right]+o(\Delta x). \quad (2.5.20)\end{aligned}$$

Taking the limit as $\Delta x \to 0$ (and $k \to \infty$) gives the PDE

$$\frac{\partial M^{k(t)-1}}{\partial t}+rx\frac{\partial M^{k(t)-1}}{\partial x}=-rM^{k(t)-1}, \quad (2.5.21)$$

which can be recognised as the PDE (2.4.4) in the absence of diffusion. This approach is generalisable to the case where we allow cell movement due to diffusion and active movement as a result of signal-sensing, as we shall see in the following section.

2.5.1 Reintroducing cell movement

Recall from Section 2.1 that in the absence of domain growth we were able to show that the stochastic simulation gave rise to the series of equations (2.1.4)-(2.1.6). When deriving the RDME and the relationship for the mean cell numbers on the growing domain (as noted in the previous section) it is trivial to insert the additional terms due to cell movement into the derivation as they are not affected by interval-splitting. It is a simple step to combine the results for the simplifications of the RDMEs from Section 2.1 and 2.5 for the mean cell numbers,

$$\begin{aligned}\frac{\partial M^{k(t)-1}}{\partial t}(x_j,t) &= -\frac{k(t)r}{(k(t)-1)}\left[x_j\frac{\partial M^{k(t)-1}}{\partial x}(x_j,t)-M^{k(t)-1}(x_j,t)\right]\\ &+\left[T^+_{j-1}M^{k(t)-1}_{j-1}-\left(T^+_j+T^-_j\right)M^{k(t)-1}_j+T^-_{j+1}M^{k(t)-1}_{j+1}\right]+o(\Delta x), \quad (2.5.22)\end{aligned}$$

which, in the limit as $k \to \infty$, gives

$$\frac{\partial M^{k(t)-1}}{\partial t} + rx\frac{\partial M^{k(t)-1}}{\partial x} = \frac{\partial\left(\mathcal{J}\left(M^{k(t)-1}, s(x)\right)\right)}{\partial x} - rM^{k(t)-1}, \tag{2.5.23}$$

which is analogous to the PDE (2.4.4) derived from analytical considerations. We must remember, when solving PDE (2.5.23) on the transformed, normalised domain, that the signal profile must also be transformed to $s(\eta) = \exp(-a\eta e^{rt})$. It may be more biologically realistic to have the original signal gradient expand as the domain expands i.e. $s(x) = \exp(-axe^{-rt}) = \exp(-a\eta)$, which would be equally easy to implement.

Fig. 2.7 shows a comparison between stochastic simulations of cell migration on a growing domain with a constant rate of interval-splitting, $r = 0.001$, and the solution of the diffusion equation (2.4.4) on an exponentially growing (with rate r) domain. The corresponding PDE (as will be the other PDEs on the growing domain) is solved numerically using `Matlab` and the `Numerical Algorithms Group` (`NAG`) solver `D03PE` (The Numerical Algorithms Group, 2009). The similarity between the two simulations corroborates the theory. In order to quantify this comparison we have also plotted the evolution of the HDE in Fig. 2.7 (d). The HDE remains low throughout the duration of the simulation and is the same order of magnitude as the HDEs found in the case of the stationary domain (see Fig. 2.2 (f)) although it does increase over time.

In Figs. 2.7 (a)-(c), it appears that the stochastic domain length is considerably longer than the predicted continuum model. In some of the stochastic simulations the domain will have grown more and in some less than the expected exponential growth derived (see Section 2.5.2) for the PDE. As a consequence of the way we average cell numbers (summing all the numbers of cells in a specific interval at a specific time over 40 repeats but only dividing through by the number of simulations for which the domains that have grown to or past the relevant length, rather than 40 for each interval) intervals which may have only been in existence in a relatively small number of the repeats can appear to have a larger average cell density than perhaps they should, giving the impression of a longer domain in the stochastic simulations. This effect will be seen to a greater or lesser extent throughout the figures in the remainder of this and subsequent chapters.

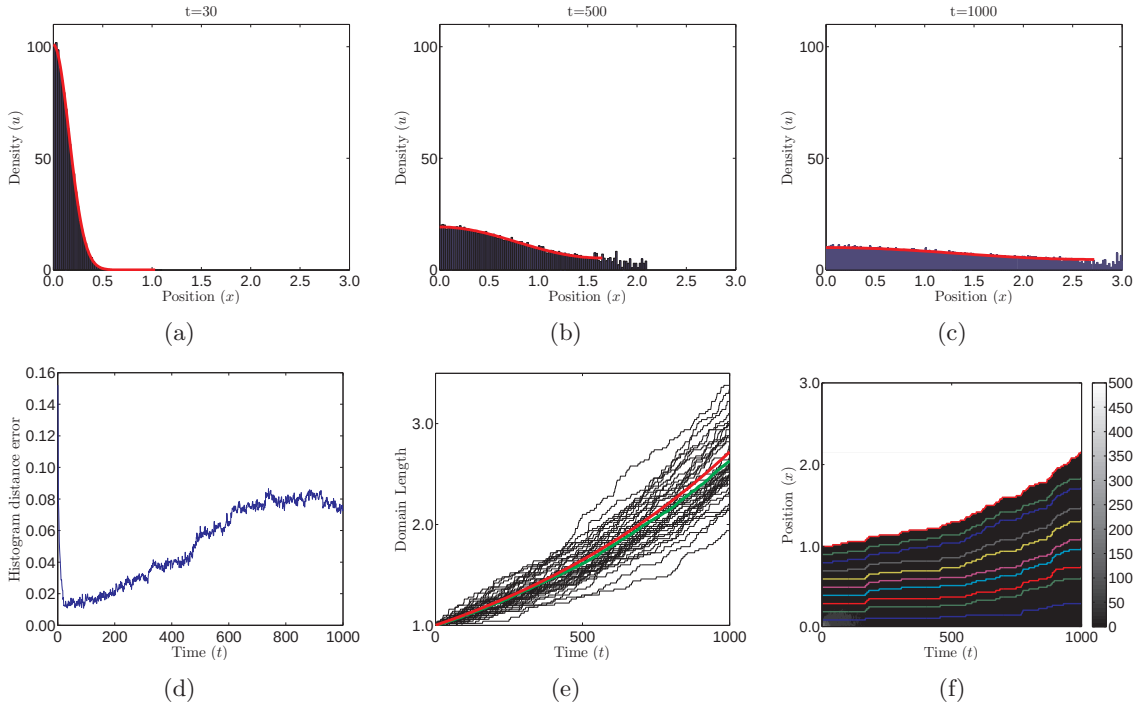


Figure 2.7: *Cells diffusing with constant rate, $D = d(\Delta x)^2$, on an exponentially growing domain at several time points. (a)-(c) The histograms represent an average of 40 stochastic realisations of the system using Gillespie's exact DM SSA (Gillespie, 1977) with transition rates $T_i^{\pm} = d$. The red curves represent the solution of the PDE (2.4.4) with flux operator $\mathcal{J} = D\partial u/\partial x$. All $\mathcal{N} = 1000$ cells are released from the centre of the first interval of the domain. (d) Evolution of the HDE. (e) Evolution of domain length for each of the 40 realisations, with their average in green and the predicted deterministic domain length of the continuous model plotted in red. (f) The results of a typical stochastic simulation. The coloured lines indicate the trajectories of 10 individual intervals initially positioned evenly across the domain. The parameter values are $k_0 = 50$, $r = 0.001$, $d = 1.0$ and $\Delta x = 0.02$.*

2.5.2 A master equation for domain length

In the case of constant interval-splitting probabilities, r, we would like to derive an expression for the length of the domain at time t. Let P_k denote the probability that the growing domain is divided into k intervals at time t. The master equation describing the number of intervals at time t is

$$\frac{\mathrm{d}P_k}{\mathrm{d}t} = (k-1)rP_{k-1} - krP_k, \qquad \text{for} \qquad k = k_0, k_0+1, \ldots, \tag{2.5.24}$$

where, as before, k_0 is the initial number of intervals. The initial conditions for the ordinary differential equations (ODEs) are given by

$$P_k(0) = \begin{cases} 1 & \text{if} \quad k = k_0, \\ 0 & \text{if} \quad k \neq k_0. \end{cases}$$

It is possible to solve ODEs (2.5.24) inductively subject to the above initial conditions to obtain:

$$P_k(t) = \binom{k-1}{k_0-1}(1-\exp(-rt))^{k-k_0}\exp(-k_0 rt), \quad \text{for} \quad k > k_0. \tag{2.5.25}$$

The mean number of intervals, $K(t) = \langle k(t) \rangle$, is given by

$$K(t) = \sum_{k=k_0}^{\infty} kP_k(t) = \sum_{k=k_0}^{\infty} k \begin{pmatrix} k-1 \\ k_0-1 \end{pmatrix} (1-\exp(-rt))^{k-k_0}\exp(-k_0 rt). \tag{2.5.26}$$

Taking out a factor of $k_0\exp(rt)$, the terms to the right of the summation can be recognised (after some relabelling) as belonging to the probability mass function (PMF) of the negative binomial distribution:

$$f(x; s, p) = \binom{x+s-1}{s-1} p^s (1-p)^x, \tag{2.5.27}$$

where $x = k - k_0$, $s = k_0 + 1$ and $p = \exp(-rt)$. Therefore, the sum of these terms equates to unity and we are left with

$$K(t) = k_0\exp(rt), \tag{2.5.28}$$

implying that the mean domain length grows exponentially in time (see Fig. 2.7 (e)).

2.6 More general domain growth

In this section we consider locally specified domain growth of the form $f(x, t, u(x, t))$. This functional form means that growth can depend on position, can alter throughout time and can be affected by cell density. As briefly mentioned earlier, density-dependent domain growth, in particular, will be of interest to us due to its biological motivation (Lieberman

and Glaser, 1981). We choose to conserve cells and take (time- and space- independent) transition rates $T_i^{\pm} = d$. Note that, using results from previous sections, it is simple enough to reintroduce the time- and space- dependent transition rates.

2.6.1 Derivation of the population-level PDE

We aim to find the functional form of the PDE describing domain growth when a general form of the domain growth function is considered. We can employ our previous results as far as equation (2.4.3) (see Section 2.4.1). For simplicity, we will neglect cell proliferation and decay and hence assume $R(u) = 0$, leaving

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u) &= \nabla \cdot \mathcal{J}(u, s), \\ \boldsymbol{x} \in \Omega(t), \qquad & t \in [0, \infty). \end{aligned} \tag{2.6.1}$$

We will work in one dimension, but it should be noted that the results presented are extendable to higher dimensions. As before, we require expressions for the domain length, $L(t)$, and for the flow due to growth, v. Consider dividing the domain into k intervals of length Δx_i. For a small time interval δt we can express the domain length at time $t + \delta t$ as

$$\begin{aligned} L(t + \delta t) &= \sum_{i=1}^{k} \left[1 + f(x_i, t, u(x_i, t))\delta t\right] \Delta x_i, \\ &= L(t) + \sum_{i=1}^{k} f(x_i, t, u(x_i, t))\delta t \Delta x_i. \end{aligned} \tag{2.6.2}$$

Rearranging and taking the limit as $\delta t \to 0$ gives

$$\frac{dL(t)}{dt} = \sum_{i=1}^{k} f(x_i, t, u(x_i, t))\Delta x_i. \tag{2.6.3}$$

Taking the limit as $\Delta x_i \to 0$ for $i = 1, \ldots, k$ the summation becomes an integration and gives the following integro-differential equation (IDE) for $L(t)$:

$$\frac{dL(t)}{dt} = \int_0^{L(t)} f(x, t, u(x, t))\mathrm{d}x. \tag{2.6.4}$$

This IDE can be integrated to give

$$L(t) = \int_0^t \int_0^{L(\bar{t})} f(x, \bar{t}, u(x, \bar{t}))\mathrm{d}x\mathrm{d}\bar{t}. \tag{2.6.5}$$

This formulation can be used in order to derive the form of the flow due to domain growth, $v(x, t, u(x, t))$:

$$v(x, t, u(x, t)) = \frac{dx}{dt} = \int_0^x f(\hat{x}, t, u(\hat{x}, t))\mathrm{d}\hat{x}. \tag{2.6.6}$$

Substituting this expression for the flow due to domain growth into the one-dimensional version of PDE (2.6.1) gives us the general form of the PDE for cell migration on a growing, one-dimensional domain:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(u \int_0^x f(\hat{x}, t, u(\hat{x}, t))\mathrm{d}\hat{x}\right) = \frac{\partial \mathcal{J}}{\partial x}, \quad x \in [0, L(t)], \quad t \in [0, \infty). \tag{2.6.7}$$

2.7 Time-dependent domain growth

In this section we allow the interval-splitting rates to be purely time-dependent and show that we can produce linear and logistic domain growth in both the individual-based and population-based models. These extensions further increase the versatility of our equivalence framework and, especially in the case of logistic growth, serve to increase its biological applicability.

2.7.1 Linear growth

Consider a domain whose length is given by

$$L(t) = L_0\alpha(t), \text{ with } \alpha(t) = 1 + \rho t, \tag{2.7.1}$$

where L_0 is the initial domain length (which we will take to be unity) and ρ is the rate of growth. Considering equation (2.6.4) we note that this type of domain growth is equivalent to setting $f(x, t, u(x, t)) = \rho/\alpha(t)$. Given $f(x, t, u(x, t))$ we can find a flow using equation

(2.6.6):

$$v(x,t) = \int_0^x f(\hat{x}, t, u(\hat{x}, t))\mathrm{d}\hat{x} = \frac{\rho x}{1+\rho t}. \tag{2.7.2}$$

This formulation of the strain gives rise to the continuum PDE for linear domain growth in the absence of cell movement:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -\frac{\partial}{\partial x}\left(\frac{\rho x u}{1+\rho t}\right), \\ &= -\frac{\rho x}{1+\rho t}\frac{\partial u}{\partial x} - \frac{\rho}{1+\rho t}u, \quad (x,t) \in [0, L_0(1+\rho t)] \times [0,\infty). \end{aligned} \tag{2.7.3}$$

The next question we must ask is what forms of stochastic interval-splitting probabilities give rise to this type of domain growth in the continuum case? There are two possible answers to this question. We could simply choose the interval-splitting probabilities for each interval to be $r(t) = \rho/(1+\rho t)$. We can derive the PDE as before in an equivalent manner. However, a more interesting approach is to formulate the linear growth rate stochastically, so that the rate of interval-splitting is inversely proportional to the stochastic variable $k(t)$, the current number of intervals:

$$r(t) = \frac{\phi}{k(t)}. \tag{2.7.4}$$

Excluding cell movements, as usual, we can formulate the RDME by substituting this expression for r into equation (2.5.2):

$$\frac{\partial P^{k(t)}(\boldsymbol{n},t)}{\partial t} = \frac{\phi}{k(t)-1}\sum_{i=1}^{k(t)-1}\pi(n_i, n_{i+1}|n_i+n_{i+1}) \times P^{k(t)-1}(G_i\boldsymbol{n},t) - \frac{\phi}{k(t)}\sum_{i=1}^{k(t)}P^{k(t)}(\boldsymbol{n},t), \tag{2.7.5}$$

where π and G_i, $i = 1,\ldots,n$ are as defined earlier. By analogy to the process which takes us from (2.5.2) to (2.5.17) we can derive a system of mean equations away from the end intervals:

$$\frac{\partial M_j^{k(t)}}{\partial t} = \frac{\phi}{k(t)-1}\left[(j-3/2)M_{j-1}^{k(t)-1} + (k(t)-j-1/2)M_j^{k(t)-1}\right] - \phi M_j^{k(t)}. \tag{2.7.6}$$

As always we search for correspondence of this RDME with a deterministic PDE, in this

case, of the form of equation (2.7.3). It will be useful to consider the mean number of intervals, $K(t)$. As before, let $P_k(t)$ denote the probability that the domain has grown to k intervals at time t. We can write down a master equation satisfied by P_k:

$$\frac{dP_k}{dt} = \phi P_{k-1} - \phi P_k, \quad \text{for} \quad k = k_0, k_0 + 1, \ldots, \tag{2.7.7}$$

with initial conditions

$$P_k(0) = \begin{cases} 1 & \text{if} \quad k = k_0, \\ 0 & \text{if} \quad k \neq k_0. \end{cases}$$

We can solve this infinite set of ODEs inductively to give $P_k(t) \equiv 0$ for $k < k_0$, as expected, and

$$P_k(t) = \frac{1}{(k-k_0)!}(\phi t)^{k-k_0} e^{-\phi t}, \tag{2.7.8}$$

for $k \geq k_0$. Therefore, the mean number of intervals is given by

$$K(t) = \sum_{k=k_0}^{\infty} k P_k(t) = \sum_{k=k_0}^{\infty} \frac{k}{(k-k_0)!}(\phi t)^{k-k_0} e^{-\phi t}. \tag{2.7.9}$$

By a simple relabelling and separation of the summand, it can be shown the mean domain length increases linearly with time

$$K(t) = k_0 + \phi t. \tag{2.7.10}$$

Now we return to equation (2.7.6), which provides a relationship between mean cell numbers in neighbouring intervals, and attempt to draw a comparison with the continuum model (equation (2.7.3)) for the same process. In order to relate the coefficients of the term $\partial u/\partial x$ in the two models, we must choose

$$\frac{\phi}{k(t)-1} = \frac{\phi}{k_0 + \phi t - 1} = \frac{\rho}{1+\rho t}, \tag{2.7.11}$$

where the first equality follows from substituting $K(t)$, the mean number of intervals given by equation (2.7.10), for $k(t)$. The second equality is satisfied by setting $\phi = \rho(k_0 - 1)$. We next have to match the coefficients of u. We can do this by making the simple heuristic

approximation:

$$M_j^{k(t)-1} \approx M_j^{k(t)}, \tag{2.7.12}$$

which essentially implies that cell density should vary smoothly over the domain. With these assumptions in place the PDE for mean cell numbers becomes

$$\frac{\partial M_j^{k(t)-1}}{\partial t} + \frac{\rho x}{1+\rho t}\frac{\partial M_j^{k(t)-1}}{\partial x} = -\frac{\rho}{1+\rho t} M_j^{k(t)-1}, \tag{2.7.13}$$

which is equivalent to the PDE (2.7.3) for cell density on a linearly growing domain in the absence of cell movement. We can easily reintroduce cell movement, as before, to give the resultant PDE,

$$\frac{\partial u}{\partial t} + \frac{\rho x}{1+\rho t}\frac{\partial u}{\partial x} = \frac{\partial \mathcal{J}}{\partial x} - \frac{\rho}{1+\rho t} u. \tag{2.7.14}$$

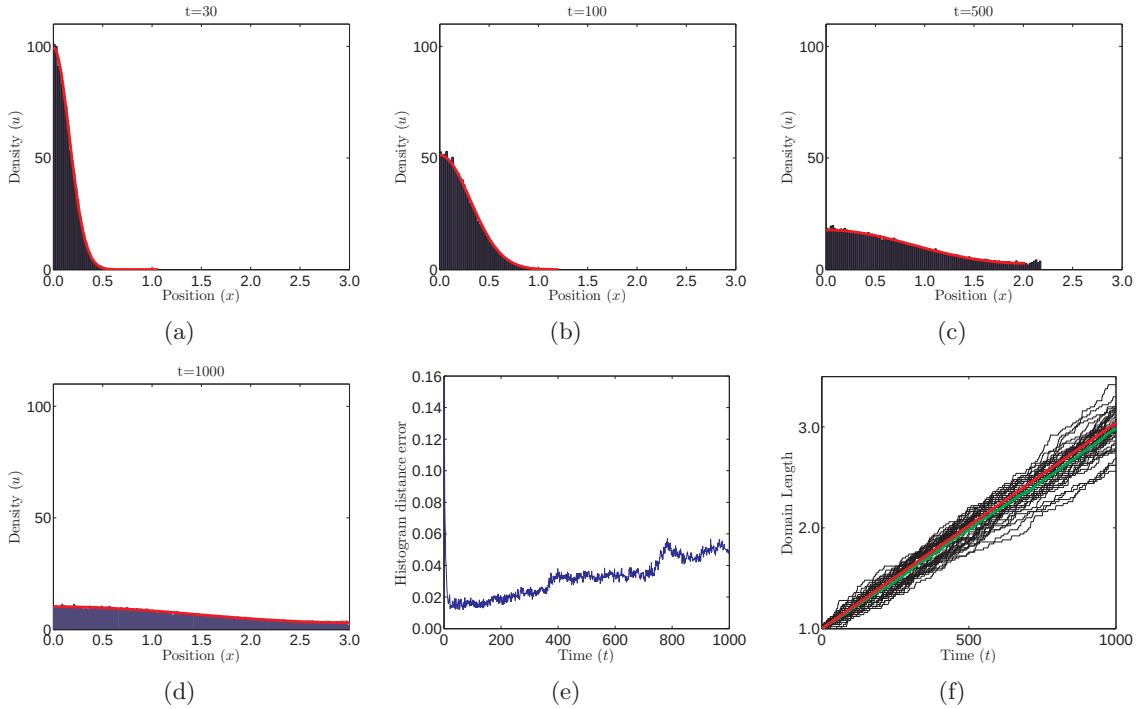


Figure 2.8: *Cells diffusing with constant rate, $D = d(\Delta x)^2$, on a growing domain with linear domain growth at several time points. (a)-(d) Histograms represent an average of 40 stochastic realisations of the system with transition rates $T_i^{\pm} = d$. The red curves represent the solution of the PDE (2.7.14) with flux operator $\mathcal{J} = D\partial u/\partial x$. All $\mathcal{N} = 1000$ cells are initially positioned in the centre of the left-hand-most interval of the domain and the boundaries are assumed to be reflecting. (e) Evolution of the HDE. (f) Evolution of domain length for each of the 40 realisations, with their average in green and the predicted deterministic domain length of the continuous model plotted in red. The parameter values are $k_0 = 50$, $\phi = 0.1$, $\rho = \phi/(k_0 - 1) \approx 0.00204$, $d = 1.0$ and $\Delta x = 0.02$.*

The results of simulating the above system with diffusion as the mechanism for cell movement are shown in Fig. 2.8. As expected there is a good correlation between the stochastic cell density and the PDE solution for the duration of the simulations (see Fig. 2.8 (a)-(e)) and between the deterministic domain length and that given by the stochastic model (see Fig. 2.8 (f)).

2.7.2 Logistic growth

In a realistic model of, for example, embryo growth, we must be able to build stochasticity into domain growth and, eventually, the deceleration and cessation of growth. So far the models we have considered (for exponential and linear growth) do not have this functionality built in. It has been found in experiments on turtle and albatross embryos (Leshem et al. (1991) and Pettit et al. (1982) resp.) that the mass of the embryo grows according to a logistic curve. It is also postulated that logistic growth is a good model for the underlying tissue growth in cell migration models in several mathematical papers (Crampin et al., 1999; Landman et al., 2003). The proposed underlying causes of this growth are mitosis, causing an exponential-type growth term, and competition for essential nutrients or contact inhibition of growth giving rise to a negative term proportional to the square of the number of cells. Combining these two terms gives rise to logistic growth, with carrying capacity related to the speed of division and the strength of competition.

Consider logistic growth such that the domain length obeys the following ODE:

$$\frac{dL}{dt} = \rho L \left(1 - \frac{L}{L_0 \xi}\right), \tag{2.7.15}$$

where L_0 is the initial domain length, ρ represents the rate of growth and ξ is the ratio between final and initial domain sizes. The solution of equation (2.7.15) is

$$L(t) = L_0 \alpha(t), \quad \text{with} \quad \alpha(t) = \frac{\exp(\rho t)}{1 + \frac{1}{\xi}(\exp(\rho t) - 1)}, \tag{2.7.16}$$

In a similar manner to the previous section we find $f(x, t, u(x, t)) = \rho(1 - \alpha(t)/\xi)$ using

(2.6.4). This can then be used in equation (2.6.6) to find the flow due to domain growth

$$v(x,t) = \int_0^x f(\hat{x},t,u(\hat{x},t))\mathrm{d}\hat{x} = \rho x\left(1-\frac{\alpha(t)}{\xi}\right), \tag{2.7.17}$$

and hence the continuum representation of the PDE for cell density in the absence of cell movement:

$$\begin{aligned}\frac{\partial u}{\partial t} &= -\frac{\partial}{\partial x}\left(\rho x u\left(1-\frac{\alpha(t)}{\xi}\right)\right),\\ &= -\rho x\left(1-\frac{\alpha(t)}{\xi}\right)\frac{\partial u}{\partial x} - \rho\left(1-\frac{\alpha(t)}{\xi}\right)u, \quad (x,t)\in[0,L_0\alpha(t)]\times[0,\infty).\end{aligned} \tag{2.7.18}$$

Returning to stochastic considerations, if we substitute the constant r in equation (2.5.2) by

$$r(t) = \begin{cases} \phi\left(1-\frac{k(t)}{\xi k_0}\right) & \text{for } k(t) < \xi k_0,\\ 0 & \text{otherwise,}\end{cases} \tag{2.7.19}$$

then we can explicitly write the RDME as follows,

$$\begin{aligned}\frac{\partial P^{k(t)}(\boldsymbol{n},t)}{\partial t} &= \phi\left(1-\frac{k(t)-1}{\xi k_0}\right)\sum_{i=1}^{k(t)-1}\pi(n_i,n_{i+1}|n_i+n_{i+1})\times P^{k(t)-1}(G_i\boldsymbol{n},t)\\ &- \phi\left(1-\frac{k(t)}{\xi k_0}\right)\sum_{i=1}^{k(t)}P^{k(t)}(\boldsymbol{n},t),\end{aligned} \tag{2.7.20}$$

for $k = k_0,\ldots,k_f$, where π and G_i, $i = 1,\ldots,n$ are defined as before and $k_f = \xi k_0$ is the maximum number of intervals to which the domain can grow. This results in an RDME of the form

$$\begin{aligned}\frac{\partial M_j^{k(t)}}{\partial t} &= \phi\left(1-\frac{k(t)-1}{\xi k_0}\right)\left[(j-3/2)M_{j-1}^{k(t)-1} + (k(t)-j-1/2)M_j^{k(t)-1}\right]\\ &- \phi k(t)\left(1-\frac{k(t)}{\xi k_0}\right)M_j^{k(t)}.\end{aligned} \tag{2.7.21}$$

In order to equate the above with a PDE we need to derive an expression for $K(t)$, the mean number of intervals at time t. This can be done directly via the master equation for

the number of intervals at time t:

$$\frac{dP_k}{dt} = \phi(k-1)\left(1-\frac{k-1}{k_f}\right)P_{k-1} - \phi k\left(1-\frac{k}{k_f}\right)P_k, \quad \text{for} \quad k = k_0, k_0+1, \ldots, k_f. \tag{2.7.22}$$

An expression for the mean number of intervals, $K(t)$, at time t is given by

$$\begin{aligned}
\frac{dK}{dt} &= \sum_{k=0}^{\infty} k\frac{dP_k}{dt}, \\
&= \sum_{k=1}^{\infty} \phi k(k-1)\left(1-\frac{k-1}{k_f}\right)P_{k-1} - \sum_{k=0}^{\infty}\phi k^2\left(1-\frac{k}{k_f}\right)P_k, \\
&= \phi\left(\langle k\rangle - \frac{\langle k^2\rangle}{k_f}\right),
\end{aligned} \tag{2.7.23}$$

where $\langle k^n\rangle$ denotes the n^{th} moment of k. Specifically $\langle k\rangle = K(t)$. Since the master equation is non-linear in k we cannot derive $K(t)$ in closed form. However, if we make the approximation, as we have done several times before, that $\langle k^2\rangle = \langle k\rangle^2$ then we can derive a closed form ODE for $K(t)$:

$$\frac{dK}{dt} = \phi K\left(1-\frac{K}{k_f}\right) \quad \Longrightarrow \quad K(t) = \frac{k_0 e^{\phi t}}{1+\frac{1}{\xi}(e^{\phi t}-1)}. \tag{2.7.24}$$

This can be compared directly with the deterministic case (see equations (2.7.16) and (2.7.15)). Returning to the RDME (2.7.21) we can substitute $M_{j-1}^{k(t)-1}$ for its Taylor expansion in the usual manner to give

$$\begin{aligned}
\frac{\partial M_j^{k(t)}}{\partial t} &= \phi\left(1-\frac{k(t)-1}{\xi k_0}\right)\left[(j-3/2)\left(M_j^{k(t)-1} - \Delta x\frac{\partial M_j^{k(t)-1}}{\partial x} + o(\Delta x)\right)\ldots\right. \\
&+ \left.(k(t)-j-1/2)M_j^{k(t)-1}\right] - \phi k(t)\left(1-\frac{k(t)}{\xi k_0}\right)M_j^{k(t)}.
\end{aligned} \tag{2.7.25}$$

In order to derive equivalence between the RDME (2.7.25) and the continuum PDE (2.7.18) we must make two crucial assumptions. The first is needed in order to equate the coefficients of $\partial M_j^{k(t)-1}/\partial x$ in the master equation (2.7.25) and $\partial u/\partial x$ in the PDE (2.7.18).

We must assume that $k(t) \gg 1$ which allows us to make the approximation

$$1 - \frac{k(t)-1}{\xi k_0} \approx 1 - \frac{k(t)}{\xi k_0} = 1 - \frac{\alpha(t)}{\xi}, \tag{2.7.26}$$

so that the coefficients of $\partial u/\partial x$ will be comparable, providing we choose $\phi = \rho$. The second assumption we have to make is in order to relate the coefficients of $M_j^{k(t)-1}$ (in the RDME) and u (in the PDE). Making use of the previous assumption (2.7.26) this requires

$$\rho(k(t)-2)\left(1-\frac{\alpha(t)}{\xi}\right)M_j^{k(t)-1} - \rho k(t)\left(1-\frac{\alpha(t)}{\xi}\right)M_j^{k(t)} = -\rho\left(1-\frac{\alpha(t)}{\xi}\right)M_j^{k(t)-1}, \tag{2.7.27}$$

which simplifies significantly to the usual heuristic approximation of equation (2.5.18):

$$(k(t)-1)M_j^{k(t)-1} = k(t)M_j^{k(t)}. \tag{2.7.28}$$

This assumes that cell density tends to dilute in a natural way, conserving cell numbers across the domain. Using the two approximations above allows us to show

$$\frac{\partial M_j^{k(t)-1}}{\partial t} + \rho x\left(1-\frac{\alpha(t)}{\xi}\right)\frac{\partial M_j^{k(t)-1}}{\partial x} = -\rho\left(1-\frac{\alpha(t)}{\xi}\right)M_j^{k(t)-1}, \tag{2.7.29}$$

for $(x,t) \in [0, L(t)] \times [0,\infty)$. This PDE is clearly in correspondence with the PDE (2.7.18) derived through continuous considerations. As usual we can reintroduce cell movement to this equation to give the continuum approximation:

$$\frac{\partial u}{\partial t} + \rho x\left(1-\frac{\alpha(t)}{\xi}\right)\frac{\partial u}{\partial x} = \frac{\partial \mathcal{J}}{\partial x} - \rho\left(1-\frac{\alpha(t)}{\xi}\right)u, \quad (x,t) \in [0, L(t)] \times [0,\infty). \tag{2.7.30}$$

Results from the numerical simulation of this system in the case of simple diffusion transition rates are shown in Fig. 2.9. There is good correlation between the stochastic density (averaged over 40 repeats) and the continuum PDE solution in Figs. 2.9 (a)-(d). This is corroborated by the consistently low HDE in Fig. 2.9 (e). It is clear from Fig. 2.9 (f)

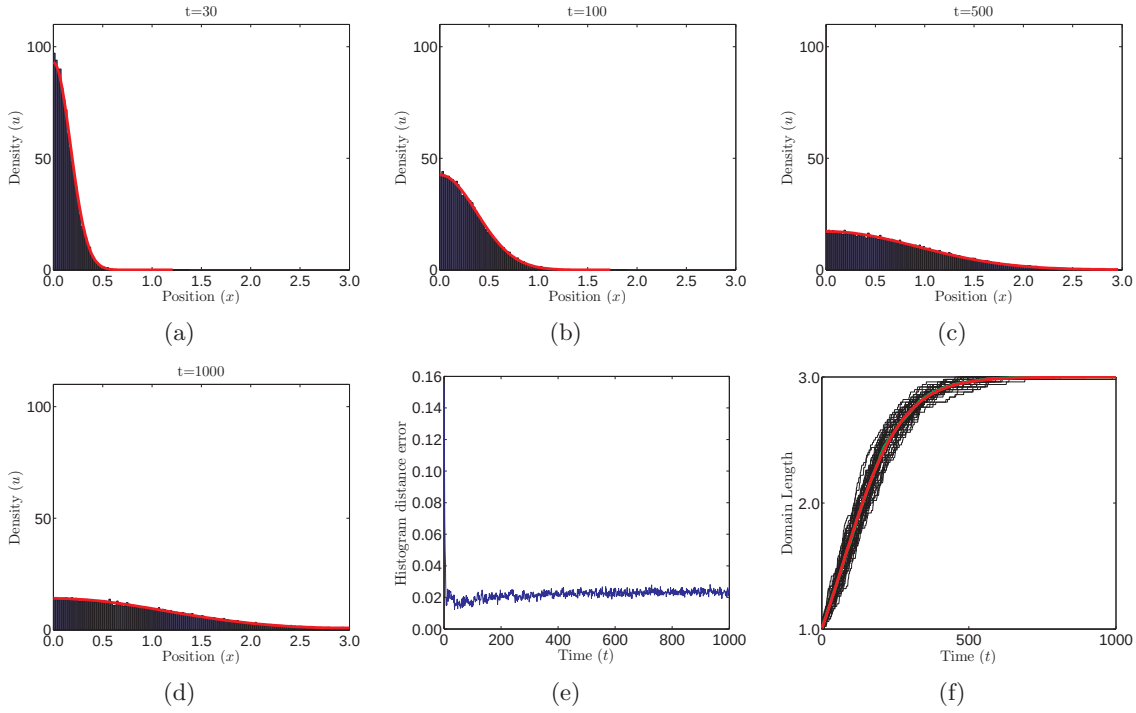


Figure 2.9: *Cells diffusing with constant rate, $D = d(\Delta x)^2$, on a growing domain with logistic domain growth at several time points. The red curves represent the solution of PDE* (2.7.30) *with flux operator $\mathcal{J} = D\partial u/\partial x$. Other figure descriptions, initial and boundary conditions are as for Fig. 2.8. The parameter values are $k_0 = 50$, $\xi = 3$, $\rho = 0.05$, $d = 1.0$ and $\Delta x = 0.02$.*

that the correspondence between the domain lengths in the two models is excellent (the red deterministic curve largely obscuring the green stochastic curve).

2.8 Density-dependent domain growth

In Section 2.6 we derived a PDE which can be used to model cell migration on a domain with density-dependent growth. In this section we show that the individual-based model describing the same process can be considered to be equivalent to the PDE model and corroborate our theory with numerical simulations.

2.8.1 Individual-level behaviour

In the individual-based regime, we use the same stochastic framework as before (see Sections 2.4.3 and 2.5) only now we assume that the interval-splitting rate can be dependent on cell density, time and space. We ignore any explicit temporal or spatial dependence of the

transition rates as these have already been partially explored in previous sections and can easily be appended to this work.

Suppose that the rate of interval-splitting is some function, f, of the number of cells in that interval. Then, neglecting cell movement for the meantime, we can write down the RDME:

$$\begin{aligned}\frac{\partial P^{k(t)}(\boldsymbol{n},t)}{\partial t} &= \sum_{i=1}^{k(t)-1} f(n_i+n_{i+1})\pi(n_i,n_{i+1}|n_i+n_{i+1})P^{k(t)-1}(G_i\boldsymbol{n},t) \\ &- \sum_{i=1}^{k(t)} f(n_i)P^{k(t)}(\boldsymbol{n},t), \end{aligned} \tag{2.8.1}$$

where the 'inverse growth' operators G_i, $i = 1,\ldots,k(t)$, are defined as before. As usual, the superscript, $k(t)$, on the joint probability density function (PDF) indicates the number of intervals to which the domain has grown at time t and π is the symmetric probability distribution which enables the unbiased redistribution of cells upon interval-splitting. Define the means, as usual, for the growing domain (see equation (2.5.6)) and also recall that $\langle\cdot\rangle$ is the stochastic mean of the term in angle brackets. We can use equation (2.8.1) to give us the following relationship involving the mean cell numbers,

$$\begin{aligned}\frac{\partial M_j^{k(t)}}{\partial t} &= \sum_{i=1}^{k(t)-1}\sum_{\boldsymbol{n}} n_j f(n_i+n_{i+1})\pi(n_i,n_{i+1}|n_i+n_{i+1})P^{k(t)-1}(G_i\boldsymbol{n},t) \\ &- \sum_{i=1}^{k(t)}\sum_{\boldsymbol{n}} n_j f(n_i)P^{k(t)}(\boldsymbol{n},t), \quad \text{for} \quad j=1,\ldots,k(t)-1. \end{aligned} \tag{2.8.2}$$

In a manner entirely analogous to the case of constant interval splitting we can consider the inner summation of the first summation in equation (2.8.2):

$$\sum_{\boldsymbol{n}} n_j f(n_i+n_{i+1})\pi(n_i,n_{i+1}|n_i+n_{i+1})P^{k(t)-1}(G_i\boldsymbol{n},t), \tag{2.8.3}$$

and rewrite the sum over $\boldsymbol{n}$ as the sum over the vector of cell values before the splitting of interval i (i.e. $G_i\boldsymbol{n}$) and then sum over all the possible ways the cells could be redistributed

once the split occurs i.e.

$$\sum_{G_i\boldsymbol{n}} f(n_i+n_{i+1})\left[\sum_{\bar{n}_i=0}^{n_i+n_{i+1}} \pi(\bar{n}_i, n_i+n_{i+1}-\bar{n}_i|n_i+n_{i+1})\right] n_j P^{k(t)-1}(G_i\boldsymbol{n},t). \tag{2.8.4}$$

The term in square brackets sums to unity and we can evaluate the remaining terms as follows,

$$\sum_{G_i\boldsymbol{n}} n_j f(n_i+n_{i+1}) P^{k(t)-1}(G_i\boldsymbol{n},t) = \begin{cases} \langle n_{j-1}^{k(t)-1} f(n_i^{k(t)-1})\rangle & \text{for } i < j-1, \\ \langle n_j^{k(t)-1} f(n_i^{k(t)-1})\rangle & \text{for } i > j. \end{cases} \tag{2.8.5}$$

However, we must consider terms where $i = j-1, j$ individually. Using the symmetry assumption on π, we can conclude

$$\begin{aligned} &\sum_{\boldsymbol{n}} \quad n_j f(n_j+n_{j+1})\pi(n_j, n_{j+1}|n_j+n_{j+1}) P^{k(t)-1}(G_j\boldsymbol{n},t), \\ &= \quad \frac{1}{2}\langle n_j^{k(t)-1} f(n_j^{k(t)-1})\rangle, \end{aligned} \tag{2.8.6}$$

and

$$\begin{aligned} &\sum_{\boldsymbol{n}} \quad n_{j-1} f(n_{j-1}+n_j)\pi(n_{j-1}, n_j|n_{j-1}+n_j) P^{k(t)-1}(G_{j-1}\boldsymbol{n},t), \\ &= \quad \frac{1}{2}\langle n_{j-1}^{k(t)-1} f(n_{j-1}^{k(t)-1})\rangle. \end{aligned} \tag{2.8.7}$$

Gathering together these results and substituting into (2.8.2) gives

$$\begin{aligned} \frac{\partial M_j^{k(t)}}{\partial t} \quad = \quad & \sum_{i=1}^{j-2}\langle n_{j-1}^{k(t)-1} f(n_i^{k(t)-1})\rangle + \frac{1}{2}\langle n_{j-1}^{k(t)-1} f(n_{j-1}^{k(t)-1})\rangle + \frac{1}{2}\langle n_j^{k(t)-1} f(n_j^{k(t)-1})\rangle \\ & + \sum_{i=j+1}^{k(t)-1}\langle n_j^{k(t)-1} f(n_i^{k(t)-1})\rangle - \sum_{i=1}^{k(t)}\langle n_j^{k(t)} f(n_i^{k(t)})\rangle. \end{aligned} \tag{2.8.8}$$

This expression is analogous to equation (2.5.17) and reduces exactly to it if we take $f \equiv r$. As in (2.5.17) we will have to make moment closure approximations in order to derive correspondence with a PDE, only this time they will be complicated by the fact that f is

not constant, but is a function of cell density. We make the approximations

$$\langle n_j^k f(n_i^k)\rangle \approx \langle n_j^k\rangle f(\langle n_i^k\rangle), \tag{2.8.9}$$

and

$$\sum_{i=1}^{k}\langle n_j^k\rangle f(\langle n_i^k\rangle) \approx \sum_{i=1}^{k-1}\langle n_j^{k-1}\rangle f(\langle n_i^{k-1}\rangle), \tag{2.8.10}$$

the latter of which is analogous to assumption (2.5.18) made in the derivation for constant interval-splitting. For a further discussion of the validity of these moment closure approximations see Appendix D in Baker et al. (2010). Using the above assumptions allows us to greatly simplify things so that, using $M_j^{k(t)}$ to denote $\langle n_j^{k(t)}\rangle$, as before, we can rewrite equation (2.8.8) as

$$\begin{aligned}\frac{(k(t)-1)}{k(t)}\frac{\partial M_j^{k(t)-1}}{\partial t} &= \sum_{i=1}^{j-1} M_{j-1}^{k(t)-1} f(M_i^{k(t)-1}) + \sum_{i=j}^{k(t)-1} M_j^{k(t)-1} f(M_i^{k(t)-1}) \\ &- \sum_{i=1}^{k(t)-1} M_j^{k(t)-1} f(M_i^{k(t)-1}) - \frac{1}{2} M_{j-1}^{k(t)-1} f(M_{j-1}^{k(t)-1}) \\ &- \frac{1}{2} M_j^{k(t)-1} f(M_j^{k(t)-1}).\end{aligned} \tag{2.8.11}$$

Using a Taylor expansion we have

$$\begin{aligned}\frac{(k(t)-1)}{k(t)}\frac{\partial M_j^{k(t)-1}}{\partial t} &= -\frac{\partial M^{k(t)-1}}{\partial x}\sum_{i=1}^{j}\Delta x f(M_j^{k(t)-1}) \\ &- M_j^{k(t)-1} f(M^{k(t)-1}) + o(\Delta x).\end{aligned} \tag{2.8.12}$$

Taking the limit as $\Delta x \to 0$ and recognising the summation term in this limit to be an integral we arrive at the PDE

$$\frac{\partial M^{k(t)-1}}{\partial t} + M^{k(t)-1} f(M^{k(t)-1}) + \left(\int_0^x f(M^{k(t)-1}(\hat{x},t))\mathrm{d}\hat{x}\right)\frac{\partial M^{k(t)-1}}{\partial x} = 0, \tag{2.8.13}$$

in the absence of cell movement. We can reintroduce cell movement using the arguments of Section 2.5.1: the cell movement contributions in the RDME will be unrelated to those contributions pertaining to domain growth and hence their continuum approximations can

be derived separately and combined as follows:

$$\frac{\partial M^{k(t)-1}}{\partial t}+M^{k(t)-1}f(M^{k(t)-1})+\left(\int_0^x f(M^{k(t)-1}(\bar{x},t))\mathrm{d}\bar{x}\right)\frac{\partial M^{k(t)-1}}{\partial x}=\frac{\partial\left(\mathcal{J}\left(M^{k(t)-1},s\right)\right)}{\partial x}, \tag{2.8.14}$$

with the usual zero-flux boundary conditions. This PDE is clearly the analogue of equation (2.6.7) derived using continuous considerations in Section 2.6.1.

2.8.2 Diffusion on the linearly density-dependent growing domain

Fig. 2.10 shows a comparison of the results of numerical simulations of the stochastic individual-based models and the corresponding PDE (2.6.7) for domain growth of the form $f(u) = ru$. The correspondence between the individual-based and population-based descriptions appears to be good. The HDE (see Fig 2.10 (e)) exhibits similar dynamics to the case of the exponentially growing domain and the two models also compare well with regards to domain length (*c.f.* Fig. 2.7 (e)).

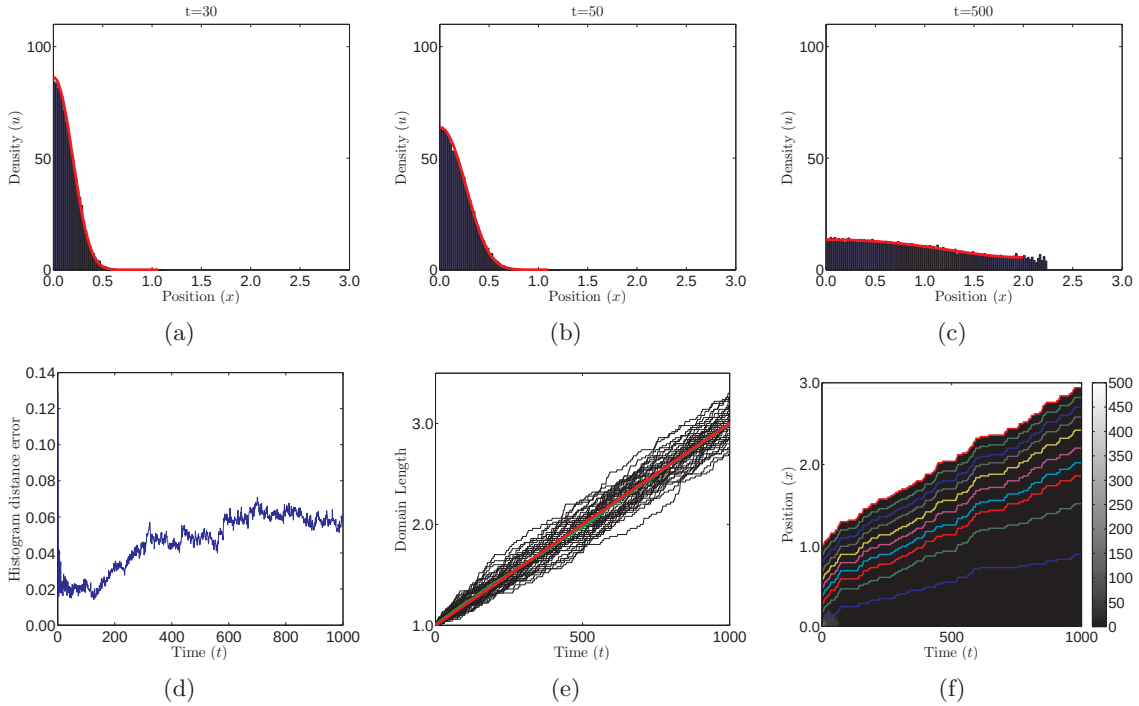


Figure 2.10: *Cells diffusing with constant rate, $D = d(\Delta x)^2$, on a growing domain with domain growth of the form $f = ru$ at several time points. The red curves represent the solution of PDE (2.8.14) with flux operator $\mathcal{J} = D\partial u/\partial x$. Other figure descriptions and initial conditions are as for Fig. 2.7. The parameter values are $k_0 = 50$, $r = 0.0001$, $d = 1.0$ and $\Delta x = 0.02$.*

2.8.3 An explicit formulation of domain length for linear dependence of growth on density

Although in general it is not possible to write down explicitly an expression for domain length in the case of more general density-dependent domain growth, when considering linear dependence of growth on density, $u(x,t)$, we can find an expression for domain length in closed form. Recall equation (2.6.4) from Section 2.6.1:

$$\frac{dL(t)}{dt} = \int_0^{L(t)} f(x,t,u(x,t))\mathrm{d}x.$$

In the case of linear dependence of domain growth on density (and no explicit dependence on space or time), i.e. $f(x,t,u(x,t)) = ru(x,t)$, we have

$$\frac{dL(t)}{dt} = r\int_0^{L(t)} u(x,t)\mathrm{d}x. \tag{2.8.15}$$

Since we have specified zero flux boundary conditions we will have conservation of total cell numbers across the domain and can evaluate the integral using the initial condition, which implies

$$\frac{dL(t)}{dt} = r\mathcal{N}\Delta x, \tag{2.8.16}$$

in terms of the total cell number, $\mathcal{N}$, and the interval width, Δx, of the stochastic simulation. Hence domain length at time t is given by

$$L(t) = L_0 + r\mathcal{N}t\Delta x. \tag{2.8.17}$$

This linear increase in domain length with time can clearly be observed in Fig. 2.10 (e). Despite linear density-dependence of growth on cell density giving a linear increase of domain length with time, this does not mean that the flow $v(x,t)$ is constant across the domain:

$$v(x,t) = r\int_0^x u(\hat{x},t)\mathrm{d}\hat{x}, \tag{2.8.18}$$

for a point x in the domain.

2.8.4 Diffusion on the quadratically density-dependent growing domain

Fig. 2.11 shows the comparison between the numerical simulations of the individual-based model and the population-based model where the domain grows as a quadratic function of the density at the current point. It appears that there is good agreement between the two models in Figs. 2.11 (a)-(c), however, these figures belie the obvious differences in the domain length in the two models as the domain becomes large (see Fig. 2.11 (e)). The disparity between the two models is further corroborated by the evolution of the HDE given in Fig 2.11 (d). By the end of the simulation the HDE is roughly twice as large for quadratic density-dependent growth as it is for linear density-dependent growth. We expect that this is a result of the decreasing validity of the moment closure approximations (equations (2.8.9) and (2.8.10)) combined with the increased importance of stochastic effects as the number of cells per interval becomes smaller as the domain grows. These hypotheses remain to be tested.

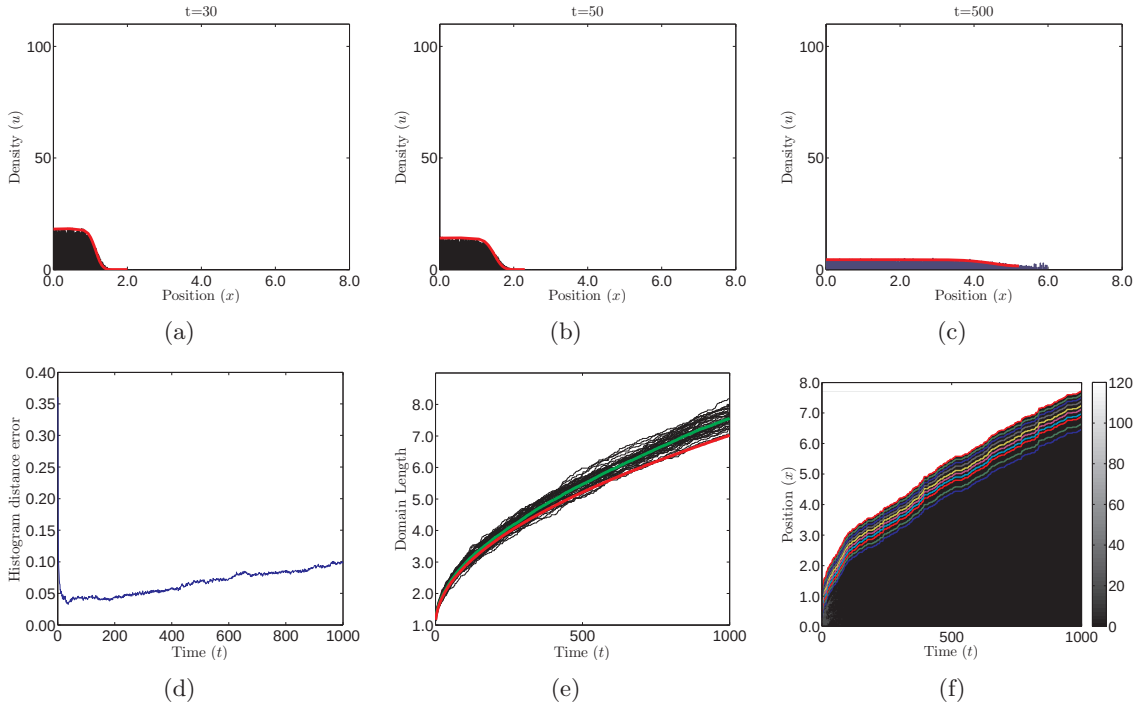


Figure 2.11: *Cells diffusing with constant rate, $D = d(\Delta x)^2$, on a growing domain with domain growth of the form $f = ru^2$ at several time points. The red curves represent the solution of PDE* (2.8.14) *with flux operator $\mathcal{J} = D\partial u/\partial x$. Other figure descriptions and initial conditions are as for Fig. 2.7. The parameter values are $k_0 = 50$, $r = 5 \times 10^{-5}$, $d = 1.0$ and $\Delta x = 0.02$.*

Summary

In this chapter we outlined a stochastic scheme for considering cell movement. In their most basic form the models use constant transition rates for cell movement between intervals in order to simulate diffusion. By introducing and employing an RDME approach we were able to derive equations describing the evolution of the mean cell number in each interval. These equations were linked, by means of Taylor series expansions, to PDEs describing cell density in continuous space for various signal sensing scenarios. Some scenarios were accompanied by corresponding numerical simulations to corroborate our findings. We extended this most basic framework by introducing various biologically motivated extensions to the models.

Motivated by the large number of biological processes in which cell migration on a growing domain is inherent, we incorporated various forms of domain growth into our modelling framework. Initially we considered the simplest case: each interval splitting at the same time-independent rate as every other. We then incorporated the dependence of the rate of interval-splitting on the density of cells in that interval. We were able to find PDE equivalents describing the variation of cell density across the growing domain for each of these stochastic models. We were also able to derive PDE equivalents for stochastic models which include linear and logistic domain growth. In all cases, in order to derive the appropriate correspondences, we were forced to make simplifying, heuristic, moment closure approximations. We tested our results (and hence the validity of our heuristics) with numerical simulations of the discrete and continuum models in all cases and, where the two models appeared to deviate significantly, discussed the possible reasons and implications[11].

This chapter lays the foundations for a modelling framework in which we can segue between discrete and continuum descriptions of cell density in a wide variety of biologically

[11] In the majority of simulations presented in this chapter the agreement between the individual simulations and the solutions of the corresponding PDEs was good. However, there are situations in which we can foresee that the correspondence will break down. We have already seen that non-linearity in the dependence of domain growth on cell density can cause domain lengths in the individual- and population-based models to diverge. In addition, stochastic models in which transition rates depend on cell densities (such as those presented in Section 2.3.2) can lead, in certain circumstances, to steady state cell densities which are vastly different from those predicted by a deterministic 'equivalent' (Anguige and Schmeiser, 2009). Another possible cause of disparity between the discrete and continuum models is steeply varying signalling profiles. The sampling of such a signal profile at finite regularly spaced intervals can mean that the transition rates in the individual model do not accurately represent the movement of cell density predicted by the population-level model.

plausible situations, including on growing domains. We can be confident that the macroscopic models are accurate representations of the microscopic models and hence faithfully adhere to microscopic observations, which the model can be used to represent, in an experimental context.

Although we have attempted to make the model as plausible as possible by incorporating biologically motivated features such as cell sensing and domain growth, we have not taken into account the biological mechanism by which these processes occur. Consider domain growth for example: although it is convenient, mathematically, to allow underlying tissue elements to double in size and divide instantaneously, this is not representative of the mechanisms by which biological tissues increase in size. An incremental interval growth process, more akin to continuum growth, would seem more appropriate. In the next chapter we incorporate such a mechanism for domain growth into our stochastic models. Since the intervals grow incrementally (and independently) they differ in size. This compels us to consider more carefully what it means for a cell to move from one interval to another and thus what the appropriate transition rates between intervals should be. Using the theory of first passage processes we derive appropriate transition rates for cell migration on a non-uniform domain and use these rates to simulate cell migration with incremental domain growth. By considering an altered master equation we demonstrate an improved correspondence between the macroscopic and microscopic models under this stochastic growth regime.

Chapter 3

More appropriate domain growth on a non-uniform domain

Overview

In the previous chapter we laid the foundations for an equivalence framework which allows us to segue between an individual-based, discrete, stochastic model of cell migration and a population-based, continuum, deterministic analogue. We were able to include several biologically motivated features in the framework, including cell proliferation and death, signalling profiles (that interact with and bias the motion of the cells) and, fundamentally, domain growth.

We implemented domain growth in the stochastic model by allowing a lattice-tissue element to instantaneously double in size and divide to form two daughter elements of the same size as the parent. This method of domain growth is unrealistic from a biological point of view: underlying tissue elements grow continuously rather than discretely. From a mathematical point of view, a growth process which is so evidently discrete makes motivating a continuum equivalent difficult. It also causes the domain length to have a large standard deviation so that many stochastic realisations are required in order to compare the individual-level and population-level models accurately (compare Fig. 2.7 (e) of Chapter 2 with its equivalent, Fig. 3.10 (e) in this chapter).

As such, in this chapter, we consider a more appropriate approach to domain growth, both mathematically and biologically. We allow the underlying tissue elements to grow by small increments and to divide upon reaching a pre-defined threshold size, rather than instantaneously doubling in size and dividing (see Fig. 3.1). Incorporating incremental stochastic interval growth necessitates the consideration of tissue elements of different sizes. This will affect transition rates between the elements (since, for example, it will take longer, on average, for a particle exhibiting a random walk to exit a larger element) and cause us to define, more carefully, what we mean by 'cell density' in the individual-based model.

We begin, in Section 3.1, by demonstrating that individual-based stochastic models for cell migration on a non-uniform domain partition are equivalent to PDE models of the same phenomenon on non-growing domains, providing the transition rates (which we derive) are chosen correctly and that we partition the domain in the 'correct' manner. In Section 3.2 we extend this analysis to the case where the domain is allowed to change in size, altering the transition rates as necessary. Through a novel application of the master equation, in Section 3.3 we derive a PDE for cell density on this growing domain and corroborate our findings in Section 3.4 with numerical simulations. We demonstrate that our incremental domain growth process causes, in general, less variation between individual realisations than our previous method and corresponds better to the PDE model.

Movies corresponding to the simulations presented in this chapter can be found at http://people.maths.ox.ac.uk/yatesc/Thesis_movies/Chapter_3.

The majority of the work in this chapter appears and is reprinted with permission from C.A. Yates, R.E. Baker, R. Erban and P.K. Maini. Going from microscopic to macroscopic on non-uniform growing domains. *Physica A*, Under Review, 2011.

3.1 Cell migration on a non-uniform domain

In the previous chapter we considered a stationary (time-independent) domain $x \in [0, 1]$ discretised into k intervals each of length $\Delta x = 1/k$, known as the 'standard interval length'. On this uniform grid, cell *density* scales with cell *number* by the constant factor, Δx, so that

we can use the terms almost interchangeably. However, when considering a non-uniform domain partition the numbers of cells in each interval must be divided by the length of that interval in order to calculate the cell density.

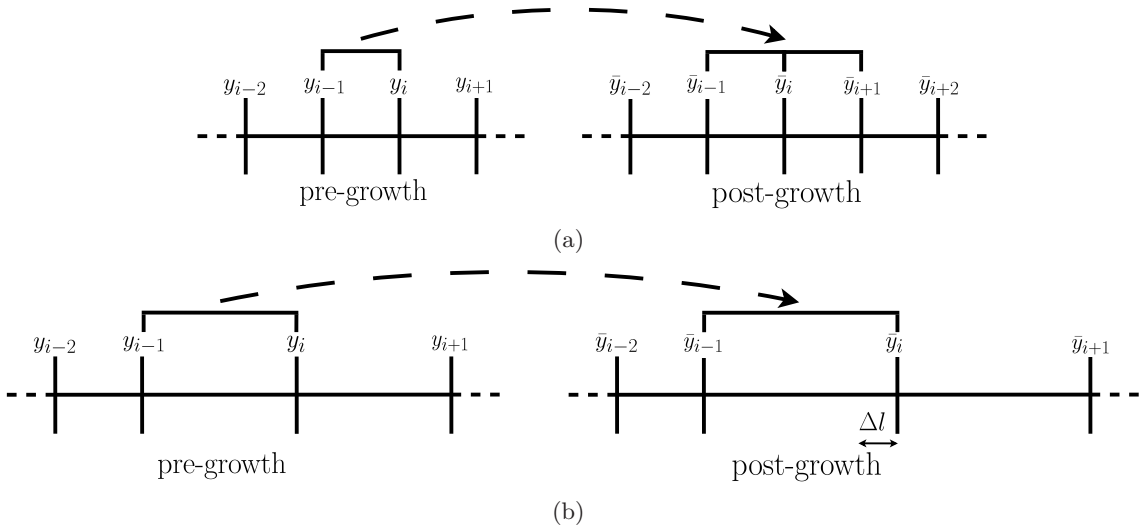


Figure 3.1: *Two different implementations of discrete domain growth. (a) Originally the interval instantaneously doubles and splits. Interval i is selected to grow. This interval doubles in size and splits down the middle creating two daughter intervals of the same size as the original. All intervals with index greater than i are shifted to the right by one interval length, Δx, and their indices are increased by one. All intervals remain the same size as each other. (b) The new incremental growth method. Interval i is selected to grow. The interval is made larger by a small amount, Δl, and all intervals with index greater than i are shifted to the right by Δl. Interval division does not take place until an interval grows to a pre-specified size (see Fig. 3.5). This method introduces inhomogeneity in interval size.*

3.1.1 Cell migration on a stationary, non-uniform grid

Now that we are considering a non-uniform grid we must be careful about how we define our domain. There are two natural ways to do this (see Fig 3.2).

1. Points $x_1, x_2, \ldots, x_k$ are chosen and are associated with each interval, $1, \ldots, k$, respectively. The interval edges, $y_0 = 0$, $y_k = 1$ and $y_i = (x_i + x_{i+1})/2$, for $i = 1, \ldots, k-1$, are then naturally defined in a Voronoi neighbourhood sense: a point on the domain is defined to lie in the interval i if it is nearer to x_i than any other x_j for $j = 1, \ldots, k$, $j \neq i$.

2. The edges of the intervals, $y_0, y_1, \ldots, y_k$, are chosen and the point x_i associated with interval i, for $i = 1, \ldots, k$, is defined to be the centre of that interval (i.e. $x_i = (y_{i-1} + y_i)/2$ for $i = 1, \ldots, k$).

On the uniform grid, these two interval definitions are equivalent, but there is an important distinction to be made on the non-uniform grid: the centres of each interval will no longer correspond to the Voronoi points. The second method of defining intervals gives increased control over the sizes and shapes of the individual intervals but we show that it leads to incorrect cell densities when implementing stochastic simulations (see Appendix B.1). It is, therefore, important to implement the Voronoi property in order to choose transition rates which lead both to the correct cell densities in each interval[1] for stochastic simulations, and to the correct corresponding macroscopic PDE. In what follows we primarily use the first method to partition our domain into intervals (known hereafter as Voronoi partitioning). We will, however, give comparisons of the two partition methods on both fixed and growing domains demonstrating the propriety of the Voronoi partition. In Section 3.2.3 we also discuss when it might be more appropriate to use the interval-centred domain partition to increase our control over the interval sizes.

The Voronoi partition implies that the boundaries for interval i will be at $y_{i-1} = (x_{i-1} + x_i)/2$ (left-hand boundary) and $y_i = (x_i + x_{i+1})/2$ (right-hand boundary). As in the case of the uniform domain, cells are considered to be positioned at $x_1, x_2, \ldots, x_k$ for intervals $1, \ldots, k$, respectively (see Fig. 3.2).

Previously, the scalings of the transition rates for cells to move between intervals have been specified somewhat artificially by considering equations (2.1.4)-(2.1.6), Taylor expanding the terms at $i \pm 1$ and choosing the scaling so that we return to a macroscopic PDE. Now that we are considering a non-uniform domain it is not so simple to see what the requisite scaling for each transition rate should be. In order to ensure correspondence with a macroscale PDE the transition rates should depend on the sizes of the intervals between which a cell moves. By assuming a form for the macroscopic equations, discretising and comparing them to equations (2.1.4)-(2.1.6) it is possible to find the correct transition rates. However, it would be preferable to have a microscale justification of these scalings. The

[1]Given a set of points, $x_1, x_2, \ldots, x_k$, associated with each interval, the number of cells at x_i is independent of how we define the boundaries of the intervals, since (as we will demonstrate) transition rates are only dependent on the distances between neighbouring points. However, when we come to calculating cell densities, the interval boundaries become important and we can show (see Appendix B.1) that the Voronoi domain partition gives smoothly varying cell densities, which correspond to the derived PDE, because the interval sizes are related to the transition rates in a unique way.

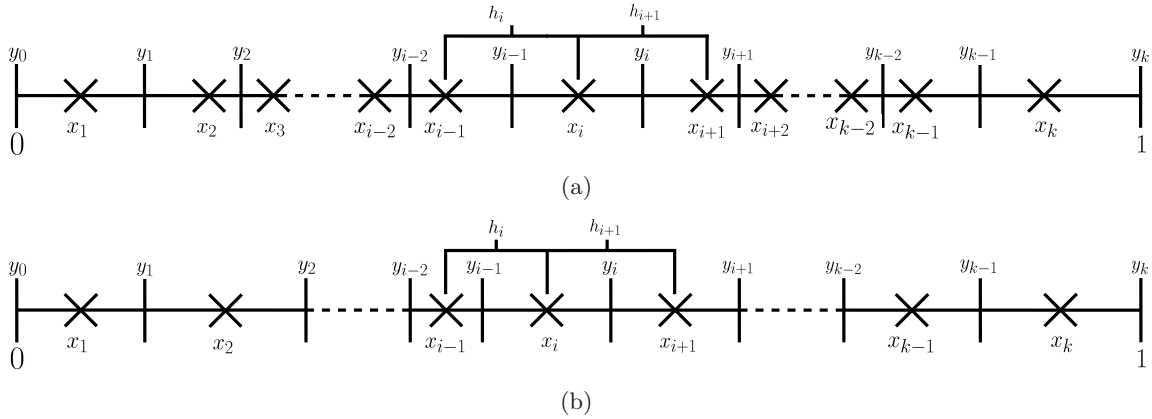


Figure 3.2: *Two different domain partitions. (a) The Voronoi partition method. Cell positions, x_i, are chosen first and interval i is defined to be all the points which lie closer to x_i than any of the other cell positions x_j for $j \neq i$. This gives interval edges at $y_i = (x_i + x_{i+1})/2$. The Voronoi partition method allows for the natural incorporation of transition rates which are inversely proportional to interval size, which cannot be done within the interval-centred framework. (b) The interval-centred definition. Interval boundaries, y_i, are defined first and cells are assumed to lie in the centres of these intervals. In both partitions the distance between neighbouring cell positions, x_i and x_{i-1}, is denoted $h_i = x_i - x_{i-1}$ and, similarly, $h_{i+1} = x_{i+1} - x_i$. For the end intervals indexed 1 and k, h_1 and h_{k+1} are not defined, so we choose $h_1 = 2x_1$ and $h_{k+1} = 2(y_k - x_k)$ for consistency with later results.*

transition rate for a cell at x_i should depend on the distance to the associated domain definition points in the neighbouring intervals, x_{i-1} and x_{i+1} (Engblom et al., 2009). As such, we consider a microscale migration process in the interval $[x_{i-1}, x_{i+1}]$. Initially we will consider this microscale process to be simple Brownian motion and expect to derive transition rates which lead us, via our position jump process, to the diffusion equation on the macroscale. Similar analyses of microscopic processes relating to different signal sensing mechanisms allow derivation of the correct transition rates corresponding to the appropriate macroscale chemotaxis equations.

3.1.2 Deriving transition rates from an underlying microscopic process

A particle moving according to Brownian motion, with position $X(t)$, obeys a stochastic differential equation (SDE)

$$\mathrm{d}X(t) = \sqrt{2D}\mathrm{d}W_t, \tag{3.1.1}$$

where $\mathrm{d}W_t$ is a standard Wiener process and D is the diffusion coefficient of the Fokker-Planck equation (FPE) corresponding to this SDE. The probability density function of the particle, $p(x,t)$, evolves according to the classical diffusion equation:

$$\frac{\partial p(x,t)}{\partial t} = D\frac{\partial^2 p(x,t)}{\partial x^2}. \tag{3.1.2}$$

Given that we know the initial position of the particle, x_i, we have a δ function initial condition in probability, $p(x,0) = \delta(x - x_i)$. Finding the transition rates for the position-jump process reduces to a first passage problem. In order to find the transition rate for moving out of interval i in the position-jump model, we consider the process in which a particle starting at x_i exits the interval $[x_{i-1}, x_{i+1}]$ and find the probability for it to do so at either end. The absorbing boundary conditions $p(x_{i-1},t) = p(x_{i+1},t) = 0$ complete the formulation of the above problem. This is a classic first passage problem, the likes of which are dealt with thoroughly by Redner (2001).

We now briefly summarise the derivation of the correct transition rates and refer the interested reader to Redner (2001) for a more detailed description. We first calculate the probabilities for the particle to leave at either end of the interval, known as the *eventual hitting probabilities*,

$$\varepsilon_-(x_i) = \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}, \tag{3.1.3}$$

$$\varepsilon_+(x_i) = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}. \tag{3.1.4}$$

We next calculate the *conditional mean exit times* to leave the interval at either end,

$$\langle t(x_i)\rangle_- = \frac{(x_i - x_{i-1})(2x_{i+1} - x_i - x_{i-1})}{6D}, \tag{3.1.5}$$

$$\langle t(x_i)\rangle_+ = \frac{(x_{i+1} - x_i)(x_{i+1} + x_i - 2x_{i-1})}{6D}, \tag{3.1.6}$$

which we can use in conjunction with the eventual hitting probabilities (see equations (3.1.3)

and (3.1.4)) to calculate the *unconditional mean exit time* from the interval,

$$\begin{aligned}\langle t(x_i)\rangle &= \varepsilon_-(x_i)\langle t(x_i)\rangle_- + \varepsilon_+(x_i)\langle t(x_i)\rangle_+, &(3.1.7)\\ &= \frac{1}{2D}(x_i - x_{i-1})(x_{i+1} - x_i). &(3.1.8)\end{aligned}$$

We can invert this to give the *unconditional mean exit rate* and by multiplying through by the eventual hitting probabilities calculate the *conditional mean exit rates* or transition rates (for $i = 2, \ldots, k$),

$$\begin{aligned}T_i^- &= \frac{2D}{h_i(h_i + h_{i+1})}, &(3.1.9)\\ T_i^+ &= \frac{2D}{h_{i+1}(h_i + h_{i+1})}, &(3.1.10)\end{aligned}$$

where, for brevity, we denote $h_i = x_i - x_{i-1}$ and similarly $h_{i+1} = x_{i+1} - x_i$. These transition rates correspond to those stated in Engblom et al. (2009).

Recall the equations relating the mean numbers of cells in each interval (2.1.4)-(2.1.6), which we derived from the master equation in Chapter 2:

$$\begin{aligned}\frac{\partial M_1}{\partial t} &= T_2^- M_2 - T_1^+ M_1,\\ \frac{\partial M_i}{\partial t} &= T_{i-1}^+ M_{i-1} - \left(T_i^+ + T_i^-\right) M_i + T_{i+1}^- M_{i+1}, \qquad i = 2, \ldots, k-1,\\ \frac{\partial M_k}{\partial t} &= T_{k-1}^+ M_{k-1} - T_k^- M_k.\end{aligned}$$

Although these equations were derived initially for a uniform mesh, they remain true for the non-uniform mesh. We can re-write this equation in terms of cell densities, $u_i = M_i/l_i$, where l_i is the length of interval i:

$$\begin{aligned}\frac{\partial u_1}{\partial t} &= \frac{1}{l_1}\left(T_2^- u_2 l_2 - T_1^+ u_1 l_1\right), &(3.1.11)\\ \frac{\partial u_i}{\partial t} &= \frac{1}{l_i}\left(T_{i-1}^+ u_{i-1} l_{i-1} - \left(T_i^+ + T_i^-\right) u_i l_i + T_{i+1}^- u_{i+1} l_{i+1}\right), \quad i = 2, \ldots, k-1, &(3.1.12)\\ \frac{\partial u_k}{\partial t} &= \frac{1}{l_k}\left(T_{k-1}^+ u_{k-1} l_{k-1} - T_k^- u_k l_k\right). &(3.1.13)\end{aligned}$$

We can now use Taylor series expansions about position x_i on the appropriate terms. For

example,

$$u_{i+1} = u(x_{i+1}) = u(x_i) + (h_{i+1})\frac{\partial u}{\partial x}(x_i) + \frac{1}{2}(h_{i+1})^2\frac{\partial^2 u}{\partial x^2}(x_i) + \dots. \tag{3.1.14}$$

Allowing the number of domain elements, k, to tend to infinity on the Voronoi domain partition, implying $h_i, h_{i+1} \to 0$, we obtain the diffusion equation for cell density $u(x,t)$:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}, \quad \text{for} \quad (x,t) \in [0,1] \times [0,\infty), \tag{3.1.15}$$

with the usual zero flux boundary conditions. For a more detailed derivation and a justification of the necessity of the Voronoi domain partition see Appendix B.1.

Fig. 3.3 shows a numerical comparison of the stochastic simulations and the derived PDE (3.1.15). Qualitatively, the PDE for cell density matches the cell density of the stochastic simulations well. In Fig. 3.3 (f), in order to give a more quantitative comparison of the two simulation types, we have plotted the variation of the HDE between the stochastic cell density and the PDE over time. Since our cell densities are defined at non-regular lattice points, x_i, we interpolate the value of the PDE at each lattice point in order to compare the curves. Fig. 3.3 (f) shows a low HDE (of comparable magnitude to the HDE for cell migration on the uniform domain, *c.f.* Fig. 2.2 (f)) for the duration of the simulation, indicating good agreement between the two simulation types.

3.1.3 Transition rates for the local sensing equation

An advantage of the formalism used by Redner (2001) to derive the transition rates is that the corresponding transition rates for non-diffusive processes can be derived with only small adjustments to the method. The probability density of cells moving according to local signal sensing in an underlying microscopic stochastic model obeys an FPE of the form

$$\frac{\partial p(x,t)}{\partial t} = D^l\frac{\partial^2(s(x)p(x,t))}{\partial x^2}. \tag{3.1.16}$$

We have reintroduced a time-independent signalling profile, $s(x)$, which cells can sense and respond to, as in Chapter 2. The superscript l on the diffusion coefficient, D^l, alludes to the

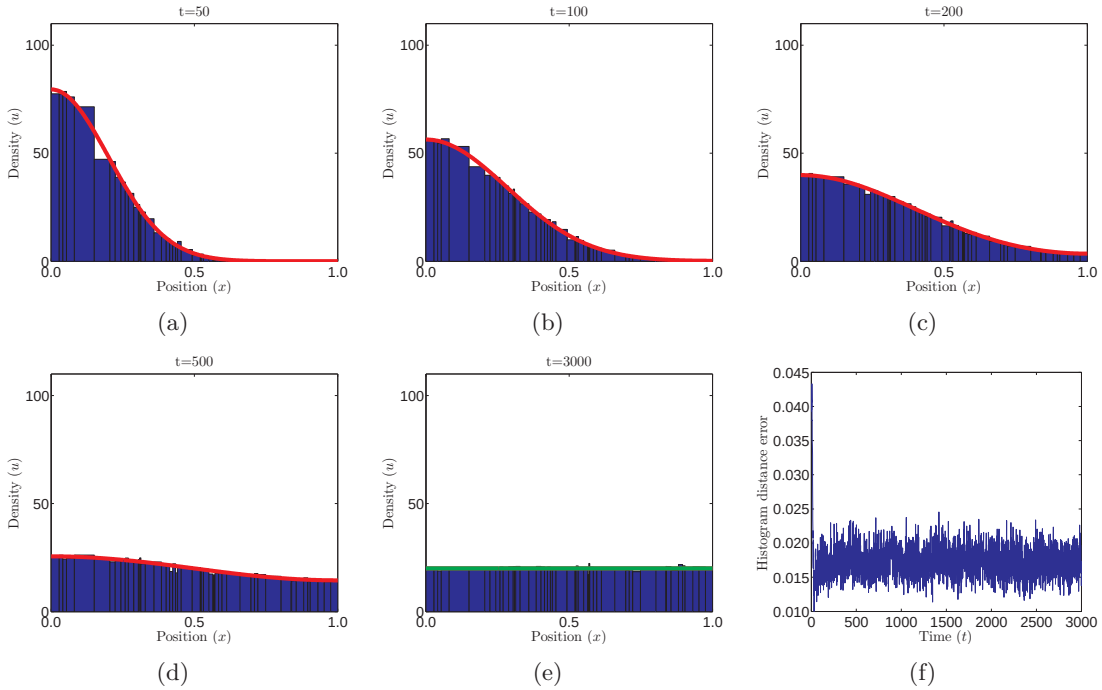


Figure 3.3: *Cells diffusing with constant rate at several time points. (a)-(e) The histograms represent an average of 40 stochastic realisations of the system with transition rates given by equations* (3.1.9) *and* (3.1.10)*. The red curves represent the numerical solution of the PDE* (3.1.15) *and the green curve in (e) represents the steady state solution of the PDE found analytically. (f) The evolution of the HDE between the stochastic simulations and the PDE. All* $\mathcal{N} = 1000$ *cells are released from the Voronoi point associated with the first interval. The macroscopic diffusion coefficient takes the value* $D = \Delta x^2$*, where* $\Delta x = 1/k$ *and* $k = 50$*.*

fact that this underlying microscopic process corresponds to *local* sensing in the position jump model and, as we will show, also to the macroscopic *local* sensing PDE (2.2.3) derived in Chapter 2.

As a first approximation, assume the value of signal sensed across each interval to be a piecewise constant function of space. In the local sensing case we will assume that the level of signal that can be sensed throughout the interval corresponds to the value of the signalling profile at x_i, the initial position of the particle. A simple argument leads us to the conclusion that, if $s(x_i)$ is constant across the interval, then we can find the transition rates for local sensing by substituting $D^l s(x_i)$ for D in the diffusion transition rates (3.1.9)

and (3.1.10). This gives the following transition rates

$$T_i^- = \frac{2D^l s(x_i)}{h_i(h_i + h_{i+1})}, \tag{3.1.17}$$

$$T_i^+ = \frac{2D^l s(x_i)}{h_{i+1}(h_i + h_{i+1})}, \tag{3.1.18}$$

which replicate the correct macroscopic PDE (2.2.3) (upon Taylor expansion and taking appropriate limits) when applied to equations (2.1.4)-(2.1.6). Its agreement with both microscopic and macroscopic processes provides confirmation that our position jump model with these transition rates is a consistent description of cell migration with local signal sensing.

Transition rates for the non-local and average sensing cases can also be derived given the corresponding microscopic SDE (results not shown). These transition rates can also be shown to be consistent with the macroscopic PDEs for the appropriate sensing mechanism, as presented in Chapter 2.

3.1.4 Using a mixed boundary interval to derive transition rates for the end intervals

In order to implement zero flux boundary conditions on the macroscopic domain we designate the left-hand end of the first interval and the right-hand end of the last interval, at zero and one, respectively, to be reflecting boundaries. We can carry out a similar analysis to that above to find the correct transition rates from an underlying microscopic process. Without loss of generality we will assume that we are considering the right-hand end interval, $[y_{k-1},y_k]$, of the position-jump process. In order to derive the transition rates we consider the behaviour of a particle initially at x_k on the domain $[x_{k-1},y_k]$ where we impose an absorbing boundary condition at $x = x_{k-1}$ and a reflecting boundary condition at $x = y_k = 1$. These conditions impose the reflecting boundary condition at the right-hand end of the domain, but allow cells to leave interval k to the left as with other (interior) intervals.

Employing a similar method as was used in the case of two absorbing boundaries (Red-

ner, 2001), we can calculate the conditional hitting probabilities at $x = x_{k-1}$ and $x = 1$ to be

$$\varepsilon_-(x_k) = 1, \qquad \varepsilon_+(x_k) = 0, \tag{3.1.19}$$

which implies we are certain to exit the interval at the left-hand boundary, which is to be expected. We can calculate the conditional mean exit time at the left-hand boundary (as in Berg, 1993) as

$$\langle t(x_k)\rangle_- = \frac{1}{2D}(2(y_k - x_{k-1})h_k - h_k^2). \tag{3.1.20}$$

Rearranging gives the transition rate as[2]

$$\begin{aligned} T_k^- &= \frac{2D}{h_k(2(y_k - x_{k-1}) - x_k + x_{k-1})}, \\ &= \frac{2D}{h_k(h_k + h_{k+1})}. \end{aligned} \tag{3.1.21}$$

In an entirely analogous manner we can derive the transition rate to move right out of the first interval as

$$\begin{aligned} T_1^+ &= \frac{2D}{h_2\,(x_2 + x_1)}, \\ &= \frac{2D}{h_2\,(h_1 + h_2)}. \end{aligned} \tag{3.1.22}$$

On the uniform domain these transition rates reduce to

$$T_k^- = T_1^+ = \frac{D}{h^2}, \tag{3.1.23}$$

as we might reasonably expect.

3.1.5 Comparison with the interval-centred domain partition

In order to demonstrate the propriety of the Voronoi domain partition in comparison to the interval-centred domain partition we have carried out similar simulations to those detailed

[2]Recall the definitions $h_1 = 2x_1$ and $h_{k+1} = 2(y_k - x_k)$.

in Fig. 3.3 with the interval-centred domain partition. The transition rates are the same as those derived in previous subsections (see equations (3.1.9) and (3.1.10) in Section 3.1.2 and (3.1.21) and (3.1.22) in Section 3.1.4) only now the positions x_i, where cells are assumed to reside, are the centres of the corresponding intervals i (for $i = 1, \ldots, k$). It is evident from glancing at Fig. 3.4 (a)-(e) that the cell densities deviate significantly from the mean-field description for the vast majority of the intervals. This is reinforced quantitatively by considering Fig. 3.4 (f): for the majority of the simulation the HDE takes values which are more than an order of magnitude larger than those in the analogous Voronoi partition simulations.

Initialising each repeat of the stochastic simulation with the same interval-centred partition gives aberrant cell densities. However, if the domain partition is different for each repeat of the simulation then the errors can balance themselves out leading to an improved comparison to the cell density given by the PDE. We give further details of this phenomenon for the stationary domain in Appendix B.2 and for the growing domain in Section 3.4.

3.2 Cell migration with domain growth

In Chapter 2, growth was implemented using interval-splitting, where an interval doubles in length and divides instantaneously. In order to represent the growth of the underlying tissue elements more realistically it is preferable to allow intervals to grow by small increments (more akin to a continual growth process) and to divide upon reaching a pre-determined size. Different methods for this more incremental domain growth description are detailed below.

3.2.1 Deterministic domain growth

Each time a cell moves, we allow each of the intervals to grow an amount proportional to its length. This produces exponential domain growth of each interval and hence exponential domain growth of the whole domain. We allow intervals to grow to roughly (given the stochastic nature of the time stepping) twice the standard interval size (i.e. to $2\Delta x$) before dividing. Upon division, an interval splits into two equally sized daughter intervals and

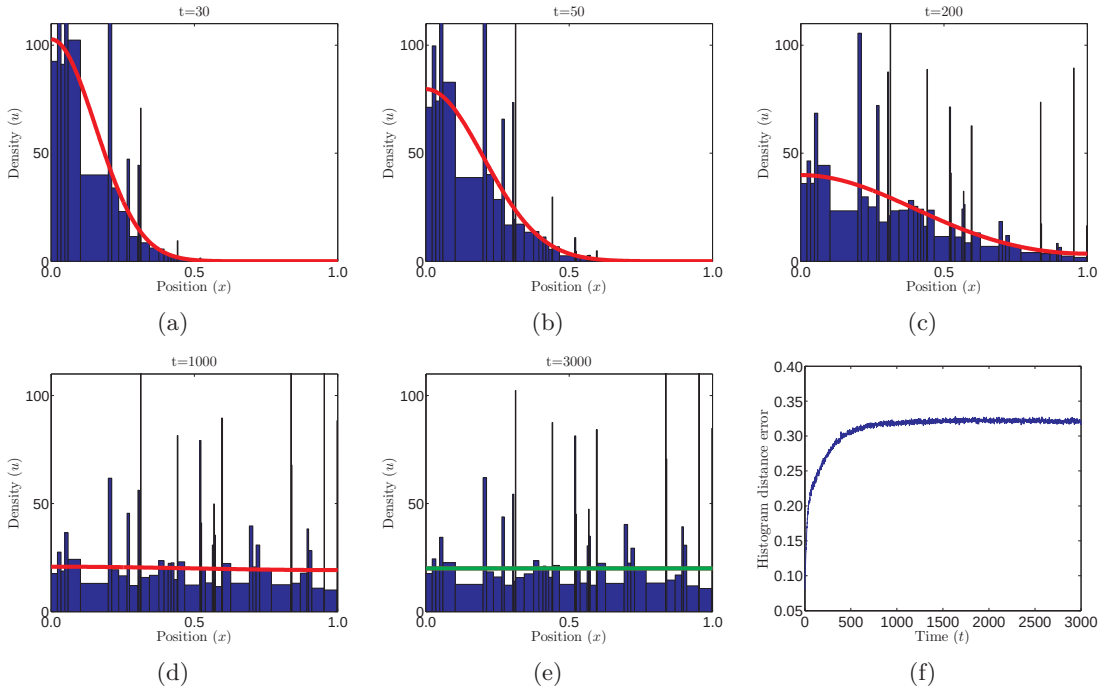


Figure 3.4: *Cells undergoing simple diffusion at several time points on an interval-centred domain partition. Panel descriptions and initial and boundary conditions are as in Fig. 3.3. With the microscopically derived transition rates given by equations* (3.1.9) *and* (3.1.10)*, which correspond to the macroscopic PDE, the interval-centred description of cell migration leads to cell densities which do not match the mean-field description. In some of the panels we have cut off the tops of some of the histograms representing cell density so that a detailed comparison of the PDE and stochastic cell density can still be seen.*

the cells which resided in that interval are divided randomly between these two 'daughter' intervals using a number drawn from a binomial random variable, $\mathcal{B}(n_i, 0.5)$, where n_i is the number of cells in parent interval i. We call this a 'division event'.

In this regime, if all intervals are initialised with the same length then they will continue to have the same length as each other for the duration of the simulation and, as such, we are not required to consider non-uniform domain partitions. Since all intervals are initialised with the same size and grow at the same rate, division of intervals will be synchronous and simple to implement.

However, if we allow initial inhomogeneity in the interval sizes we must divide the domain using the Voronoi partition so that allowing cell movement according to the transition rates derived above gives cell densities in the stochastic simulations which match those of the mean-field description (see Appendix B.1). Allowing all intervals to grow proportionally to their length, l_i, at each time-step, as above, maintains the Voronoi property of the domain.

The main issue, then, is how to appropriately repartition the domain once an interval has become large enough to divide (a 'division event'). Ideally we would divide the parent interval into two equally sized daughter intervals, each half the length of the original parent interval. Unfortunately, in general, it is not possible to do this whilst maintaining the Voronoi partition. In order to maintain the Voronoi nature of the partition it is necessary to redefine the boundaries of the two neighbouring intervals (see Fig. 3.5). We are, however, at least able to choose daughter intervals that are of equal sizes.

Begin by considering interior intervals. We can express these restrictions mathematically in terms of the positions of the pre-division Voronoi point of dividing interval i, x_i, those of the two neighbouring intervals, x_{i-1} and x_{i+1}, and the positions of the Voronoi points after division $\bar{x}_{i-1}$, $\bar{x}_i$, $\bar{x}_{i+1}$, $\bar{x}_{i+2}$. We have relabelled the Voronoi points on the post-division domain with an 'over-bar' in order to avoid confusion in what follows (see Fig. 3.5).

In order to ensure that only the intervals $i-1$, i and $i+1$ are affected by the division event we must ensure that the Voronoi points x_{i-1} and x_{i+1} remain unchanged i.e. $x_{i-1} = \bar{x}_{i-1}$ and $x_{i+1} = \bar{x}_{i+2}$. We can express the condition that the two daughter intervals should be of the same size as

$$2l = \bar{x}_{i+1} - \bar{x}_{i-1} = \bar{x}_{i+2} - \bar{x}_i, \tag{3.2.1}$$

where l is the length of the daughter intervals. We must also maintain the strict ordering of the post-division Voronoi points:

$$\bar{x}_{i-1} < \bar{x}_i < \bar{x}_{i+1} < \bar{x}_{i+2}. \tag{3.2.2}$$

These inequalities can be shown (after manipulation) to bound the length of the daughter intervals above and below:

$$\frac{\bar{x}_{i+2} - \bar{x}_{i-1}}{4} < l < \frac{\bar{x}_{i+2} - \bar{x}_{i-1}}{2}. \tag{3.2.3}$$

Choosing the daughter intervals to be half as long as the parent interval, $l = (\bar{x}_{i+2} - \bar{x}_{i-1})/4$, would mean choosing the two new Voronoi points to be coincident, $\bar{x}_i = \bar{x}_{i+1}$. At the other extreme, choosing each daughter interval to be the same length as the parent interval,

$l = (\bar{x}_{i+2} - \bar{x}_{i-1})/2$, requires each new point to be coincident with an already existing point, $\bar{x}_i = \bar{x}_{i-1}$ and $\bar{x}_{i+1} = \bar{x}_{i+2}$. Neither of these extreme situations is acceptable. An unbiased choice, therefore, would be

$$l = \frac{3}{8}\left(\bar{x}_{i+2} - \bar{x}_{i-1}\right). \tag{3.2.4}$$

This choice fully determines the position of the new Voronoi points (see Fig. 3.5):

$$\bar{x}_i = \frac{3}{4}\bar{x}_{i-1} + \frac{1}{4}\bar{x}_{i+2}, \tag{3.2.5}$$

$$\bar{x}_{i+1} = \frac{1}{4}\bar{x}_{i-1} + \frac{3}{4}\bar{x}_{i+2}. \tag{3.2.6}$$

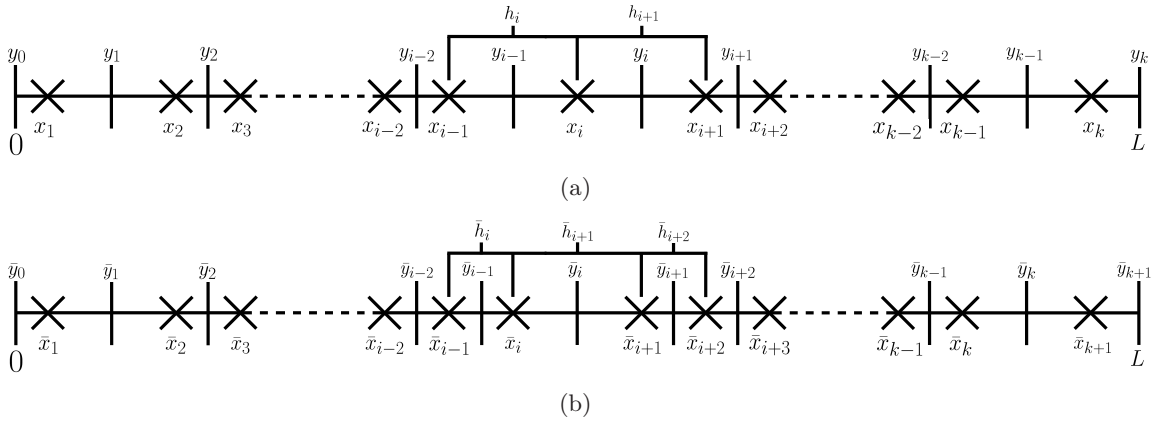


Figure 3.5: *The implementation of (interior) interval division whilst maintaining the Voronoi partition. Interval i becomes large enough to divide due to a growth event. (a) The pre-division domain. (b) The domain after the division event and the consequential repartitioning of the affected intervals. It should be noted that the domain does not grow during a division event.*

We must be careful when dividing the end intervals, 1 and k. In the case of the first interval, for example, we do not need to determine the left-hand end of the interval in relation to another Voronoi point. This boundary is fixed ($\bar{y}_0 = y_0 = 0$). The right-hand end Voronoi point, $x_2 = \bar{x}_3$ also remains fixed. Clearly, we still require the ordering condition of the post-division Voronoi points:

$$0 < \bar{x}_1 < \bar{x}_2 < \bar{x}_3. \tag{3.2.7}$$

Again the two daughter intervals are chosen to be the same size as each other, but, as an additional constraint, the first new Voronoi point, $\bar{x}_1$, is chosen to lie in the centre of the

first interval, $[\bar{y}_0, \bar{y}_1]$. This prescribes that the second Voronoi point, $\bar{x}_2$, must also lie in the centre of the second interval, $[\bar{y}_1, \bar{y}_2]$ (since the intervals are chosen to be of the same length). This determines positions of the new Voronoi points

$$\bar{x}_1 = \frac{\bar{x}_3}{5}, \tag{3.2.8}$$

$$\bar{x}_2 = \frac{3\bar{x}_3}{5}, \tag{3.2.9}$$

and hence the repartitioning of the domain upon splitting of the first interval. In an analogous manner, when we split interval k the new Voronoi points must depend on the distance between the end of the domain ($y_k = \bar{y}_{k+1} = L$) and the last unaltered Voronoi point, $x_{k-1} = \bar{x}_{k-1}$, in the following way:

$$\bar{x}_k = \frac{3\bar{x}_{k-1}}{5} + \frac{2\bar{y}_{k+1}}{5}, \tag{3.2.10}$$

$$\bar{x}_{k+1} = \frac{\bar{x}_{k-1}}{5} + \frac{4\bar{y}_{k+1}}{5}. \tag{3.2.11}$$

For the purpose of cell redistribution upon splitting we can assume that cells are distributed evenly across each interval. When interval boundaries are redrawn upon the splitting of interval i the number of cells in the new intervals are chosen as close as possible (given the integer nature of cell numbers) to

$$\bar{n}_{i-1} = \left(\frac{\bar{y}_{i-1} - y_{i-2}}{y_{i-1} - y_{i-2}}\right) n_{i-1}, \tag{3.2.12}$$

$$\bar{n}_i = \left(\frac{y_{i-1} - \bar{y}_{i-1}}{y_{i-1} - y_{i-2}}\right) n_{i-1} + \left(\frac{\bar{y}_i - y_{i-1}}{y_i - y_{i-1}}\right) n_i, \tag{3.2.13}$$

$$\bar{n}_{i+1} = \left(\frac{y_i - \bar{y}_i}{y_i - y_{i-1}}\right) n_i + \left(\frac{\bar{y}_{i+1} - y_i}{y_{i+1} - y_i}\right) n_{i+1}, \tag{3.2.14}$$

$$\bar{n}_{i+2} = \left(\frac{y_{i+1} - \bar{y}_{i+1}}{y_{i+1} - y_i}\right) n_{i+1}. \tag{3.2.15}$$

The numbers of cells in each new interval can be found by summing the proportion of each original interval which lies in that new interval multiplied by the number of cells in the corresponding original interval. Alternatively, for each original interval j ($j = i - 1, i, i + 1$, where interval i is the interval chosen to split) we can draw a random integer, m, between 0 and n_j from a binomial distribution with parameters $N = n_j$ and $p =$

$(\bar{y}_j - y_{j-1}) \,/\, (y_j - y_{j-1})$. We allow m cells to remain in the new interval $\bar{j}$ and the remaining $(n_j - m)$ cells will be redistributed to the new interval $\bar{j} + 1$ (see Fig. 3.6). In this way, although cell redistribution will be random, on average the cells will be redistributed as in the proportional splitting method described above. As in Chapter 2, the cell redistribution method can be shown to have minimal effect on the model results and, as such, in all simulations that follow we employ the binomial redistribution method.

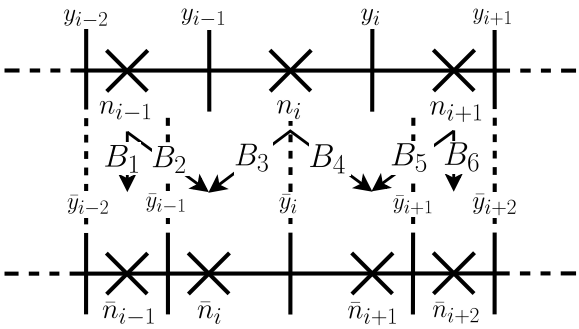


Figure 3.6: *Cell redistribution after interval division. Each cell is redistributed to an interval on the post-division domain with probability proportional to the overlap between the new intervals and the old intervals using random numbers drawn from a binomial distribution:* $B_1 = \mathcal{B}\,(n_{i-1}, (\bar{y}_{i-1} - y_{i-2})/(y_{i-1} - y_{i-2}))$, $B_2 = n_{i-1} - B_1$, $B_3 = \mathcal{B}\,(n_i, (\bar{y}_i - y_{i-1})/(y_i - y_{i-1}))$, $B_4 = n_i - B_3$, $B_5 = \mathcal{B}\,(n_{i+1}, (\bar{y}_{i+1} - y_i)/(y_{i+1} - y_i))$ *and* $B_6 = n_{i+1} - B_5$.

On the interval-centred domain division is much simpler. A new interval boundary, $\bar{y}_i$, is drawn at position x_i, the centre of the interval that is dividing, and new interval centres are defined at $\bar{x}_i = (y_{i-1} + x_i)/2 = (\bar{y}_{i-1} + \bar{y}_i)/2$ and $\bar{x}_{i+1} = (y_i + x_i)/2 = (\bar{y}_i + \bar{y}_{i+1})/2$. All interval centres and edges to the right of the interval that is dividing are relabelled by increasing their index by one i.e. $x_j = \bar{x}_{j+1}$ and $y_j = \bar{y}_{j+1}$ for $j = i + 1, \ldots, k$. The n_i cells that previously resided in interval i are redistributed evenly into the two daughter intervals, in a manner analogous to that used in Chapter 2.

3.2.2 Stochastic domain growth

An alternative method for implementing domain growth is to allow intervals to grow in pairs[3]. We call this a 'growth event'. As in Chapter 2, growth events are implemented as any other 'reaction' in the stochastic simulation algorithm, with propensity functions

[3]Intervals must grow in pairs in order to preserve the Voronoi property of the domain. However, when considering the interval-centred domain partition it is possible, and indeed preferable, to allow intervals to grow individually.

proportional to their probability of occurrence. Growth must be implemented carefully in order to preserve the Voronoi property of the domain. Intervals i and $i+1$ are chosen to grow with a probability proportional (with constant of proportionality, r) to the size of interval i: $l_i = y_i - y_{i-1}$. All the Voronoi points to the right of x_i ($x_{i+1}, \ldots, x_k$) and interval boundaries to the right of x_{i+1} ($y_{1+1}, \ldots, y_k$) move by a constant amount Δl to the right, where Δl is some small fraction[4] of the standard interval length, Δx. The boundary, y_i, is then re-drawn $\Delta l/2$ to the right of its original position. Thus growth causes intervals i and $i+1$ to grow by $\Delta l/2$ each (see Fig. 3.7). The only exception to this rule is for the right-hand-most interval of the domain where we can simply move the right-hand boundary of the interval to the right by Δl. Upon growing to a predetermined size, Δx_{split}, intervals are divided and cells redistributed in the same manner as described above. By considering a master equation for the domain length, $L(t)$, on the Voronoi domain partition we can show that, on average, this process leads to exponential domain growth; $L(t) = \exp(r\Delta l t)$ (but not to the exponential growth of each interval).

If we are considering an interval-centred domain partition (which may be justified in some circumstances, see Appendices B.2 and B.3) we can implement domain growth in a much more straightforward manner. Growth of interval i occurs with rate proportional to its length, l_i. Interval edges to the right of the growing interval (y_j for $j = i+1, \ldots, k$) are shifted to the right by Δl and point x_i is shifted to the right by $\Delta l/2$ in order to preserve its position in the centre of interval i. Only one interval changes in size. By considering a master equation for domain length (see Section 3.2.3) we can show that each interval (and hence the whole domain) grows exponentially.

3.2.3 A master equation for domain length on the interval-centred domain partition with intervals growing stochastically

For ease of working consider the interval-centred domain partition, although we can also derive similar results for the Voronoi partition. It is possible to repartition the interval-centred domain in order to calculate the cell densities which correspond to the mean-field

[4] Δl is defined such that, when taking the continuum limit, the ratio $\Delta l/\Delta x \ll 1$ remains constant as $\Delta x \to 0$ so that terms that are of order Δl^2 may still be neglected in comparison to terms that are of order l_j, the length of the j^{th} interval (see Section 3.3.3).

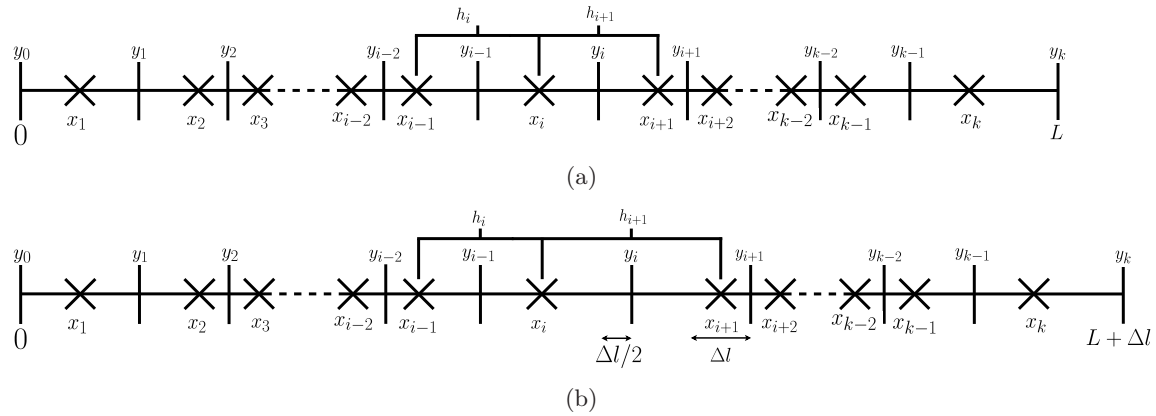


Figure 3.7: *The implementation of individual domain growth on the Voronoi domain partition. Interval i is selected to grow with probability proportional to its length, l_i. (a) The pre-growth domain. (b) The domain after the implementation of the growth of intervals i and $i+1$, each by $\Delta l/2$ as described above.*

PDE and, as such, the interval-centred domain partition may be as valid as the Voronoi domain partition for simulating cell migration on growing domains (see Appendix B.3).

Consider a time interval small enough that the probability of more than one growth event occurring in $[t, t+\delta t)$ is $O(\delta t)$ and ignore, for the meantime, the movement of cells. Define $\boldsymbol{L} = (L_1, \ldots, L_k)$ to be the vector of random variables representing the length of each interval. We express the rate of change of the probability that the j^{th} component of $\boldsymbol{L}$, representing the length of interval j, takes the value l_j at time t via the following master equation:

$$\frac{d\Pr(\boldsymbol{L}=\boldsymbol{l},t)}{dt} = r\sum_{i=1}^{k}\Pr\left(\boldsymbol{L}=(l_1,\ldots,l_i-\Delta l,\ldots,l_k),t\right)(l_i-\Delta l) - r\sum_{i=1}^{k} l_i \Pr(\boldsymbol{L}=\boldsymbol{l},t), \tag{3.2.16}$$

where $\boldsymbol{l} = (l_1, \ldots, l_k)$ is the current state of the vector of random variables representing the length of each interval. In order to find the mean length of interval j we multiply through by l_j and sum over all the (finite number of) possible values that $\boldsymbol{l}$ can take. Upon simplification we arrive at an ODE which describes how the mean of each interval grows:

$$\frac{d\langle l_j\rangle}{dt} = r\Delta l\langle l_j\rangle \quad \text{for} \quad j=1,\ldots,k. \tag{3.2.17}$$

We can solve this ODE to show that each interval grows exponentially:

$$\langle l_j \rangle (t) = l_j(0) \exp(r\Delta l t) \quad \text{for} \quad j = 1, \ldots, k, \tag{3.2.18}$$

where $l_j(0)$ is the initial size of interval j. By taking the sum of ODEs (3.2.17) we can arrive at an ODE for the average domain length, $\langle L \rangle$:

$$\frac{d \langle L \rangle}{dt} = r\Delta l \langle L \rangle , \tag{3.2.19}$$

which can also be trivially solved to show that the domain length increases exponentially:

$$\langle L \rangle (t) = L(0) \exp(r\Delta l t), \tag{3.2.20}$$

where $L(0) = \sum_{j=1}^{k} l_j(0)$ is the original length of the domain.

3.3 Derivation of the PDE for growth from the master equation

First we introduce some notation: n_j will denote the *density* of cells in interval j after the growth event and N_j will denote the *number* of cells in interval j after the growth event[5]. l_j will denote the post-growth length of interval j. Clearly these three quantities are related by

$$n_j = \frac{N_j}{l_j}, \tag{3.3.1}$$

for each post-growth interval, j.

3.3.1 A conceptual point

In order to derive an equation for the mean density of cells in interval j, n_j, we must multiply through the associated master equation (see Section 3.3.3, below) by n_j, and sum over all the possible values that n_j can take. Since we have defined n_j as being the density

[5]The *number* of cells in interval j before and after growth events will be the same for $j = 1, \ldots, k$ since growth events do not change the number of cells at each point.

of cells in interval j on the post-growth domain, it is important that we express all other terms pertaining to cell density in terms of densities on the post-growth domain. As such we would like to repartition the pre-growth cell densities from the pre-growth domain partition to the post-growth domain partition. In order to do this we assume that cells are distributed evenly across each interval.

3.3.2 Pre-growth densities on the post-growth domain partition

To repartition cells from the pre-growth domain to the post-growth domain we must first repartition the pre-growth cell numbers on the pre-growth domain partition, denoted by the vector $\boldsymbol{N}$, to the post-growth domain partition. This repartitioned vector will be denoted $\boldsymbol{N}'$. Since the pre- and post-growth domain partitions are the same up to the interval i, which grows, we have $N'_j = N_j$ for $j < i$. Since interval i grows, the number of cells in the pre-growth domain that lie in the post-growth interval i is $N'_i = N_i + N_{i+1}\Delta l/l_{i+1}$. This corresponds to all the cells that lie in the pre-growth interval i added to the fraction $\Delta l/l_{i+1}$ of the cells which lie in the pre-growth interval $i+1$, N_{i+1} (see Fig. 3.8). This fraction is the amount of overlap between the post-growth interval i and the pre-growth interval $i+1$ and our repartitioning assumes cells are spread homogeneously across each interval. In a similar manner, the number of cells in the pre-growth domain that lie in post-growth interval j, for $j > i$, is $N'_j = N_j\,(1 - \Delta l/l_j) + (\Delta l/l_{j+1})\,N_{j+1}$. The factor $\Delta l/l_j$ corresponds to the fraction of cells in pre-growth interval j lost to post-growth interval $j-1$ (hence $1 - \Delta l/l_j$ corresponds to the fraction of cells in *pre*-growth interval j that are also in *post*-growth interval j). Correspondingly $\Delta l/l_{j+1}$ is the number of cells requisitioned by *post-growth* interval j from *pre*-growth interval $j+1$ (see Fig. 3.8). Finally, the number of cells that lie in post-growth interval k is $N'_k = N_k\,(1 - \Delta l/l_k)$, where $1 - \Delta l/l_k$ represents the overlap fraction of pre- and post-growth intervals, k.

Now that we have repartitioned the cell *numbers* to the post-growth domain partition, it is a simple matter to find the pre-growth *densities* on the post-growth domain partition, q_j. For all $j < i$ we have

$$q_j = \frac{N_j}{l_j} = n_j, \tag{3.3.2}$$

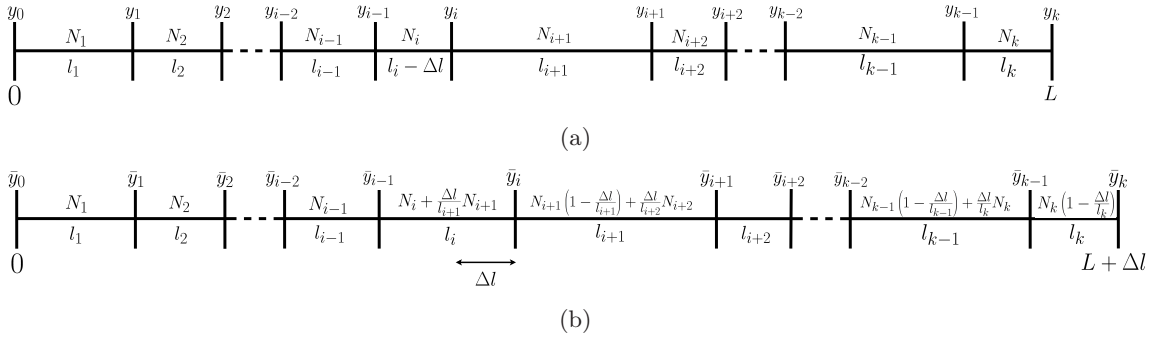


Figure 3.8: *The pre-growth cell numbers partitioned on (a) the pre-growth domain and (b) the post-growth domain. The number of cells in each interval is given in the upper half of the interval and the length of that interval is given in the lower half. Repartitioning the cell numbers as in (b) allows us to easily write down the pre-growth cell densities on the post-growth domain partition.*

since the boundaries of these intervals are the same on the post- and pre-growth domain partition (see Fig. 3.8). For $j = i$:

$$q_i = \frac{N_i + N_{i+1}\left(\Delta l / l_{i+1}\right)}{l_i} = n_i + \frac{\Delta l}{l_i} n_{i+1}. \tag{3.3.3}$$

Similarly, if $i < j < k$ then we can write the pre-growth densities on the post-growth domain partition, q_j, associated with interval j as

$$q_j = \frac{N_j\left(1 - \Delta l / l_j\right) + \left(\Delta l / l_{j+1}\right) N_{j+1}}{l_j} = n_j\left(1 - \Delta l / l_j\right) + \frac{\Delta l}{l_j} n_{j+1}. \tag{3.3.4}$$

Finally for the last interval, k, the density will become

$$q_k = \frac{N_k\left(1 - \Delta l / l_k\right)}{l_k} = n_k\left(1 - \Delta l / l_k\right). \tag{3.3.5}$$

3.3.3 The master equation

Now that we have formulated the pre-growth densities in terms of the post-growth densities on the post-growth grid we are in a position to consider the master equation for domain growth. Consider the probability of having cell density vector, $\mathbf{n}$, and interval length vector, $\mathbf{l}$, at time $t + \delta t$ where the time increment δt is small enough that the probability of more than one growth event occurring is $O(\delta t)$. Ignoring, for the meantime, cell movement and the effects of signal sensing, and considering all the possible ways we could have arrived in

this state space from time t provides us with a formulation of the master equation for cell density on a growing domain:

$$\begin{aligned}\frac{\partial \Pr(\boldsymbol{n},\boldsymbol{l},t)}{\partial t} &= r\sum_{i=1}^{k}\Pr(q_1,\ldots,q_i,q_{i+1},\ldots,q_k,l_1,\ldots,l_i-\Delta l,\ldots,l_k,t)\,(l_i-\Delta l)\\ &- r\sum_{i=1}^{k}\Pr(\boldsymbol{n},\boldsymbol{l},t)l_i. \end{aligned} \tag{3.3.6}$$

Multiplying through by the cell density in interval j, n_j, and summing over all the possible values that cell density can take[6] we arrive at

$$\begin{aligned}\frac{\partial \langle n_j\rangle}{\partial t} &= r\sum_{i=1}^{k}\sum_{\boldsymbol{n}} n_j\Pr(q_1,\ldots,q_i,q_{i+1},\ldots,q_k,l_1,\ldots,l_i-\Delta l,\ldots,l_k,t)\,(l_i-\Delta l)\\ &- r\sum_{i=1}^{k}\langle n_j l_i\rangle, \end{aligned} \tag{3.3.7}$$

where $\langle n_j\rangle = \sum_{\boldsymbol{n}} n_j \Pr(\boldsymbol{n},\boldsymbol{l},t)$ represents the mean density of cells in interval j and $\langle n_j l_i\rangle = \sum_{\boldsymbol{n}} n_j l_i \Pr(\boldsymbol{n},\boldsymbol{l},t)$. For the first sum on the RHS of equation (3.3.7) we must carefully consider each value of i. For $i > j$ (i.e. the splitting box is to the right of box j) the terms are of the form

$$r\sum_{\boldsymbol{n}} n_j\Pr(q_1,\ldots,q_i,q_{i+1},\ldots,q_k,l_1,\ldots,l_i-\Delta l,\ldots,l_k,t)\,(l_i-\Delta l) = r\langle n_j l_i\rangle, \tag{3.3.8}$$

since $q_j = n_j$ for $j < i$. However, if $i \le j$ then the growth event occurs at, or to the left of, interval j and things are not so simple.

For $j = i$:

$$\begin{aligned} & r\sum_{\boldsymbol{n}} n_j\Pr\Big(n_1,\ldots,n_{j-1},n_j+\frac{\Delta l}{l_j}n_{j+1},n_{j+1}(1-\Delta l/l_{j+1})+\frac{\Delta l}{l_{j+1}}n_{j+2},\ldots,\\ & n_k(1-\Delta l/l_k),l_1,\ldots,l_j-\Delta l,\ldots,l_k,t\Big)(l_j-\Delta l)\\ = \; & r\sum_{\boldsymbol{n}}\left(n_j+\frac{\Delta l}{l_j}n_{j+1}\right)\Pr\Big(n_1,\ldots,n_{j-1},n_j+\frac{\Delta l}{l_j}n_{j+1},n_{j+1}(1-\Delta l/l_{j+1})+\frac{\Delta l}{l_{j+1}}n_{j+2},\\ & \ldots,n_k(1-\Delta l/l_k),l_1,\ldots,l_j-\Delta l,\ldots,l_k,t\Big)(l_j-\Delta l)\end{aligned}$$

[6]The values that cell density can take are determined by the number of cells in the interval and the width of that interval. We use the shorthand notation, $\sum_{\boldsymbol{n}}$, to represent the double sum over all possible cell numbers and all possible interval widths, $\sum_{\boldsymbol{N}}\sum_{\boldsymbol{l}}$.

$$
\begin{aligned}
- \quad & r\sum_{\boldsymbol{n}} \frac{\Delta l}{l_j} n_{j+1} \Pr\Big(n_1, \ldots, n_{j-1}, n_j + \frac{\Delta l}{l_j} n_{j+1}, n_{j+1}\left(1 - \Delta l/l_{j+1}\right) + \frac{\Delta l}{l_{j+1}} n_{j+2}, \ldots, \\
& n_k \left(1 - \Delta l/l_k\right), l_1, \ldots, l_j - \Delta l, \ldots, l_k, t \Big) (l_j - \Delta l) \\
= \quad & r\left\langle n_j l_j \right\rangle - r\sum_{\boldsymbol{n}} \frac{\Delta l}{l_j} \left(\frac{l_{j+1}}{l_{j+1} - \Delta l} \right) \left\{ n_{j+1}\left(1 - \Delta l/l_{j+1}\right) + \frac{\Delta l}{l_{j+1}} n_{j+2} \right\} \Pr\Big(n_1, \ldots, n_{j-1}, \\
& n_j + \frac{\Delta l}{l_j} n_{j+1}, n_{j+1}\left(1 - \Delta l/l_{j+1}\right) + \frac{\Delta l}{l_{j+1}} n_{j+2}, \ldots, n_k \left(1 - \Delta l/l_k\right), l_1, \ldots, l_j - \Delta l, \\
& \ldots, l_k, t \Big) (l_j - \Delta l) + O(\Delta l^2) \\
= \quad & r\left\langle n_j l_j \right\rangle - r\Delta l \left\langle n_{j+1} \right\rangle + O(\Delta l^2). \qquad (3.3.9)
\end{aligned}
$$

To go between the first and second equality in this statement we have added and subtracted terms that are $O(\Delta l^2)$, namely:

$$
r\sum_{\boldsymbol{n}} \frac{\Delta l}{l_j} \left(\frac{l_{j+1}}{l_{j+1} - \Delta l} \right) \frac{\Delta l}{l_{j+1}} n_{j+2} \Pr(\ldots) \left(l_j - \Delta l\right), \qquad (3.3.10)
$$

where the term we have added is incorporated into the $O(\Delta l^2)$ term. To go between the second and third equalities we have used the Taylor expansion of $1/(l_{j+1} - \Delta l)$ and grouped all $O(\Delta l^2)$ terms together:

$$
\frac{1}{l_{j+1} - \Delta l} = \frac{1}{l_{j+1}} + \frac{\Delta l}{(l_{j+1})^2} + O(\Delta l^2), \qquad (3.3.11)
$$

so that $\Delta l(l_{j+1}/(l_{j+1} - \Delta l))$ may be approximated by $\Delta l + O(\Delta l^2)$, and made use of the following Taylor expansion of $1/l_j$:

$$
\frac{1}{l_j} = \frac{1}{(l_j - \Delta l) + \Delta l} = \frac{1}{l_j - \Delta l} - \frac{\Delta l}{\left(l_j - \Delta l\right)^2} + O(\Delta l^2), \qquad (3.3.12)
$$

so that $\Delta l/l_j$ may be approximated as $\Delta l/\left(l_j - \Delta l\right) + O(\Delta l^2)$.

In general, for $1 \leq i < j$ all terms will take a similar form:

$$
\begin{aligned}
& r\sum_{\boldsymbol{n}} n_j \Pr\Big(q_1, \ldots, q_{j-1}, n_j\left(1 - \Delta l/l_j\right) + \frac{\Delta l}{l_j} n_{j+1}, n_{j+1}\left(1 - \Delta l/l_{j+1}\right) + \frac{\Delta l}{l_{j+1}} n_{j+2}, \ldots, \\
& n_k \left(1 - \Delta l/l_k\right), l_1, \ldots, l_i - \Delta l, \ldots, l_k, t \Big) (l_i - \Delta l)
\end{aligned}
$$

$$
\begin{aligned}
&= r\sum_{\boldsymbol{n}}\left(\frac{l_j}{l_j-\Delta l}\right)\left(n_j\left(1-\frac{\Delta l}{l_j}\right)+\frac{\Delta l}{l_j}n_{j+1}\right)\Pr\Big(q_1,\ldots,q_{j-1},n_j\left(1-\Delta l/l_j\right)+\frac{\Delta l}{l_j}n_{j+1},\\
&\quad n_{j+1}\left(1-\Delta l/l_{j+1}\right)+\frac{\Delta l}{l_{j+1}}n_{j+2},\ldots,n_k\left(1-\Delta l/l_k\right),l_1,\ldots,l_i-\Delta l,\ldots,l_k,t\Big)(l_i-\Delta l)\\
&- r\sum_{\boldsymbol{n}}\left(\frac{\Delta l}{l_j-\Delta l}\right)n_{j+1}\Pr\Big(q_1,\ldots,q_{j-1},n_j\left(1-\Delta l/l_j\right)+\frac{\Delta l}{l_j}n_{j+1},\\
&\quad n_{j+1}\left(1-\Delta l/l_{j+1}\right)+\frac{\Delta l}{l_{j+1}}n_{j+2},\ldots,n_k\left(1-\Delta l/l_k\right),l_1,\ldots,l_i-\Delta l,\ldots,l_k,t\Big)(l_i-\Delta l)\\
&= r\sum_{\boldsymbol{n}}\left(1+\frac{\Delta l}{l_j-\Delta l}\right)\left(n_j\left(1-\Delta l/l_j\right)+\frac{\Delta l}{l_j}n_{j+1}\right)\Pr\Big(q_1,\ldots,q_{j-1},\\
&\quad n_j\left(1-\Delta l/l_j\right)+\frac{\Delta l}{l_j}n_{j+1},n_{j+1}\left(1-\Delta l/l_{j+1}\right)+\frac{\Delta l}{l_{j+1}}n_{j+2},\ldots,n_k\left(1-\Delta l/l_k\right),\\
&\quad l_1,\ldots,l_i-\Delta l,\ldots,l_k,t\Big)(l_i-\Delta l)\\
&- r\sum_{\boldsymbol{n}}\left(\frac{\Delta l}{l_j-\Delta l}\right)\left(\frac{l_{j+1}}{l_{j+1}-\Delta l}\right)\left\{n_{j+1}\left(1-\Delta l/l_{j+1}\right)+\frac{\Delta l}{l_{j+1}}n_{j+2}\right\}\Pr\Big(q_1,\ldots,q_{j-1},\\
&\quad n_j\left(1-\Delta l/l_j\right)+\frac{\Delta l}{l_j}n_{j+1},n_{j+1}\left(1-\Delta l/l_{j+1}\right)+\frac{\Delta l}{l_{j+1}}n_{j+2},\ldots,n_k\left(1-\Delta l/l_k\right),l_1,\\
&\quad \ldots,l_i-\Delta l,\ldots,l_k,t\Big)(l_i-\Delta l)+O(\Delta l^2)\\
&= r\left\langle n_jl_i\right\rangle+r\Delta l\left\langle\frac{n_jl_i}{l_j}\right\rangle-r\Delta l\left\langle\frac{n_{j+1}l_i}{l_j}\right\rangle+O(\Delta l^2). \qquad (3.3.13)
\end{aligned}
$$

Again, to arrive at the last equality we have used a Taylor expansion of $1/(l_j-\Delta l)$:

$$
\frac{1}{l_j-\Delta l}=\frac{1}{l_j}+\Delta l\frac{1}{(l_j)^2}+O(\Delta l^2), \qquad (3.3.14)
$$

and the Taylor expansion of $1/(l_{j+1}-\Delta l)$ from the case $i=j$, above.

We substitute the expressions from equations (3.3.8), (3.3.9) and (3.3.13) into the RHS of the master equation (3.3.7):

$$
\begin{aligned}
\frac{\partial\left\langle n_j\right\rangle}{\partial t} &= r\sum_{i=1}^{k}\left\langle n_jl_i\right\rangle-r\sum_{i=1}^{k}\left\langle n_jl_i\right\rangle\\
&+ r\Delta l\sum_{i=1}^{j-1}\left\langle\frac{n_jl_i}{l_j}\right\rangle-\left\langle\frac{n_{j+1}l_i}{l_j}\right\rangle\\
&+ r\Delta l\left\langle\frac{n_jl_j}{l_j}\right\rangle-r\Delta l\left\langle\frac{n_{j+1}l_j}{l_j}\right\rangle\\
&- r\Delta l\left\langle n_j\right\rangle+O(\Delta l^2), \qquad (3.3.15)
\end{aligned}
$$

where we have added and subtracted $r\Delta l \langle n_j l_j / l_j \rangle = r\Delta l \langle n_j \rangle$ to the RHS. This can be written more simply as:

$$\frac{\partial \langle n_j \rangle}{\partial t} = -r\Delta l \langle n_j \rangle - r\Delta l \sum_{i=1}^{j} \left(\left\langle \frac{n_{j+1} l_i}{l_j} \right\rangle - \left\langle \frac{n_j l_i}{l_j} \right\rangle \right). \tag{3.3.16}$$

We make a moment closure approximation in order to make these terms manageable:

$$\left\langle \frac{n_j l_i}{l_j} \right\rangle = \frac{\langle n_j \rangle \langle l_i \rangle}{\langle l_j \rangle}. \tag{3.3.17}$$

We justify the use of this moment closure approximation using numerical simulations in Section 3.4.3. Upon applying this moment closure approximation we can re-write equation (3.3.16) as

$$\frac{\partial \langle n_j \rangle}{\partial t} = -r\Delta l \langle n_j \rangle - xr\Delta l \frac{\langle n_{j+1} \rangle - \langle n_j \rangle}{\langle l_j \rangle}, \tag{3.3.18}$$

where we have recognised that $\sum_{i=1}^{j} \langle l_i \rangle = x$. Taylor expanding the term $\langle n_{j+1} \rangle$ and taking the limit of the interval sizes tending to zero[7] i.e. $l_j \to 0$ yields a PDE formulation:

$$\frac{\partial \langle n \rangle}{\partial t} = D \frac{\partial^2 \langle n \rangle}{\partial x^2} - r\Delta l \langle n \rangle - xr\Delta l \frac{\partial \langle n \rangle}{\partial x}, \tag{3.3.19}$$

where we have reintroduced terms due to diffusion since their inclusion does not affect the derivation of the domain growth terms.

These are precisely the terms we would expect, as per equation (1.3.1), when considering exponential domain growth. The term $r\Delta l \langle n \rangle$ corresponds to local dilution due to volume increase and $xr\Delta l \partial \langle n \rangle / \partial x$ corresponds to material being transported along the domain by the flow induced by growth.

Here by defining mean cell density in interval j to be $\langle n_j \rangle = \langle N_j / l_j \rangle$ we have derived a PDE corresponding to the individual-level model. It should be noted that it may be simpler to derive a PDE for the mean number of cells in each interval divided by the mean interval length $\langle N_j \rangle / \langle l_j \rangle$, which can in some senses be thought to be the mean density. Such a derivation for the interval-centred domain partition is given in Appendix B.5.

[7] In order to maintain a constant rate of domain growth, as $l_j \to 0$, we allow $\Delta l \to 0$ in such a way that $r\Delta l$ remains constant.

The derivation of the PDE on the Voronoi domain partition is similar to that of the PDE on the interval-centred domain partition, illustrated above, and as such is not presented. The only differences come when considering intervals i and $i+1$ which both grow by $\Delta l/2$, as opposed to the case above where only interval i grows (by Δl). As a result the macroscopic terms derived for the Voronoi domain partition are not exactly the same as for the interval-centred partition. This is because the domain growth is not uniformly exponential across each interval of the Voronoi domain partition.

3.4 Numerical simulations

To corroborate our findings we have carried out extensive numerical simulations using different types of signal sensing, domain growth mechanisms and interval partitions.

We note the difficulty in representing average cell densities over several repeats: in the stochastic implementations of domain growth the partition of the domain in each of the repeat simulations will be different. In the deterministic implementations of domain growth (see Section 3.2.1) all the intervals grow at the same rate, implying that the domain partition should be the same at all the sampled time points. However, due to the stochastic nature of the cell movement and the time-step chosen for each reaction, simulations cannot be sampled with exactly the same domain partition. These factors require us to rethink what we mean by average cell density. To circumvent this problem we define a uniform mesh over the domain and redistribute cell density into these regions at time-points where we wish to record cell density. This relies on the assumption that cells are spread homogeneously across each interval.

3.4.1 Cells diffusing on an initially uniform domain partition with deterministic exponential interval growth

In Fig. 3.9 we implement deterministic domain growth on an initially uniform domain partition, as described in Section 3.2.1. In this case the interval-centred and Voronoi domain partitions are equivalent. Cell densities in both the stochastic and PDE models are found

to match well, as evidenced quantitatively by the evolution of the HDE[8] with time (see Fig. 3.9 (d)). The domain in the individual-level model grows, on average, at the same rate as the domain of the population-level model, as exhibited in Fig. 3.9 (e). This is what we expect for the deterministic implementation of domain growth. In Fig. 3.9 (f) each interval tracked progresses in an exponential manner as the domain grows. A slight jump is seen at around $t \approx 700$ when all the boxes divide simultaneously and we continue to track the right-hand-most of the daughter intervals. The HDE in Fig. 3.9 (d) is low and remains so for the duration of the simulation. For all the numerical simulations presented in this section we have found that the HDE is at least as small as its counterpart simulated using the instantaneous interval-splitting and doubling of Chapter 2 (*c.f.* Fig. 2.7 (d) in Section 2.5). In some cases it is significantly lower, especially later in simulations when a significant number of interval division events have occurred.

3.4.2 Cells diffusing on an initially non-uniform domain partition with stochastic interval growth

In Fig. 3.10 we implement stochastic domain growth on an initially non-uniform *Voronoi* domain partition, as described in Sections 3.2.1 and 3.2.2. The average stochastic domain grows at a similar rate to the deterministic domain, as exhibited in Fig. 3.10 (e). This also shows that the variation in the individual realisations of the stochastic domain is reduced in comparison to the interval-splitting and doubling approach (*c.f.* Fig. 2.7 (f)). Both the deterministic and stochastic domains exhibit exponential growth, as expected. In Fig. 3.10 (f) each interval tracked progresses in an exponential manner as the domain grows with occasional jumps relating to interval division events. The evolution of the HDE with time (see Fig. 3.10 (d)) demonstrates the similarity of the two model types throughout the simulations.

For comparison, in Fig. 3.11, we show the stochastic evolution of cell density on an initially non-uniform *interval-centred* domain partition. For each repeat of the simulation

[8] In all the HDE figures when cells are initialised in the first interval the error starts off large as an artefact of the way we are forced to initialise our PDE to replicate this condition. When we initialise with an exponentially decaying profile of cells we do not see this initial disparity since this initial profile is straightforward to replicate in the continuum model.

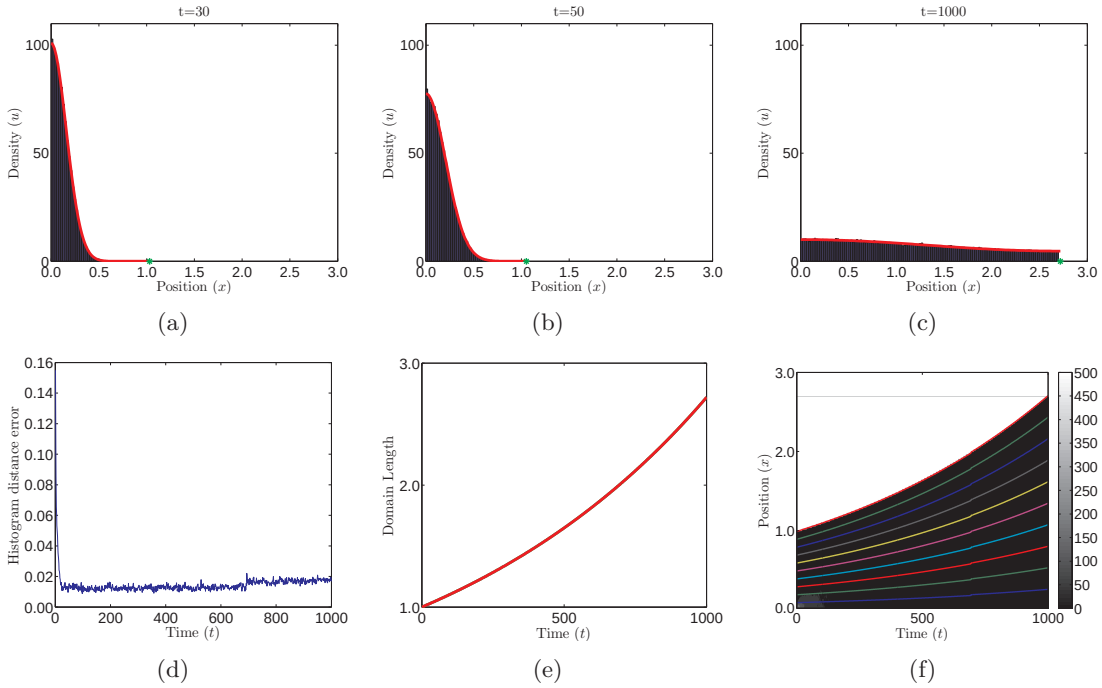


Figure 3.9: *Cells undergoing simple diffusion at several time points. (a)-(c) The histograms represent an average of 40 stochastic realisations. The underlying stochastic model has an initially uniform domain partition and undergoes deterministic growth. The solid lines exhibit the result of numerical simulation of the corresponding diffusion equation on an exponentially growing domain using the* NAG *routine* `D03PE` *(The Numerical Algorithms Group, 2009). The boundaries are assumed to be reflecting. The green dot on the x-axis represents the average stochastic domain length. (d) Evolution of the HDE with time. (e) Evolution of the average stochastic domain length plotted in green with the deterministic PDE domain length overlaid in red. Due to the deterministic implementation of domain growth in the individual-based model, the green curve lies exactly underneath the red curve and cannot be seen. (f) The progression of every 5th interval of the initial 50. Upon division we track the right-hand-most of the daughter intervals. The coloured lines indicate the trajectories of these ten boxes and the shading the average cell density (black low, white high). All cells are initialised in the first interval. The transition rates are given by equations* (3.1.9) *and* (3.1.10)*. Parameters are as follows: $k_0 = 50$, $\Delta x = 1/k_0 = 0.02$, $D = \Delta x^2$, $r = 0.0001$, $\Delta x_{split} = 2\Delta x = 0.04$.*

the same initial partition is used. The stochastic cell density gives a fair comparison to the PDE, but there are clearly discrepancies in the cell density at several points. The randomness of the growth process ensures a smoothing of the average stochastic cell density, which is why the differences between the stochastic and continuum models are not as stark as in the case of the interval-centred partition on the stationary domain. However, if the initial partitions are different for each repeat of the stochastic simulation we see that the stochastic cell density is an even better match to the PDE and the HDE is correspondingly reduced (see Fig. B.5 in Appendix B.4). The behaviour of the system averaged over many repeats is different to the behaviour we would expect to see from a single simulation. A related

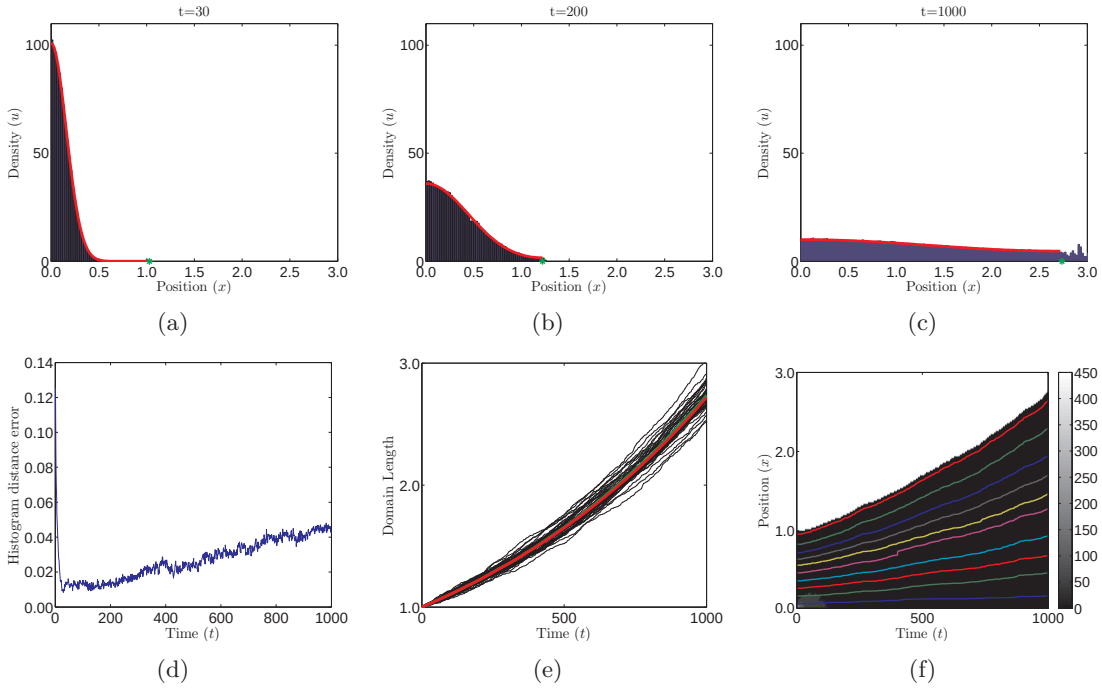


Figure 3.10: *Cells undergoing diffusion at several time points. The underlying stochastic model has an initially non-uniform Voronoi domain partition and undergoes stochastic interval growth. Figure descriptions are as in Fig. 3.9. In (e) the green curve representing the evolution of the average stochastic domain length is just visible underneath the red curve which corresponds to the deterministic PDE domain length. All cells are initialised in the first interval. Parameters and transition rates are as in Fig. 3.9 with* $\Delta l = 0.1 \times \Delta x = 0.002$ *in addition.*

phenomenon has also been observed when modelling exclusion processes for discrete agents undergoing random walks on a two-dimensional lattice (Simpson et al., 2009b; Spitzer, 1970). Although the interval-centred domain partition appears to perform quite well, the Voronoi domain partition has a lower HDE for the majority of the simulation, due to the erroneous calculation of the cell densities for each of the individual simulations on the interval-centred domain partition.

3.4.3 Justification of the moment closure approximation

In order to derive the equivalence between the stochastic and continuum descriptions of cell migration it is necessary to make the moment closure approximation of equation (3.3.17). We attempt to justify this moment closure approximation using the results of our stochastic simulations by plotting

$$\left(\left\langle \frac{n_j l_i}{l_j} \right\rangle - \frac{\langle n_j \rangle \langle l_i \rangle}{\langle l_j \rangle} \right) \Big/ \left\langle \frac{n_j l_i}{l_j} \right\rangle, \tag{3.4.1}$$

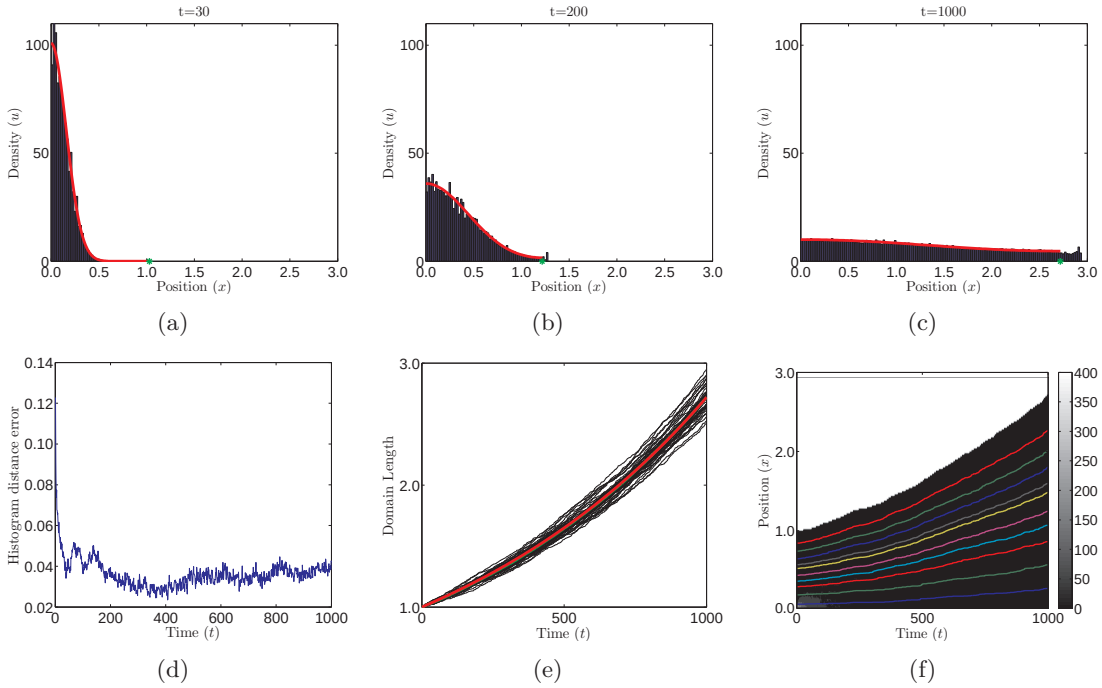


Figure 3.11: *Cells undergoing diffusion at several time points. The underlying stochastic model has an initially non-uniform interval-centred domain partition and undergoes stochastic interval growth. All repeats are initialised with the same domain partition. Figure descriptions are as in Figs. 3.9 and 3.10. All cells are initialised in the first interval. Parameters and transition rates are as in Fig. 3.10.*

for all values of i and j. Fig. 3.12 shows the results of our investigations of the diffusion process for both the Voronoi and interval-centred domain partitions. In both simulations the domains are initially *uniform* and become non-uniform as a result of the stochastic growth of the domain. In both cases the moment closure approximation is extremely accurate throughout the field although, in general, the approximation is more accurate for the Voronoi partition (see Figs. 3.12 (a) and (b)) than for the interval-centred partition (see Figs. 3.12 (c) and (d)).

Fig. 3.13 shows the results for diffusion with an initially *non-uniform* domain partition for both the interval-centred and Voronoi partitions. In both cases the moment closure approximation is less accurate than for the initially uniform domain. This is because an initially non-uniform domain is likely to give rise to small intervals. These intervals tend to grow less often than other intervals and remain small throughout the simulation as a consequence. Small intervals, in comparison to larger intervals, have larger fluctuations in cell density (as a consequence of the integer nature of cell numbers) and thus represent the

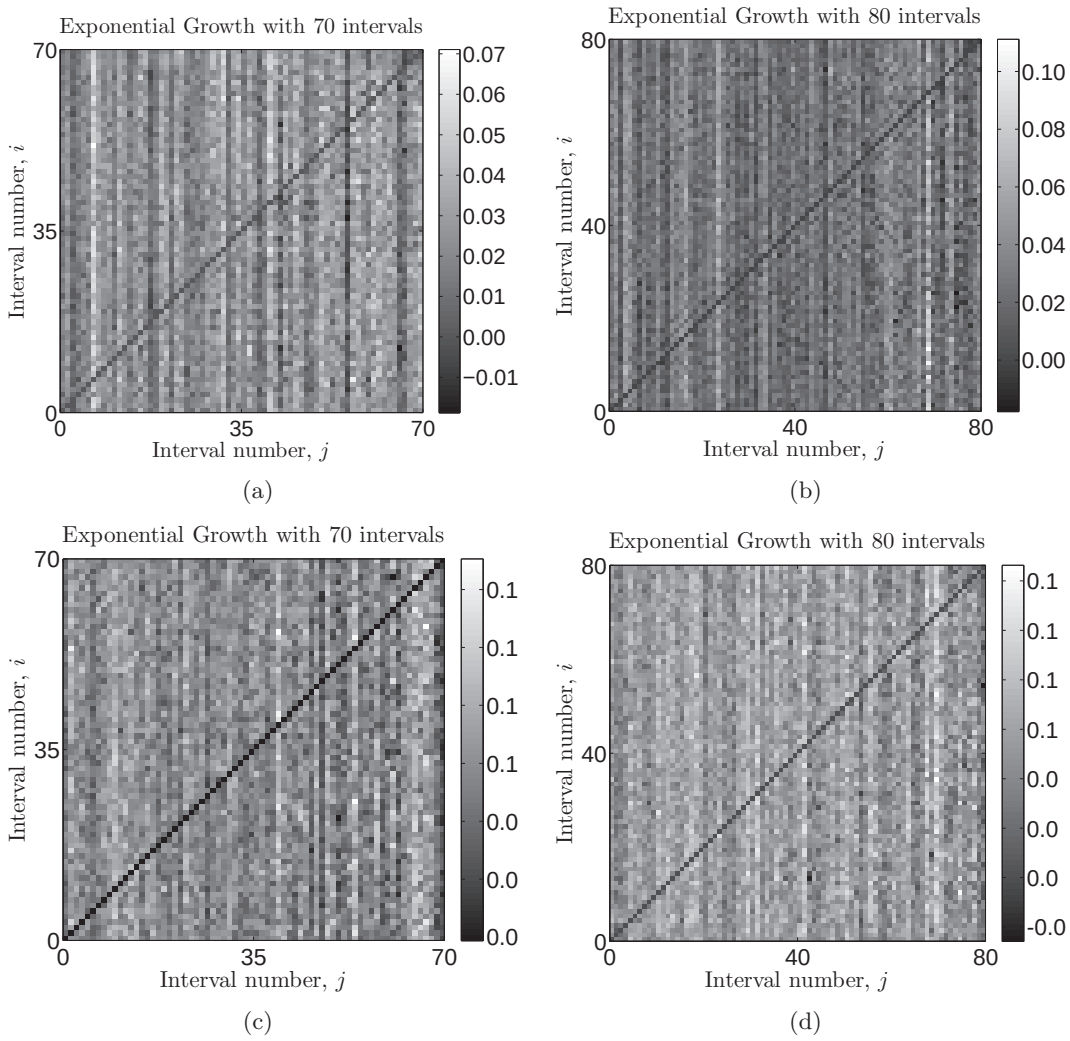


Figure 3.12: *Justification of the moment closure approximation. (a) and (b) show, for two different numbers of intervals, the relative error between the LHS and RHS of equation* (3.3.17)*, given by equation* (3.4.1)*, for an initially uniform domain which subsequently maintains a Voronoi partition. (c) and (d) show the same quantity for the same numbers of intervals, again with an initially uniform domain partition, but with an interval-centred domain partition being maintained throughout the simulations. In all of the figures a dark band can be seen along the diagonal (i.e.* $i = j$*). For these values of* i *and* j *the values of* l_i *and* l_j *are the same, as are the values of* $\langle l_i \rangle$ *and* $\langle l_j \rangle$*. This means that the value of the relative error given by equation* (3.4.1) *is zero.*

mean-field cell density less well. In Figs. 3.13 (a) and (b) the moment closure approximation is sound. In the case of the interval-centred partition small intervals have erroneously high cell numbers which distort the values for the average density and hence the moment closure approximations are generally slightly worse, but still good. There are light vertical bands of intervals in Figs. 3.13 (c) and (d) for which the moment closure approximation is poor. These results suggest that it is sensible to initialise the domain with a uniform or near-

uniform partition so as to avoid small intervals and the associated decrease in accuracy that they harbour.

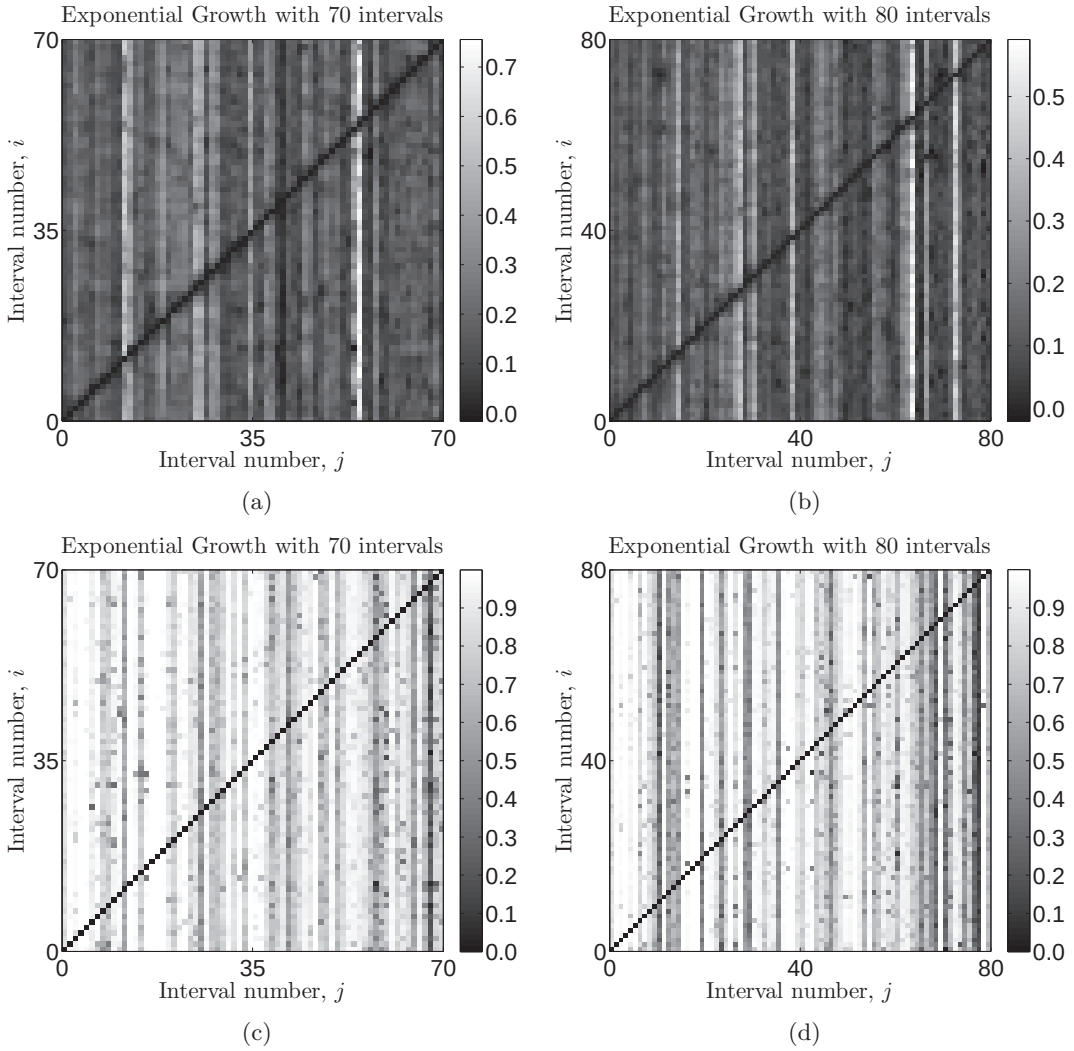


Figure 3.13: *As in Fig. 3.12 with the exception that both simulations are initialised with an initially non-uniform domain partition.*

Summary

We began this chapter, in Section 3.1, by generalising the early results of Chapter 2 to a stochastic framework in which the intervals of the domain partition were no longer the same size. In order to derive the correct transition rates between non-uniform intervals we solved the first passage problem for an underlying microscopic process defined by an SDE and corresponding FPE. The derived transition rates were found to be a generalisation of the

transition rates on the uniform domain partition and we were able to derive transition rates corresponding to different mechanisms of sensing an external signal from the appropriate SDEs. Numerical simulations were used to confirm the validity of our results.

The main motivation for the consideration of non-uniform domain partitions was a more realistic incorporation of domain growth than in the previous chapter. Instead of allowing intervals to instantaneously double in size and divide, in Section 3.2 we made growth a more continuous process and loosened the coupling between interval growth and interval division, with intervals dividing only upon reaching a certain size. As a toy model we introduced deterministic domain growth in order to verify that domain growth could be implemented in an incremental fashion. However, this chapter's main result pertains to the incorporation of stochastic, incremental domain growth into our equivalence framework in Section 3.3. We used a master equation formalism in a novel way, considering pre-growth cell densities in terms of post-growth densities on the post-growth domain, to show that the stochastic process of domain growth introduced in Section 3.2 was equivalent to a classic PDE describing continuous exponential domain growth, under some mild moment closure assumptions. We justified these assumptions, in Section 3.4, as well as corroborating our theoretical derivations with numerical simulations demonstrating the qualitative and quantitative equivalence of the stochastic and continuum models for cell migration.

In the stochastic model, as the domain grows the number of intervals increases causing an increase in the number of propensity functions and a consequent increase in the computational complexity of the accompanying stochastic simulations. In attempting to extend our equivalence framework into higher dimensions (see Chapter 5) we will find that this problem is exacerbated still further: in two dimensions, on a square domain, in order to maintain the spatial resolution of the individual-based system we will have to square the number of intervals we are considering. As a consequence, we will also have to increase the number of cells on the domain by the same factor in order to maintain a reasonable steady state cell profile of large enough magnitude to be comparable to a PDE equivalent. Such increases in computational complexity will require faster and more efficient stochastic simulations. In the next chapter we compare the increases in the speed of the SSA brought about by several different streamlining adaptations. In some situations we may find the

formalism of the SSA too restrictive to achieve the increases in speed that we require. As such, we investigate and adapt a variety of so-called 'accelerated' SSAs whereby a small amount of statistical accuracy is sacrificed in order to achieve a large increase in simulation speed.

Chapter 4

Efficient and accelerated stochastic simulation algorithms

Overview

In previous chapters we considered comparisons between stochastic individual-level models for cell migration and deterministic population-level descriptions. We simulated the stochastic model using Gillespie's direct method (DM), an exact stochastic simulation algorithm (ESSA), and noted that, even in one dimension, the simulation times are almost prohibitively long. Excessively long simulation times provide a good justification for our equivalence framework (the PDE being much faster to simulate than the stochastic model, in most cases). However, when attempting to validate our equivalence framework in higher dimensions in the next chapter, implementing stochastic simulations will become infeasible. In Section 4.1 we briefly outline some ways in which the SSA can potentially be made more efficient.

Cell migration algorithms in more than one dimension may have such a large number of reaction channels that simply increasing efficiency may not suffice. In light of this, we may be forced to sacrifice the accuracy of ESSAs in order to gain speed. This can be achieved in so-called accelerated stochastic simulation algorithms (ASSAs), which we introduce in Section 4.2. ASSAs were first introduced by Gillespie (2001) in the form of the 'τ-leaping'

method (TLM). ASSAs are only an approximation to the exact simulation and as such it is possible that the approximated species numbers can become negative, a non-physical situation which can cause simulations to break down. This problem provided a serious challenge to the credibility of the ASSA, especially in the context of low copy numbers of one or more of the reactant species[1]. This problem was quickly rectified by Chatterjee et al. (2004) and, independently, by Tian and Burrage (2004), who implemented 'binomial' τ-leap methods (BTLMs), which explicitly prevented the populations of reactant species becoming negative. However, due to a poor correspondence between the Binomial and the Poisson distributions for the relevant parameter regimes, the BTLM has been largely superseded.

In Section 4.3 we review the current τ-leaping gold standard, the 'modified' Poisson τ-leaping method of Cao et al. (2005), which reverts to the use of Poisson random numbers, rather than binomial. This method classifies as critical those reactions in immediate danger of sending species populations negative. At most, one of these critical reactions is allowed to occur in the next time-step via a process akin to the ESSA. Despite this extra caution, there remains a small but finite probability of each of the non-critical reactant species becoming negative.

In Section 4.4 we argue that the criticality of a reactant species and its dependent reaction channels should be related to the probability of the species number becoming negative. This way, only reactions that, if fired, produce a high probability of driving a reactant population negative are labelled as critical. This allows us to approximate the number of firings of more reaction channels using Poisson random variables and fewer using the adapted ESSA technique thus speeding up the simulation whilst maintaining accuracy. Numerical simulations that demonstrate the improved efficiency of our method of criticality selection are performed in Section 4.5 and we conclude the chapter by discussing the implications of our adapted technique.

The majority of the work in this chapter appears in the paper: C.A. Yates and K. Burrage. Look before you leap: A confidence-based method for selecting species criticality while avoiding negative populations in τ-leaping. *J. Chem. Phys.*, 134(8):084109, 2011.

[1]Tian and Burrage (2004) found that the simulation of the expression of LacY and LacZ genes and the activity of the corresponding proteins in *E. coli*, first considered in the context of SSAs by Kierzek (2002), consistently produced negative populations.

This paper was also selected for the February 2011 issue of *JCP: BioChemical Physics*.

4.1 Efficient, exact stochastic simulation

Introduced by Gillespie (1976, 1977) (and in a different form by Kurtz, 1972), the stochastic simulation algorithm can be used to simulate a well-stirred system of $N \geq 1$ chemical species, $\{S_1, \ldots, S_N\}$, interacting stochastically through M reaction channels $\{R_1, \ldots, R_M\}$. We denote the state of the system at time t by the vector $\boldsymbol{X}(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ represents the number of molecules of S_i at time t. We aim to simulate the evolution of $\boldsymbol{X}(t)$ given some initial condition, $\boldsymbol{X}(t_0) = \boldsymbol{x_0}$. The dynamics of each reaction channel, R_j, (for $j = 1, \ldots, M$) are completely characterised by a propensity function, $a_j(\boldsymbol{x})$, and a stoichiometric vector, $\boldsymbol{\nu}_j = (\nu_{1j}, \ldots, \nu_{Nj})$. The propensity functions are such that $a_j(\boldsymbol{x})dt$ defines the probability that, given $\boldsymbol{X}(t) = \boldsymbol{x}$, one R_j reaction will 'fire' during the next infinitesimal time interval, dt. ν_{ij} represents the change in the population of S_i caused by one 'firing' of the R_j reaction channel. For each reaction channel, R_j, the state change vector, $\boldsymbol{\nu}_j$, is defined once at the beginning of simulation but the propensity functions will usually change dynamically with the state vector, $\boldsymbol{X}(t)$, and thus require updates at each time-step.

One implementation of this mathematically exact procedure for simulating the progression of a system (which we have been using for our one-dimensional simulations) is known as the DM (Gillespie, 1977). Given a system at time t in state $\boldsymbol{X}(t)$, a time interval τ, until the next reaction occurs, is generated. Along with it a reaction, with index j, is chosen to occur at time $t+\tau$. The changes in the numbers of molecules caused by the firing of reaction channel R_j are implemented, the propensity functions are altered accordingly and the time is updated, ready for the next (τ, j) pair to be selected. A method for the implementation of this algorithm is given in Algorithm 1.

Algorithm 1

(1a) Initialise the time, $t = t_0$, and the species numbers, $\boldsymbol{X}(t_0) = \boldsymbol{x}_0$.

(1b) Evaluate the propensity functions, $a_j(\boldsymbol{x})$, associated with the reaction channels R_j $(j = 1, \ldots, M)$ and their sum $a_0(\boldsymbol{x}) = \sum_{j=1}^{M} a_j(\boldsymbol{x})$.

(1c) Generate two random numbers $rand_1$ and $rand_2$ uniformly distributed in $[0, 1]$.

(1d) Use $rand_1$ to generate a time increment, τ, from the exponential distribution with mean $1/a_0(\boldsymbol{x})$ i.e.

$$\tau = \frac{1}{a_0} \ln\left(\frac{1}{rand_1}\right). \tag{4.1.1}$$

(1e) Use $rand_2$ to generate a reaction index j with probability $a_j(\boldsymbol{x})/a_0(\boldsymbol{x})$, in proportion to its propensity function. i.e. find j such that

$$\sum_{j'=1}^{j} a_{j'}(\boldsymbol{x}) < a_0(\boldsymbol{x}) \cdot rand_2 < \sum_{j'=1}^{j+1} a_{j'}(\boldsymbol{x}). \tag{4.1.2}$$

(1f) Update the state vector, $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{\nu}_j$, and the time, $t = t + \tau$.

(1g) If $t < t_{final}$, the desired stopping time, then go to step **(1b)**. Otherwise stop.

4.1.1 Efficient propensity searching

The normalised vector of propensity functions, $\boldsymbol{a}/a_0$, forms a partition of the unit interval. In order to ascertain which reaction should take place, in step **(1e)** of Algorithm 1, we find the interval into which a uniform random number falls. This linear searching process requires $O(M)$ operations and has been identified, in many simulations (especially those with a large number of reaction channels), as being computationally intensive and hence causing a bottleneck in the simulations (Gibson and Bruck, 2000; Li and Petzold, 2006).

There exist a variety of methods to mitigate this problem.

Motivated by the fact that, in our basic cell migration simulations, we have two sets of propensity functions; one set corresponding to moving cells left and the other to moving cells right, we introduce the 'split search'. We keep track of the sum of the two sets of the propensity functions separately. We easily find which set to consider and carry out a linear search in that set. The number of operations we carry out is still $O(M)$, but now our coefficient has halved.

Our split searching approach is a special case of the 'binning method' of Maksym (1988) for fast Monte Carlo simulations. Consider a data structure of $\lceil M/p \rceil$ partial sums[2], each containing the sum of the propensity functions corresponding to approximately p reactions. The search step now has two parts. Firstly searching for the correct bin which takes $O(M/p)$ operations, then searching within that bin, which takes $O(p)$ operations. In order to minimise the number of operations we choose $p = \alpha M^{1/2}$, where α should depend on the relative efficiency of the two search processes. The whole procedure, therefore, takes $O(M^{1/2})$ operations.

This 'two-level' search method is, in turn, a special case of the 'K-level' search presented by Blue et al. (1995), again for general Monte Carlo simulations. We again begin by binning the propensity functions into $\lceil M/p \rceil$ partial sums of size p, where p is now $O(M^{1/K})$. These, in turn, are binned into partial sums containing p of the original partial sums and, therefore, p^2 propensity functions. We continue this binning process until we are left with a set of p partial sums, the K levels of which we can search recursively to find the appropriate propensity function. The total operation count for searching is, therefore, $O(KM^{1/K})$ and updating the data structure takes $O(K)$ operations. The number of operations for searching takes a minimum value of $\ln(M)$ when $K = \ln(M)$. Practically, however, we can take $p = M^{1/K} = 2$ so that each of the lowest-level partial sums has two elements. The data structure becomes a binary tree and both searching and updating the data structure take $O(\log_2(M))$ operations (Blue et al., 1995). A logarithmic search of propensity functions in the SSA has been introduced by Li and Petzold (2006) and shown to be of comparable speed to other efficient versions of the SSA.

[2] $\lceil x \rceil$ denotes the *ceiling* function (i.e. the smallest integer greater than x).

4.1.2 Algorithmic modifications

In his original formulation of the SSA, the First Reaction Method (FRM), Gillespie (1976) generated a putative time for each reaction. The reaction whose time is first is implemented, completely avoiding the need to search the list of propensity functions. Exploiting the memoryless property of Poisson processes, new putative times are calculated using a freshly generated random number for each reaction and the process continues. Gibson and Bruck (2000) devised the Next Reaction Method (NRM) which reuses the random numbers, originally generated for the reactions whose propensity functions had changed, in order to calculate new putative reaction times, thus saving computational effort on random number generation. They stored the putative reaction times in an 'indexed priority queue' tree-structure, in which the time and index of the next reaction to occur are always easily accessible at the root of the tree. The number of operations required in order to maintain the indexed priority queue at each iteration is $O(\log(M))$. They also increased the efficiency of the algorithm by introducing a dependency graph which lists the propensity functions that depend on the outcome of each reaction, enabling us to identify the set of propensity functions which require updating and alter only them. As well as increasing the speed of the SSA, the NRM reformulation is important conceptually. Adaptations of the NRM have been used in order to deal with propensity functions which change during the course of a reaction (due to volume or temperature fluctuations) and to incorporate time delays into the SSA (Anderson, 2007). The concepts underlying the NRM have also been incorporated into an algorithm for efficiently simulating spatially extended systems in the Next Sub-volume Method (NSM) (Elf and Ehrenberg, 2004).

In response to the NRM, Cao et al. (2004) introduced the Optimised Direct Method (ODM). They made two key alterations to the original DM introduced by Gillespie (1977). The first borrows the dependency graph of Gibson and Bruck (2000), updating only those reactions whose propensity functions change. The second involves structuring the order of propensity functions in the search list so that reactions occurring more frequently are higher up, reducing the 'search depth' of the linear search. In order to initialise the search list appropriately, Cao et al. (2004) introduced a pre-simulation step whereby a small number of

simulations are run in order to ascertain the relative frequencies of each reaction. The ODM is accepted as being of comparable efficiency to the NRM in a wide variety of situations (Gillespie, 2007), but is expected to outperform the NRM in systems of reactions that are tightly coupled and therefore require a large amount of computational effort to maintain the indexed priority queue (Cao et al., 2004).

Inspired by the idea of sorting propensity functions, McCollum et al. (2006) introduced the Sorting Direct Method (SDM). Using a simple bubble-type sort algorithm the SDM accounts for possible transient changes in reaction frequency by dynamically changing the reaction selection order. This means that reactions which occur more often, at a specific part of the simulation, find themselves higher in the search order, thus reducing the average search depth. The SDM also benefits from not having to implement pre-simulations. We suggest that occasional complete sorting of the list of the propensity functions may have a similar effect to the SDM.

4.1.3 Stochastic simulation on the growing domain

When simulating cell migration on a growing domain the number of propensity functions, $M(t)$, is time-dependent. Of the methods outlined above, the easiest to implement will clearly be the linear search DM, which essentially requires no adaptation. Methods such as the SDM and the ODM, which order the propensity functions according to size, will be similarly easy to implement, although, since the ODM does not implement dynamic sorting it may lose optimality in comparison to the SDM as the domain grows. Split searching will also be straightforward to implement, although, instead of splitting the number of propensity functions into two sets, we should split them into three, corresponding to the three reactions (left movement, right movement and growth). Any of the methods which have associated, non-trivial data structures, ranging from square searching through the K-level searching algorithms to the logarithmic direct method and the NRM will be increasingly difficult to implement since they will require the efficient alteration of the associated data structures.

4.2 Accelerated stochastic simulation algorithms

The SSA records the exact history of a system, stepping between individual reaction events in the finest detail. This approach is simultaneously the Herculean strength and the Achilles' heel of the SSA. The construction of every reaction event gives a complete history of $\boldsymbol{X}(t)$, but simulating every reaction is time-consuming and computationally expensive.

If some losses in accuracy are acceptable, then a faster, but approximate, method - the explicit τ-leaping algorithm (Gillespie, 2001) - is available. Instead of stepping gingerly between individual reactions, the accelerated method leaps temeritously through time, allowing several reaction events from each channel to occur between updates of the state vector. This accelerated method is not, however, completely reckless as it is constrained by the so-called 'leap condition' (Gillespie, 2001). This states that all leaps should be sufficiently small that the change in the state vector will be so slight that no propensity functions have their values changed 'significantly' during a leap. The number of firings, k_j, of each reaction channel, R_j, during a leap can be approximated using samples of the Poisson random variable, $\mathcal{P}(a_j(\boldsymbol{x})\tau)$, for sufficiently small τ. This provides us with a formula by which to update the state vector,

$$\boldsymbol{X}(t+\tau) = \boldsymbol{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}(a_j(\boldsymbol{x})\tau). \tag{4.2.1}$$

The accuracy of this approximation depends on the extent to which the leap condition is observed. If propensity functions change appreciably during the leap then we will be simulating reactions from the wrong Poisson distribution. What exactly constitutes an appreciable change in a propensity function and consequently how to choose τ is the subject of much debate (Anderson, 2008; Cao et al., 2006; Gillespie, 2001; Gillespie and Petzold, 2003). The original method of Gillespie (2001) bounds the change in each propensity function by $\varepsilon a_0(\boldsymbol{x})$, where $0 < \varepsilon \ll 1$ is known as the error control parameter. Throughout the rest of this chapter and in all numerical simulations we fix $\varepsilon = 0.03$, a canonical value used by Cao et al. (2005, 2006). This bound on the change in propensity function gives a method of bounding τ and completes the description of the basic explicit τ-leaping algorithm (see

Cao et al. (2006); Gillespie (2001)).

4.2.1 Negative populations and Binomial τ-leaping

Since the Poisson random variable can have arbitrarily large sample values, it is possible that a reaction channel may fire so many times that it will deplete one or more of its reactants by larger numbers than are actually available. One method of preventing this unphysical behaviour, presented by Chatterjee et al. (2004) and independently by Tian and Burrage (2004), is known as Binomial τ-leaping. This relies on approximating the number of firings, k_j, of a reaction channel, R_j, by a sample from a Binomial distribution whose values are naturally bounded. Due to the poor correspondence between the Poisson and Binomial distributions in the case of small species numbers, x_i, (the case with which we will most often be concerned when attempting to avoid negative populations) and over-cautious restrictions on the number of reactions that can fire, the Binomial τ-leap method tends to be less accurate than the more recent 'modified Poisson τ-leaping' procedure (Cao et al., 2005), which also circumvents the problem of negative species.

4.3 Modified Poisson τ-leaping

The modified Poisson τ-leaping algorithm[3] of Cao et al. (2005), which we refer to as the 'CGP method' after the principal authors, arose from the observation that, typically, negative populations are caused by multiple firings of reactions that are only a few firings away from exhausting the numbers of one or more of their reactants. It relies on identifying reactions that, if fired more than a predefined 'critical' number of times, will cause one or more of their contingent species populations to become negative. The number of 'allowed' firings of a reaction, R_j, will be denoted by L_j. Reactions are partitioned into critical and non-critical subsets with, at most, one of the critical reactions being able to fire per leap, by a process akin to the exact SSA. This ensures the reactant species depleted by the critical reaction do not become negative. The more critical reactions we have, the slower the simulation becomes. Defining critical reactions should, therefore, be an important part of any

[3] An algorithm for the modified τ-leaping method is given in Cao et al. (2005).

algorithm. However, the size of the critical number of firings, n_c, is difficult to determine and Cao et al. (2005) suggest a value between 2 and 20.

Widely varying reaction rate constants can cause reaction channels to have very different propensity functions and hence probabilities of driving the species population negative. Even when reactions have the same reaction rate constants, differing numbers of molecules undergoing the same reaction will have widely varying probabilities of becoming negative. If we choose n_c to be small then we run the unacceptable risk of ignoring reactions that have a high probability of causing their species populations to become negative. However, choosing n_c too large risks classifying as critical reactions that have near-infinitesimal probabilities of sending their contingent species populations negative, slowing the simulation[4]. Including rates of reaction that may vary over orders of magnitude and the fact that species numbers may be re-populated during a reaction complicates matters still further. A method for deciding the criticality of a reaction based on the *probability* of any one of its *contingent species* becoming negative would be preferable.

4.4 Incorporating confidence measures for criticality selection

Consider a simple degradation reaction, $S_1 \xrightarrow{c_1} \phi$. The expected number of firings for this reaction is $\lambda = x_1 c_1 \tau$, where x_1 is the number of molecules of S_1. Taking an arbitrary value for the rate constant multiplied by the time-step, e.g. $c_1\tau = 0.1$, we find that the probability of species numbers becoming negative, when the number of firings of the reaction is drawn from a Poisson distribution with mean, λ, is drastically different depending on the number of molecules of the species, x_1. For example, if $x_1 = 10$, the probability of the species becoming negative is approximately 10^{-8}. However, if $x_1 = 2$, the probability increases to 10^{-3}, five orders of magnitude larger. Choosing the critical species number to be $n_c = 1$, our second example is not flagged as critical despite the relatively high probability of the population becoming negative. Conversely, if we choose n_c to be larger, i.e. $n_c = 10$, then

[4]Even with a large value of n_c there is still a small possibility (due to the non-critical reactions) of negative species populations occurring. On these rare occasions a leap can simply be rejected and repeated with a reduced τ (Cao et al., 2005).

species which have very small probabilities of becoming negative are labelled as critical and their dependent reactions are restricted unnecessarily, slowing the simulation.

Even this simplified view presents problems for the CGP method of determining criticality. This example ignores the issues of differing reaction rates and the fact that reactant species may be re-populated during a leap, making it less likely for their population to become extinct. Different reactant species require levels of criticality dependent on their population size and on the rates of reactions that deplete and add to their numbers. The criticality of each reaction will depend on the criticality of its contingent species. This hints that taking a single critical value for all reactions is not appropriate.

If we are to draw a Poisson random number of firings of a reaction channel we would like to ensure that the probability that any of the contingent species' populations become negative is less than some threshold value, $P_c = \alpha/100$, for example. This is the basis for a one-sided $\alpha\%$ confidence interval[5]. If the probability of the population becoming negative goes above P_c then we regard this reaction as critical. This way we eliminate reactions being unnecessarily labelled as critical when in fact they can undergo Poisson τ-leaping with an infinitesimally small probability of sending the populations of their contingent reactant species negative.

4.4.1 Some useful results

We have already, informally, introduced L_j as the number of firings of a reaction, R_j, before it sends one of its contingent species negative, providing that no new molecules of that reactant species are created. More formally this has been expressed as

$$L_j = \min_{i=1,\ldots,N;\nu_{ij}<0} \left\lfloor \frac{x_i}{-\nu_{ij}} \right\rfloor, \tag{4.4.1}$$

where, as before, $\lfloor x \rfloor$ represents the *floor* function (i.e. the largest integer smaller than x) (Cao et al., 2005, 2006).

Johnson et al. (2005) introduce Property 1:

[5] $\alpha = 5$, for example, denotes a 5% confidence interval.

Property 1: If $\mathcal{P}_1 = \mathcal{P}(\lambda_1)$ and $\mathcal{P}_2 = \mathcal{P}(\lambda_2)$ are two independent Poisson distributions with means λ_1 and λ_2, respectively, then $\mathcal{P}_1 + \mathcal{P}_2$ is also Poisson distributed, $\mathcal{P}(\lambda_1 + \lambda_2)$, with mean $\lambda_1 + \lambda_2$.

This property can be found in any basic text on univariate discrete distributions, (see, for example, Johnson et al., 2005). We also introduce two new properties employed when determining the distribution of the change of species numbers, X_i:

Property 2: If $\mathcal{P}_1 = \mathcal{P}(\lambda_1)$ and $\mathcal{P}_2 = \mathcal{P}(\lambda_2)$ are two independent Poisson distributions with means λ_1 and λ_2, respectively, then their difference $\mathcal{P}_1 - \mathcal{P}_2$ is given by a Skellam distribution, with mean $\lambda_1 - \lambda_2$ and variance $\lambda_1 + \lambda_2$. The probability mass function (PMF) of the Skellam distribution takes the form

$$\mathcal{SK}(k; \lambda_1, \lambda_2) = e^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} I_{|k|}(2\sqrt{\lambda_1\lambda_2}), \tag{4.4.2}$$

where $I_k(z)$ is the modified Bessel function of the first kind.

The proof of Property 2 is given in Appendix C.1.1. Using Property 1 recursively in combination with Property 2 we can also show that the sum and/or difference of arbitrarily many Poisson distributions is also Skellam distributed.

If $\mathcal{P}^m$ is Poisson distributed but with support $0, m, 2m, 3m, \ldots$, for $m \in \mathbb{Z}$, instead of integer support, then we call it a Poisson m-let. i.e. $m = 1$ gives the usual Poisson distribution or Poisson singlet, $m = 2$ gives a Poisson doublet etc.

Property 3: If $\mathcal{P}^m = m \times \mathcal{P}(\lambda_1)$ and $\mathcal{P}^n = n \times \mathcal{P}(\lambda_2)$ are a Poisson m-let and a Poisson n-let, respectively, where $m, n \in \mathbb{Z}$, then their sum $\mathcal{P} = \mathcal{P}^m + \mathcal{P}^n$ is distributed as a Poisson stopped sum distribution.

The proof of Property 3 is given in Appendix C.1.2. Since, in the statement of Property 3, we assume $m, n \in \mathbb{Z}$, the formulae apply equally well to negative values of m and n as to positive values and, hence, to differences as well as sums of Poisson m-lets. It is also a trivial extension to prove that the sum or difference of Poisson stopped sum distributions is also a Poisson stopped sum distribution. From the individual Poisson m-lets we can find the probability generating function (PGF) of the Poisson stopped sum distribution and from the PGF we can find the PMF and hence the cumulative mass function (CMF).

4.4.2 Determining the species change probability distribution

In order to find the probability of a reactant species' population, X_i, becoming negative we can rearrange equation (4.2.1) in order to consider the distribution of its change:

$$\Delta_\tau X_i = X_i(t+\tau) - X_i(t) = \sum_{j \in J_{ncr}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau), \tag{4.4.3}$$

where J_{ncr} is the set of non-critical reactions. A problem occurs in that the value of τ is unknown and cannot be chosen until we have decided which reactions are critical and which are not. This 'chicken and egg' situation can be resolved by invoking the leap condition, which states that the propensities should not change significantly between steps. This implies their sum should not change too much and hence the size of the time-step should, on average, be approximately the same. We can, therefore, evaluate $\Delta_\tau X_i$ using the value of the previous time-step, τ.

Property 3 states that the sum of Poisson m-lets is given by a Poisson stopped sum distribution. This is precisely the situation when considering the distribution (4.4.3) of $\Delta_\tau X_i$, where m is replaced by ν_{ij} and we assume that the Poisson random variables, $\mathcal{P}_j(a_j(\boldsymbol{x})\tau)$, describing the number of firings of the reactions dependent on reactant S_i, are uncorrelated. The $2q$ parameters (where q is the highest order of the reaction that reactant S_i is involved in) of the Poisson stopped sum distribution, given in equation (4.4.3), all depend linearly on τ. Given a predefined confidence interval we can use the PMF of this distribution to calculate the CMF and hence to calculate a criticality surface for species numbers, below which species will be defined as critical.

Figure 4.1 (a) displays a criticality surface for the simplest case $\Delta_\tau X_i = -\sum \mathcal{P}_j(a_j(\boldsymbol{x})\tau)$, where species i is depleted by linear reaction channels (i.e. $\nu_{ij} = -1$). It also shows for comparison the effective criticality curve for the CGP method of τ-selection, with $n_c = 20$. When simulating our revised method for values of the parameter that would give species criticality values above 20 we take the species criticality value to be exactly 20. This threshold is necessitated by the fact that we cannot calculate the value of the species criticality curve for all possible (infinitely many) values of the parameter of the Poisson distribution. As such, we assign a maximum criticality threshold of 20 which, as the maximum value

of n_c considered by Cao et al. (2005), has previously been shown to be robust in avoiding negative populations in τ-leaping.

In a more complicated scenario, in which a species undergoes linear degradation and creation reactions during each τ-leap, the species is depleted by a Poisson distribution with mean λ_1, but replenished by a Poisson distribution with mean λ_2. The random variable describing the number of molecules of the species that are *depleted* is drawn from a Skellam distribution with mean $\lambda_1 - \lambda_2$ and variance $\lambda_1 + \lambda_2$ given by $-\Delta_\tau X_i = \mathcal{P}_1(\lambda_1) - \mathcal{P}_2(\lambda_2)$. For the two parameters of this Skellam distribution, λ_1 and λ_2, we can calculate a critical *surface*, below which species will be considered critical (see Fig. 4.1 (b)). This criticality surface is sufficient for all sets of reactions for which $|\nu_{ij}| = 1$. Similar curves/surfaces can be calculated for non-linear reactions using the more general Poisson stopped sum distributions.

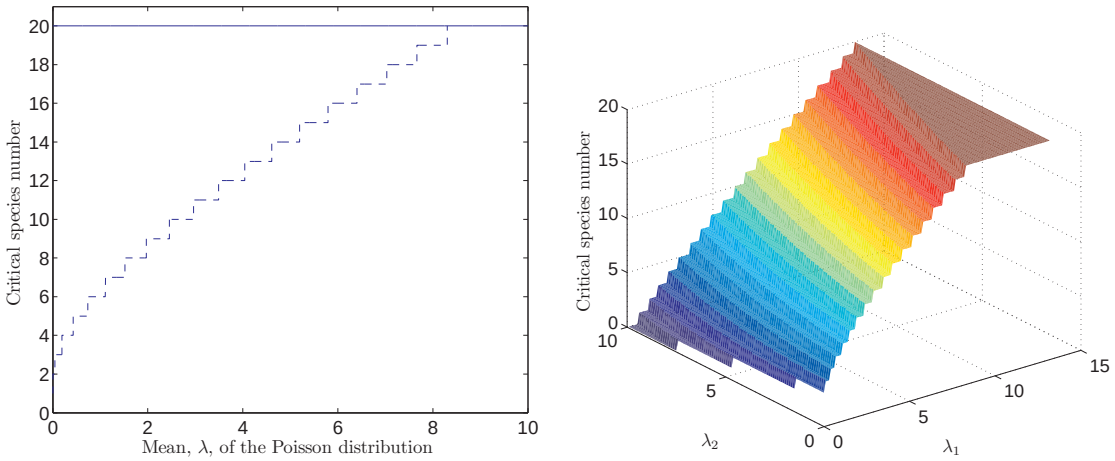


Figure 4.1: *Criticality surfaces for the confidence-based τ-leap method. (a) An example critical curve for purely linear species depletions with criticality level $\alpha = 0.1$. The values on the x-axis represent the mean, λ, of the Poisson distribution depleting the species S_i and the dashed curve represents the value of x_i above which species S_i is not classified as critical. The full line represents the effective criticality curve of the CGP method ($n_c = 20$). (b) An example critical surface for a Skellam distribution with criticality level $\alpha = 0.1$. The values on the x-axis represent the mean, λ_1, of the Poisson distribution depleting the species whereas the values on the y-axis represent the mean, λ_2, of the Poisson distribution regenerating the species. The surface represents the value of x_i above which species S_i is not classified as critical.*

This revised method of deciding criticality eliminates the problem of reactions that have an infinitesimally small probability of driving their reactant species populations negative

being labelled as critical. It also ensures that reactions that have a high probability of sending reactant species negative are always labelled as critical. This allows us to approximate the number of firings of more reaction channels using Poisson random variables and fewer using the adapted SSA technique.

An algorithm for the implementation of the confidence-based methods is given in Algorithm 2.

Algorithm 2

(2a) Initialise the time $t = t_0$ and the species numbers $\boldsymbol{X}(t_0) = \boldsymbol{x}_0$.

(2b) Evaluate the propensity functions $a_j(\boldsymbol{x})$ for $j = 1, \ldots, M$, and their sum, $a_0(\boldsymbol{x}) = \sum_{j=1}^{M} a_j(\boldsymbol{x})$, and identify which reactions are currently critical using the method described in this section.

(2c) With a pre-specified value of ε, compute a candidate leap time, τ', using the method of Cao et al. (2006). τ' is the largest permissible time-step for non-critical reactions.

(2d) If the value of τ' chosen is less than some small multiple (say 2) of $1/a_0$ then reject it and implement a number (say 100) of successive steps of the SSA before returning to step **(2b)**. If τ is larger than our chosen small multiple of $1/a_0$ then accept it and proceed to step **(2e)**.

(2e) Compute the sum, $a_0^c(\boldsymbol{x})$, of the critical propensity functions. Generate a second candidate leap time, τ'', in a method akin to the SSA, as a sample of an exponential random variable with mean $1/a_0^c(\boldsymbol{x})$. τ'' is a tentative estimate of the time to the next critical reaction.

(2f) Take $\tau = \min\{\tau', \tau''\}$. For all non-critical reactions, R_j, generate k_j as a sample of the Poisson random variable with mean $a_j(\boldsymbol{x})\tau$.

1. If $\tau' < \tau''$, set $k_j = 0$ for all critical reactions (i.e. no critical reactions fire).

2. If $\tau'' < \tau'$, use *rand*, a uniform random number in $[0, 1]$, to generate a reaction index, j_c, with probability $a_{j_c}(\boldsymbol{x})/a_0^c(\boldsymbol{x})$, in proportion to its propensity function. i.e. find j_c such that

$$\sum_{j'=1}^{j_c} a_{j'}(\boldsymbol{x}) < a_0^c(\boldsymbol{x}) rand < \sum_{j'=1}^{j_c+1} a_{j'}(\boldsymbol{x}). \tag{4.4.4}$$

Set $k_{j_c} = 1$, and for all other critical reactions set $k_j = 0$.

(2g) If there is a negative component in $\boldsymbol{x} + \sum_{j=1}^{M} k_j \boldsymbol{\nu}_j$, reduce τ' by half and return to step **(2d)**. Otherwise update $\boldsymbol{x} = \boldsymbol{x} + \sum_{j=1}^{M} k_j \boldsymbol{\nu}_j$ and $t = t + \tau$.

(2h) If $t < t_{final}$, the desired stopping time, then go to step **(2b)**. Otherwise stop.

4.5 Numerical Examples

In order to test our revised τ-leaping method we have applied it to four different systems of reactions. The first is a set of 100 isomerisation reactions, the second a similar set of 100 dimerisation reactions. The third is the standard LacY/LacZ system (Krebs et al., 2010) using the model developed by Kierzek (Carrier and Keasling, 1999; Kierzek, 2002; Kierzek et al., 2001) as employed by Tian and Burrage (2004) and Cao et al. (2005, 2006) to test the accuracy and efficiency of their τ-leaping algorithms. The fourth and final comparison is a one-dimensional position-jump model of cell migration as detailed in Chapter 2. When calculating a critical curve/surface in these numerical simulations we first consider the distribution of the change in species number due solely to reactions that deplete the species number:

$$\sum_{\nu_{ij}<0, j\in J_{ncr}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau). \tag{4.5.1}$$

In the comparisons that follow we will denote this the 'Confidence-Based (CB)-depletion' method, indicating that the distribution we consider for the change in species is purely from reactions which deplete the species number. This method will use a criticality surface

similar to that shown in Fig. 4.1 (a). We recognise that this does not take into account the possible re-population of the species by reactions for which $\nu_{ij} > 0$ and as such, in this method, we are overly cautious in our classification of critical reactions. For comparison, we also consider the distribution of the change in species number due to those reactions that deplete and those that regenerate the species number:

$$\sum_{j \in J_{ncr}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau). \tag{4.5.2}$$

We denote this the 'Confidence-Based (CB)-regeneration' method. This method will employ a criticality curve similar to that shown in Fig. 4.1 (b). In general the 'regeneration' method will be more efficient than the 'depletion' method since by considering possible regeneration of species numbers during reactions we will obtain a tighter bound on the necessary level of criticality for each species and will classify fewer reactions, unnecessarily, as critical. However, the depletion method may be simpler to implement since the CMF of the Poisson distribution has a closed form and the CMF of the Skellam distribution does not. This makes calculating the criticality curve for the depletion method using the CMF of the Poisson distribution simpler than for the regeneration method, where the probability density function of the Skellam distribution must be summed for each value of the parameter to calculate the desired curve.

In the following simulations we compare the times of the basic DM implementation of the SSA, the CGP with an extensive range of values of the criticality parameter, $n_c = 0, \ldots, 20$ (approximately the range of values suggested by Cao et al., 2005), and the CB methods with confidence level, $\alpha = 1 \times 10^{-5}$. The accuracy of the CB methods is found to be comparable to that of the CGP method in all simulations. All simulations detailed in this section were written in `C++` performed on a 'no name' brand machine with 4 AMD (Advanced Micro Devices) Phenom(tm) II 945 processors (3GHz clock speed, 2MB L2 cache, 6MB L3 cache), 64-bit kernel running Ubuntu Linux 10.04 LTS.

4.5.1 Isomerisation loops

In this somewhat contrived set of reactions there are 100 reactant species and 100 reactions. These reactions form two isomerisation loops. In both loops every species decays to the next until the last species decays back to the first:

$$\begin{aligned} S_1 &\xrightarrow{c_1} S_2, \\ &\vdots \\ S_{50} &\xrightarrow{c_{50}} S_1, \\ S_{51} &\xrightarrow{c_{51}} S_{52}, \\ &\vdots \\ S_{100} &\xrightarrow{c_{100}} S_{51}. \end{aligned} \tag{4.5.3}$$

All the isomerisation reactions occur with the same rate constant, $c_i = 0.1$ for $i = 1, \ldots, 100$. Initially each species in the first isomerisation loop has 10 molecules and each species in the second loop has 1000 (i.e. $S_i(0) = 10$ for $i = 1, \ldots, 50$ and $S_i(0) = 1000$ for $i = 51, \ldots, 100$). We simulate over the time interval $t \in [0, 10]$ and repeat the simulations 1000 times.

	CPU time (s)	Steps	Critical reactions
SSA	72.93	50507096	N/A
CGP ($n_c = 11$)	18.93	264192	8232365
CB-depletion	2.77	31080	66544
CB-regeneration	2.74	29461	45493

Table 4.1: *A comparison, for model* (4.5.3)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition ($S_i(0) = 10$ for $i = 1, \ldots, 50$ and $S_i(0) = 1000$ for $i = 51, \ldots, 100$) for 1000 repeats using the SSA and three τ-leaping methods. None of the τ-leaping methods required the rejection of any time-steps due to negative species numbers. We choose the criticality parameter $n_c = 11$ as the median value of the range 2 to 20 suggested by Cao et al. (2005).*

The CB methods are the fastest for these simulations (see Table 4.1) with the 'regeneration' method being slightly faster than the 'depletion' method, as expected. There are sufficiently many of each species to make τ-leaping more effective than the SSA, but there are a number of species which are below the criticality threshold, $n_c = 11$, of the CGP τ-

leaping method. The CB methods determine that these reaction channels are not, in fact, critical as they have a minuscule probability of sending their contingent species negative. We can alter the initial conditions to show that the CB methods perform consistently well in comparison to the SSA and the CGP τ-leaping method.

For example, if we set the initial number of all species to be 10 (i.e. $S_i(0) = 10$ for $i = 1, \ldots, 100$) then we see that the SSA out-performs all three τ-leaping algorithms (see Table 4.2). It out-performs the CGP τ-leaping method by more than an order of magnitude, but only marginally out-performs the CB methods.

	CPU time (s)	Steps	Critical reactions
SSA	1.40	1001664	N/A
CGP ($n_c = 11$)	37.75	529277	33053214
CB-depletion	2.65	34017	127148
CB-regeneration	2.41	31666	83179

Table 4.2: *A comparison, for model* (4.5.3)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval* ($t \in [0, 10]$) *from a common initial condition* ($S_i(0) = 10$ *for* $i = 1, \ldots, 100$) *for 1000 repeats using the SSA and three* τ*-leaping methods. The SSA only marginally out-performs the CB methods. Both the CB method and the SSA drastically out-perform the CGP method.*

If we set the initial number of each species to be 1000 (i.e. $S_i(0) = 1000$ for $i = 1, \ldots, 100$) we see that the CB methods are nearly two orders of magnitude faster than the SSA and of comparable (or slightly increased) efficiency with the CGP method (see Table 4.3). For this reaction system the CB methods demonstrate more versatility than the traditional one-size-fits-all approach of the CGP algorithm.

	CPU time (s)	Steps	Critical reactions
SSA	127.32	100004782	N/A
CGP ($n_c = 11$)	1.38	3830	0
CB-depletion	1.37	3818	0
CB-regeneration	1.36	3789	0

Table 4.3: *A comparison, for model* (4.5.3)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval* ($t \in [0, 10]$) *from a common initial condition* ($S_i(0) = 1000$ *for* $i = 1, \ldots, 100$) *for 1000 repeats using the SSA and three* τ*-leaping methods.*

Figure 4.2 demonstrates that the CB methods perform consistently well in comparison

to the CGP technique over a range of values of the criticality parameter, n_c. At worst the speeds of the two algorithms are comparable and at best the CB algorithms are considerably faster than the CGP algorithm. These plots demonstrate that the CB methods are a robust choice in the face of differing initial conditions and crucially their performance is independent of any tuning parameters.

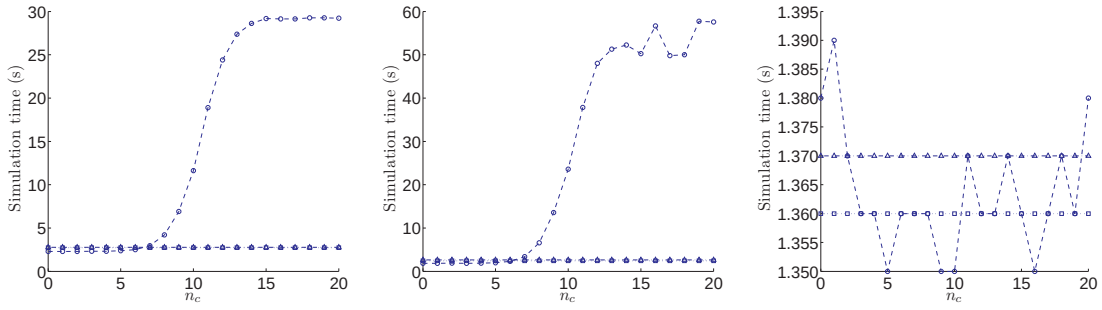


Figure 4.2: *A comparison of simulation times for 1000 repeats of each of the three τ-leaping methods for each of the three initial conditions described in Tables 4.1, 4.2 and 4.3: (a) $S_i(0) = 10$ for $i = 1, \ldots, 50$ and $S_i(0) = 1000$ for $i = 51, \ldots, 100$, (b) $S_i(0) = 10$ for $i = 1, \ldots, 100$, (c) $S_i(0) = 1000$ for $i = 1, \ldots, 100$, over the range of values of the parameter $n_c = 0, \ldots, 20$. Circles joined by a dashed line correspond to the CGP method, triangles joined by a dot-dash line to the CB-depletion method and squares joined by a dotted line to the CB-regeneration method. In (a) and (b), for lower values of n_c, the simulation times of all three algorithms are comparable, however, for larger values of n_c the CB methods are significantly faster. In (c) the times of all three methods are approximately comparable independent of the value of n_c.*

4.5.2 Dimerisation loops

In order to show that the CB methods can be applied in a straightforward way to systems of non-linear reactions we have altered the above reaction system to be dimerisations (instead of isomerisations) where two molecules of each reactant species are required for each reaction. These reactions form two dimerisation loops. In each loop one molecule of a species combines with another of the same species and decays to two molecules of the next species until molecules of the last species decay back to the first:

$$\begin{aligned} S_1 + S_1 &\xrightarrow{c_1} 2S_2, \\ S_2 + S_2 &\xrightarrow{c_2} 2S_3, \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
S_{50} + S_{50} &\xrightarrow{c_{50}} 2S_1, \\
S_{51} + S_{51} &\xrightarrow{c_{51}} 2S_{52}, \\
S_{52} + S_{52} &\xrightarrow{c_{52}} 2S_{53}, \\
&\vdots \\
S_{100} + S_{100} &\xrightarrow{c_{100}} 2S_{51}.
\end{aligned} \tag{4.5.4}
$$

All the dimerisation reactions occur with the same rate constant, $c_i = 0.1$ for $i = 1, \ldots, 100$. Initially each species in the first dimerisation loop has 10 molecules and each species in the second loop has 1000 molecules (i.e. $S_i(0) = 10$ for $i = 1, \ldots, 50$ and $S_i(0) = 1000$ for $i = 51, \ldots, 100$). We use this set of reactions to test the speed and reliability of the algorithms. We simulate over the time interval $[0, 10]$ and repeat the simulations 1000 times.

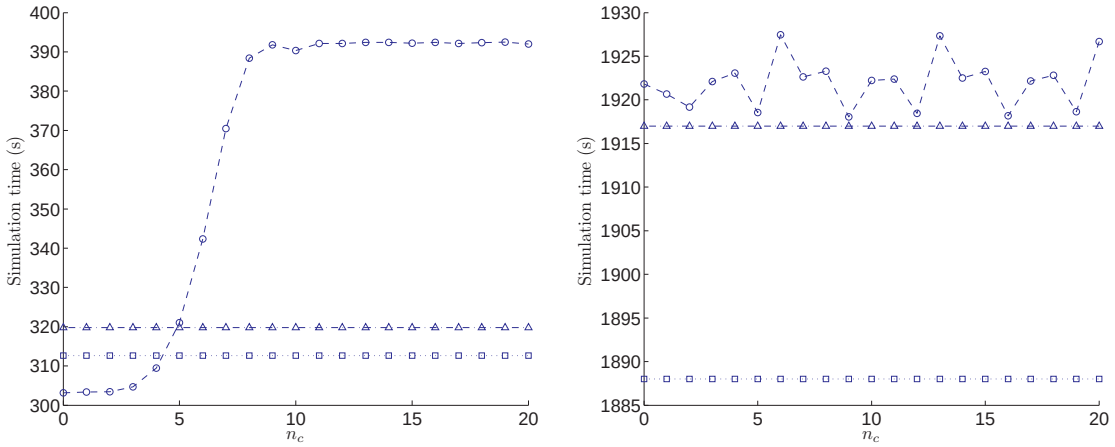
	CPU time (s)	Steps	Critical reactions
SSA	39706	3463378894	N/A
CGP ($n_c = 11$)	398	2468173	123345100
CB-depletion	320	972062	13492169
CB-regeneration	313	898838	7982245

Table 4.4: *A comparison, for model* (4.5.4)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition ($S_i(0) = 10$ for $i = 1, \ldots, 50$ and $S_i(0) = 1000$ for $i = 51, \ldots, 100$) for 1000 repeats using the SSA and three τ-leaping methods. None of the τ-leaping methods required the rejection of any time-steps due to negative species numbers.*

In this situation the CB methods are faster than the CGP (see Table 4.4) for similar reasons to those given for the isomerisation loop system. Altering the initial conditions we will see that the CB methods perform consistently well in comparison to the SSA and the CGP τ-leaping method. If we set the initial number of each species to be large for both dimerisation loops; 1000, for example (i.e. $S_i(0) = 1000$ for $i = 1, \ldots, 100$) then we see that all three τ-leaping methods are well over an order of magnitude faster than the SSA. Both CB methods remain marginally more efficient than the CGP method (see Table 4.5). Fig. 4.3 demonstrates the consistently good performance of the CB method in comparison to the CGP technique as n_c varies for the dimerisation loop described above.

	CPU time (s)	Steps	Critical reactions
SSA	70402	2516518253	N/A
CGP ($n_c = 11$)	1923	19979926	0
CB-depletion	1917	19980419	0
CB-regeneration	1888	19979825	0

Table 4.5: *A comparison, for model* (4.5.4)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval* ($t \in [0, 10]$) *from a common initial condition* ($S_i(0) = 1000$ *for* $i = 1, \ldots, 100$) *for 1000 repeats using the SSA and three* τ*-leaping methods.*



(a) (b)

Figure 4.3: *A comparison of simulation times for 1000 repeats of each of the three* τ*-leaping methods for each of the initial conditions described in Tables 4.4 and 4.5: (a)* $S_i(0) = 10$ *for* $i = 1, \ldots, 50$ *and* $S_i(0) = 1000$ *for* $i = 51, \ldots, 100$ *and (b)* $S_i(0) = 1000$ *for* $i = 1, \ldots, 100$*, over a range of values of the parameter* $n_c = 0, \ldots, 20$*. The results of the different methods are as denoted in Fig. 4.2. In (a) we see that for lower values of* n_c *the CGP method does slightly better than the CB methods. However, for larger values of* n_c *the CB methods are significantly faster. In (b) the CB methods are marginally faster than the CGP method independent of the value of* n_c*.*

4.5.3 LacY and LacZ expression model

The genetic regulation of the lac operon was one of the earliest complex genetic regulatory mechanisms to be described in detail. The lac operon consists of a promoter, a terminator, regulator, operator and three structural genes. The two most important of these structural genes are LacY and LacZ, necessary for lactose catabolism (Krebs et al., 2010). The other constituents of the lac operon control the expression of LacY and LacZ via a series of interactions with transcription regulators in the cytosol. The number of transcription regulators in the system may be as low as ten and these transcription regulators bind to

a single 'molecule' of the DNA regulatory region (Krebs et al., 2010). Clearly, stochastic fluctuations of molecule numbers and time intervals will be important in such a system and the near-zero copy numbers of some species make this system a good test of τ-leaping methods aiming to avoid negative populations. The LacY/LacZ model, as developed by Kierzek et al. (2001), consists of 18 reactant species, one product (a result of the catabolisation of lactose which does not take part, as a reactant, in any of the system's reactions) and 22 reaction channels. A list of reaction channels and rates of the chemical kinetics is given in Table 4.6.

	Reaction Channel	Reaction rate
R_1	PLac+RNAP→ PlacRNAP	0.17
R_2	PLacRNAP→ Plac+RNAP	10
R_3	PLacRNAP→ TrLacZ1	1
R_4	TrLacZ1→ RbsLacZ+PLac+TrLacZ2	1
R_5	TrLacZ2→ TrLacY1	0.015
R_6	TrLacY1→ RbsLacY+TrLacY2	1
R_7	TrLacY2→ RNAP	0.36
R_8	Ribosome+RbsLacZ → RbsRibosomeLacZ	0.17
R_9	Ribosome+RbsLacY → RbsRibosomeLacY	0.17
R_{10}	RbsRibosomeLacZ→ Ribosome+RbsLacZ	0.45
R_{11}	RbsRibosomeLacY→ Ribosome+RbsLacY	0.45
R_{12}	RbsRibosomeLacZ→ TrRbsLacZ+RbsLacZ	0.4
R_{13}	RbsRibosomeLacY→ TrRbsLacY+RbsLacY	0.4
R_{14}	TrRbsLacZ→ LacZ	0.015
R_{15}	TrRbsLacY→ LacY	0.036
R_{16}	LacZ→ dgrLacZ	6.42×10^{-5}
R_{17}	LacY→ dgrLacY	6.42×10^{-5}
R_{18}	RbsLacZ→ dgrRbsLacZ	0.3
R_{19}	RbsLacY→ dgrRbsLacY	0.3
R_{20}	LacZ+Lactose→ LacZlactose	9.52×10^{-5}
R_{21}	LacZlactose→ product+LacZ	431
R_{22}	LacY→ lactose+LacY	14

Table 4.6: A list of the reaction channels and deterministic reaction rates for the LacY/LacZ model of Kierzek (2002)

Running the SSA for this reaction set from $t = 0$ to $t = 2000$ (the time interval used for the comparison of algorithms in Tian and Burrage, 2004) takes over 6 minutes on our computer. Since running large numbers of repeats will require a large amount of computer time we instead run the SSA once from $t = 0$ to $t = 1000$ to find an 'initialization state' for the system and run 1000 repeats of each simulation algorithm from $t = 1000$ to $t = 1001$,

as in Cao et al. (2006). The numbers of molecules of each species in the initialization state are given in Table 4.7.

Species	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Number	0	28	0	1	0	22	0	1	0	392
Species	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	
Number	59	57	1661	662	17760	15832	6814877	155	78727744	

Table 4.7: *Initial numbers of molecules of each of the 19 species in the LacY/LacZ model taken from one run of the SSA from $t = 0$ to $t = 1000$.*

The results in Table 4.8 demonstrate that the CGP method is the quickest, closely followed by the CB-depletion and CB-regeneration τ-leaping algorithms. This result remains true for a range of values of the criticality parameter n_c (see Fig. 4.4). Both the τ-leaping methods are an order of magnitude faster than the SSA. Very few of the species have the sorts of numbers of molecules for the CB methods of criticality-selection to make a considerable difference over the CGP method (see Table 4.7). Some are too large, meaning that both CGP and CB methods will classify the dependent reactions as non-critical, and some are too small, meaning that both methods are likely to classify the dependent reactions as critical. The CB methods turn out to be slightly slower since they do not gain any advantage from differential classifications of criticality (see the fourth column (titled Critical reactions) of Table 4.8). However, all three τ-leaping methods are still considerably faster than the SSA over the chosen time-period (see Table 4.8).

	CPU time (s)	Steps	Critical reactions
SSA	254.32	448740253	N/A
CGP ($n_c = 11$)	24.56	3904290	13662309
CB-depletion	26.73	3902985	13588212
CB-regeneration	27.09	3905514	13608898

Table 4.8: *A comparison, for the LacY/LacZ model of Kierzek (2002), of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval ($t \in [1000, 1001]$) from a common initial condition (see Table 4.7) for 1000 repeats using the SSA and three τ-leaping methods.*

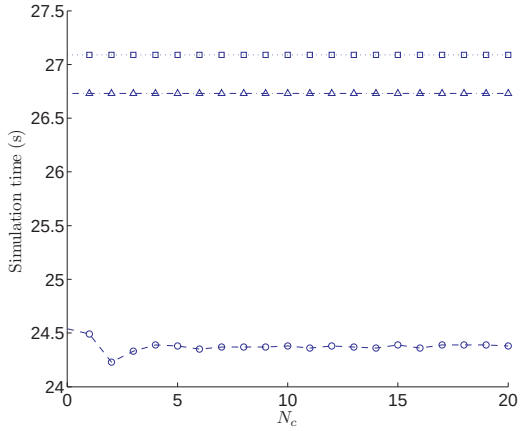


Figure 4.4: *A comparison of simulation times for 1000 repeats of each of the three τ-leaping methods for the LacY/LacZ model over a range of values of the parameter $n_c = 0, \ldots, 20$. The results of the different methods are denoted as in Fig. 4.2. For all values of n_c the CGP method is marginally quicker than either of the CB methods.*

4.5.4 A one-dimensional position-jump process

The final system we consider is a familiar one. We take, as a representative example of the simulations considered in Chapters 2, and 3, the stochastic simulation of diffusion via a position-jump process. Explicitly, we divide the unit domain $[0, 1]$ into $k = 50$ intervals of equal length and implement migration by allowing cells to move left and right from their current interval with constant transition rate, $d = 1$. We assume there is no cell death or division on the time-scale of interest so that, in combination with the zero-flux boundary conditions, the total cell numbers remain constant. This example reduces to a system of $k = 50$ species reacting through $2 \times k - 2 = 98$ reaction channels (cells moving left and right from each interval except for at the end intervals where cells are only allowed to move in one direction to implement zero flux boundary conditions). The number of cells in each interval corresponds to the number of molecules of each species. As such, all reactions are isomerisations and the propensity functions depend only on the transition rate, d, and the number of cells in the corresponding interval. We can write the reaction system as follows:

$$\begin{aligned} S_1 &\xrightarrow{d} S_2, \\ S_1 \xleftarrow{d} S_2 &\xrightarrow{d} S_3, \\ &\vdots \end{aligned}$$

$$S_{k-2} \xleftarrow{d} S_{k-1} \xrightarrow{d} S_k,$$
$$S_k \xrightarrow{d} S_{k-1}. \tag{4.5.5}$$

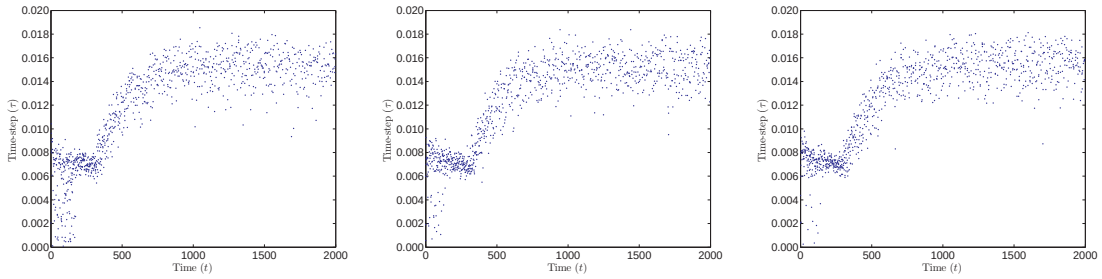
We consider a variety of conditions to determine how the algorithms perform against each other. In the first example we place 5000 cells in the first interval of the domain (i.e. $S_1(0) = 5000$ and $S_i(0) = 0$ for $i = 2, \ldots, 50$). The system is run over the time interval $[0, 2000]$ and results are given in Table 4.9.

	CPU time (s)	Steps	Critical reactions	Mean HDE
SSA	34834	2.72×10^9	N/A	N/A
CGP ($n_c = 11$)	7542	1.71×10^8	5.20×10^8	0.120
CB-depletion	7310	1.66×10^8	6.86×10^8	0.121
CB-regeneration	7160	1.65×10^8	3.31×10^8	0.122

Table 4.9: *A comparison, for model* (4.5.5)*, of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval ($t \in [0, 2000]$) from a common initial condition ($S_1(0) = 5000$ and $S_i(0) = 0$ for $i = 2, \ldots, 50$) for 1000 repeats using the SSA and three τ-leaping methods. The mean HDE of the species numbers at the end of the simulation is also compared.*

Table 4.9 demonstrates the improved efficiency of the CB methods in comparison to the CGP method. The simulation time is reduced by approximately 3-5% and accuracy remains comparable (see Figs. 4.7 (c) and (d)). This is evidenced by the smaller number of steps taken by the CB methods. Importantly our method does this in a parameter-independent manner, which removes the possibility of sub-optimal performance through incorrect parameter choice. Although it appears that the CGP has fewer critical reactions, this is an artefact of our recording procedure: critical reactions are not recorded during the time the τ-leaping simulations spend implementing the SSA (when the proposed time-step, τ', given by non-critical reactions is smaller than twice the average time-step of the SSA, $2/a_0$, (see Step **(2d)** of the algorithm given in Section 4.4)). The CGP algorithm spends far more time implementing the SSA, as evidenced by the increased simulation time and number of steps in comparison to the CB methods. Similar reasoning applies to the results in Table 4.10.

Fig. 4.5 displays the size of the time-step, τ, against the time it was chosen. Early in the simulations there is usually at least one species with low molecular numbers (i.e. an



(a) (b) (c)

Figure 4.5: *For a simulation of the one-dimensional cell migration reactions of model* (4.5.5) *the values of the chosen time-step, τ, vs. the time it was chosen, for (a) the CGP τ-selection procedure, (b) the CB-depletion procedure and (c) the CB-regeneration procedure. τ values were recorded every 1000^{th} leap (up to a maximum of 1000 samples). Other conditions are as given in Table 4.9.*

interval with a small number of cells) and the CGP method marks a large proportion of reactions dependent on low copy-number species as being critical, despite their minuscule probability of driving the species number negative. This results in a choice of small τ (see Fig. 4.5 (a)). However, the CB methods are freer to classify reactions dependent on low copy number species as non-critical. Far fewer small values of τ are selected in the initial stages of the simulations using the CB methods (see Fig. 4.5 (b) and 4.5(c)) allowing the simulations to run to completion in a shorter time (see Table 4.9).

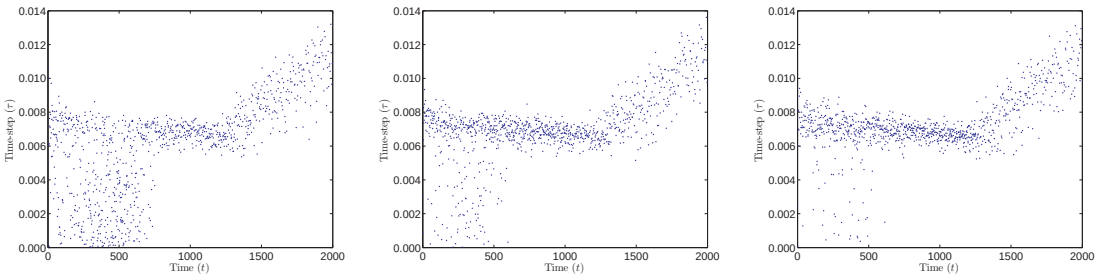
The difference between the criticality classification methods is even more stark when we simulate the same reaction model with slightly altered conditions. Consider doubling the spatial resolution of the interval (by doubling the number of intervals and halving the interval width) i.e. $k = 100$. Consequently, in order to maintain the average number of cells per interval, we also double the initial number of cells i.e. $S_1(0) = 10000$ and $S_i(0) = 0$ for $i = 2, \ldots, 100$. We now have a reaction system of 100 chemical species reacting through 198 reaction channels. Table 4.10 highlights the improved efficiency of the CB methods in comparison to the CGP method whilst at the same time maintaining its advantage over the SSA. This further demonstrates the robustness of the CB methods as a parameter-independent choice for stochastic acceleration.

Fig. 4.6 shows the analogous comparison of time-step, τ, to Fig. 4.5 for the revised conditions described above. The CGP method is overly restrictive in its choice of time-step for an even larger proportion of the simulation than in Fig. 4.5. This trend continues as

	CPU time (s)	Steps	Critical reactions	Mean HDE
SSA	13538	4.00×10^9	N/A	N/A
CGP ($n_c = 11$)	4497	3.27×10^7	5.53×10^8	0.353
CB-depletion	3851	2.78×10^7	5.96×10^8	0.351
CB-regeneration	3762	2.71×10^7	2.79×10^8	0.352

Table 4.10: *A comparison, for model* (4.5.5) *(now with* $k = 100$ *intervals), of the total CPU time, number of steps taken and number of reactions deemed to be critical over a common time interval (*$t \in [0, 2000]$*) from a common initial condition (*$S_1(0) = 10000$ *and* $S_i(0) = 0$ *for* $i = 2, \ldots, 100$*) for 100 repeats using the SSA and three* τ*-leaping methods. The mean HDE of the species numbers at the end of the simulation is also compared.*

we increase the spatial resolution of the system until the CGP method is overly restrictive in time-step choice for the duration of the simulation, further increasing the discrepancy in times between the CGP and CB τ-leaping methods.



(a) (b) (c)

Figure 4.6: *For a simulation of the one-dimensional cell migration reactions of model* (4.5.5) *the values of the chosen time-step,* τ*, vs. the time it was chosen, for (a) the CGP* τ*-selection procedure, (b) the CB-depletion procedure and (c) the CB-regeneration procedure.* τ *values were recorded every* 1000^{th} *leap (up to a maximum of 1000 samples). Other conditions are as in Table 4.10.*

As with the other model systems, in Figs. 4.7 (a) and (b), we provide a comparison of the speeds of three τ-leaping algorithms for varying values of the criticality parameter, n_c. The speeds of the CB methods are shown to be either comparable or superior to the CGP method as n_c varies.

In order to compare the accuracy of the τ-leaping methods the distributions of the final numbers of each species are compared quantitatively, using the HDE metric, to the equivalent distributions given by the SSA. The value found for each species is plotted against species number in Figs. 4.7 (c) and (d). The HDE for each method is similar for each species. The similarity of the mean HDE, across species, given in the final column of Tables 4.9 and

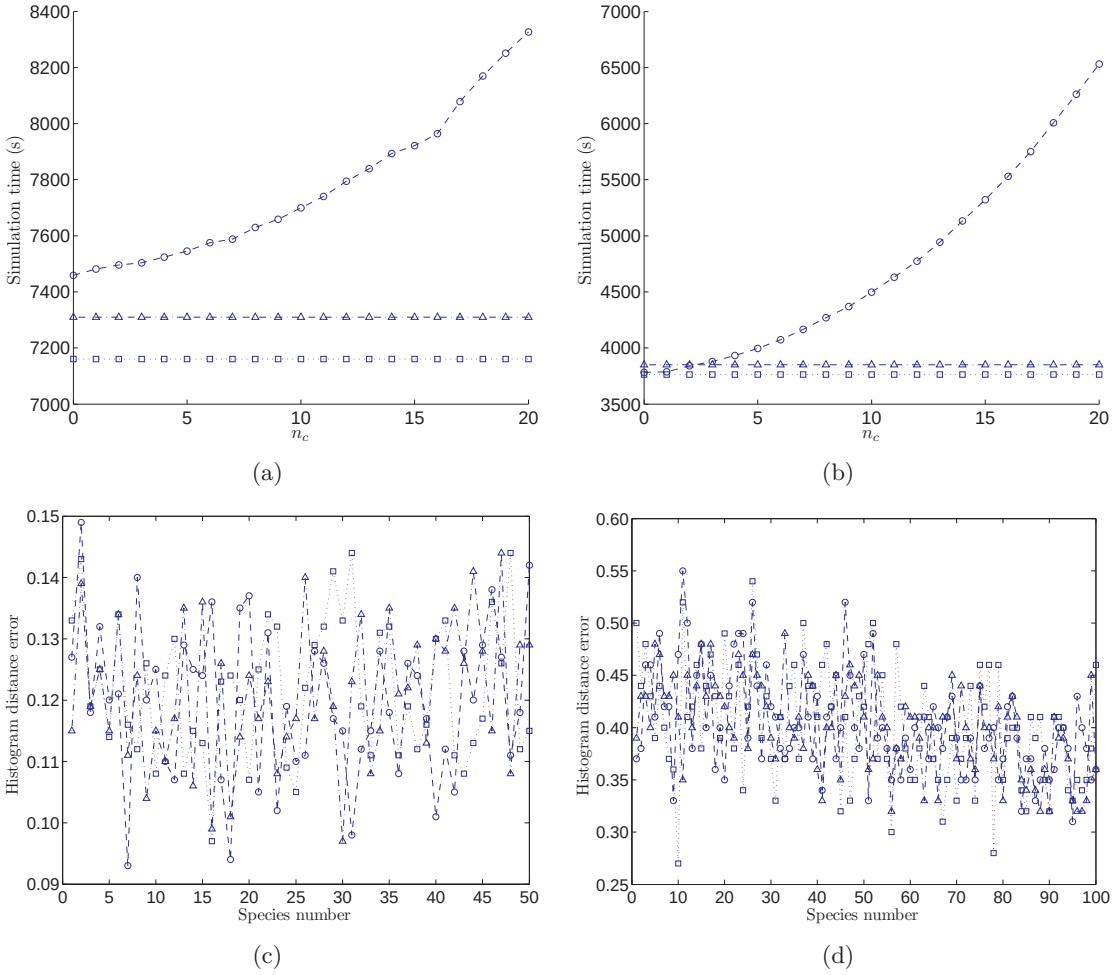


Figure 4.7: *A comparison of simulation times and accuracies for each of the three τ-leaping methods for both of the initial conditions described above: (a) and (c) $k = 50$, $S_1(0) = 5000$ and (b) and (d) $k = 100$, $S_1(0) = 10000$. The simulation times are compared over a range of values of the parameter $n_c = 0, \ldots, 20$ and accuracies for a representative value, $n_c = 11$. The results of the different methods are denoted as in Fig. 4.2. In (a) the CB methods are significantly faster than the CGP method. Times given are for 1000 repeats of the simulation. The higher the value of n_c the less efficient the CGP method becomes in comparison to the CB methods. In (b) we see that for very low values of n_c the CGP and CB methods are of comparable speed, however, for larger values of n_c the CB methods are significantly faster. Times given are for 100 repeats of the simulation. (c) and (d) Comparisons of the accuracies of the three methods for each species using the HDE metric.*

4.10, demonstrates that the methods are of comparable accuracy.

As a final demonstration of the efficiency of the CB methods, Fig. 4.8 shows the average time taken per simulation for varying numbers of cells initially positioned in the first interval of 50. The CB-depletion method is quicker in all simulations and in some cases substantially so.

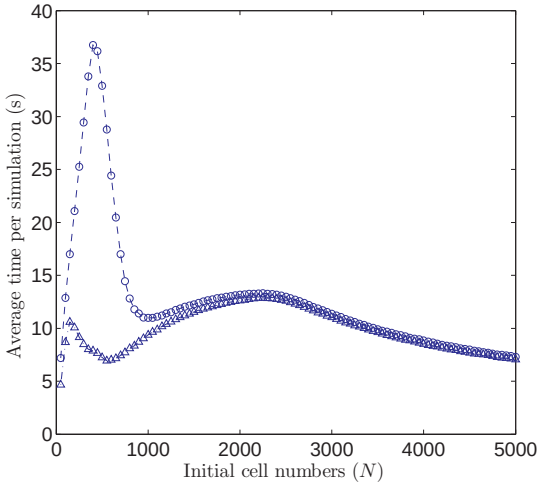


Figure 4.8: *Average CPU time taken (in seconds) given different numbers of cells initially positioned in the first interval on a domain partitioned into $k = 50$ intervals. The initial cell numbers range from $N = 50$ to $N = 5000$ in increments of 50. All times are averaged over 10 repeats. Circles with a dashed line represent the CGP method ($n_c = 11$) whilst triangles with a dot-dashed line the CB-depletion method. The CB-regeneration method is of comparable speed to the CB-depletion method and, for clarity, is not shown.*

Summary

We began this chapter by detailing a variety of different methods for implementing ESSAs. Considerable increases in speed can be achieved using intelligent algorithm choices, but it must be stressed that there is no absolute 'best' method. Different methods will perform with varying levels of success depending on the dynamics and initial conditions of the reaction system.

We may find that not even the most efficient SSA is fast enough to produce reasonable simulation times. In this case it may be necessary to sacrifice the accuracy of the simulation in order to accelerate the algorithms. The CGP method for determining criticality takes a one-size-fits-all approach by allocating a critical value, n_c, for L_j values below which reaction channel j is classified as critical. As demonstrated in Section 4.4, by considering a simple degradation reaction, the speed and accuracy of the CGP method is dramatically dependent on the value of this parameter. If chosen too high it can be overly restrictive. If chosen too low there is a danger of species numbers becoming negative. We have presented a more versatile method of classifying criticality of reaction channels. We consider the

species numbers and calculate how likely it is that a species will be driven negative, given the reactions that deplete and replenish its numbers. We can then classify an individual species as critical and consequently all reactions dependent on that species are also classified as critical. Our confidence-based approach allows us to be more discriminatory about which species, and hence reactions, are considered critical. Only those species having at least a pre-specified probability of becoming negative are classified as critical. This flexible approach leads to the classification of fewer reactions as critical, enabling the simulation to select more and larger time-steps from the non-critical regime. This method does not face the problem of having to tune the value of n_c for the relevant simulation so we need not worry that we will send the population of a reaction species negative through incaution. This makes our method a consistent choice, robust to a wide variety of initial conditions.

Test simulations carried out on four model reaction systems have indicated that the CB τ-leaping procedures can be orders of magnitude faster than the CGP method in some situations. It appears that in all situations the CB methods will be at least of comparable speed to the CGP method with comparable accuracy. It is important to emphasise that the CB methods achieve this optimal or near-optimal performance (optimal with respect to the the most efficient choice of n_c in the CGP method) without the necessity for the tuning parameter, n_c. This will obviate the need for running expensive test simulations to optimise the value of n_c.

We suggest that our CB method of species, and hence reaction criticality classification, is sufficient to increase the speed of many systems of reactions and will, at the very least, be a robust choice of algorithm in the face of widely varying species populations. The confidence-based method of choosing criticality will perform particularly well when there are some species with large numbers of molecules and some with smaller (formerly CGP sub-critical (i.e. $S_i < n_c$)) numbers. This will confer the benefits of τ-leaping whilst avoiding the handicap of classifying too many reaction channels as critical and decelerating the algorithm. Such a situation often occurs during our cell migration simulations, depending on the initial condition and the sensing mechanism used.

As we move into higher dimensions (and especially when considering domain growth) in the following chapter, the increased number of cells and domain elements will accentuate

the increasing importance of making our simulations as fast as possible in order to achieve realistic simulation times. The SSA efficiency adaptations and $\tau-$leaping improvements outlined in this chapter will help us to achieve this goal.

Chapter 5

Simulating cell migration in two dimensions

Overview

In addition to the problem of increased computational complexity considered in the previous chapter, moving into higher dimensions presents many new challenges, which we must overcome in order to maintain the integrity of our equivalence framework. Among them are the choice of domain discretisation, domain shape and the appropriate method for implementing domain growth in higher dimensions. In this chapter we will attempt to tackle these issues.

In Section 5.1 we will show that a series of regular domain discretisations (squares, triangles and hexagons) all support equally valid lattices on which the cells of our individual-based model can move. Indeed we are able to show that the cell densities for these tessellations correspond to the same PDEs in the continuum limit. Another important consideration is the underlying shape of the domain that our equivalence framework can incorporate. We demonstrate, by considering a circular arena tessellated with squares, that non-square domains can be incorporated into the equivalence framework.

This stationary circular domain turns out to be useful when attempting to incorporate domain growth in two dimensions into our equivalence framework. In order to achieve

this, in Section 5.2 we test a range of cluster aggregation models (mainly drawn from the tumour growth literature) in an attempt to grow regularly shaped domains using random and realistic growth processes. In particular we focus on the Eden model (Eden, 1958), a canonical example of near-circular cluster growth. By studying this model we are able to introduce concepts used for model classification and analysis (resp.), including universality classes and anisotropy metrics (resp.). We use the canonical example of the Eden model in order to compare our implementations of these metrics to those in the literature.

We find that even growing something as simple as a circular domain is fraught with problems such as the anisotropy induced by regular lattices. We consider a variety of methods from the literature in order to combat these problems and highlight the downfalls of certain anisotropy metrics which can themselves be affected by the underlying lattice. After reviewing a large number of candidate models, in Section 5.3, we implement our own adaptation which we show to be isotropic and, in the low noise limit, to give rise to near-perfect circular clusters.

In Section 5.4 we use this model for domain growth in our stochastic cell migration simulations and compare the results with those of an appropriate PDE on a radially growing domain.

Movies accompanying all the simulations mentioned in this chapter can be found at http://people.maths.ox.ac.uk/yatesc/Thesis_movies/Chapter_5.

5.1 Stationary domain

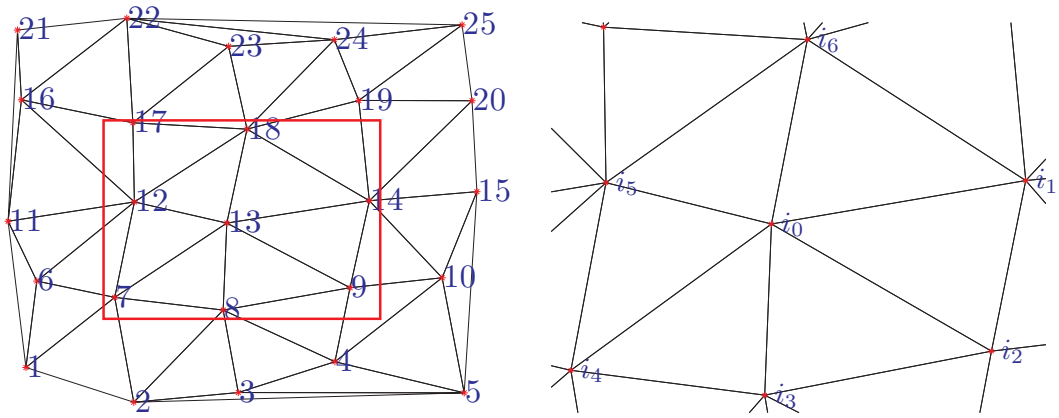
The first challenges thrown up by higher dimensions are those related to the appropriate choice of domain discretisation. How does the domain discretisation affect the PDE that we derive? How must the diffusion coefficients in the PDE model be altered in order to match the results of the stochastic simulations?

We consider each of the three regular domain tessellations in two dimensions (squares, triangles and hexagons) (Coxeter, 1973), deriving the appropriate PDE counterpart in each case. We go on to discuss the possibility of semi-regular tessellations or even irregular lattices upon which the cells can move. In all the regular tessellations that follow we assume that

cells reside at the geometric centre of the elements of the tessellation and move along the edges of the dual of that tessellation. For example, cells in a square domain tessellation move on a square lattice (since the square tessellation is self-dual) but cells in a regular triangular tessellation move on a regular hexagonal lattice, and *vice versa*.

5.1.1 A general lattice

Consider cells moving on a general (possibly irregular) lattice. Associated with each node of the lattice is a domain element or area which encloses the node. The set of domain elements forms a tessellation of the domain. Cells reside at the nodes and are able to move between elements along the edges of the lattice. We introduce a global numbering system in which all the nodes of the lattice take a number from 1 to k (see Fig. 5.1 (a)). We also introduce a local labelling system whereby the neighbouring nodes of a central node with global label $i = i_0$ are labelled $i_1, i_2, \ldots, i_{J(i_0)}$, where $J(i_0)$ is the number of neighbours of node i_0 (see Fig. 5.1 (b)). The vectors connecting vertex i_0 to its neighbours are labelled



(a) (b)

Figure 5.1: *(a) Global node labelling from* 1 *to* $k(= 25)$*. (b) Local node labelling starting with the central node,* i_0*, and proceeding clockwise to* $i_{J(i_0)}$*, where* $J(i_0)\,(= 6)$ *is the number of neighbours of node* i_0*. The domain plotted in (b) corresponds to the area enclosed by the red rectangle in (a).*

locally as $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{J(i_0)}$ and, in two dimensions, can be expressed in Cartesian coordinate form as $(x_1, y_1), (x_2, y_2), \ldots, (x_{J(i_0)}, y_{J(i_0)})$. We denote the transition rates to go from node i_0 to one of its neighbours, i_j, for $j = 1, \ldots, J(i_0)$, as $T_{i_0}^{i_j}$ and the transition rate from that neighbour back to node i_0 as $T_{i_j}^{i_0}$. A vector of the number of cells at each node can be

written as follows:

$$\boldsymbol{n} = (n_1, n_2, \ldots, n_{i0}, \ldots, n_{i_{J(i_0)}}, \ldots, n_k), \tag{5.1.1}$$

where we have replaced some of the global labels with their local counterparts (with respect to central node i_0) for ease of notation in what follows[1]. Define operators $M_j^{i_0}$ for central node i_0 and neighbouring nodes i_j for $j = 1, \ldots, J(i_0)$ by their actions on the vector $\boldsymbol{n}$, as follows:

$$M_j^{i_0}\boldsymbol{n} = (n_1, n_2, \ldots, n_{i0} + 1, \ldots, n_{i_j} - 1, \ldots, n_{i_{J(i_0)}}, \ldots, n_k). \tag{5.1.2}$$

Operator $M_j^{i_0}$ takes a cell from node i_j and moves it to node i_0. As usual, let $P(\boldsymbol{n}, t)$ denote the probability that the cells take the configuration $\boldsymbol{n}$ at time t. Then we can write down an RDME as follows:

$$\frac{\partial P(\boldsymbol{n}, t)}{\partial t} = \sum_{i_0=1}^{k} \sum_{j=1}^{J(i_0)} \left\{ (n_{i_0} + 1) T_{i_0}^{i_j} P(M_j^{i_0}\boldsymbol{n}) - n_{i_0} T_{i_0}^{i_j} P(\boldsymbol{n}, t) \right\}. \tag{5.1.3}$$

In order to find an equation for the evolution of the mean number of cells at general node l_0 in terms of the mean numbers of cells at its neighbouring nodes we multiply equation (5.1.3) by n_{l_0} and sum over all the possible values the vector of cell densities can take. Upon simplification we derive the following equations:

$$\frac{\partial \langle n_{l_0} \rangle}{\partial t} = \sum_{j=1}^{J(l_0)} \left\{ \langle n_{l_j} \rangle T_{l_j}^{l_0} - \langle n_{l_0} \rangle T_{l_0}^{l_j} \right\}, \qquad \text{for} \qquad l_0 = 1, \ldots, k, \tag{5.1.4}$$

where $\langle n_{l_0} \rangle$ denotes the mean number of cells at node l_0 and $\langle n_{l_j} \rangle$ is defined similarly. In order to find an equation for the evolution of cell *density* rather than cell *number* we divide through by the total area/volume of the element associated with each node[2] (denoted A_{l_j}

[1] The global ordering we have used in equation (5.1.1) is purely for notational convenience. Generally, locally neighbouring nodes will not appear consecutively in the globally ordered vector of cell numbers.

[2] The area/volume can be defined using the Voronoi domain partition in two/three dimensions given the grid of points upon which cells move. Alternatively element boundaries can be defined first and the points on which cells move can be considered as the centroids of these elements in analogy to the situation in one dimension.

for node l_j). Equation (5.1.4) becomes

$$\frac{\partial u_{l_0}}{\partial t} = \frac{1}{A_{l_0}} \sum_{j=1}^{J(l_0)} \left\{ u_{l_j} A_{l_j} T_{l_j}^{l_0} - u_{l_0} A_{l_0} T_{l_0}^{l_j} \right\}, \qquad \text{for} \qquad l_0 = 1, \ldots, k, \tag{5.1.5}$$

where $u_{l_0} = \langle n_{l_0} \rangle / A_{l_0}$ is the mean cell density in interval l_0 with u_{l_j} defined similarly for $j = 1, \ldots, J(l_0)$. Given transition rates and the appropriate grid geometry, by Taylor expanding these densities about l_0 in equation (5.1.5), we can determine whether the individual-level model corresponds to a population-level PDE. Alternatively we can use equation (5.1.5) to derive the appropriate transition rates corresponding to a particular PDE[3]. For example, if we want to find transition rates which correspond to the diffusion equation in two dimensions then, after Taylor expansion of the appropriate terms, we can equate the coefficients of the derivatives of u_{l_0} to find a system of equations which the transition rates must satisfy:

$$\sum_{j=1}^{J(l_0)} \left\{ A_{l_j} T_{l_j}^{l_0} - A_{l_0} T_{l_0}^{l_j} \right\} = 0, \tag{5.1.6}$$

$$\sum_{j=1}^{J(l_0)} A_{l_j} T_{l_j}^{l_0} x_j = 0, \tag{5.1.7}$$

$$\sum_{j=1}^{J(l_0)} A_{l_j} T_{l_j}^{l_0} y_j = 0, \tag{5.1.8}$$

$$\sum_{j=1}^{J(l_0)} A_{l_j} T_{l_j}^{l_0} x_j^2 = 2DA_{l_0}, \tag{5.1.9}$$

$$\sum_{j=1}^{J(l_0)} A_{l_j} T_{l_j}^{l_0} x_j y_j = 0, \tag{5.1.10}$$

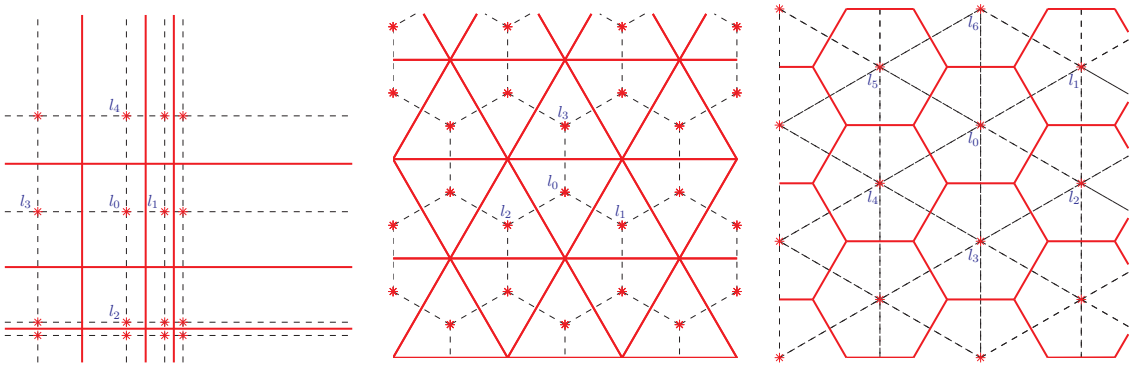
$$\sum_{j=1}^{J(l_0)} A_{l_j} T_{l_j}^{l_0} y_j^2 = 2DA_{l_0}. \tag{5.1.11}$$

Equation (5.1.6) corresponds to coefficients of u_{l_0}, equations (5.1.7) and (5.1.8) to coefficients of the first derivatives and equations (5.1.9)-(5.1.11) to coefficients of the second derivatives.

We now consider some special cases where we can derive the transition rates explicitly. For all the simulations in the subsequent sections we have employed zero-flux boundary

[3]Note that this approach for deriving appropriate transition rates on an unstructured mesh is significantly different to that of Engblom et al. (2009). They use a Finite Element Method (FEM) discretisation of the diffusion equation, mass lumping the mass matrix so that it is easily invertible. The resulting discretisation is then directly compared to equations (2.1.4)-(2.1.6) of Chapter 2 for the evolution of mean cell numbers.

conditions on the boundaries of the domains.



(a) (b) (c)

Figure 5.2: *For each tessellation the element boundaries are demarcated in red, the grid on which the cells move is displayed using a dashed black line and the points at which the cells reside in each element are represented by a red star (∗). (a) The rectangular tessellation. The grid upon which cells move is also rectangular. (b) The triangular tessellation. The grid upon which cells move is hexagonal. (c) The hexagonal tessellation. The grid upon which cells move is triangular.*

5.1.2 Rectangular tessellation

If the domain is tessellated using rectangles then the grid upon which the cells move is also rectangular (see Fig 5.2 (a)). Using equations (5.1.6)-(5.1.11) we can derive transition rates for the diffusion equation consistent with this domain partition:

$$T_{l_0}^{l_j} = \frac{2D}{\boldsymbol{r}_j \cdot \left(\boldsymbol{r}_j - \boldsymbol{r}_{(j+2)(\mathrm{mod}4)}\right)}, \tag{5.1.12}$$

$$T_{l_j}^{l_0} = \frac{2DA_{l_0}}{\boldsymbol{r}_j \cdot \left(\boldsymbol{r}_j - \boldsymbol{r}_{(j+2)(\mathrm{mod}4)}\right) A_{l_j}}, \qquad \text{for} \qquad j = 1, \ldots, 4, \tag{5.1.13}$$

where we recall that $\boldsymbol{r}_j$ is the vector connecting local node l_0 to its neighbour l_j. It is interesting to note that, as in the one-dimensional non-uniform grid, it is necessary for the domain to be tessellated using a Voronoi domain partition in order for the transition rates to be self-consistent (see Appendix B.1). In the special case of a regular grid consisting of squares, the transition rates are all the same:

$$T = \frac{D}{h^2}, \tag{5.1.14}$$

where h is the distance between neighbouring nodes and consequently also the length of the sides of the squares. Fig. 5.3 displays a snapshot comparison of individual-level and population-level models of diffusion on a square domain tessellated with squares. Fig. 5.3 (c) shows a low HDE indicating that the individual-level model and the population-level model correspond closely.

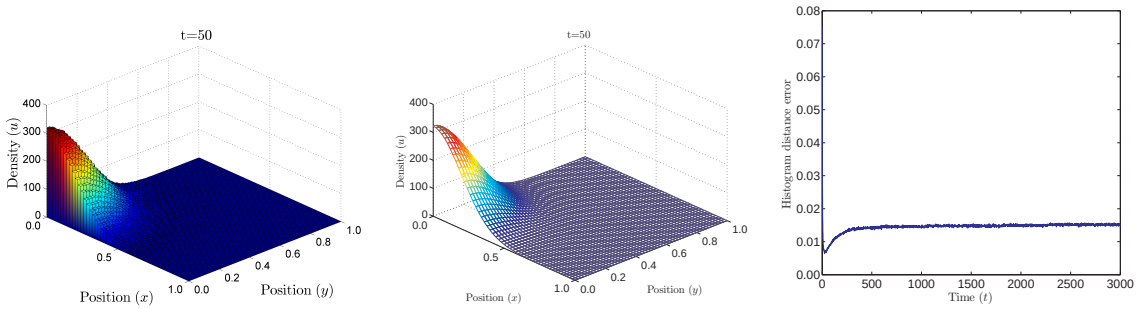
Consider the corresponding transition rates for average sensing:

$$T_{l_0}^{l_j} = \frac{2\left(D^l s(\boldsymbol{r}_0) + D^n s(\boldsymbol{r}_0 + \boldsymbol{r}_j)\right)}{\boldsymbol{r}_j \cdot \left(\boldsymbol{r}_j - \boldsymbol{r}_{(j+2)(\mathrm{mod}4)}\right)}, \tag{5.1.15}$$

$$T_{l_j}^{l_0} = \frac{2\left(D^l s(\boldsymbol{r}_0 + \boldsymbol{r}_j) + D^n s(\boldsymbol{r}_0)\right) A_{l_0}}{\boldsymbol{r}_j \cdot \left(\boldsymbol{r}_j - \boldsymbol{r}_{(j+2)(\mathrm{mod}4)}\right) A_{l_j}}, \qquad \text{for} \qquad j = 1, \ldots, 4, \tag{5.1.16}$$

where $\boldsymbol{r}_0$ is defined to be the absolute position of local node l_0 and $s(\boldsymbol{x})$, as in previous chapters, is the signalling profile, which is a function of space but not time. We can use equation (5.1.5) to verify, upon taking the appropriate limits, that we regain the general average sensing PDE in two dimensions:

$$\frac{\partial u}{\partial t} = \left(D^l \nabla^2 \left(s(\boldsymbol{x})u\right) + D^n \nabla \cdot \left(s(\boldsymbol{x})\nabla u - u \nabla s(\boldsymbol{x})\right)\right). \tag{5.1.17}$$



(a) (b) (c)

Figure 5.3: *A comparison of the individual-level (with square domain tessellation) and population-level models of cells diffusing. $N = 50000$ cells are initialised in the left-most interval of the domain. The domain is tessellated by k^2 squares, where $k = 50$. (a) The histograms represent the density of cells in each square of the domain tessellation in the individual-level model. (b) The mesh surface represents the solution of the diffusion equation in two dimensions, with diffusion coefficient $D = h^2$, where $h = 1/k$. (c) The evolution of the HDE. As in previous chapters, the initially large value of the HDE is due to the manner in which we are forced to implement the initial condition in the continuum model. For a movie comparing the density of the two models as time evolves please see MOVIE_1.avi.*

5.1.3 Triangular tessellation

On a domain tessellated with equilateral triangles (with sides of length h) the areas of the elements are the same as each other (see Fig. 5.2 (b)), which simplifies equations (5.1.6)-(5.1.11) significantly. The distance between grid points (at the centre of each triangle) is $\Delta x = h/\sqrt{3}$. Equations (5.1.6)-(5.1.11) again allow us to derive the transition rates which recapture the diffusion equation:

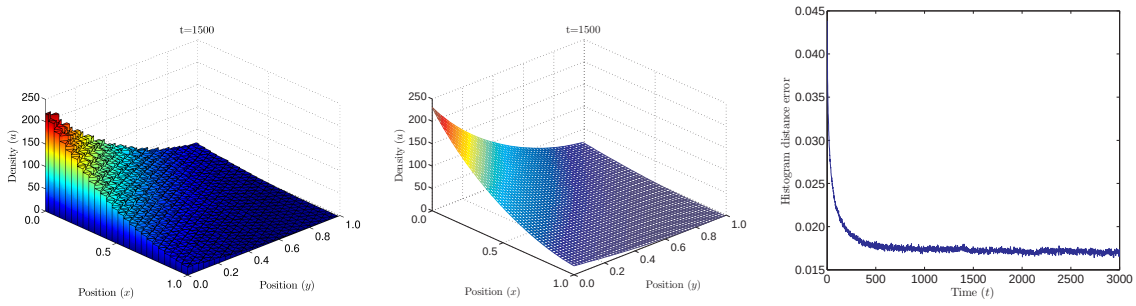
$$T_{l_0}^{l_j} = T_{l_J}^{l_0} = \frac{4D}{3(\Delta x)^2} = \frac{4D}{h^2}, \qquad \text{for} \qquad j = 1, 2, 3. \tag{5.1.18}$$

We can also make use of equation (5.1.5) to show that the transition rates

$$T_{l_0}^{l_j} = \frac{4\left(D^l s(\boldsymbol{r}_0) + D^n s(\boldsymbol{r}_0 + \boldsymbol{r}_j)\right)}{h^2}, \tag{5.1.19}$$

$$T_{l_j}^{l_0} = \frac{4\left(D^l s(\boldsymbol{r}_0 + \boldsymbol{r}_j) + D^n s(\boldsymbol{r}_0)\right)}{h^2}, \qquad \text{for} \qquad j = 1, 2, 3, \tag{5.1.20}$$

correspond to the general signal sensing PDE (5.1.17).



(a) (b) (c)

Figure 5.4: *A comparison of the individual-level (with triangular domain tessellation) and population-level models of cells moving according to non-local signal sensing. The signalling profile is a decaying exponential from the left-most corner of the domain* $s(x, y) = \exp(-2(x + y))$. $N = 50000$ *cells are initialised with an exponentially decaying profile from the left-most corner of the domain. (a) The histograms represent the density of cells in each element of the domain tessellation in the individual-level model. (b) The mesh surface represents the solution of PDE* (5.1.17) *with* $D^l = 0$ *and* $D^n = h^2/4$, *where* $h = 2/49$ *is the length of the sides of the triangles which tessellate the domain in the individual-based model. (c) The evolution of the HDE. For a movie comparing the density of the two models as time evolves please see MOVIE_2.avi.*

Fig. 5.4 displays a snapshot comparison of the densities of the individual-level model of non-local sensing (with triangular domain tessellation) and the equivalent population-level

model. They can be seen to be qualitatively similar. Fig. 5.4 (c) shows the HDE which is low, indicating good quantitative agreement between the models.

5.1.4 Hexagonal tessellation

Hexagonal domain tessellations have been used previously by Deroulers et al. (2009) in a discrete-continuum modelling framework in an attempt to find a continuum representation which captures the contact-mediated migrations of cancer cells. They chose a hexagonal tessellation since it introduced the least anisotropy of any of the possible regular tessellations (Frisch et al., 1986; Wolfram, 1986). Here we consider hexagons of side h. The distance between the centres of these hexagons is, therefore, $\Delta x = \sqrt{3}h$. Choosing transition rates

$$T_{l_0}^{l_j} = T_{l_j}^{l_0} = \frac{2D}{3(\Delta x)^2} = \frac{2D}{9h^2}, \qquad \text{for} \qquad j = 1, \ldots, 6, \tag{5.1.21}$$

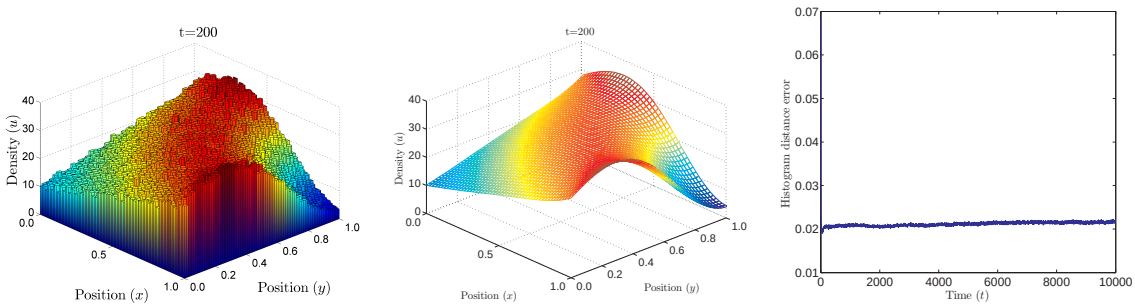
allows us to regain the diffusion equation in the continuum limit. Similarly the transition rates

$$T_{l_0}^{l_j} = \frac{2\left(D^l s(\boldsymbol{r}_0) + D^n s(\boldsymbol{r}_0 + \boldsymbol{r}_j)\right)}{9h^2}, \tag{5.1.22}$$

$$T_{l_j}^{l_0} = \frac{2\left(D^l s(\boldsymbol{r}_0 + \boldsymbol{r}_j) + D^n s(\boldsymbol{r}_0)\right)}{9h^2}, \qquad \text{for} \qquad j = 1, \ldots, 6, \tag{5.1.23}$$

correspond to the general average sensing PDE (5.1.17). It is interesting to note that the concept of local sensing in the individual-level model corresponds to the same population-level PDE independent of the domain tessellation (and hence the number of neighbours of each element). The same is true for the correspondence between the non-local and average signal sensing mechanisms and each of their equivalent PDEs.

Fig. 5.5 contrasts snapshots of the individual-level model (with hexagonal domain tessellation) and the population-level model. The two models give qualitatively similar cell density profiles. Fig. 5.5 (c) shows a low HDE, indicating good quantitative agreement between the densities generated by the two models.



(a) (b) (c)

Figure 5.5: *A comparison of the individual-level (with hexagonal domain tessellation) and population-level models of cells moving according to average sensing. The signalling profile is a decaying exponential from the left-most corner of the domain* $s(x, y) = \exp(-2(x + y))$. *The value of the local and non-local signal sensing strengths are* $d^l = 5$ *and* $d^n = 1$, *respectively. All cells are initialised in the left-most interval of the domain. (a) The histograms represent the density of cells in each element of the domain tessellation in the individual-level model. (b) The mesh surface represents the solution of PDE* (5.1.17) *with* $D^l = 45h^2/2$ *and* $D^n = 9h^2/2$, *where* $h = 2/149$ *is the length of the sides of the hexagons which tessellate the domain in the individual-level model. (c) The evolution of the HDE. For a movie comparing the density of the two models as time evolves please see MOVIE_3.avi.*

5.1.5 Octagon-square tessellation

It is also possible to find explicit formulations for the appropriate transition rates on domains covered using semi-regular tessellations such as the octagon-square tessellation. Assuming both the squares and the octagons have sides of length h then the distance between the centre of a square and the centre of a neighbouring octagon is $\Delta x_{so} = h(1 + \sqrt{2})$ and the distance between the centres of two neighbouring octagons is $\Delta x_{oo} = h(2 + \sqrt{2})$. As before, employing equations (5.1.6)-(5.1.11) for the square-centred stencil and the octagon-centred stencil sequentially we can derive the transition rates which correspond to the diffusion equation in the continuum limit:

$$T_s^o = \frac{D}{h^2(1+\sqrt{2})^2}, \tag{5.1.24}$$

$$T_o^s = \frac{DA_s}{A_o h^2(1+\sqrt{2})^2}, \tag{5.1.25}$$

$$T_o^o = \frac{D}{2h^2(1+\sqrt{2})^2}\left(1 - \frac{A_s}{A_o}\right), \tag{5.1.26}$$

where $A_s = h^2$ represents the area of the square and $A_o = 2h^2(1 + \sqrt{2})$ the area of the octagon. T_s^o represents the transition rate from the square to the octagon, T_o^s from octagon

to square and T_o^o from octagon to octagon.

5.1.6 Square tessellation on a circular domain

We now look specifically at the comparison between the individual-level model and the population-level model on a circular domain since it will be important when we come to considering growing domains later in this chapter. We tessellate the approximately circular domain with squares. Squares are of length $h = 1/49$ and a square is included as part of the domain if the coordinates of its centre (i.e. the node of the underlying lattice) lie inside a circle of radius $1/2$. The central square is centred on the origin. The relative difference between the area of a circle of radius $1/2$ and the area of our 'on-lattice' circle is small: $3.89 \times 10^{-2}\%$. The transition rates for the circular domain tessellated with squares are as given in equation (5.1.14) of Section 5.1.2. In order to compare the individual-level model with the population-level model we solve the appropriate PDE by transforming to radial coordinates. The diffusion equation, for example, becomes

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right), \qquad \text{on} \qquad r \leq L, \tag{5.1.27}$$

where r, the radial distance from the origin, and θ, the anti-clockwise angle from the x-axis, are the usual circular polar coordinates. In order to conserve mass and maintain continuity in first derivatives at the origin, it can easily be shown that the appropriate boundary conditions are

$$\left.\frac{\partial u}{\partial r}\right|_{r=0,L} = 0. \tag{5.1.28}$$

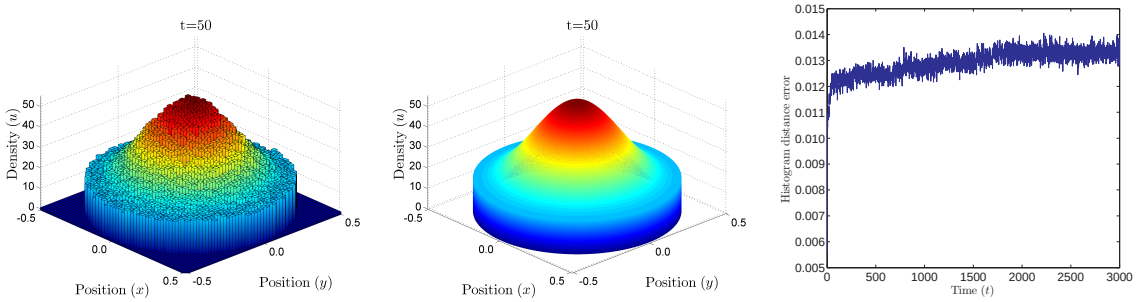
In combination with radially symmetric initial conditions, these boundary conditions ensure that the solution is independent of θ for all time. We can express the solution of the diffusion equation (5.1.27) as a linear combination of Bessel functions, as in Strauss (1992):

$$u(r,t) = \sum_{n=0}^{\infty} A_n J_0\left(\frac{k_n r}{L}\right) \exp\left(-D\left(\frac{k_n}{L}\right)^2 t\right), \tag{5.1.29}$$

where the k_n are chosen such that $\partial J_0(k_n r/L)/\partial r\,|_{r=L} = 0$. The constants, A_n, are determined by the initial condition, $u(r,0) = U_0(r)$,

$$A_n = \frac{\int_0^L U_0(r) r J_0(k_n r/L)\mathrm{d}r}{(L/k_n)^2 \int_0^{k_n} r J_0^2(r)\mathrm{d}r}. \tag{5.1.30}$$

Fig. 5.6 compares snapshots of the individual-level model (with square domain tessellation) and the population-level model. The two models give qualitatively similar cell density profiles. Fig. 5.6 (c) shows an HDE which is of comparable magnitude to those of similar simulations on a square domain. This indicates a good quantitative agreement between the densities generated by the two models.



(a) (b) (c)

Figure 5.6: *A comparison of the individual-level (with square domain tessellation) and population-level models of diffusion on a circular domain. In the individual-level model* $N = 50000$ *cells are initialised with a density profile which decays approximately exponentially from the centre to the edge of the domain. The continuum model is initialised similarly with a profile* $u = A\exp(-10r)$*, where* A*, the normalising factor, ensures that the area under the continuum initial condition corresponds to the correct number of cells in the individual-level simulation. There are a total of* 1885 *elements in the circular domain in the individual-based model. (a) The histograms represent the density of cells in each square of the domain tessellation. (b) The surface represents the radially symmetric solution of PDE* (5.1.27) *with* $D = h^2$*, where* $h = 1/49$ *is the length of an element in the individual-level model. (c) The evolution of the HDE. For a movie comparing the density of the two models as time evolves please see MOVIE_4.avi.*

In this section we have demonstrated that our equivalence framework can be extended in a straightforward manner into higher dimensions. The PDE corresponding to the individual-level model has been shown to be independent of the domain tessellation. In addition we have shown, by tessellating the domain with square elements, that we can extend our equivalence framework to non-square domains in a simple manner. This extension will prove useful when we come to considering domain growth, which will occur on inherently non-square domains.

5.2 On-lattice domain growth models

Our aim in this section is to investigate existing cluster aggregate models from the literature which might be used to simulate cell migration on a growing domain in two dimensions. We explore their suitability by considering properties such as isotropy and surface scaling. In the various terminologies in the literature we refer to the growing lattice, variably, as a 'domain', 'cluster' or 'aggregate' depending on the context, and to elements of the cluster as 'elements', 'particles' or 'sites' but never 'cells'. 'Cells' will be reserved exclusively to describe the random walkers on our growing lattice. For consistency with our one-dimensional simulations of domain growth and ease of comparison with a continuum version of the model, we make five main requirements of the model for the growth of our underlying domain:

1. elements are positioned on-lattice;
2. domain growth occurs by elemental division (i.e. mitosis where a parent element is chosen to divide randomly into two daughter elements);
3. mitosing elements can push as many neighbouring elements out of the way as is necessary to make room for daughter elements;
4. clusters are compact[4] (i.e. they do not have holes or invaginations);
5. clusters grow isotropically.

Since we have no specific biological context in mind, we have specified rules for domain growth which will be most convenient mathematically in order to help us to compare the individual-level model with a continuum equivalent. For example, the requirement for cluster compactness, convexity and isotropy will greatly simplify the solution of an equivalent PDE since circular symmetry lends itself more readily to the analytical solution of PDEs. More biologically plausible methods of domain growth may be incorporated depending on the specific biological context we wish to model. The rules we have set are relatively stringent and we hence investigate several different classes of models in order to find one which satisfies our requirements.

[4]In this context, compact refers to a cluster that is space filling i.e. the body of which is non-fractal.

5.2.1 The Eden model

One of the earliest models of cluster aggregation is known as the Eden model (Eden, 1958, 1961). This cellular automaton (CA) model, or adaptations thereof, has been used to represent such biological phenomena as wound healing (Callaghan et al., 2006), slime mold growth (Wagner et al., 1999) and tumour growth (Drasdo and Höhme, 2003) and, indeed, tumour growth is the context in which it was originally proposed (Eden, 1958). Although now largely regarded as being an unphysical model of tumour growth, it remains the most basic on-lattice model for biological domain growth. Although its clusters do not grow through mitosis[5], the Eden model is a well-studied paradigm of cluster growth and will provide a useful prototype for us to investigate the defining characteristics of such models. This will be advantageous when investigating more complicated models.

The model is often initialised with a single, fixed, square[6] 'seed' element capable of replicating and positioning a daughter element at any of the $2d$ (in the case of square domain elements) directions of its empty nearest neighbours, where d is the spatial dimension of the lattice. The cluster grows, one element at a time, by randomly filling empty lattice sites adjacent to the aggregate surface. Surface elements are defined to be those with at least one 'open' edge and an edge is said to be 'open' if the lattice site adjacent to that edge is unoccupied. An element in the aggregate which is not a surface element has no open edges and is called an 'internal' element. Jullien and Botet (1985b) note that the specific method of element addition can give rise to three variants of the Eden model which they call versions A, B and C (see Fig. 5.7):

A. the daughter element is added equiprobably to any unoccupied site adjacent to the aggregate surface;

B. an open edge of the surface is chosen equiprobably and the daughter element is added adjacent to this edge;

[5]The Eden model grows via the addition of particles to the surface. This can, in some senses, be considered as an apical growth-type model where only surface elements can mitose, but does not correspond to the general mitosis process in which any element of the cluster can divide.

[6]The shapes of the elements tessellating the domain are defined by the dual of the underlying lattice e.g. squares on a square lattice, regular hexagons on a regular triangular lattice etc.

C. a surface element is chosen equiprobably and a daughter element is added equiprobably to any of its open edges.

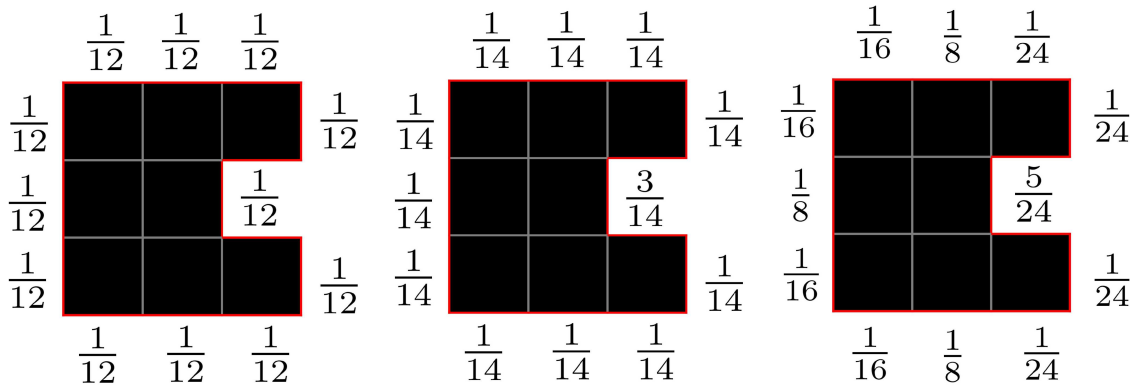


Figure 5.7: *(a)-(c) Examples of the three particle addition mechanisms of the Eden model (A-C, respectively). Elements are denoted by black squares and the boundary of the cluster is denoted with a red line. The probability of addition of a particle in a specific position is given at each surface site. (a) For the Eden A model the particle is added to any surface site equiprobably. (b) For the Eden B model an edge of the surface is chosen with equal probability, making it more likely that the particle is added to a surface sites with more neighbours.(c) For the Eden C model a surface particle is chosen equiprobably and one of its edges is chosen equiprobably to have a particle appended to it. Again surface sites with more neighbours are added to even more frequently. In addition there is a propensity to add particles so as the make the whole structure more convex.*

Surprisingly, these three different proliferation mechanisms give rise to qualitatively similar aggregates, but they differ widely in the regularity, convexity and consistency of the aggregates they form (see Fig. 5.8 (a)-(c)). The models also differ in genus (number of holes), where a hole is defined to be an unoccupied lattice site surrounded by occupied lattice sites. For our purposes, holes will be detrimental as they will be hard to model in a continuum context. The evolution of the genus of typical Eden A, B and C clusters (through 5000 elemental addition events averaged over 50 repeats) is displayed in Fig. 5.8 (d), (e) and (f), respectively. By simply looking at example clusters and the average evolution of the number of holes of the domain over time it seems evident that the version of the model most likely to satisfy our needs is version C.

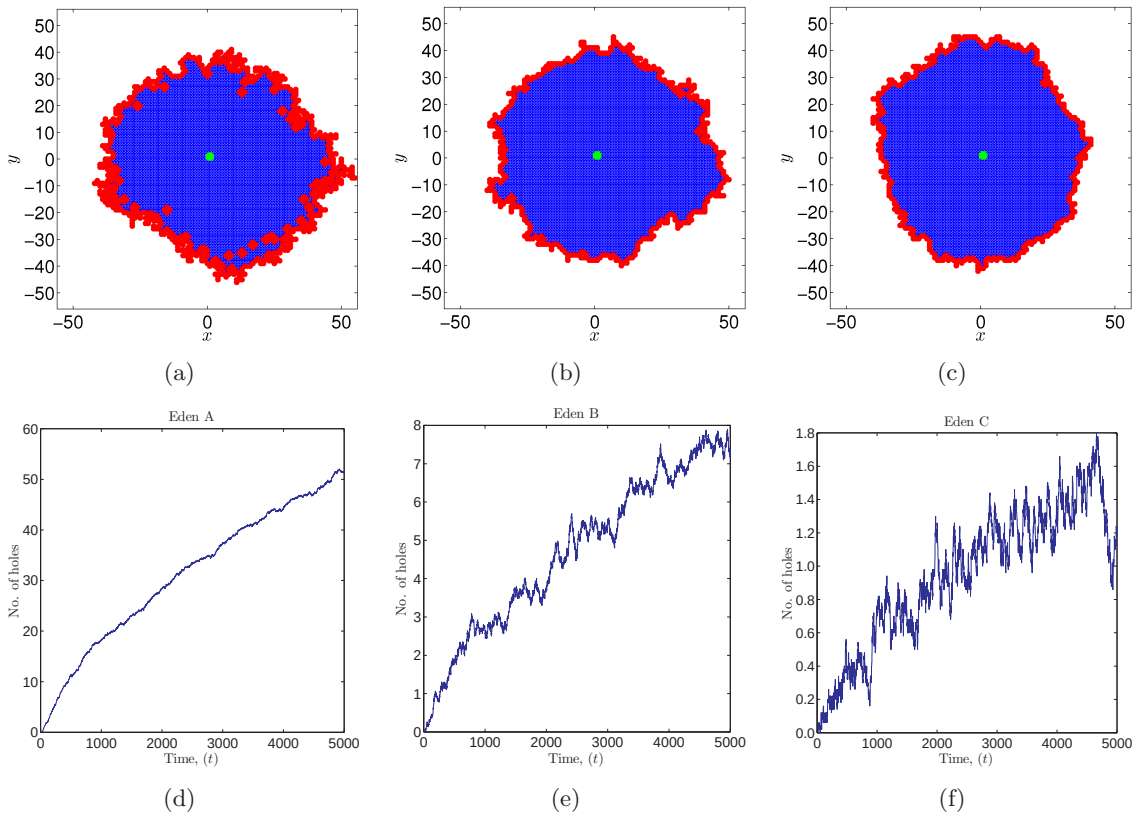


Figure 5.8: *(a)-(c) Typical examples of the aggregates formed by the three different versions of the Eden model (A-C, respectively) after 5000 proliferation events. Internal elements are marked with blue stars (∗) and surface elements with red circles (●) with the original seed element marked with a green circle (●) at the origin. Version C produces the most regular, convex and hole-free aggregates. (d)-(f) The evolution, over 5000 proliferation events, of the number of holes in the three different versions of the Eden model (A-C, respectively) averaged over 50 repeats.*

5.2.2 Scaling of the Eden model and universality classes

A standard procedure for classifying the intrinsic properties of models such as the Eden model is to implement a so-called '1+1' dimensional simulation on a 'strip geometry'. The strip geometry helps to avoid inherent biases that are often observed on aggregates grown freely from a single seed (Jullien and Botet, 1985b) (see Section 5.2.4). Clusters are grown in two dimensions from a one-dimensional substrate (i.e. a base-layer of seed elements) with linear length, l, and growth is confined to a corresponding 'strip' of length l using periodic boundary conditions on the vertical boundaries (Vicsek, 1992). Since the length of the domain, l, remains constant throughout the simulation, the 'effective height' of the surface, $\hat{h} = N/l$, will grow linearly with N, the number of particles in the simulation.

If the deposition rate is constant then $\hat{h}$ will also grow linearly with time, t. If clusters are compact (i.e. their interiors are non-fractal) and have few overhangs/holes the effective height is a good approximation to the 'average height', $\bar{h}$:

$$\hat{h} \approx \bar{h} = (1/N_s) \sum_{i=1}^{N_s} h_i, \tag{5.2.1}$$

where h_i is defined to be the height of the i^{th} surface element and N_s is the number of surface elements[7].

The fractal surface of a growing cluster tends to increase in width or 'roughen' as the cluster increases in size. This 'roughening' can be captured by considering the 'interface width', defined as the root mean square of the deviations of surface elements about their mean value:

$$w(l, \hat{h}) = \left\{ \frac{1}{N_s} \sum_{i=1}^{N_s} \left[h_i(t) - \bar{h} \right]^2 \right\}^{1/2}. \tag{5.2.2}$$

Clearly the initial substrate layer is a surface with width zero, but as particles are added to the clusters this surface gradually roughens.

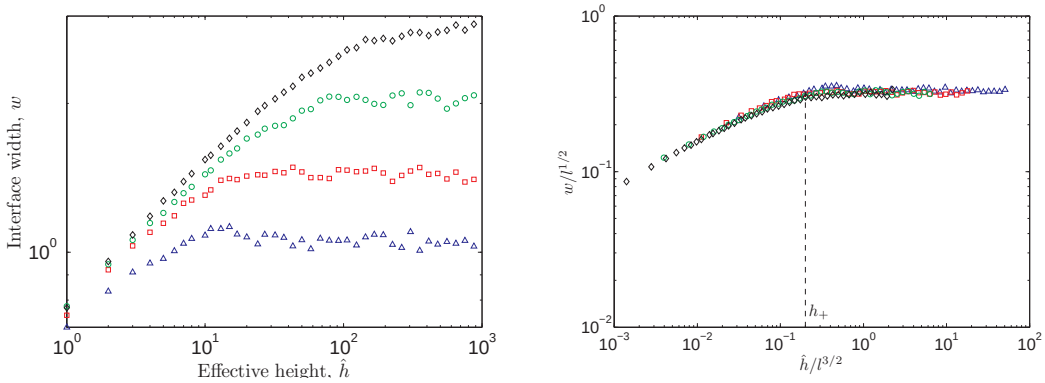
Fig. 5.9 (a) shows the increase of the surface width of the Eden model over time for a variety of different substrate lengths ($l = 10, 20, 40, 80$) on a log-log plot. The dependence of the surface width on the 'effective height', $\hat{h}$, for each value of l, has two well-demarcated regions separated by a crossover height, which we will denote h_+ (see Fig. 5.9 (b)). Although this crossover point occurs at different values of $\hat{h}$ for each different substrate length it is clear that the width of the interface increases initially as a power-law in effective height:

$$w(l, \hat{h}) \sim \hat{h}^{\beta} \qquad \text{when} \qquad \hat{h} \ll h_+. \tag{5.2.3}$$

The exponent β is known as the growth exponent and characterises the dynamical behaviour of the roughening process (Barabási and Stanley, 1995).

Eventually, due to surface correlations, the width of the surface reaches capacity for the given substrate length and the variation of the surface width with the effective height levels off (see the horizontal regions of Fig. 5.9 (a)). However, the value of the interface width

[7] Due to overhangs and holes there may be more surface sites than there are columns, hence $N_s \geq l$.



(a) (b)

Figure 5.9: *The evolution of the surface width for the Eden C model for different system sizes. (a) The logarithmic scale displays the clear initial power law dependence of surface width on effective height and the eventual saturation of surface width to the value $w_{cap}(l)$ (see equation* (5.2.4)*). (b) The evolution of surface width rescaled according to relation* (5.2.7)*. The functional form of the scaling function $f(x)$ and its properties in different regimes, as described in equation* (5.2.8)*, are clearly evident. The scaled crossover height, h_+, is marked as a vertical, black, dashed line. In both sub-figures blue triangles (△) correspond to system size $l = 10$, red squares (□) to $l = 20$, green circles (○) to $l = 40$ and black diamonds (◇) to $l = 80$.*

capacity, w_{cap}, is evidently dependent on the length of the substrate, l, (as the surface correlations take longer to spread across a wider surface). This dependence also follows a power law,

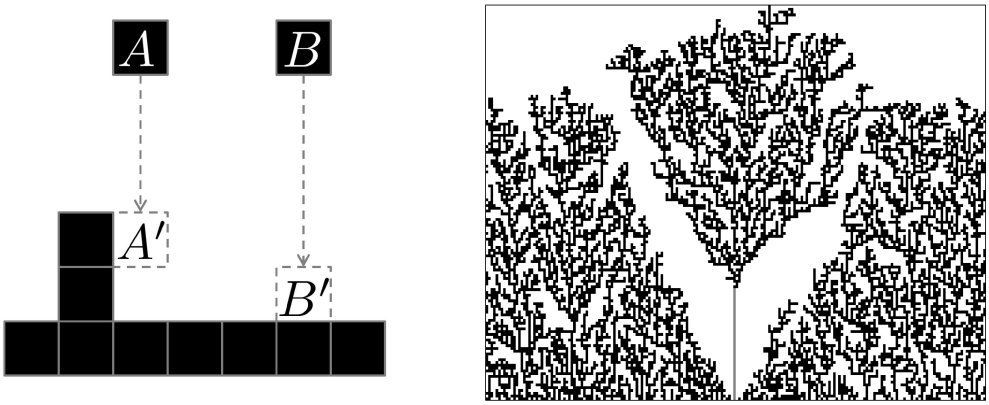
$$w_{\text{cap}}(l) \sim l^{\alpha} \qquad \text{when} \qquad \hat{h} \gg h_+. \tag{5.2.4}$$

This can be seen anecdotally from the roughly even spacing of the values of w_{cap} (on a log scale) for successively doubling substrate widths in Fig. 5.9 (a). The exponent α is known as the roughness exponent since it determines the maximum roughness of the interface.

The fact that surface width saturates at all is a combination of the surface correlations and a 'finite size effect'. Surface correlations imply that, due to the nature of the particle addition process, the height of one column is affected by the heights of its locally neighbouring columns. As the number of particles deposited increases, the distance over which these local correlations occur (the *correlation length*) increases (See Fig. 5.10 (b)) until it is the same size as the substrate length and the correlations have, therefore, become global. At this point the surface width ceases to increase and has reached its capacity. The finite size effect captures the idea that over an infinite substrate the correlations would take an infinite

amount of time to become global and hence the surface would not saturate. Indeed it is only because the substrate has a *finite size* that we see saturation (Barabási and Stanley, 1995).

The concept of surface correlations can be better understood through an example. Consider the ballistic deposition model with the nearest neighbour sticking rule. On a strip geometry (of length l) the model proceeds as follows: A random position above the substrate is chosen with equal probability and the particle falls vertically from this position towards the surface. The particle sticks at the first site along its trajectory that has an occupied nearest neighbour (where neighbours are considered in the von Neumann sense rather than the Moore sense). Fig. 5.10 (b) demonstrates the lateral spread of the aggregate from a column of particles that is initially raised higher then its neighbours.

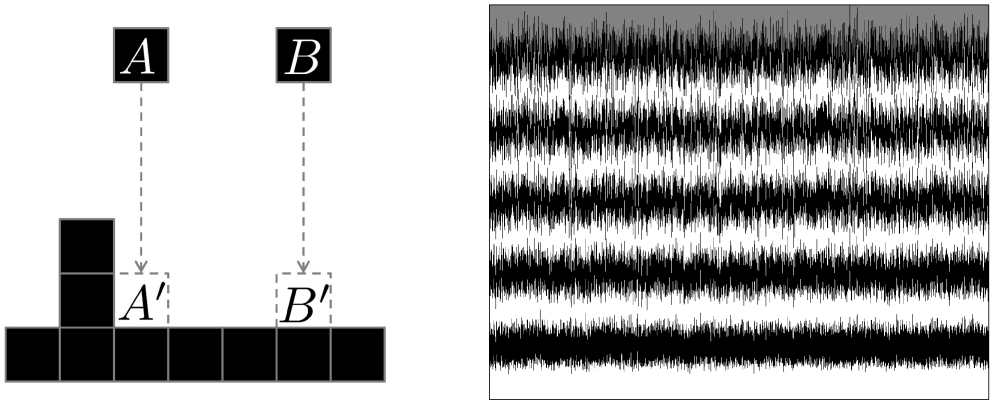


(a) (b)

Figure 5.10: *The ballistic deposition model with the nearest neighbour sticking rule. (a) Sticking rules for newly-deposited particles. Here the substrate width is chosen to be $l = 7$. According to the rules given in the text, particle A sticks at A' and particle B at B'. (b) An illustration of lateral growth and the spread of surface correlations in the ballistic deposition model. Growth is started from a flat interface except for one column which is much higher (40 rows) than its neighbours. This column captures arriving particles and in so doing grows both vertically and laterally. This illustrates the lateral spread of 'information' during growth: the excess height spreads along the surface influencing the height of neighbouring columns (Barabási and Stanley, 1995).*

If neighbouring columns are not correlated with each other then the surface width will continue to increase as the effective height of the interface grows (see Fig. 5.11 (b)). A simple example of the necessity of surface correlations for saturation is the random deposition model (or non-stick ballistic deposition model). A particle is released above a randomly chosen substrate site and falls vertically until it reaches the highest particle in that column (see

Fig. 5.11 (a)). Thus the height of each column corresponds to the number of particles allocated to it. By appealing to the multinomial distribution (for a series of independent Bernoulli trials with l possible outcomes) it is simple to see that the variance of the number of particles in each interval will depend linearly on the number of particles added and hence linearly on the effective height. This means that the interface width (which is effectively the variance of the heights of each column) grows as the square root of the effective height (i.e. a growth exponent, $\beta = 1/2$) and fails to saturate.



(a) (b)

Figure 5.11: *The non-stick ballistic deposition model. (a) Positioning rules for newly-deposited particles. Here, again, the substrate width is chosen to be $l = 7$. According to the rules given in the text, particle A sticks at A' and particle B at B'. (b) A typical interface generated by the non-stick ballistic deposition algorithm after the deposition of 1,000,000 particles on a substrate of length $l = 1000$. The shading code represents the arrival time of the particles: after the deposition of 10,000 particles the shading changes. Note the rapid increase in the roughness and the uncorrelated nature of the surface (Barabási and Stanley, 1995).*

As well as the growth exponent and the roughness exponent discussed above, there is a third critical exponent which can be used to characterise the interface roughness (Barabási and Stanley, 1995). It describes the dependence of the crossover height, h_+, (sometimes also known as the saturation height) on the substrate width[8]:

$$h_+ \sim l^z. \tag{5.2.5}$$

Here z is known as the dynamic exponent. The three characteristic exponents, α, β and z

[8]These power-law relationships (equations (5.2.3)-(5.2.5)) have been found to describe the scaling behaviour of the surface width by carrying out multiple simulations rather than through proving rigorous theorems (Barabási and Stanley, 1995; Family and Vicsek, 1985).

are not, however, independent of each other and it is simple to see, by rescaling the surface width and the effective height, that we can collapse the curves of Fig. 5.9 (a) on to each other, as in Fig 5.9 (b).

Dividing the surface width by the value at which it reaches capacity we find that, for each value of l, the normalised surface width, $w(l,\hat{h})/w_{\text{cap}}(l)$, saturates at the same value, although for different values of $\hat{h}$. By dividing the effective height by the crossover height and plotting the normalised surface width as a function of $\hat{h}/h_+$ we find that the curves all saturate at the same value of $\hat{h}/h_+$ (see Fig. 5.9 (b)). This 'collapse' is a result of the fact that the normalised surface width can be written as a function of $\hat{h}/h_+$ only:

$$\frac{w(l,\hat{h})}{w_{\text{cap}}(l)} \sim f\left(\frac{\hat{h}}{h_+}\right), \tag{5.2.6}$$

where $f(x)$ is known as the scaling function. With a slight rearrangement we obtain the Family-Vicsek scaling relation (Family and Vicsek, 1985):

$$w(l,\hat{h}) \sim l^{\alpha} f\left(\frac{\hat{h}}{l^{z}}\right). \tag{5.2.7}$$

The form of the scaling function can clearly be seen from Fig 5.9 (b):

$$f(x) \sim \begin{cases} x^{\beta}, & x \ll 1, \\ \text{constant}, & x \gg 1. \end{cases} \tag{5.2.8}$$

Using the scaling relations (5.2.3) and (5.2.4) this can equivalently be stated as

$$w(l,\hat{h}) \sim \begin{cases} \hat{h}^{\beta}, & \hat{h} \ll l^{z}, \\ l^{\alpha}, & \hat{h} \gg l^{z}, \end{cases} \tag{5.2.9}$$

from which we can derive a relationship between the critical exponents (Family and Vicsek, 1985; Jullien and Botet, 1985b; Vicsek, 1992),

$$z = \alpha/\beta, \tag{5.2.10}$$

by considering the scaling of the surface width at the crossover value $h_+ \sim l^z$.

The values of the critical exponents provide us with a mechanism for classifying models with similar growth patterns and interface dynamics into a 'universality class' (UC). Each UC has its own set of critical exponents and represents a dominant physical process of the aggregate growth and interface dynamics. There is also a canonical stochastic (partial) differential equation (SPDE) associated with each UC which describes the variation with time[9] of the interface height, $h(\boldsymbol{x}, t)$ at position $\boldsymbol{x}$ on a $d-$dimensional substrate. Inherent to all such UCs is the idea that the interfaces grow through random particle influx. In order to 'derive' a characteristic equation for the effective interface height it is assumed that the growth can be described by a continuum equation of the form

$$\frac{\partial h(\boldsymbol{x}, t)}{\partial t} = F + \eta(\boldsymbol{x}, t), \tag{5.2.11}$$

where F is the average rate of particle arrival at site $\boldsymbol{x}$ and $\eta(\boldsymbol{x}, t)$ is zero-mean, uncorrelated noise which models random fluctuations in the deposition process (Barabási and Stanley, 1995). In theory the critical exponents of the growth processes described by each UC can be derived from the corresponding canonical equation.

The simplest UC is known as Edwards-Wilkinson (EW) (Edwards and Wilkinson, 1982). The corresponding equation is:

$$\frac{\partial h(\boldsymbol{x}, t)}{\partial t} = \nu \nabla^2 h + \eta(\boldsymbol{x}, t). \tag{5.2.12}$$

The term ν is often referred to as surface tension since the term $\nu \nabla^2 h$, which corresponds to surface diffusion, has the effect of smoothing the surface.

By considering scaling arguments and the self-affine nature of the surface, as in Barabási

[9]Here we switch to using classical PDE notation with time as one independent variable. Time can be considered as a proxy for effective height, $\hat{h}$, (or *vice versa*) as the effective height will grow linearly with time (assuming a constant rate of particle deposition).

and Stanley (1995), the critical scaling exponents for the EW UC can be shown to be

$$\begin{aligned} \alpha &= \frac{2-d}{2}, \\ \beta &= \frac{2-d}{4}, \\ z &= 2, \end{aligned}$$

where d is the dimension of the substrate on which the aggregates are grown. If $d = 1$ these exponents simplify to

$$\begin{aligned} \alpha &= \frac{1}{2}, \\ \beta &= \frac{1}{4}, \\ z &= 2. \end{aligned}$$

The next most simple constitutive equation includes a non-linear term and corresponds to the Kardar-Parisi-Zhang (KPZ) universality class (Kardar et al., 1986). The RHS of the canonical KPZ equation comprises three terms:

$$\frac{\partial h}{\partial t} = \nu \nabla^2 h + \frac{\lambda}{2} (\nabla h)^2 + \eta(\boldsymbol{x}, t), \tag{5.2.13}$$

where $\nu \nabla^2 h$ is related to the surface tension, $\lambda (\nabla h)^2 / 2$ is related to lateral growth and $\eta(\boldsymbol{x}, t)$ is related to noise.

By appealing to heuristic arguments or the dynamic renormalisation group method (Barabási and Stanley, 1995) the scaling coefficients corresponding to the KPZ equation can be found, in one dimension, as $\alpha = 1/2$, $\beta = 1/3$ and $z = 3/2$. One simple, alternative way to determine the scaling coefficients of any UC is by numerically simulating its canonical SPDE.

The KPZ equation in the full two-dimensional (rather than 1+1) case has the form (in radial coordinates)

$$\frac{\partial R}{\partial t} = \frac{v}{R} \left[R^2 + \left(\frac{\partial R}{\partial \theta} \right) \right]^{1/2} - \nu \frac{R^2 + 2(\partial R/\partial \theta)^2 - R \partial^2 R / \partial \theta^2}{[R^2 + (\partial R/\partial \theta)^2]^{3/2}} + \eta(\theta, t), \tag{5.2.14}$$

where R is the radius of the cluster, which, in general, depends on θ. The first term on the RHS of equation (5.2.14) corresponds to growth normal to the surface with speed v and the second term encompasses the effects of surface tension which are assumed to be proportional to curvature. For this reason ν is often referred to as the surface tension coefficient. The noise term, $\eta(\theta, t)$, is drawn from a circularly wrapped Gaussian distribution. Batchelor et al. (1998) show that the radial version of the KPZ equation has both circular and regular polygonal solutions, but that it is only the circular solutions which are stable. As such, we expect models in the KPZ UC to exhibit circular symmetry when grown in two dimensions.

5.2.3 Practical determination of universality class

Following Jullien and Botet (1985a,b), in order to calculate the scaling of the surface width in the regime $\hat{h} \gg l$ we have carried out '1+1' dimensional simulations of Eden clusters up to heights $\hat{h} = 20l$ for $10 < l < 100$ and averaged over 20 repeats. By considering the gradient of the log-log plot of Fig. 5.12 (a), the roughness exponents are found to be $\alpha_B = 0.46 \pm 0.10$, $\alpha_C = 0.48 \pm 0.09$, where the subscript corresponds to the version of the model being simulated[10]. Similarly we considered $\hat{h} = l/10$ for $10 < \hat{h} < 100$ in order to approximate the regime $l \gg \hat{h}$. The value of the growth exponent was calculated, using Fig. 5.12 (b), to be $\beta_B = 0.33 \pm 0.04$, $\beta_C = 0.32 \pm 0.04$. Recall that once the roughness and growth exponents (α and β respectively) are known, the dynamic exponent z can be calculated as $z = \alpha/\beta$.

These values are consistent with previous findings that the Eden model falls into the KPZ UC (Devillard and Stanley, 1989; Ferreira Jr. and Alves, 2006; Kertész and Wolf, 1988; Vicsek, 1992) with characteristic exponents $\alpha = 1/2$, $\beta = 1/3$ and $z = 3/2$. Combined with the results of Batchelor et al. (1998), this suggests that the unconstrained Eden model may be an appropriate first step towards a circularly symmetric, on-lattice growth model.

Although the strip version of the Eden model appears to fall into the KPZ UC, it is not clear that the model in an unconstrained two-dimensional geometry will do the same. In

[10] Jullien and Botet (1985b) found that Eden A clusters give inconsistent values for growth and roughness coefficients due to the irregularity of their shape and the large numbers of holes they permit. By carrying out simulations of our own we have confirmed these findings and have therefore not presented the data for this version of the model.

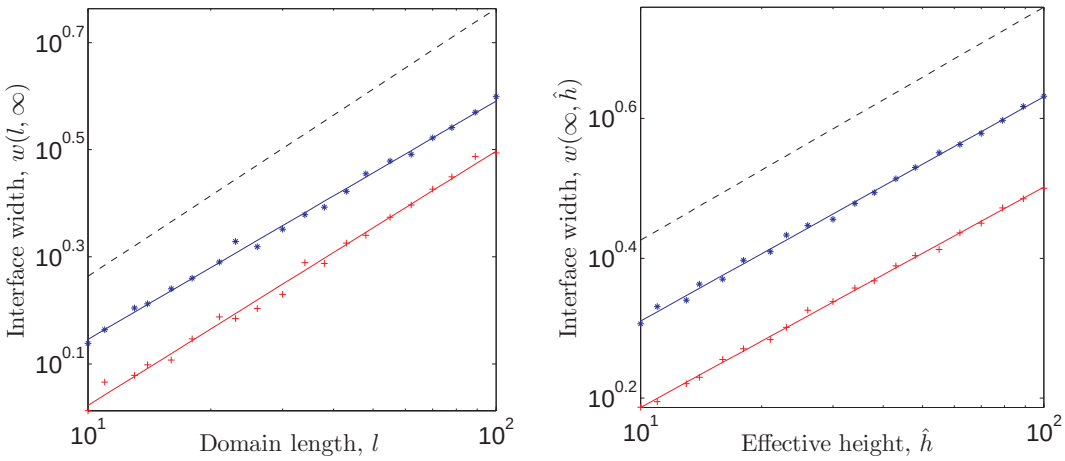


Figure 5.12: *Scaling of the surface width of Eden clusters in '1+1' dimensions. (a) The variation of $w(l,\infty)$ (i.e. $\hat{h} \gg l$) with domain length, l, for models B (blue stars (∗)) and C (red crosses (+)). The straight (black dashed) line plotted for comparison has gradient* $1/2$. *(b) The variation of $w(\infty,\hat{h})$ (i.e. $l \gg \hat{h}$) with effective cluster height, $\hat{h}$. Descriptions as for (a). The straight (black dashed) line plotted for comparison has gradient* $1/3$. *The scaling of the Eden A model is not included in this plot due to its poor convergence (Jullien and Botet, 1985a; Meakin and Witten Jr, 1983).*

an unconstrained geometry the interface width becomes a function of the average cluster radius, R, (which is the equivalent of $\hat{h}$, the effective distance from the seed layer, in the strip geometry):

$$w(l,R) = \left\{ \frac{1}{N_s} \sum_{i-1}^{N_s} [r_i(t) - \langle r_i \rangle]^2 \right\}^{1/2}, \tag{5.2.15}$$

where N_s is the number of surface elements and r_i is the distance of surface element i from the seed particle, or a suitable proxy[11] when the seed particle is unknown.

In order to verify that the full two-dimensional Eden model grown from a single seed particle is also in the KPZ UC we require methods of determining the critical exponents of the model on a non-strip geometry. Such methods will also be useful for many physical applications, such as determining the UC of tumours or other higher dimensional aggregates (Brú et al., 2003) and also for determining the UC of a cluster aggregation model where it is not clear what the analogous version of the model on a strip geometry should be.

[11]The centre of mass of the whole aggregate has been shown by Ferreira Jr. and Alves (2006) to be an appropriate proxy for the seed particle when measuring Eden clusters since, asymptotically, its fluctuations grow more slowly than those of the boundary width. However, the centre of mass of the surface elements themselves has been shown to be unsuitable since the fluctuations in this quantity grow faster than those of the surface width and hence dominate the asymptotic calculations of surface scaling.

As in the strip geometry, the interface width of the cluster grows as a power law with the mean radius of the cluster, R, or some suitable proxy (e.g. $N^{1/d}$ in a d-dimensional simulation, where N is the number of particles in the cluster) with a characteristic critical exponent β i.e. $w \sim R^{\beta}$. Continuing the analogy with the strip geometry, the dependence of the surface width on the circumference, L, also grows as a power law with roughness exponent α i.e. $w \sim L^{\alpha}$ (Brú et al., 2003).

It is straightforward to calculate the growth coefficient, β, for two-dimensional models. We can employ the method of consecutive slopes in order to mitigate the finite size effects of the system which limit the scaling regimes (Barabási and Stanley, 1995): instead of employing a linear fit on a plot of $\log(w(l,R))$ against $\log(R)$ we calculate the value of the slope of the curve on the log-log plot over a region of size s (i.e. fit a straight line to the data points between R and $R+s$). The values of these slopes are plotted against $\log(R)$ and, if there were no upper or lower cutoffs[12], we would see the value of these slopes clustered around the true value of β (Barabási and Stanley, 1995).

In order to calculate the roughness exponent, α, we assume that $w \sim l^{\alpha}$ where l is a local length scale (Barabási and Stanley, 1995). Specifically we study the variation of the surface width with the local arc length:

$$w_L^2(l,R) \equiv \left\langle [R(\boldsymbol{x},R) - \bar{R}_l(\boldsymbol{x},R)]^2 \right\rangle_x . \tag{5.2.16}$$

The subscript l refers to the fact that $\bar{R}_l(\boldsymbol{x},t)$ is the average radius of the surface in the arc of length l and the subscript L (on $w_L^2(l,R)$) indicates that we are determining the scaling of surface width with local length, l, for a cluster of fixed circumference, L. The angle brackets indicate that for each arc we average over all the values of $R(\boldsymbol{x},t)$ on the surface and then take the ensemble average over all the different arcs considered. For sufficiently small values of l the following relationship holds true:

$$w_L(l,R) \sim l^{\alpha} \quad \text{provided} \quad l \ll \xi_{||}, \tag{5.2.17}$$

[12]Cutoffs are sizes of the cluster for which the scaling relationship is not valid (e.g. an upper cutoff will be when the surface begins to saturate and a lower cutoff may be the size of the lattice).

where $\xi_{||}$ is the correlation length (Barabási and Stanley, 1995). Providing the system is saturated the scaling relationship is valid up to $l = L$ (where L is the current circumference of the cluster) since the correlation length will be the same as the system length, by the definition of saturation.

Zabolitzky and Stauffer (1986) fail to find saturation even with Eden clusters of size 10^9. Since determining the roughness exponent, α, would be difficult even with clusters of this size, we save computational effort by limiting ourselves to relatively modest cluster sizes ($\sim 10^5$ particles) which, nevertheless, make it possible to predict the characteristic scaling behaviour of the surface width with the average cluster radius as $w \sim R^{1/3}$ (see Fig. 5.13 (b) and (d)). Upon simulation of an isotopic version of the Eden A model we found $\beta = 0.35 \pm 0.09$. Similarly for the isotropic Eden C model[13] we found $\beta = 0.31 \pm 0.10$. These values are consistent with the hypothesis that $\beta = 1/3$ and therefore that the Eden model falls into the KPZ universality class (Batchelor et al., 1998; Jullien and Botet, 1985b; Paiva and Ferreira Jr, 2007). We also attempt to determine the roughness exponent, α, by considering the scaling of the local surface width, $w_L(l, R)$, (see equation (5.2.16)) with the characteristic length scale, l, (see Fig. 5.13 (a) and (c)). A clear power-law relationship is difficult to obtain since the clusters are not saturated. This becomes even more apparent when attempting to determine α via the method of local slopes (see Fig. 5.13 (c)). A true power law dependence would show small oscillations about the correct value, as exhibited for the growth exponent, β, (see Fig. 5.13(d)) or, with ideal data, a plateau at the correct value. Another possible difficulty in determining the critical exponents of clusters in unconstrained geometries is the inherent anisotropy induced by the lattice on which they grow. Here we have attempted to avoid the effects of anisotropy by using the isotropic versions of the Eden model proposed by Paiva and Ferreira Jr (2007).

5.2.4 Measuring anisotropy

We consider two different methods for measuring anisotropy. The first measure, the 'angular surface anisotropy' is given by the normalised square of the distance to the surface from the

[13] In order to determine the characteristic coefficients as accurately as possible, we have used isotropic versions of the Eden A and C models as detailed by Paiva and Ferreira Jr (2007). No such equivalent version of the Eden B model is available.

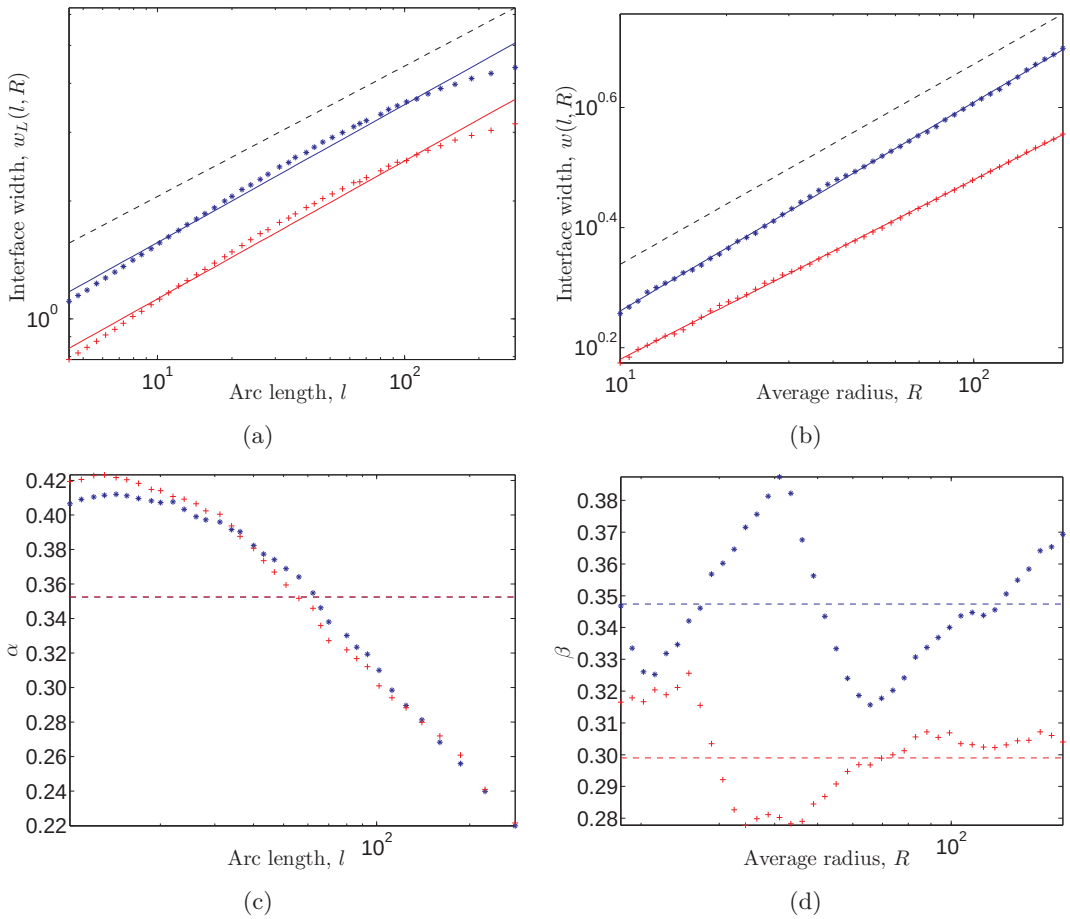


Figure 5.13: *Scaling of the surface width of Eden clusters in two unconstrained dimensions. (a) The variation of $w_L(l,R)$ with arc length, l, (for fixed circumference size, L) for models A (blue stars (∗)) and C (red crosses (+)). The straight (black dashed) line plotted for comparison has gradient* 1/3*. (b) The variation of $w(l,R)$ with average cluster radius, R. Legend as for (a). The straight (black dashed) line plotted for comparison has gradient* 1/3*. (c) Evaluation of the roughness exponent, α, using the method of local gradients. Model A is represented by blue stars (∗) and model C by red crosses (+). The dashed red and blue lines correspond to the value of the gradient fitted using all points. The version C value is very close to the version A value and hence obscures the model A gradient. (d) Evaluation of the growth exponent, β, using the method of local gradients. Legend as for (c). The scaling of the Eden B model is not included in this plot since an isotropic version of the model has not yet been determined (see Section 5.2.4) (Paiva and Ferreira Jr, 2007).*

centre in a particular direction (Batchelor and Henry, 1991; Freche et al., 1985). For each surface element, i, calculate the angle, ϕ_i, subtended by the x-axis and the line connecting the centre of the seed particle (or centre of mass, translating the cluster so that this lies at the origin) to the element's centre (see Fig. 5.14). Also calculate the element's distance from the seed/centre of mass, r_i (see Fig. 5.14). The angular surface anisotropy in the

direction θ_k is then given by

$$P(\theta_k) = \frac{1}{N_s^{\theta_k}} \sum_{i=1}^{N_s} r_i^2 \mathbb{I}_{\theta_k - \delta\theta < \phi_i < \theta_k + \delta\theta}, \tag{5.2.18}$$

where the sum is over the N_s surface elements and $N_s^{\theta_k} = \sum_{i=1}^{N_s} \mathbb{I}_{\theta_k - \delta\theta < \phi_i < \theta_k + \delta\theta}$ is the number of those surface elements with angles in the appropriate region $[\theta_k - \delta\theta, \theta_k + \delta\theta]$ (green stars on Fig. 5.14). $\mathbb{I}$ represents the indicator function given by

$$\mathbb{I}_{A(x)} = \begin{cases} 1 & x \in A, \\ 0 & x \notin A. \end{cases}$$

K values of θ_k are chosen so as to cover the whole circle. This necessarily defines $\delta\theta = \pi/K$. We choose K to be a power of two[14] and $\theta_1 = 0$ in order to capture the anisotropy that might be caused by the square lattice. The angular surface anisotropy is then normalised by dividing $P(\theta_k)$ by $\sum_{k=1}^{K} P(\theta_k)$ for each value of k.

The second measure, the 'angular distribution anisotropy', counts the number of elements which fall into the angular regions defined above (green and yellow stars in Fig. 5.14) and is, in some sense, an 'angular distribution function' (Meakin, 1986):

$$Q(\theta_k) = \frac{K}{N} \sum_{i=1}^{N} \mathbb{I}_{\theta_k - \delta\theta < \phi_i < \theta_k + \delta\theta}, \tag{5.2.19}$$

where we now sum over the total number of elements, N, rather than just the surface elements.

In order to compare these anisotropy measures and to determine whether they are themselves influenced by the lattice, we tested them on a series of artificial clusters: the circle, the square and the diamond (see Figs. 5.15 (a), (d) and (g) respectively). We made these test clusters representative of the size of cluster we will be simulating (i.e. $\sim 10^5 - 10^6$ elements).

Both measures discern the strong diagonal anisotropy in the square cluster and the axial

[14] Choosing K to be a sufficiently large power of two ensures that each axis and diagonal has a sector of the circle centred upon it. This ensures that we measure the anisotropy in each of these directions in a consistent manner.

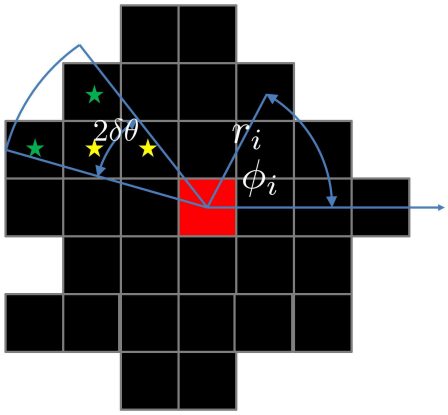


Figure 5.14: *Measuring anisotropy of particle aggregates. Elements of the aggregate are represented by black squares (■). The angle to the x-axis, ϕ_i, and the distance from the seed particle (red square, ■), r_i, are determined for each element. In each segment of the circle the mean square distance of the centres of mass of the surface elements (green stars, ⋆) is calculated and then normalised to find the 'angular surface anisotropy'. In order to find the 'angular distribution anisotropy' the number of elements whose centres of mass lie in each segment (yellow and green stars, ⋆ and ⋆) is counted and divided by the mean number of elements per segment (averaged over all segments) in order to normalise.*

anisotropy in the diamond cluster (see Fig. 5.15). We should note that the measures themselves are, potentially, to some small degree, affected by the anisotropy of the underlying lattice: clear periodicity is observed in both measures even when applied to the circular cluster[15]. For the angular surface anisotropy measure, $P(\theta)$, the maximum anisotropy is around 0.2% and for the angular distribution anisotropy, $Q(\theta)$, it is nearer 0.8%. We bear this in mind when drawing conclusions about the isotropy or otherwise of clusters.

We measured the anisotropy of Eden clusters and found good agreement with the values found by Batchelor and Henry (1991) of $\sim +2\%$ in the axial directions and $\sim -2\%$ in the diagonal direction (see Fig. 5.16). These results verify the consistency of our choice of anisotropy metric.

[15]Consider the box counting anisotropy metric, $Q(\theta)$. In order to reach the same radius, fewer boxes will be required in the diagonal directions (since their centres will be spaced $\sqrt{2}h$ apart) than in the axial directions (where centres will be spaced a distance h apart).

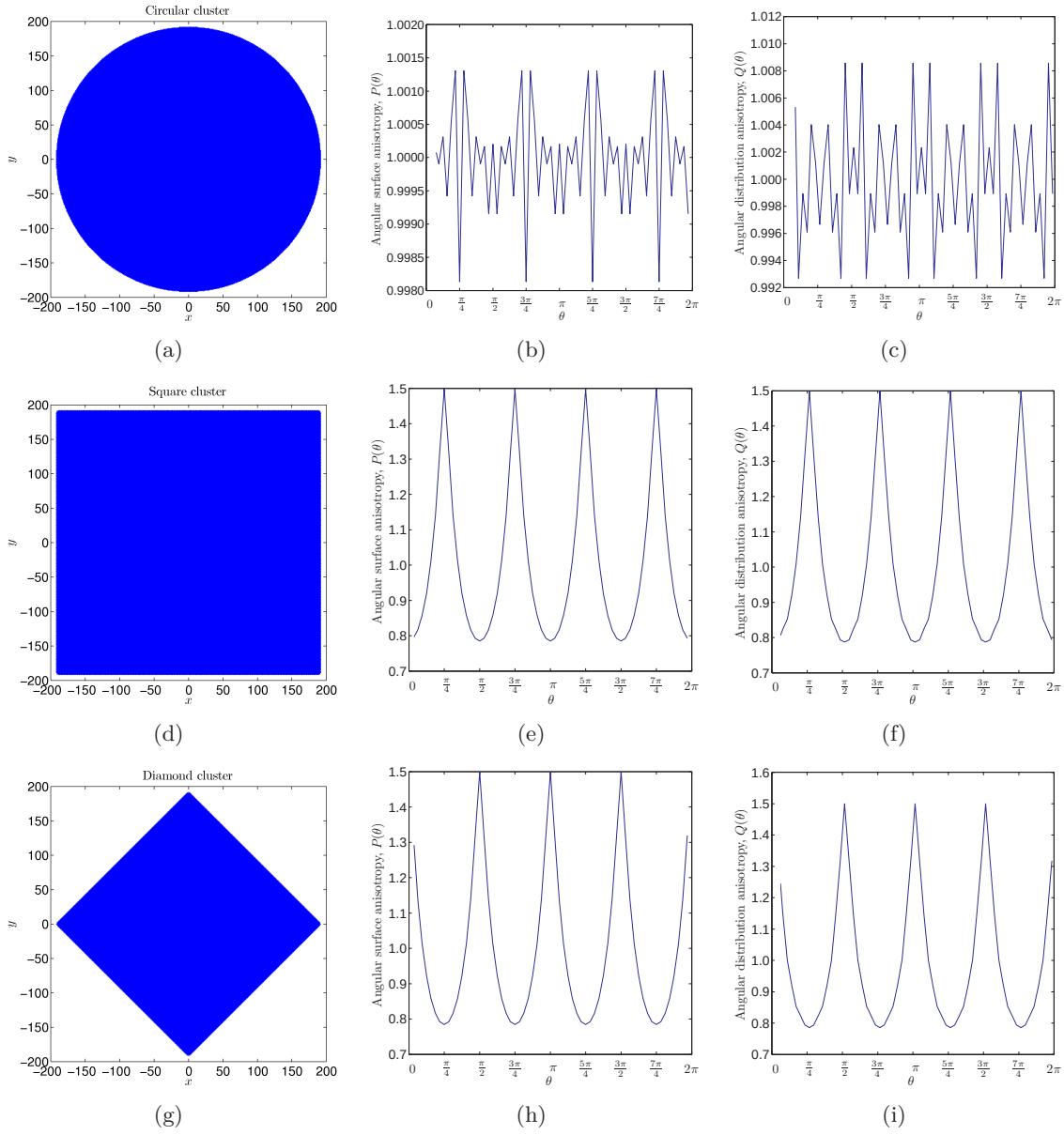
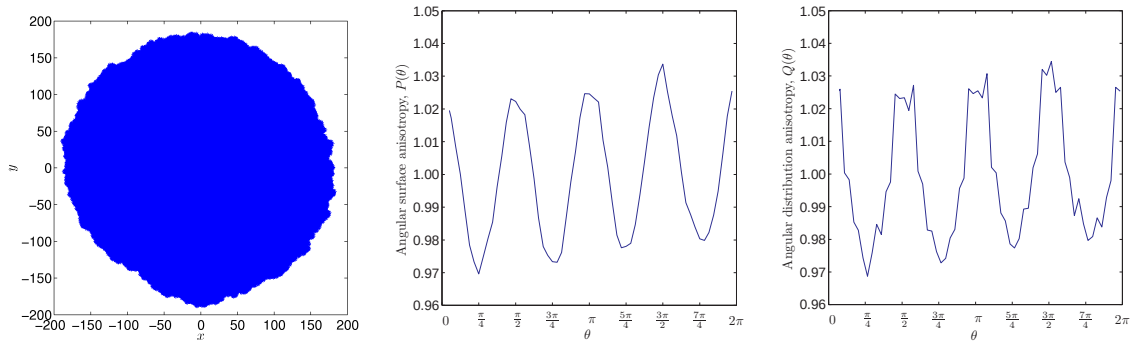


Figure 5.15: *Determining the effectiveness of anisotropy measures on some sample clusters. Both the angular surface anisotropy measure, $P(\theta)$, ((b), (e) and (h)) and angular distribution anisotropy measure, $Q(\theta)$, ((c), (f) and (i)) are able to detect strongly isotropic clusters such as the square ((d) $N = 143641$) and the diamond ((g) $N = 71821$). Both metrics give relatively constant values for the circular cluster ((a) $N = 112825$), but a small amount of anisotropy is inherent (note the different scales on the x-axis in (e), (f), (h) and (i) in comparison to (b) and (c)). This anisotropy decreases the larger the cluster size (data not shown).*

Noise reduction

Noise reduction is a technique which is used to understand better the scaling behaviour of a cluster aggregation model. New growth sites are selected as usual, but instead of filling



(a) (b) (c)

Figure 5.16: *The anisotropy of Eden clusters. (a) A typical example of an Eden C cluster displaying slightly flatter edges in the diagonal directions. (b) The angular surface anisotropy, $P(\theta)$. (c) The angular distribution anisotropy, $Q(\theta)$. Clear axial anisotropy of the clusters is observed in both measures. Clusters were grown to $N = 100001$ elements and anisotropy values are averaged over 200 repeats.*

a selected site with an element, a counter associated with that site is incremented by one. Only when the counter for a site reaches the 'noise reduction parameter', m, is an element added to that site. Clusters, therefore, cannot grow too rapidly in the same area, as 'older' perimeter sites (with a higher counter value than the newly created perimeter sites) grow in preference. The waiting time distribution at each site becomes more and more peaked as the parameter, m, increases. If, in the non-noise-reduced model, element addition can be considered to be a Poisson process, then waiting times will be exponentially distributed:

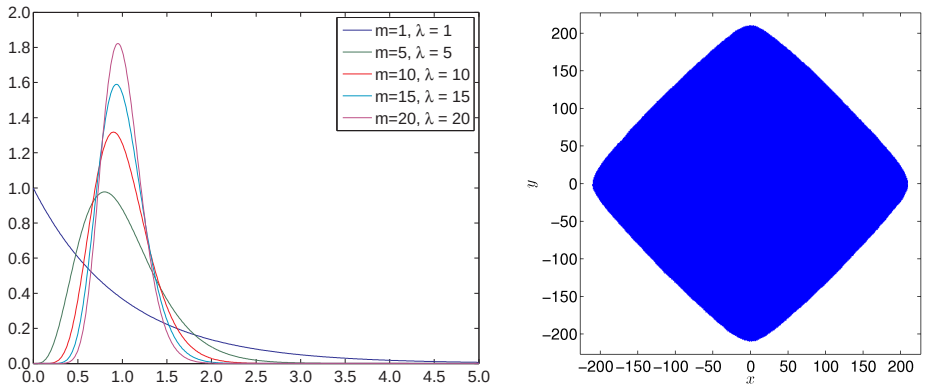
$$p(t) = \lambda e^{-\lambda t} \quad \text{for } t \geq 0,$$

where $\lambda > 0$ is the reciprocal of the expected waiting time or 'rate parameter'. The waiting time for the m^{th} occurrence of an exponentially distributed random variable is Erlang distributed:

$$p(t) = \frac{\lambda^m t^{m-1} e^{-\lambda t}}{(m-1)!} \quad \text{for } t \geq 0.$$

We set the rate parameter, $\lambda = m$, so that the mean waiting time, λ/m, remains unchanged. The larger the value of m (known as the 'shape parameter'), the more sharply peaked the Erlang distribution becomes (see Fig. 5.17 (a)). In the noise-reduced limit, $m \to \infty$, the waiting times become δ-distributed and the model becomes deterministic. Therefore, in the

Eden model, all available surface sites are filled at exactly the same time. In this limit, it is easy to see that clusters are built up by the simultaneous addition of layers of elements, one layer at a time. This leads to diamond shaped Eden clusters in two dimensions or octahedral shaped clusters in three dimensions, clearly exposing the anisotropic nature of the on-lattice Eden model (see Fig. 5.17 (b)). As such, noise reduction will be a useful tool for determining asymptotic cluster properties without having to simulate large clusters.



(a) (b)

Figure 5.17: *The noise-reduced Eden model. (a) The Erlang distribution becomes more and more peaked as the shape parameter (i.e. the noise reduction parameter), m, and the rate parameter, $\lambda(= m)$, increase. (b) A typical noise-reduced Eden C cluster ($m = 60$). The underlying axial anisotropy is clearly evident.*

5.3 New models for mitotic growth

Thus far we have focussed almost exclusively on the Eden model as the archetypal model for domain growth. However, as we have seen, even the Eden C model, which has relatively few holes, does not satisfy two of the essential criteria for a model of domain growth that we laid out at the start of this chapter, namely, domain growth by elemental division and isotropic growth. However, there are adaptations of the Eden model which are able to satisfy one or both of these demands, unfortunately at the expense of one of the other criteria on the list.

Paiva and Ferreira Jr (2007), for example, have found on-lattice versions of the Eden A and C models which are isotropic. They ensure isotropy by altering the probability of growth events, dependent on the neighbourhood of the free space into which a new element is to be positioned. Although isotropic, this model does not implement growth by mitosis

so cannot be considered a candidate for our growing domain.

Inspired by off-lattice versions of the ballistic deposition model (Ball and Brady, 1985; Jullien and Meakin, 1987) and the diffusion limited aggregation (DLA) model for cluster aggregation (Tolman and Meakin, 1989), Jullien and Botet (1987) introduced an off-lattice version of the Eden A model using circular elements. Wang et al. (1995) followed this by presenting an off-lattice version of the Eden C model and Ho and Wang (1999) extended this model to grow via mitosis. These models can also be shown to be isotropic, however, their off-lattice nature precludes them from use as our underlying domain growth model.

Despite the Eden model's inability to satisfy all of our five key criteria simultaneously, its study has provided us with tools which will be invaluable in the investigation and classification of the domain growth models we present below.

Von Neumann neighbourhood

Our first attempt at including mitosis in an on-lattice model of domain growth is simply to allow all elements of the cluster to be selected with equal probability to divide. Once selected, an element divides randomly in one of the four von Neumann neighbour directions. In order to make room for a daughter element, as many elements as necessary are pushed one element's width linearly in the chosen direction. Unsurprisingly, this simple model gives rise to a severe axial anisotropy (data not shown).

Moore Neighbourhood

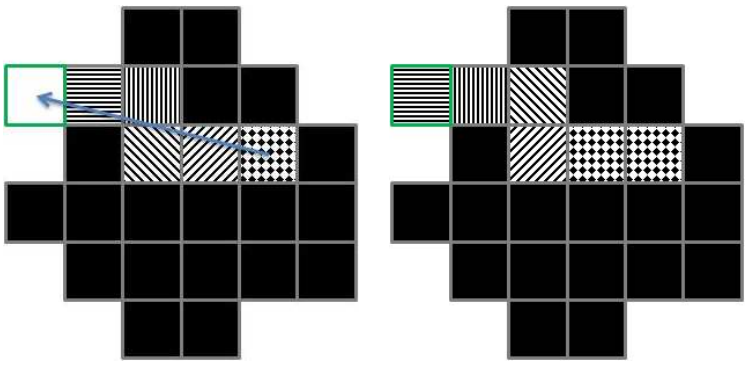
In an attempt to reduce this axial bias we allowed division in the direction of all eight of the Moore neighbours of a randomly selected element. Again, as many elements as necessary were pushed out of the way in order to make room for a daughter element. Although this markedly reduced the axial anisotropy it also introduced a diagonal anisotropy (data not shown).

Division in all directions

Drasdo (2005) introduced an on-lattice mitosis model. An element of the cluster is chosen randomly to be a division candidate and, providing it is within k elements of the boundary,

it divides. Division is such that the two daughter elements are positioned adjacent to each other and the elements of the cluster are rearranged so that the nearest empty site to the original parent element is filled. This model has been shown to be inherently anisotropic and, in the noise-reduced limit, this anisotropy is realised. It is hypothesised that in the limit $k \to \infty$ the model becomes isotropic, but this result is not tested (Drasdo, 2005). We adapt this model to make it isotropic, even in the noise-reduced limit.

We first remove the restriction on the number of neighbouring elements a dividing element can displace. In addition, we generate a uniform random number (representing a clockwise angle with the x-axis) between 0 and 2π (the 'direction of division'). Upon division a daughter element is placed in a lattice site neighbouring the parent interval in the direction of division and other elements are displaced sequentially to make room. Displaced elements follow the direction of an arrow drawn from the centre of the parent element in the direction of division to a neighbouring lattice site (if, for example, the arrow passes through one lattice site into the site above, then the element in the first lattice site moves upwards displacing the element above. See Fig. 5.18). Elements are sequentially displaced in this way until a free lattice site is encountered.



(a) (b)

Figure 5.18: *The rearrangement protocol for division in a random direction from a random parent interval. (a) An element of the cluster is chosen randomly to divide (checked pattern) and a uniform random angle is used to determine the direction in which the element will divide. A line is drawn from the parent interval in this direction until the first free site is encountered (green boundary, □). (b) Elements (various patterns) are pushed in this direction in such a way that one of them occupies the free lattice site. The daughter interval (also checked) is placed on the lattice site adjacent to the original parent interval in the direction of division.*

The angular distribution anisotropy, $Q(\theta)$, appears to indicate an axial anisotropy and

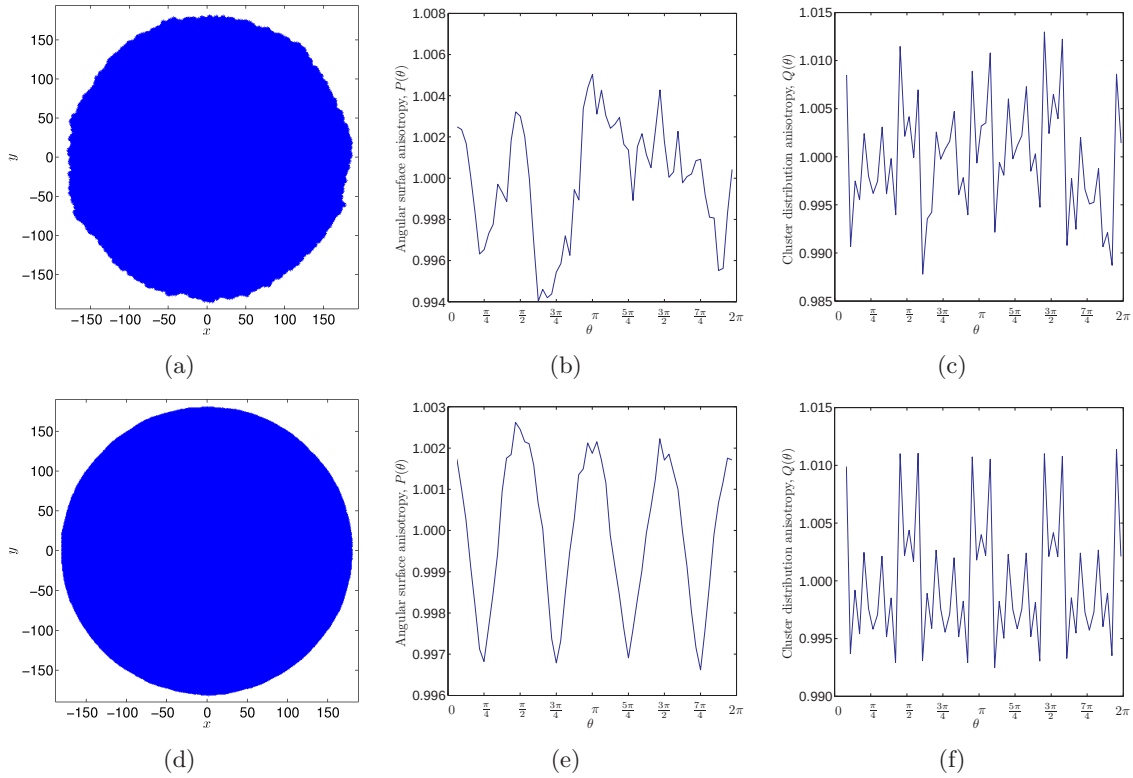
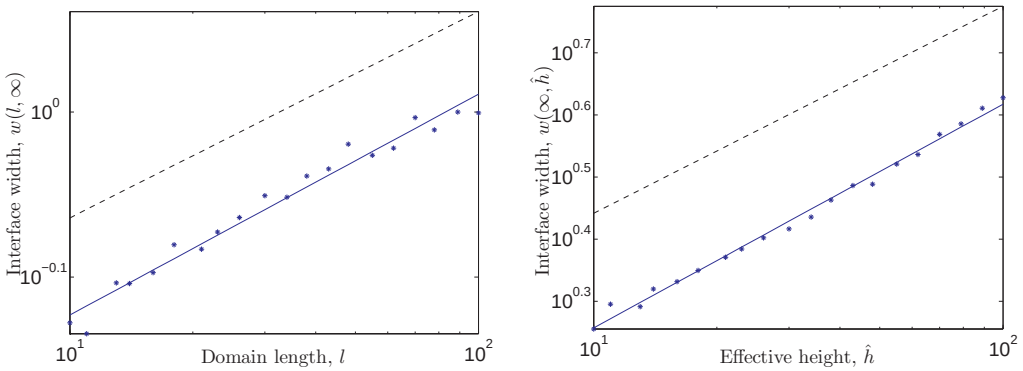


Figure 5.19: *The anisotropy of clusters grown by elements dividing in random directions. (a) A typical example of such a cluster which looks relatively circular, displaying no obvious anisotropy. (b) The angular surface anisotropy, $P(\theta)$. (c) The angular distribution anisotropy, $Q(\theta)$. (d)-(f) As for (a)-(c) but for the noise-reduced variant of the model with $m = 60$. Clusters were grown to $N = 100001$ elements and anisotropy values are averaged over 200 repeats.*

a slight diagonal anisotropy (see Fig. 5.19 (c)). However, as we have already seen, the angular surface anisotropy (see Fig. 5.19 (b)) is a more accurate measure of anisotropy when the values get close to unity. Since there is no clear periodicity to the anisotropy metric, $P(\theta)$, which we would expect with a genuine anisotropy, we conclude that this model is isotropic. We also study the noise-reduced variant of the model with parameter $m = 60$ (see Fig. 5.19 (d) for a sample cluster). We find that the noise-reduced clusters are even more isotropic than their random counterparts with a lower overall anisotropy of maximum absolute value around 0.3%. Although there appear to be periodic variations in the anisotropy we found similar variations of comparable magnitude for a truly circular cluster (see Fig 5.15 (b)). The very fact that the clusters generated by this method become more isotropic, rather than less, in the noise-reduced limit (in contrast to the Eden model (see Fig. 5.17 (b))) is a clear indication that this method of domain growth is suitable for

our purposes.



(a) (b)

Figure 5.20: *Scaling of the surface width of random direction mitosis clusters in '1+1' dimensions. (a) The variation of* $w(l,\infty)$ *(i.e.* $\hat{h} \gg l$*) with domain length,* l*, for the random directional mitosis model. The straight (black dashed) line plotted for comparison has gradient* $1/8$*. (b) The variation of* $w(\infty,\hat{h})$ *(i.e.* $l \gg \hat{h}$*) with effective height,* $\hat{h}$*. Descriptions as for (a). The straight (black dashed) line plotted for comparison has gradient* $1/3$*.*

This model appears to exhibit a classic power law dependence of surface width on domain length, l, (see Fig. 5.20 (a)) and effective height, $\hat{h}$, (see Fig. 5.20 (b)) in the appropriate limits which might lead us to conclude that the model follows the classic Family-Vicsek scaling relationship (see equation (5.2.7)). However, by studying the scaling behaviour of the surface we can see that the surface width exhibits a much more complicated non-monotonic dependence on the effective height. Fig. 5.21 (a) shows the variation of interface width with effective height for four different values of domain length. Critical exponents $\alpha = 0.15 \pm 0.07$ and $\beta = 0.35 \pm 0.07$ have been calculated in the usual manner. When the surface width is rescaled (using the Family-Vicsek scaling relationship) by the critical exponents derived, from the scaling relationships in Fig. 5.20, the data no longer collapse onto themselves (data not shown). This suggests a different functional form for the scaling relationship for this model.

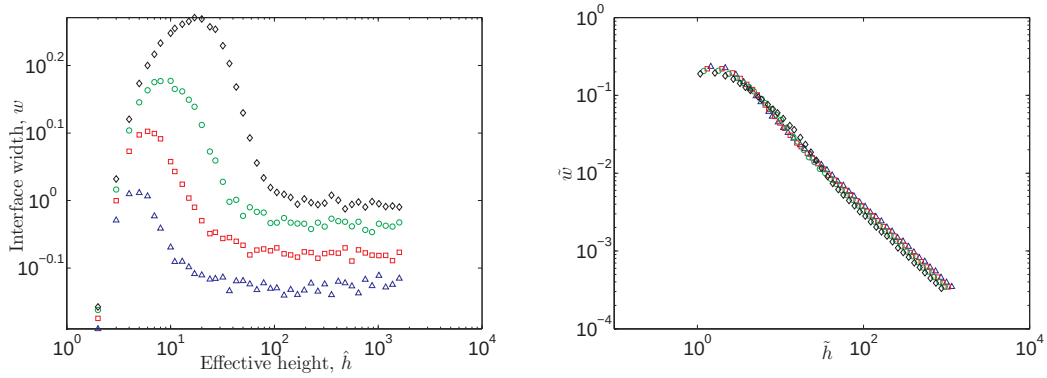
The fact that the surface width saturates at all is an artefact of the periodic boundary conditions of the strip geometry. Division events with shallow division angles displace large numbers of elements and effectively explore the surface in search of the lowest empty lattice site to occupy. This effect introduces the surface correlations required to reduce surface width, while division events with steep division angles effectively increase the surface

width. When the average surface height is low, particles chosen for division are at or near the surface a large proportion of the time. This reduces the effect of shallow division events since the surface cannot be explored fully and hence the surface width grows beyond its saturation value. As the effective height increases, a higher proportion of division events are initiated sufficiently far below the surface for shallow angle division events to allow surface exploration. This explains the non-monotonic behaviour of the surface width displayed in Fig. 5.21 (a). In a similar division model where only elements on the bottom-most layer are allowed to divide this non-monotonicity is not observed since the surface width cannot grow beyond its saturation value (data not shown).

The surface thickness of these aggregates no longer obeys the Family-Vicsek scaling relationship. Instead we propose that the thickness obeys a new scaling relationship:

$$w(l, \hat{h}) \sim l^{\alpha} \hat{h}^{\beta} f\left(\frac{\hat{h}}{l^{\gamma}}\right). \tag{5.3.1}$$

In order for the scaling of the saturation thickness, w_{cap}, to be correct we must choose $\alpha = 0.15$, the value of the roughness exponent found previously. In this case we also chose $\gamma = \alpha$ and $\beta = 1$ in order to achieve a tight collapse of the data points (see Fig. 5.21 (b)).

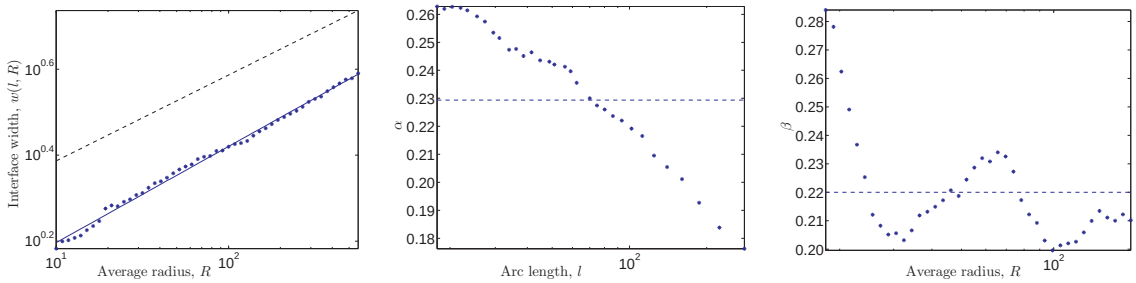


(a) (b)

Figure 5.21: *The evolution of the surface width for our random direction model for different system sizes. (a) The surface exhibits an initial power law dependence on effective height, but quickly peaks, then falls and finally plateaus at the saturation value $w_{cap}(l)$. (b) The evolution of surface width rescaled according to equation* (5.3.1). *Axis labels are short hand where $\tilde{w} = w(\hat{h}, l)/l^{\alpha}\hat{h}^{\beta}$ and $\tilde{h} = \hat{h}/l^{\gamma}$. A tight collapse of the data points can be seen. In both sub-figures blue triangles (△) correspond to system size $l = 10$, red squares (□) to $l = 20$, green circles (○) to $l = 40$ and black diamonds (◇) to $l = 80$.*

We also investigated the surface scaling of this model in a full two-dimensional geometry. The width is found to grow with critical exponent $\beta = 0.22 \pm 0.07$. The small oscillations of the local slopes about the average value of β in Fig. 5.22 (c) demonstrate that the value we have found for β is a reliable one. This value of β differs from the value of the growth exponent in the strip geometry, $\beta = 0.35 \pm 0.07$.

An even more stark difference between the strip geometry and the full two-dimensional geometry is the saturation behaviour of the surface. In the strip geometry the surface was found to saturate with roughness coefficient $\alpha = 0.15$, but in the two-dimensional geometry the surface width does not appear to saturate. Fig. 5.22 (a) demonstrates unbounded growth of the surface width for clusters of up to $N = 10^6$ particles. We should expect the surface width not to saturate since additions to the surface are completely uncorrelated. A consistent value for the roughness exponent, α, is unobtainable (see Fig. 5.22 (b)). Since the critical exponent describes the scaling of the surface thickness with a local length scale at saturation, $w_{cap} \sim l^{\alpha}$, the inability to determine α is consistent with our hypothesis that the surface width does not saturate.



(a) (b) (c)

Figure 5.22: *Scaling of the surface width of the random direction mitosis model in full two dimensions. (a) The variation of surface width, $w(l, R)$, with average cluster radius, R, for clusters up to $N = 10^6$ particles. Simulations are averaged over 20 repeats. Saturation of the surface width is not evident. The straight (black dashed) line is plotted for comparison and has gradient $1/5$. (b) Evaluation of the roughness exponent, α, using the method of local gradients (method as described in Fig. 5.13). The mean value is plotted as a dashed blue line. (c) Evaluation of the growth exponent, β, using the method of local gradients. The mean value is plotted as a dashed blue line.*

Our findings suggest that the traditional mechanisms for classifying growing aggregates into UCs, based on the saturation behaviour of their surface widths (often in a strip geometry), may be inadequate for partitioning aggregate growth models which grow by internal

elemental division (rather than apical growth, through the addition of particles to the surface). In particular, the model for isotropic domain growth presented above does not exhibit Family-Vicsek scaling behaviour and, as such, cannot be placed in a UC.

5.4 Comparison of cell migration models on a growing lattice

We have invented a model of cluster aggregation for which the traditional methods of analysing and classifying surface scaling are not sufficient. This does not, however, prevent us from employing this model in order to simulate cell migration on a two-dimensional domain which grows radially. We allow cells to diffuse on a square lattice which grows as described in Section 5.3. In addition, when an element divides the cells that originally resided in the parent element are split roughly evenly between the two daughter elements using the binomial distribution as in previous chapters. Although we only consider diffusion here, we note that it would be simple to incorporate an external signalling profile in the same manner as we did on the stationary domain. Cells in the individual-level model are not allowed to leave the domain. This corresponds to zero flux boundary conditions in the continuum model.

Given the constant rate of interval splitting, ϱ, in the individual-level model, we can write down a probability master equation for the evolution of the number of domain elements at time t. This master equation determines that the number of domain elements grows exponentially. This, coupled with the fact that each element is equally likely to split and that the growth model is isotropic, leads us to assume uniformly exponential domain growth in the continuum model. The growth occurs at a rate, ρ, which we can easily relate to the splitting rate, ϱ, in the discrete model: the increase in area due to a division event must produce a corresponding increase in area in the continuum model and (assuming growth occurs isotropically) this corresponds to an increase in radius in the continuum model.

Since the PDE, boundary and initial conditions are circularly symmetric the population-level solution will be circularly symmetric for all time. As in the case of the non-growing

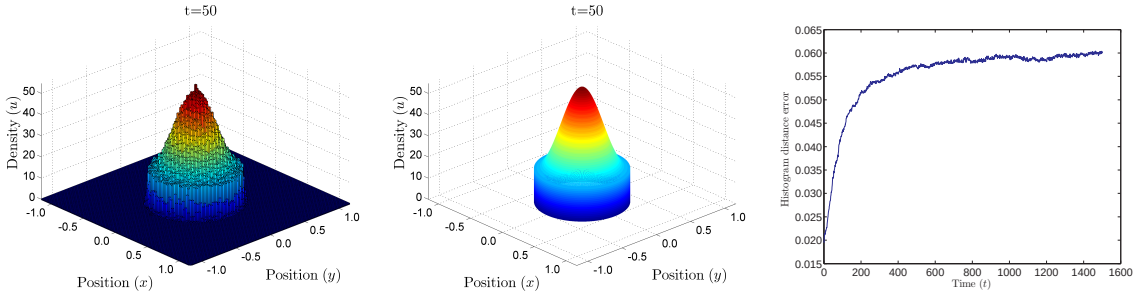
circular domain, this means we need only solve a one-dimensional PDE:

$$\frac{\partial u}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(rvu) = D\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}\right) \quad \text{for} \quad r \in [0, \exp(\rho t)/2], \quad t \in [0, \infty). \tag{5.4.1}$$

As in one dimension, the flow is given by $v = dr/dt = \rho r$, where ρ is the strain, which is determined from the element splitting rate, ϱ, in the individual-level model.

We chose the initial domain to have radius 1/2. We again solve this equation using the `NAG` solver `D03PE` (The Numerical Algorithms Group, 2009).

A snapshot comparison of the densities of the two models is given in Fig. 5.23. The models appear to give qualitatively similar results. We also plot the evolution of the HDE in Fig. 5.23 (c). Although the HDE plateaus at a larger value than for the non-growing circular domain (see Fig. 5.6 (c)), the HDEs are of the same order of magnitude, indicating a good quantitative comparison between the individual-level and population-level domain growth models in two dimensions.



(a) (b) (c)

Figure 5.23: *A comparison of the individual-level (with square domain tessellation) and population-level models of diffusion on a growing circular domain. In the individual-level model* $N = 50000$ *cells are initialised with a density profile which decays approximately exponentially from the centre to the edge of the domain. The continuum model is initialised similarly with a profile* $u = A\exp(-10r)$*, where* A*, the normalising factor, ensures that the area under the continuum initial condition corresponds to the correct number of cells in the individual-level simulation. There are initially* 1885 *elements in the circular domain of the individual-level model, as in the stationary case. Over a simulation time of 1500 time units, with an interval splitting rate of* $\varrho = 0.001$*, this number grows to* ~ 8448*. (a) The histograms represent the density of cells in each square of the domain tessellation. (b) The surface represents the radially symmetric solution of PDE* (5.4.1) *with* $D = h^2$*, where* $h = 1/49$ *is the length of an element in the individual-level model. (c) The evolution of the HDE. For a movie comparing the density of the two models as time evolves please see MOVIE_5.avi*

In this section we have employed our discrete domain growth mechanism in order to compare individual-based and population-based models of cell migration on a growing do-

main in two-dimensions. We have not, however, attempted to draw a rigorous mathematical link between the two models. This will be necessary if we are to extend our equivalence framework to include more general forms of domain growth. Our simulations can be considered a proof of concept, that the individual-level models and the population-level model can be seen to give quantitatively similar results, and provides encouragement in the search for a mathematical link between the two modelling regimes.

Summary

Modelling cell migration in more than one dimension presents a series of choices pertaining to domain discretisation and domain shape that were not pertinent to the simple line-geometry of the one-dimensional domain.

We began this chapter by demonstrating that, providing the scalings are chosen correctly, the general concept of signal sensing gives rise to the same PDE, independent of the way in which the domain is tessellated. As part of this work we presented a method for determining the correct scaling of these individual-level transition rates given the appropriate population-level PDE. We also demonstrated the extension of our equivalence framework to non-square domain geometries.

Motivated by the requirement to include growth in the two-dimensional equivalence framework we investigated on-lattice growth models. Upon finding that none of these models were able to satisfy the five key criteria for a growing domain that we set out (with isotropic growth on the regular lattice being a stumbling block for a large number of models) we presented our own domain growth model. We were able to show that our model is capable of producing circular clusters, even in the noise-reduced limit. Upon analysing our model's surface scaling properties we found that the traditional scaling laws and classification procedures for cluster aggregation models are insufficient to describe our model.

Finally, we made use of this cluster aggregation process as an underlying model for domain growth in our individual-based models. Through simulations we were able to show that the models corresponded to a population-based PDE on a radially growing domain.

We have extended the equivalence framework we have been building for the duration of this thesis to two dimensions. Extensions into still higher dimensions are relatively easy to implement based on the work laid out in this chapter.

The model we have presented here, for domain growth on a regular lattice, has many advantages. Given a PDE, the derivation of the corresponding transition rates in the individual-level model is relatively straightforward. Conversely, once we have postulated transition rates, we are able to determine the corresponding PDE by Taylor expanding and taking the relevant limit. We do not need to worry about the effects of the areas of the elements on the transition rates, since they are all the same. The regular lattice also makes it easy to convert from mean cell numbers to mean cell densities on the domain.

In Chapter 3 we presented a cell migration algorithm on a one-dimensional non-uniform lattice. This was motivated by the fact that we deemed the instantaneous element doubling and splitting domain growth method presented in Chapter 2 to be unrealistic. In the present chapter, because of the advantages outlined above, we have resorted to our original method of domain growth. This method is the obvious first-step towards building domain growth into our two-dimensional equivalence framework. However, if we are to be self-consistent, we should attempt to consolidate this first attempt by implementing domain growth via a method akin to the incremental, stochastic domain growth mechanism of Chapter 3.

Such an implementation necessitates the consideration of unstructured meshes, such as the general mesh presented in Section 5.1.1. Unstructured meshes will also enable us to deal more accurately with complex domain geometries including curved boundaries (Kieri, 2011). However, rather than a method for determining whether the appropriate transition rates (in order that the mean cell densities correspond to a particular PDE in the continuum limit) have been chosen, as presented in Section 5.1.1, we require a method which will determine those transition rates in the first place (Engblom et al., 2009). Despite their obvious advantages in terms of realism, these unstructured meshes do not come without their challenges. They require careful consideration in order to determine transition rates so that the individual-level model corresponds to a PDE (Engblom et al., 2009; Kieri, 2011). This is complicated, in part, by the fact that the areas of all the domain elements will, in general, be different, which makes it difficult to determine the correct transition rates for

cell density rather than just cell number. Finally, as the lattice adapts to take account of growth events the propensity functions will change dynamically and it will be necessary to recalculate them, which may be computationally expensive.

The incorporation of unstructured meshes into our equivalence framework is one of the many extensions we will discuss in more detail in the further work section in the next chapter.

Chapter 6

Discussion

Cell migration is crucial for a variety of developmental and homeostatic mechanisms in the human body from conception through to death. It has been documented as early as the stage of gastrulation (Keller, 2005) in embryogenesis and is important for wound healing and many aspects of the immune response, amongst other biological functions. Cell migration is also implicated in many pathologies including vascular disease, chronic inflammatory diseases and rheumatoid arthritis. A more thorough appreciation of cell migration offers the potential to cure life-threatening and debilitating diseases as well as improving our understanding of homeostatic mechanisms in order that we might correct them when they go wrong. These applications make a clear understanding of the detailed mechanisms of cell migration one of the most important current topics in biology. It is imperative, therefore, that the scientific community charged with investigating cell migration make use of all the tools available to them, both experimental and theoretical. One tool which is being used more and more frequently in order to test and verify experimental hypotheses, as well as making predictions about and guiding the experiments themselves, is mathematical modelling.

In general there is a dichotomy between the types of models used for the mathematical representation of cell migration. On one side of the divide are continuum models capable of representing the population-level characteristics of a group of cells and often amenable to mathematical analysis. The solutions of such models are fast to simulate numerically (if necessary) and can often be linked explicitly to the model parameters. However, it is

difficult for these models to capture fine-scale detail and be linked directly to experimental data; abilities which are inherent to the discrete models, on the other side of the divide. Discrete models are often intuitive to formulate and are able to incorporate stochasticity naturally. However, it is often difficult to link the results of such models directly to the model parameters in order to gain a population-level overview and their simulation can be time-consuming. Clearly there are gains to be made by bridging the divide between these two modelling regimes in order to exploit their complementary strengths and avoid their complementary weaknesses.

6.1 Conclusions

The central theme that runs through this thesis is connecting these two disparate modelling regimes in what we have called an 'equivalence framework'. We begin with an individual-level description of the specific phenomenon we are trying to model (perhaps inspired directly by experiments) and from this we are able to determine a continuum equivalent which will replicate the population-level characteristic of the model in the appropriate limits. We can use the individual-level representation to parametrise the model and use the continuum level-description to link the population-level results back to the parameters of interest. The key step in this process is to be able to link the models in such a way that the parameters of the individual-level model feed directly into the formulation of the continuum-level model.

There are a range of models in the literature which may be used to represent migrating cells as discrete entities. In order to derive a useful continuum model we must, when choosing an individual-level representation of cell migration, strike a balance; the model must be sufficiently sophisticated that we can capture accurately the detailed movement of individual cells but also be sufficiently straightforward that it can be readily linked to a continuum equivalent.

The on-lattice position-jump model employed throughout this thesis can be considered to be a fair compromise. Transition rates between lattice sites can be determined directly from experimental data and each cell represented as an independent entity. The on-lattice formulation establishes, in a relatively simple manner (through the use of the RDME and

Taylor expansions) a direct link to a corresponding PDE. In addition, by using the RDME to generate relationships between the mean number of cells in each interval, we are able to account for non-linear interactions (such as density-dependent cell movement) in the way that simply deriving an FPE for occupation probability from an SDE which describes the motion of an individual cell, for example, could not.

We have focused our efforts, throughout this thesis, on expanding the previously limited range of scenarios which our equivalence framework (both at the individual and continuum levels) is capable of representing: where we have encountered difficulties in representing the appropriate situations in our individual-level models, we have adapted the models to fit our needs; where we have been frustrated by time-consuming simulations, we have introduced methods for their acceleration; and where we have discovered difficulties in transitioning between discrete and continuum models we have invented new techniques to help us bridge the gap.

6.1.1 One dimension

At the beginning of Chapter 2 we outlined the equivalence framework in its most basic form: the density of cells obeying an unbiased random walk on a regular lattice in one dimension can be shown, using the RDME, to evolve according to the diffusion equation, providing we have enough cells and a sufficiently fine lattice. We built on this foundation by introducing time-varying signal profiles (which the cells can sense, respond to and alter), cell proliferation and death and density-dependent cell migration. However, the main focus of the chapter was on introducing domain growth to the equivalence framework. For the first time we introduced growth, via interval doubling and splitting, to an on-lattice position-jump model. We envisage this being of use, not just to the community of cell migration modellers, but also to the reaction-diffusion modelling community who are realising that it is important to be able to segue between discrete-stochastic and continuum-deterministic models in order to determine the effects of noise on their systems. Indeed, it has already begun to be used by Woolley et al. (2011) to investigate the effects of domain growth on stochastic reaction-diffusion models. We should note, however, that the implementation of bimolecular reactions (in order to model interacting chemical species) is more complicated

than for the simple unimolecular reactions we have used to model the diffusion/advection of cells. In particular the propensity functions should depend, in a non-trivial manner, on the size of the sub-compartments into which the domain is divided (Erban and Chapman, 2009).

Importantly, we were able to demonstrate the equivalence of this individual-based model on a growing domain, through an adapted RDME, to a PDE. We were then able to incorporate logistic, linear and even density-dependent growth into our equivalence framework. In each case we compared the two models by carrying out extensive numerical simulations of both the position-jump and the PDE models. It was possible to show quantitatively, in most cases, that there was a good agreement between the two modelling paradigms. There were, however, instances, such as the case of quadratic density-dependent domain growth, where the two models did not appear to agree. In these cases the disparity was assumed to be due to the inaccuracy of our moment closure assumptions and the effects of low copy numbers in the individual-based model, causing the divergence of the growth rates of the stochastic and continuum models.

Unfortunately, although mathematically convenient, the interval-splitting method of domain growth seems flawed from a physical point of view: lattice elements should not be allowed to double in size instantaneously. In Chapter 3 we set about correcting this problem. We introduced an incremental growth process in which domain elements can grow independently by arbitrarily small amounts. Such a growth process introduces variable lattice spacing, even in an initially uniform lattice. Before deriving a continuum equivalent to the individual-level model incorporating domain growth, we first considered a similar derivation on a stationary, non-uniform domain. This required us to define precisely what is meant by 'cell density', a concept that was not required on the uniform domain, since cell numbers and cell density were directly proportional. In turn, this required us to specify the element boundaries. This can be achieved naturally in two ways: the Voronoi domain partition and the interval-centred domain partition. We showed, by deriving the transition rates from an assumed underlying SDE, which determines the cells' movements at the microscale, that only the Voronoi domain partition affords cell densities in the individual-level model that can be shown to be equivalent to those of a population-level PDE. Upon rein-

troducing domain growth it was necessary for us to use a novel interpretation of the RDME in order to derive a continuum equivalent. Our theoretical results were again corroborated with extensive numerical simulations and the quantitative correspondence between the two models was found to be slightly improved. We expect that this improvement is due to the fact that the new incremental growth process mirrors more closely the continuum growth process.

The findings of this chapter (including the necessity of the Voronoi domain partition and the derivation of a PDE corresponding to a realistic incremental domain growth process) again have implications, not only for the modelling of cell migration, but also to the wider position-jump modelling community. It will be important to define precisely what is meant by cell density if the effects of noise on reaction-diffusion systems are to be modelled using this irregular-lattice position-jump process.

6.1.2 Stochastic simulation

Motivated by the necessity to reduce the computational complexity of cell migration algorithms when moving into higher dimensions, in Chapter 4 we considered a variety of methods in the literature for increasing the efficiency of a general SSA. We also considered acceleration methods, such as τ-leaping, for a general class of models and noted the importance of species non-negativity. Building on ideas in the literature we introduced a mechanism for determining species criticality using the exact species depletion distribution to approximate our confidence in a species number remaining non-negative. For all the reaction systems we considered this method was of at least comparable speed to existing τ-leaping techniques and in some simulations (especially for simulations representative of our spatially extended cell-migration system) a significant speed increase was demonstrated, whilst accuracy was maintained.

We envisage applications of this acceleration technique beyond the simple spatially extended single species models presented in this thesis. Several methods already exist for the efficient exact (Elf and Ehrenberg, 2004) or accelerated (in the case where there are many more diffusive movements than chemical reactions) (Drawert et al., 2010; Lampoudi et al., 2009) simulation of inhomogeneous, spatially extended reaction-diffusion systems. Our al-

gorithm will complement these methods by providing an accelerated method of simulating position-jump reaction-diffusion models independent of the relative frequency of reaction and diffusion events.

6.1.3 Two dimensions

By taking a general approach on an arbitrary lattice in two dimensions, in Chapter 5 we were able to derive a set of equations which must be satisfied in order for transition rates to correspond to a macroscopic diffusion equation. We showed on a variety of regular lattices that these equations uniquely determine the correct transition rates, but can also be used to find the PDE which corresponds to predefined transition rates[1]. We found, remarkably, that the abstract concepts of local, non-local and average signal sensing give rise to the same PDE irrespective of the regular domain tessellation. These findings were confirmed via numerical simulation and quantitative comparison of the two models.

Upon investigating individual-level models for radial domain growth in two (or higher) dimensions, we found that it was necessary to create our own isotropic cluster aggregation model. We used techniques from the literature and invented our own anisotropy metric in order to determine that the model we introduced was truly isotropic, even in the low-noise limit.

We completed this chapter, in analogy to the work undertaken in Chapter 2, by incorporating this isotropic model of domain growth into the individual-based cell migration algorithm in two dimensions. Through numerical simulations we saw a good qualitative agreement between the individual-based model and the equivalent continuum model on the growing domain. Although circular/spherical domain growth may not be relevant to most biological domain growth situations, it certainly is to some (including avascular tumour growth (Folkman and Hochberg, 1973)). Above all, the comparison can be considered a proof-of-concept which can serve to inspire further work connecting individual- and continuum-level models on domains which grow in a manner which is not so straightforward (Smith et al., 2011).

[1]Transition rates might be chosen to incorporate the effects of signal sensing.

6.2 Further work

Although the additions and improvements presented in this thesis (and outlined above) have helped to make our equivalence framework more comprehensive, there are still several areas into which our framework can be extended in order to increase its versatility and applicability.

Mathematically trivial, but physically important extensions include altering the boundary conditions to incorporate a time-dependent flux of cells into and out of the domain. The ability to introduce or remove cells at the boundary will allow us to model more realistic situations. However, with Robin boundary conditions, for example, the type of individual-based model (e.g. position-jump, velocity-jump etc) has an important effect on the implementation of the boundary conditions (Erban and Chapman, 2007).

Although we have already introduced cell proliferation and death, there are perhaps more realistic ways to achieve this. For example, cells may be assigned a finite lifetime after which they reproduce or die. In a more complicated extension, each cell may be assigned a model of the cell-cycle, in the form of a set of ODEs, with which it is able to determine when to proliferate or die according inherent cell-cycle dynamics coupled to signals received from the environment (Pitt-Francis et al., 2009). Some success, in deriving population-based representations from individual-level models incorporating cell-cycle dynamics, has been had by phenomenologically incorporating the effects of such a microscale cell-cycle model as a proliferation term in a macroscale model (Murray et al., 2009; Van Leeuwen et al., 2009).

Cell proliferation is also often one of the driving forces behind domain expansion. Although we have attempted to include this phenomenon in our models by including density-dependent domain growth and cell proliferation and death, these two factors have not been linked in a consistent way. The problem of how to model domain growth due to cell proliferation deserves a more thorough consideration as it is of vital importance to many areas of development.

6.2.1 One dimension

The domain partition dichotomy between the Voronoi and the interval-centred domain partitions, introduced in Chapter 3 in order to allow incremental domain growth, is by no means perfect. It is obvious that on the stationary domain we should use the Voronoi partition to ensure the correct number of cells in each interval. However, on the growing domain we cannot derive a closed-form PDE for cell density with the Voronoi partition due to the enforced nature of interval growth: two intervals must grow at the same time in order to preserve the Voronoi property of the domain, thus preventing each interval from growing exponentially. We have seen that the interval-centred domain partition allows us to derive a master equation which corresponds closely with the expected PDE for cell density on the growing domain. We have also seen, however, that the cell densities on this partition will, ultimately, not correspond to the mean-field cell densities. This poses a question as to whether a form of domain partition duality might be more appropriate. The interval-centred domain partition may be employed to ensure the correct interval growth rates and the Voronoi domain partition used in order to find the correct cell density (see Fig. B.3 in Appendix B.3 for an example of this duality on a stationary domain). Each time we require cell density it would be simple enough, keeping interval centres as fixed points, to find the corresponding Voronoi domain partition on which we might expect cell density to be correct. This, does, however, pose questions as to what exactly we mean by cell density. Even more troubling, from a computational point of view, is the situation in which cell migration or domain growth are dependent on cell density. This would require conversion from the interval-centred to the Voronoi domain partition at every time-step which would surely lead to a large increase in simulation times.

6.2.2 Stochastic simulation

There are several possible areas in which we believe stochastic simulation can be improved. For example, random number generation is often highlighted as being a costly step in the SSA (Gibson and Bruck, 2000). Although there are versions of the SSA which use as few as one random number per time-step (Anderson, 2007; Gibson and Bruck, 2000), they also

use complex data structures in order to keep track of the reaction due to occur next and require the re-use of random numbers at each time-step. We suggest that it may be possible to use only one random number per time-step in a much simpler manner.

First generate a uniform random number, U, and determine which reaction will occur, in the usual way, by finding into which segment of the partitioned unit domain (the sizes of the segments corresponding to the values of the propensity functions for each reaction, normalised so that they sum to unity) the random number falls. The random number, U, will also be distributed uniformly in this segment. If the segment has upper and lower limits $l_{\min}$ and $l_{\max}$, respectively, then we can rescale the the uniform random number so that it again lies uniformly in the unit interval: $U' = (U - l_{\min})/(l_{\max} - l_{\min})$. U' can now be used to determine the time for the next reaction. Although sound in theory, this method may encounter problems in practice due to the finite precision of the random numbers generated, especially when some propensity functions are very small in comparison to others. The small propensity functions will take up a small segment of the unit interval and, although they will be chosen infrequently, when they are chosen they may cause the rescaled random numbers to have far fewer digits. This loss of accuracy in the uniform random number will feed through to a loss of accuracy for the time-step generated. For example, if our random numbers are generated initially with k digits of accuracy and the smallest interval is of size 10^{-j} (where $0 < j < k$) then the resulting random number we will use in order to generate the time-step will only have $k - j$ digits of accuracy, which will lead to inaccuracy when $j \sim k$. This brings to the fore a more general problem for the SSA. A reaction whose normalised propensity function is represented by a segment of the unit interval which has size less than machine precision will never be selected. This may cause a problem for systems with huge numbers of reactions and/or widely varying propensity function magnitudes. However, for the majority of simulations this will not cause a problem.

As we saw at the start of Chapter 4, efficient search methods constitute another way to increase the speed of exact SSAs. Many of the methods we outlined have yet to be tested explicitly in the context of the SSA. A comparison of the efficiency of the 'K-level' search methods presented by Blue et al. (1995), the ODM of Cao et al. (2004), the SDM of McCollum et al. (2006) and the logarithmic direct method of Li and Petzold (2006) for a

general class of reaction systems (and specifically, in the context of this work, for spatially extended systems) would be of great use.

Eukaryotic cells such as leukocytes are capable of spatial sensing, which requires the establishment of differential signalling across the cell length. However, it has recently been suggested that this does not preclude the possibility that leukocytes are also capable of temporal sensing in a manner akin to that of free-swimming bacterial cells. Potentially, if a signal gradient is too shallow to allow spatial sensing, it may be possible for cells to activate dormant temporal sensing as an alternative navigation strategy (F. Lin, *pers. com.*). Models for temporal sensing dynamics in bacteria are well-established, however, potential intracellular mechanisms for the temporal sensing in eukaryotic cells are only beginning to be postulated (Erban and Othmer, 2007; Levine et al., 2006; Swaney et al., 2010). Temporal sensing requires the cell to have a 'memory' of the signal profile it experiences as it travels on a random walk. In order to implement such a memory in our position-jump models it will be necessary to incorporate delays in SSAs. Fortunately, Anderson (2007) has recently formulated an adapted SSA which does exactly this. In combination with cell migration experiments, detailed, individual-level mathematical models of cell migration in response to shallow gradients may be able to provide insights into the temporal sensing process of leukocytes.

In the main section of Chapter 4 we introduced confidence-based methods for determining criticality of species in order to avoid negative populations in ASSAs. One possible cause of erroneous classification of criticality which may occur in our confidence-based methods (and hence a possible reduction in speed of the algorithms) is due to the fact that we are forced to use the time-step, τ, from the previous step of the algorithm in order to approximate the distribution in the change of species number for the current step. We justified this approximation by appealing to the leap-condition: that propensity functions, and hence the proposed time-step τ, will not change significantly between reactions. However, there is likely to be some degree of change which will lead to erroneous classification. One possible way to circumvent this problem would be to iteratively choose τ and species criticality in a manner similar to that implemented by Harris and Clancy (2006) when classifying reactions in their 'partitioned leaping' approach: start with the value of τ from the previous time-step

and classify reactions as critical or otherwise based on this τ. Then update τ according to the criticality status of the reaction channels. Return and reclassify the reactions using this updated value of τ. Repeat this process until reaction classifications cease to change. Whether this iterative method of τ-selection would converge quickly and whether the advantage gained from the precise classification of reactions would outweigh the cost of the iterative procedure remain unexplored areas for further investigation.

There is currently much debate about the best method for choosing τ so as to gain the maximum acceleration whilst observing the leap condition (Cao et al., 2006; Gillespie, 2001; Gillespie and Petzold, 2003). The method of τ-selection proposed by Cao et al. (2006) approximates the relative change in propensity function by the relative change in the reactant species populations. A value of τ is then determined by bounding the relative change in species to produce a relative change in propensity function which is less than a desired error tolerance, ε. The maximum absolute value of the change in the random variable, $\Delta_\tau X_i$ for $i = 1, \ldots, N$, describing the change in the number of molecules of species i, is estimated by an approximation to the absolute value of its mean, $|\langle \Delta_\tau X_i \rangle|$ plus or minus an approximation to its standard deviation $\sqrt{\text{var}\Delta_\tau X_i}$. This estimation approach leaves the method open both to over-estimation of τ, risking violation of the leap condition and hence inaccuracy, and under-estimation of τ, which risks a significant reduction in speed of the algorithm. Ideally we would have a more consistent method.

It might be possible to adapt this method of τ-selection so that instead of *estimating* $\Delta_\tau X_i$, the *precise* distributions, from which the changes in species number are drawn, can be found (as we did for determining criticality) as the sums and/or differences of Poisson m-lets. Using this distribution it would be possible to find the value of τ which ensures that the restriction on the relative change in species number holds at a pre-specified confidence level (using, for example, a two-sided $(100 - \alpha)\%$ confidence interval where α specifies our desired confidence level). This way we can describe quantitatively how often our conditions on the change in species numbers, and hence the corresponding conditions on the change in propensity functions, hold and therefore be more assured in our method of τ-selection.

The next step in the acceleration of stochastic simulation beyond ASSAs is to extend the approximation approach in order to derive SDEs for the numbers of cells in each interval

(Gillespie, 2000). The SDE that is derived from a set of ODEs by adding noise is often known as the Chemical Langevin Equation (CLE). It is possible to rigorously derive the CLEs for cell migration simulations and to show that the simulation time of these CLEs is dramatically faster than both the ESSA and the ASSA. However, when simulating, we again encounter problems with negative species numbers which, this time, are not so easy to overcome. It is also unclear how well the CLE approach will adapt to domain growth. These extensions are certainly worth exploring due to the potentially large increase in simulation speed provided.

6.2.3 Two dimensions

In Chapter 5 we spent a considerable amount of time studying (and finding isotropic versions of) the Eden model: a canonical example of cluster aggregation. We eventually ruled it out as an underlying model for the growing domain since it did not exhibit the ubiquitous domain growth we required. However, it is possible that we could employ the Eden model as a model of apical/surface domain growth (in which only the elements on or near the boundary are capable of proliferation) which is known to occur in a variety of biologically important situations including the growth of shoots and roots in many plants, limb bud growth and the growth of tumours (Drasdo, 2005) and shells (Meinhardt, 2009).

The individual-based position-jump model we have used throughout this thesis is somewhat artificial in that it restricts a cell's movement to a single jump of a fixed distance between neighbouring lattice nodes, irrespective of the time since its last movement event took place. However, one can instead postulate that random walkers can jump further than the distance between lattice sites in a single jump (Meysman et al., 2003). For example, as a first step, we could propose that a random walker at lattice site i can jump to site j, where $|i - j| \leq 2$. Using the notation introduced in Chapter 2 we can write down a probability master equation for the vector of cell numbers, $\boldsymbol{N}$, at time t. From this master equation we

can derive relationships for the evolution of the mean cell numbers, M_i, in each interval:

$$\frac{\partial M_1}{\partial t} = T_2^{-1}M_2 + T_3^{-2}M_3 - \left(T_1^{+1} + T_1^{+2}\right)M_1, \tag{6.2.1}$$

$$\frac{\partial M_2}{\partial t} = T_1^{+1}M_1 - \left(T_2^{+1} + T_2^{-1} + T_2^{+2}\right)M_2 + T_3^{-1}M_3 + T_4^{-2}M_4, \tag{6.2.2}$$

$$\frac{\partial M_i}{\partial t} = T_{i-2}^{+2}M_{i-2} + T_{i-1}^{+1}M_{i-1} - \left(T_i^{+1} + T_i^{-1} + T_i^{+2} + T_i^{-2}\right)M_i + T_{i+1}^{-1}M_{i+1} + T_{i+2}^{-2}M_{i+2}, \tag{6.2.3}$$

$$\frac{\partial M_{k-1}}{\partial t} = T_{k-3}^{+2}M_{k-3} + T_{k-2}^{+1}M_{k-2} - \left(T_{k-1}^{+1} + T_{k-1}^{-1} + T_{k-1}^{-2}\right)M_{k-1} + T_k^{-1}M_k, \tag{6.2.4}$$

$$\frac{\partial M_k}{\partial t} = T_{k-2}^{+2}M_{k-2} + T_{k-1}^{+1}M_{k-1} - \left(T_k^{-1} + T_k^{-2}\right)M_k, \tag{6.2.5}$$

where equation (6.2.3) holds for $i = 3, \ldots, k-2$. Transition rate T_a^b describes the rate at which cells move from lattice site a to lattice site $a+b$. In order to derive an appropriate PDE in the continuum limit we must choose these transition rates carefully. We would expect the mean square displacement of a particle undergoing a diffusive process, for example, to increase linearly with time. This observation can be reformulated to provide a relationship between the transition rates to neighbouring regular-lattice nodes $i+1$ and $i+2$ from node i : $T_i^{+2} = T_i^{+1}/4$. Choosing $T_i^{+1} = T_i^{-1} = T_{i-1}^{+1} = T_{i+1}^{-1} = D/(2h^2)$ and $T_i^{+2} = T_i^{-2} = T_{i-2}^{+2} = T_{i+2}^{-2} = D/(8h^2)$ allows us to recapitulate the one-dimensional diffusion equation for cell density in the continuum limit. Clearly, at least theoretically, we can extend this idea so that cells can jump between any two lattice nodes, providing the transition rates are chosen appropriately.

As noted at the end of Chapter 5, the use of unstructured meshes will also enable us to deal more accurately with complex domain geometries, including curved boundaries, and will facilitate the inclusion of arbitrary domain growth into our equivalence framework. Methods already exist for solving reaction-diffusion equations on dynamically changing meshes (Smith et al., 2011), but methods for linking these PDEs to individual-level models are only in their infancy (Engblom et al., 2009; Kieri, 2011). Engblom et al. (2009) use mass lumping in conjunction with the FEM in order to determine transition rates and to ensure that the rates between non-neighbouring nodes are zero. However, given the non-local position-jump model described above, it may not be strictly necessary to do this if

we allow jumps to non-neighbouring nodes. As well as making their method simpler and more faithful to our equivalence-framework this may help to resolve the problem of deriving negative transition rates which affects around 20% of transition rates derived using the mass lumping method. Kieri (2011) uses the finite volume method (FVM) rather than the FEM to overcome the problem of negative transition rates, but this method only works on a Voronoi domain partition in which nodes are connected by a Delaunay triangulation. On the growing domain the triangulation would presumably have to be recalculated after each growth event. Whether a complete re-tessellation is necessary after each growth event will be presumably determined by the method in which growth is implemented. It may be possible to rearrange the lattice after a growth event by introducing only local disturbances. Even so, given the current methods for determining transition rates the work required to update the local rates alone will be equivalent to the work to update them all.

6.2.4 Other extensions

Hybrid modelling

Thus far we have considered the two modelling regimes as distinct, to be used one at a time as necessary. In some situations, it may be possible that the number of cells in one part of the domain is high enough that the cells can and should (for computational purposes) be modelled as a continuum. However, in another part of the domain the number of cells may not be sufficiently large to justify such continuum assumptions and cells should, therefore, be modelled as individuals. The equivalence framework described in this thesis contains the basic ground work required for a hybrid model of cell migration which could address this problem. However, questions remain as to how to segue adaptively between the two modelling regimes and what is the appropriate limit on cell density in which to do so.

Reaction-Diffusion Modelling

The question of robust pattern formation for stochastic processes on growing domains is one of much biological interest. The individual-continuum framework we have developed in this thesis can be used to couple reaction-diffusion processes at the individual-level to

those at the population level. For example, until now there has been no method to simulate such stochastic processes robustly, without artificially changing the nature of the pattern formed by implementing instantaneous 'growth and division' events. Our new incremental interval growth method, and its corresponding PDE, will allow for the correct simulation of stochastic pattern formation and evolution, even with strong ubiquitous growth. The implementation of a reaction-diffusion system on a stationary non-uniform lattice will also be of considerable interest since it will necessitate a precise definition of concentration and will require differing reaction rates for the same reaction in each interval. Pattern formation via a reaction-diffusion mechanism on non-square meshes in two dimensions is also a candidate for exploration; how a regular domain tessellation will affect the pattern formed is not immediately obvious. Still less obvious is the effect that a semi-regular tessellation will have.

Experimentally-driven model verification

The ultimate goal of our research is to provide insights into the mechanisms underlying cell migration and domain growth. Therefore, experimentally-driven model building is vital to assess the validity of the type of models presented in this thesis. Model verification could be undertaken initially by comparison to relatively simple cell migration experiments carried out in microfluidic devices (Lin et al., 2004, 2005). It also seems prudent that further model analysis and parametrisation, followed by repeats of the 'alter-predict-test' cycle, should be carried out in order to suggest ways in which the models should be improved.

All of the extensions outlined in this section are feasible, would improve the current work and contribute towards making our stochastic equivalence framework a useful tool for biologists and mathematicians alike. We hope to implement as many of these extensions as possible.

6.3 Summary

Our equivalence framework is just one of a growing number of individual-continuum descriptions (Alber et al., 2007; Murray et al., 2009; Simpson et al., 2009a, 2010; Turner et al.,

2004) which will ultimately benefit the cell migration community as a whole: not only will the community have a wide variety of individual-based models from which to choose in order to best represent the key dynamical features of the process they are trying to model, but they will also have a continuum equivalent which they can use to ascertain the macroscale properties of their cell populations.

Given the nature of the individual-based model in our framework, we suggest it will be best suited to modelling cells that migrate via a crawling mechanism (Ridley et al., 2003) as opposed to swimming cells, like many species of planktonic bacteria, which are propelled by flagellar motors (Blair, 1995). Crawling cells tend to be sufficiently large and their sensing mechanisms sufficiently sensitive that they can detect a difference of less than 10% in the concentration of a chemoattractant between their front and rear (Ridley et al., 2003). This allows such cells to undertake spatial sensing rather than temporal sensing imposed upon bacterial cells by their smaller size and makes a position-jump-type model suitable.

We began, in Chapter 2, by outlining a relatively simple version of such a position-jump model on a regular, one-dimensional lattice, for which we were able to find a PDE equivalent. However, throughout the course of this thesis we have been able to extend and improve our equivalence framework, incorporating a variety of physically realistic processes into the individual-based model. Importantly, we were able to incorporate domain growth in a position-jump model for the first time and to confirm its place in the equivalence framework by finding a corresponding PDE. We also extended the framework into higher dimensions and demonstrated methods that will help to keep simulation times for the individual-based models reasonable, even in these higher dimensions.

By incorporating some or all of the ideas outlined in the further work section above we will continue to make our equivalence framework more versatile and ultimately better equipped to answer relevant biological questions.

Appendix A

Appendices for Chapter 2

A.1 Derivation of the stochastic mean equations from the master equation

Recall the master equation (2.1.2) from Section 2.1,

$$\begin{aligned}\frac{\partial \Pr(\boldsymbol{n},t)}{\partial t} &= \sum_{i=1}^{k-1} T_i^+ \left\{(n_i+1)\Pr(J_i^+\boldsymbol{n},\boldsymbol{s},t) - n_i \Pr(\boldsymbol{n},\boldsymbol{s},t)\right\} \\ &+ \sum_{i=2}^{k} T_i^- \left\{(n_i+1)\Pr(J_i^-\boldsymbol{n},\boldsymbol{s},t) - n_i \Pr(\boldsymbol{n},\boldsymbol{s},t)\right\}. \qquad \text{(A.1.1)}\end{aligned}$$

In order to find the mean cell numbers in interval j, M_j, we multiply the master equation by n_j and sum over all possible values that the vector of cell densities, $\boldsymbol{n}$, can take:

$$\begin{aligned}\frac{\partial M_j}{\partial t} &= \sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} n_j \left(\sum_{i=1}^{k-1} T_i^+ \left\{(n_i+1)\Pr(J_i^+\boldsymbol{n},\boldsymbol{s},t) - n_i \Pr(\boldsymbol{n},\boldsymbol{s},t)\right\} \right. \\ &+ \left. \sum_{i=2}^{k} T_i^- \left\{(n_i+1)\Pr(J_i^-\boldsymbol{n},\boldsymbol{s},t) - n_i \Pr(\boldsymbol{n},\boldsymbol{s},t)\right\} \right). \qquad \text{(A.1.2)}\end{aligned}$$

For ease of comprehension, from now on in this derivation, we will write out the action of the operators $J_i^{\pm}$ on $\boldsymbol{n}$ in long hand. Split both the inner sums on the RHS of equation

(A.1.2) into two parts each and consider them separately. First consider,

$$\sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} n_j \sum_{i=1}^{k-1} T_i^+(n_i+1)\Pr(n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_k,t). \tag{A.1.3}$$

If $i \neq j, j-1$ then each term of this expression reduces to $T_i^+\langle n_j n_i\rangle$, where $\langle n_j n_i\rangle$ represents the mean of the product $n_j n_i$. However, for $i = j, j-1$ we have to be more careful. If $i = j$, then

$$\begin{aligned}
&\sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} T_j^+ n_j(n_j+1)\Pr(n_1,n_2,\ldots,n_j+1,n_{j+1}-1,\ldots,n_k,t),\\
&= \sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} T_j^+ \left\{(n_j+1)^2-(n_j+1)\right\}\Pr(n_1,n_2,\ldots,n_j+1,n_{j+1}-1,\ldots,n_k,t),\\
&= T_j^+\left\{\langle n_j n_i\rangle - M_j\right\}.
\end{aligned} \tag{A.1.4}$$

In a similar manner, if $i = j-1$, then

$$\begin{aligned}
&\sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} T_{j-1}^+ n_j(n_{j-1}+1)\Pr(n_1,n_2,\ldots,n_{j-1}+1,n_j-1,\ldots,n_k,t),\\
&= \sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} T_{j-1}^+ \left\{(n_j-1)(n_{j-1}+1)+(n_{j-1}+1)\right\}\Pr(n_1,n_2,\ldots,n_j+1,n_{j+1}-1,\ldots,n_k,t),\\
&= T_{j-1}^+\left\{\langle n_j n_i\rangle + M_{j-1}\right\}.
\end{aligned} \tag{A.1.5}$$

When these terms are brought back together, expression (A.1.3) simplifies to

$$\sum_{i=1}^{k-1}\left(T_i^+\langle n_j n_i\rangle\right) + T_{j-1}^+ M_{j-1} - T_j^+ M_j. \tag{A.1.6}$$

Next consider the term:

$$-\sum_{n_1,n_2,\ldots,n_k=0}^{\mathcal{N}} n_j \sum_{i=1}^{k-1} n_i \Pr(n_1,\ldots,n_i,\ldots,n_k) = -\sum_{i=1}^{k-1} T_i^+\langle n_j n_i\rangle. \tag{A.1.7}$$

Adding the simplifications of the two terms from the top line of equation (A.1.2) (given by terms (A.1.6) and (A.1.7)) we are left only with $T_{j-1}^+ M_{j-1} - T_j^+ M_j$. If we carry out a similar analysis for the two terms coming from the second line of equation (A.1.2) then we

are left with $T^-_{j+1}M_{j+1} - T^-_j M_j$. Combining these results gives

$$\frac{\partial M_j}{\partial t} = T^+_{j-1}M_{j-1} - \left(T^+_j + T^-_j\right) M_j + T^-_{j+1}M_{j+1}. \tag{A.1.8}$$

We can apply a similar method to find the relationship in the special cases of interval 1 and k,

$$\begin{aligned} \frac{\partial M_1}{\partial t} &= T^-_2 M_2 - T^+_1 M_1, \\ \frac{\partial M_k}{\partial t} &= T^+_{k-1}M_{k-1} - T^-_k M_k, \end{aligned}$$

where in each case the contribution from one of the sums on the RHS of the master equation vanishes.

A.2 Derivation of consistent boundary conditions on a time-independent domain

Consider the general formulation of the PDE derived for the stationary domain in Section 2.1,

$$\frac{\partial u}{\partial t} = \frac{\partial \left(\mathcal{J}\left(u, s\right)\right)}{\partial x}, \tag{A.2.1}$$

where the flux operator is as defined in Section 2.1. If we specify that u must be conserved then, in one dimension on a domain $x \in [a, b]$, we require

$$\int_a^b u \mathrm{d}x = C, \tag{A.2.2}$$

where C is a constant with respect to time. Differentiate equation (A.2.2) with respect to t. Since the limits are independent of t we can bring the derivative inside the integral:

$$\int_a^b \frac{\partial u}{\partial t} \mathrm{d}x = 0.$$

Using the PDE (A.2.1) we can substitute for $\partial u/\partial t$ and integrate the LHS with respect to x to obtain

$$[\mathcal{J}(u,s)]_a^b = 0, \Longrightarrow \mathcal{J}(u,s)\,|_b = \mathcal{J}(u,s)\,|_a, \tag{A.2.3}$$

the periodic boundary condition. As an example consider diffusion on $[0,1]$, as in Section 2.1, so that $\mathcal{J}(u,s) = \partial u/\partial x$. The zero flux boundary condition is consistent with equation (A.2.3):

$$\left.\frac{\partial u}{\partial x}\right|_{x=0,1} = 0. \tag{A.2.4}$$

A.3 Boundary conditions on the growing domain

In analogy to our analysis of cell migration on the time-independent domain there is a general form of the PDE derived for the one-dimensional growing domain:

$$\frac{\partial u}{\partial t} = \frac{\partial\left(\mathcal{J}(u,s)\right)}{\partial x} - \frac{\partial vu}{\partial x}. \tag{A.3.1}$$

Consider the conservation of cell density, u, on the growing domain, $[0, L(t)]$:

$$\int_0^{L(t)} u\mathrm{d}x = C, \quad \Longrightarrow \quad \frac{d}{dt}\int_0^{L(t)} u\mathrm{d}x = 0, \tag{A.3.2}$$

since C is a time independent constant. Differentiate under the integral to give

$$\int_0^{L(t)} \frac{\partial u}{\partial t}\mathrm{d}x + \left.\frac{\partial l}{\partial t}u\right|_{L(t)} = 0, \tag{A.3.3}$$

and then substitute in for $\partial u/\partial t$ from equation (A.3.1) and evaluate the integral to obtain

$$[\mathcal{J}(u,s) - vu]_0^{L(t)} + \left.\frac{\partial l}{\partial t}u\right|_{L(t)} = 0. \tag{A.3.4}$$

Recall the relationship between the flow and the domain position:

$$v = \frac{\mathrm{d}x}{\mathrm{d}t}. \tag{A.3.5}$$

Evaluating this expression for v at the end of the domain allows us to substitute for $\partial l/\partial t|_{L(t)}$ in (A.3.4):

$$\left[\mathcal{J}\left(u,s\right) - vu\right]_0^{L(t)} + vu|_{L(t)} = 0. \tag{A.3.6}$$

Since $v(0) = 0$ (as the origin remain fixed) the terms in v cancel and we are left with the familiar periodic boundary conditions:

$$\left[\mathcal{J}\left(u,s\right)\right]_0^{L(t)} = 0. \tag{A.3.7}$$

which are consistent with zero-flux boundary conditions.

Appendix B

Appendices for Chapter 3

B.1 The necessity of the Voronoi partition

Recall equations (3.1.11)-(3.1.13) of Section 3.1.2 relating the mean cell densities in the stochastic model:

$$\begin{aligned}
\frac{\partial u_1}{\partial t} &= \frac{1}{l_1}\left(T_2^- u_2 l_2 - T_1^+ u_1 l_1\right), \\
\frac{\partial u_i}{\partial t} &= \frac{1}{l_i}\left(T_{i-1}^+ u_{i-1} l_{i-1} - \left(T_i^+ + T_i^-\right) u_i l_i + T_{i+1}^- u_{i+1} l_{i+1}\right), \quad i = 2, \ldots, k-1, \\
\frac{\partial u_k}{\partial t} &= \frac{1}{l_k}\left(T_{k-1}^+ u_{k-1} l_{k-1} - T_k^- u_k l_k\right),
\end{aligned} \tag{B.1.1}$$

where u_i is the density of cells in interval i and l_i is its length. Upon Taylor expansion[1] of the density terms in equation (B.1.1) about x_i we are left with a system of three equations which must be satisfied in order for the cell densities in the stochastic model to recapitulate the diffusion equation in the continuum model:

$$T_{i-1}^+ l_{i-1} h_i - \left(T_i^+ + T_i^-\right) l_i + T_{i+1}^+ l_{i+1} = 0, \tag{B.1.2}$$

$$T_{i-1}^+ l_{i-1} h_i = T_{i+1}^- l_{i+1} h_{i+1}, \tag{B.1.3}$$

$$T_{i-1}^+ h_i^2 l_{i-1} + T_{i+1}^- h_{i+1}^2 l_{i+1} = 2 l_i D. \tag{B.1.4}$$

[1] Recall the definitions of the distances between domain points: $h_i = x_i - x_{i-1}$, $h_{i+1} = x_{i+1} - x_i$, $h_1 = 2x_1$ and $h_{k+1} = 2(y_k - x_k)$.

These equations (valid for $i = 2, \ldots, k$) come from equating coefficients of u_i, $\partial u_i/\partial x$ and $\partial^2 u_i/\partial x^2$, respectively, in the RHS of equation (B.1.1) (once Taylor expansions have been carried out) and the RHS of the diffusion equation. Equations (B.1.3) and (B.1.4) can be used to find the transition probabilities T^+_{i-1} and T^-_{i+1} in terms of interval lengths:

$$T^+_{i-1} = \frac{2Dl_i}{l_{i-1}h_i\left(h_i + h_{i+1}\right)}, \tag{B.1.5}$$

$$T^-_{i+1} = \frac{2Dl_i}{l_{i+1}h_{i+1}\left(h_i + h_{i+1}\right)}. \tag{B.1.6}$$

These in turn determine the transition rates T^+_i and T^-_i by a simple relabelling of indices:

$$T^+_i = \frac{2Dl_{i+1}}{l_i h_{i+1}\left(h_{i+1} + h_{i+2}\right)}, \tag{B.1.7}$$

$$T^-_i = \frac{2Dl_{i-1}}{l_i h_i\left(h_{i-1} + h_i\right)}. \tag{B.1.8}$$

Substituting these expressions for the transition probabilities into equation (B.1.2) leaves us with the relationship:

$$\frac{l_i}{h_i\left(h_i + h_{i+1}\right)} + \frac{l_i}{h_{i+1}\left(h_i + h_{i+1}\right)} - \frac{l_{i+1}}{h_{i+1}\left(h_{i+1} + h_{i+2}\right)} - \frac{l_{i-1}}{h_i\left(h_{i-1} + h_i\right)} = 0. \tag{B.1.9}$$

In order to satisfy this equation the length of interval i must be chosen to be proportional to the distance between the neighbouring interval definition points of interval i:

$$l_i = \alpha\left(h_i + h_{i+1}\right) = \alpha(x_{i+1} - x_{i-1}), \tag{B.1.10}$$

for some constant α. The additional constraint that the intervals must be contiguous, covering the domain without any overlaps or gaps between intervals, $\sum_{i=1}^{k} l_i = y_k$, provides us with a value for the constant of proportionality: $\alpha = 1/2$. With this value of α the relationship (B.1.10) defines the Voronoi domain partition. It is of comfort to note that these choices of l_i, in combination with equations (B.1.7) and (B.1.8), recapitulate the transition rates that were derived in Section 3.1.2 from consideration of an underlying microscopic process.

B.2 Further analysis of the interval-centred partition on the stationary domain

Our aim in this section is to test whether the effects of different domain partitions for each repeat of the simulation can lead to a better correspondence between the PDE and the stochastic simulations on the interval-centred partition. In order to test this we ran similar stochastic simulations to those described in Section 3.1.5 with the alteration that the non-uniform partition was different for each repeat of the simulation. In order to take an average cell density from each simulation we mapped the cell density of each repeat onto a uniform domain partition, and averaged the cell density on this partition, as in Section 3.4. Fig. B.1 shows the results of these simulations for the simple diffusion process. Clearly the stochastic cell density profile matches the PDE cell density profile much better than in Fig. 3.4 of Chapter 3 where the domain partition is the same for each simulation. This is demonstrated qualitatively by the lower HDE in Fig. B.1 (f).

It is possible that this improved correspondence between the stochastic simulations and the PDE is an artefact of the regularised repartitioning carried out in order to take averages of the stochastic simulations. In order to test this hypothesis we re-ran the simulations with the same domain partition for each repeat, as in Section 3.1.5, but instead of using these cell densities to compare the stochastic simulation to the PDE we redistribute the cell density onto a regular domain partition as we did in Fig. B.1. The results of these simulations are shown in Fig. B.2. Although the repartitioning improves the qualitative and quantitative comparison between the stochastic simulations and the PDE it is clear that it is by no means the only contributing factor. Repartitioning alone reduces the HDE by half, but repartitioning in combination with random initial domains for each repeat reduces the HDE by an order of magnitude in comparison with the original simulations. This suggests that randomising the domain for each repeat may be a viable method of obtaining a good approximation to the PDE from the stochastic simulations on the interval-centred domain.

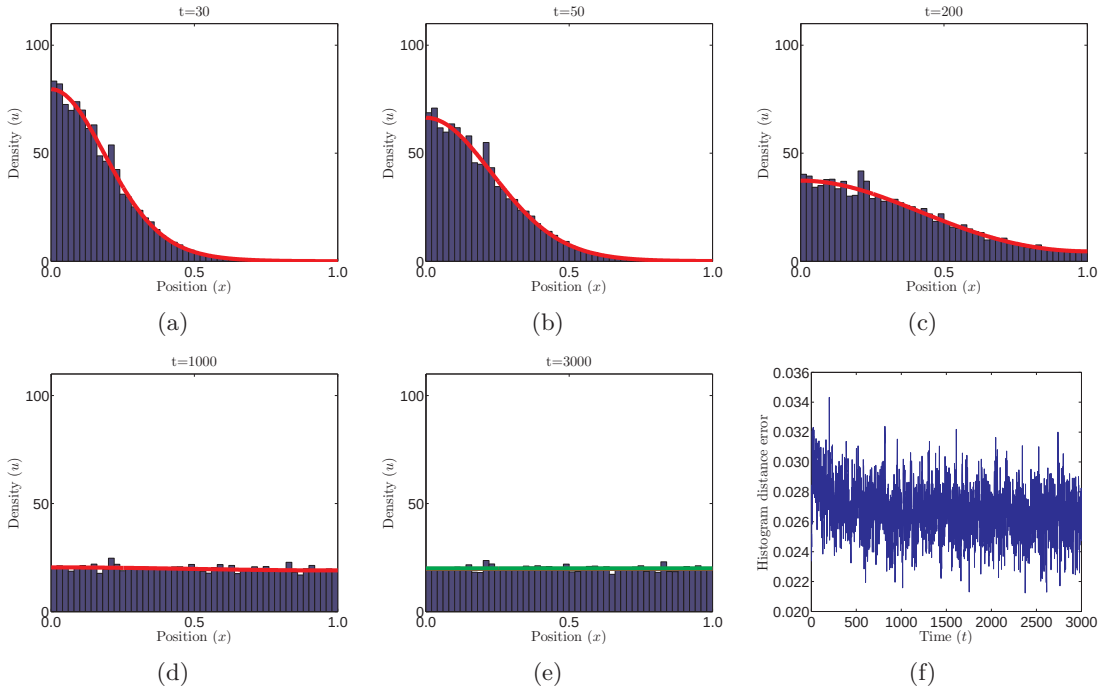


Figure B.1: *Cells undergoing simple diffusion on an interval-centred domain partition, different for every repeat, at several time points. (a)-(e) Histograms represent an average of 40 stochastic realisations. The solid lines exhibit the result of numerical simulation of the corresponding diffusion equation and the green curve in (e) represents the stationary solution derived analytically. (f) The HDE between the stochastic simulations and the PDE. The boundaries are assumed to be reflecting. Initially cell density has an exponentially decaying profile across the domain. Transition rates are given by equations* (3.1.9) *and* (3.1.10)*. Parameters are as follows:* $k = 50$, $\Delta x = 1/k = 0.02$, $D = \Delta x^2$.

B.3 Domain growth on the interval-centred domain partition with densities given by the Voronoi partition.

As previously mentioned, it might be possible to implement exponential domain growth on the interval-centred domain partition, in order to ensure the correct interval growth rates, but to use the Voronoi domain partition when considering cell densities. As a test case we consider the same principle on the stationary domain. The definition of the edges of the intervals are irrelevant to the simulation of cell numbers. It is only when we come to finding cell densities that the interval edges become important. It seems feasible, therefore, that we could implement a cell migration simulation on an interval-centred domain partition and map to the Voronoi domain partition only when we are interested in cell densities. Fig. B.3 shows this to be the case. Qualitatively, in Fig. B.3 (a)-(e) the stochastic cell density profiles

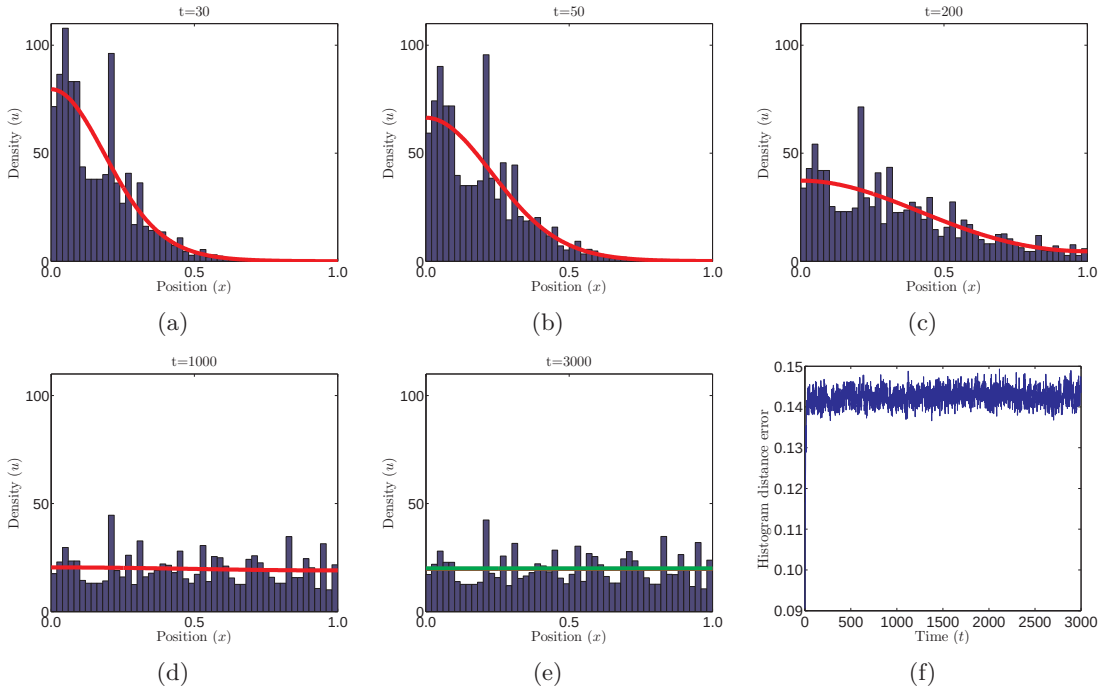


Figure B.2: *Cells undergoing simple diffusion on the same interval-centred domain partition for every repeat, at several time points. (a)-(e) Histograms represent an average of 40 stochastic realisations. The solid lines exhibit the result of numerical simulation of the corresponding diffusion equation and the green curve in (e) represents the stationary solution derived analytically. (f) The HDE between the stochastic simulations and the PDE. The boundaries are assumed to be reflecting. Initially cell density has an exponentially decaying profile across the domain. Parameters and transition rates are as in Fig. B.1.*

match the PDE much better than in Fig. 3.4 in Section 3.1.5, and the HDE in Fig. B.3 (f) is over an order of magnitude lower than when densities are calculated on the interval-centred domain partition.

The case for growing domains is somewhat different in that the lengths of intervals (and therefore the positions of interval edges) are important when determining the propensity of each interval to grow. It is not immediately clear whether using the interval-centred domain partition to implement domain growth and the Voronoi domain partition when considering densities will improve our results.

By implementing the Voronoi domain partition in order to find densities, Fig. B.4 demonstrates both a qualitative and quantitative improvement over the case when the interval-centred domain partition is used to implement growth *and* calculate cell densities (*c.f.* Fig. 3.11 of Section 3.4.2). It also appears to be at least as accurate as using the Voronoi domain partition for implementing growth and calculating cell densities (results

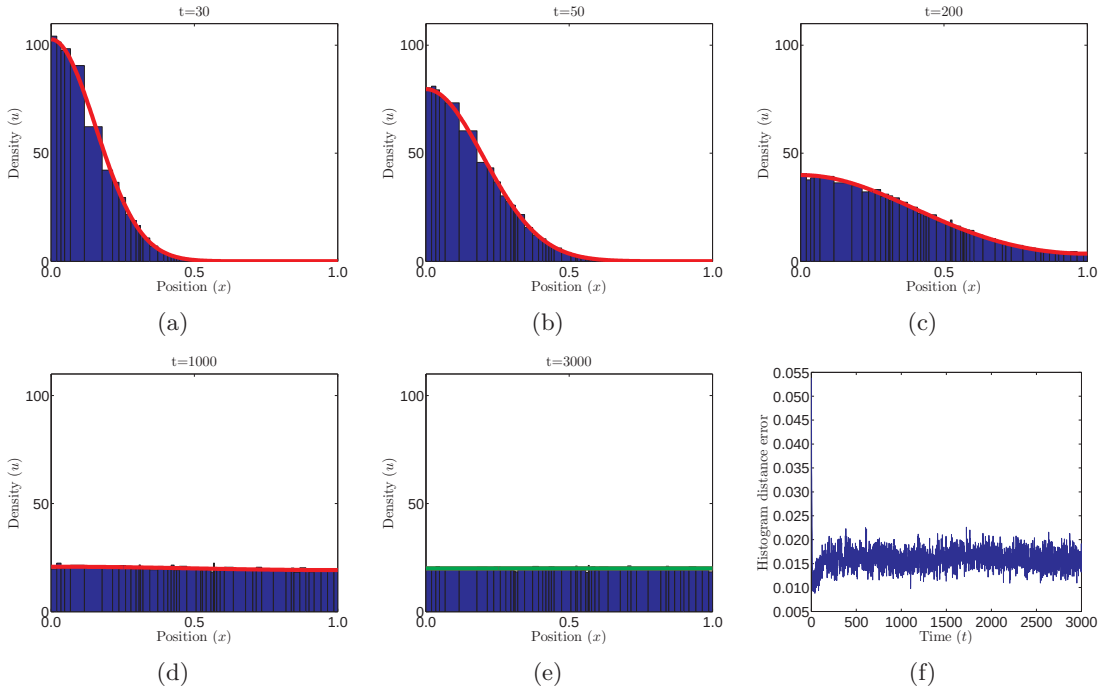


Figure B.3: *Cells undergoing simple diffusion on an interval-centred domain partition, the same for every repeat, at several time points. Densities are calculated on the Voronoi domain partition. Details are as in Fig. B.1. The boundaries are assumed to be reflecting. All cells are initialised in the first interval. Parameters and transition rates are as in Fig. B.1.*

not shown).

B.4 Further analysis of the interval-centred partition on the growing domain

Fig. B.5 shows the evolution of the average cell density for an initially non-uniform, interval-centred domain partition. The initial partition is different for each repeat of the simulation. The cell density from the discrete model is closer to the density in the PDE model than in the case when all repeats were initialised with the same domain partition (see Fig. 3.11 in Section 3.4.2). This is evidenced by the smaller HDE throughout the duration of the simulation shown in Fig. B.5 (d).

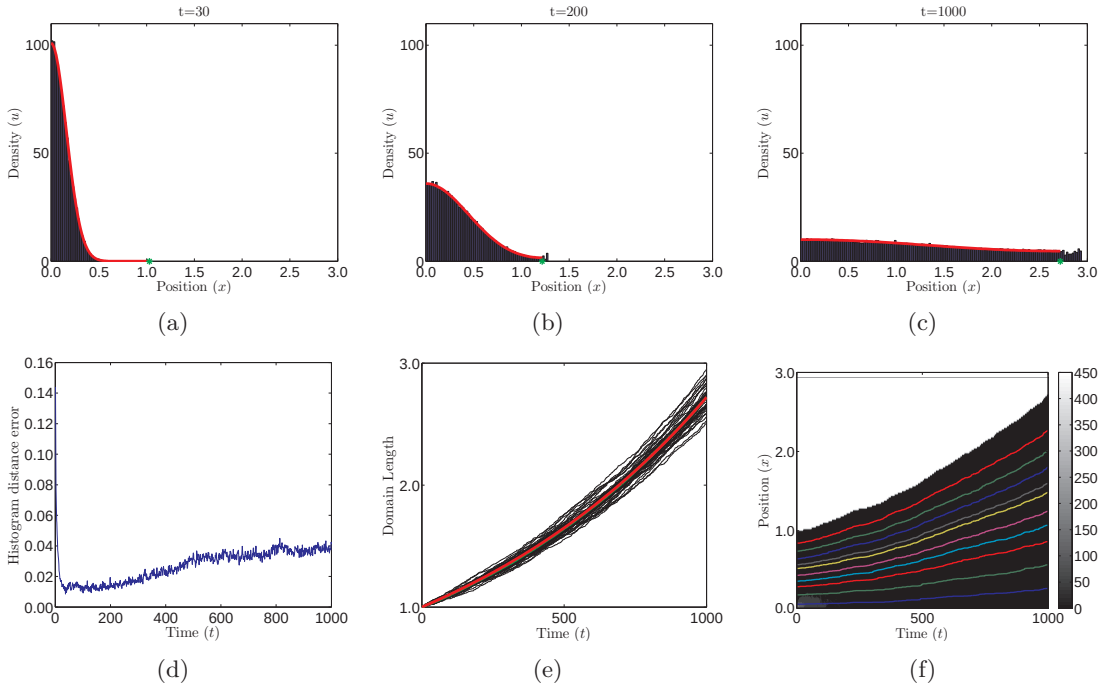


Figure B.4: *Cells undergoing diffusion at several time points. The underlying stochastic model has an initially non-uniform, interval-centred domain partition (the same for each repeat) and undergoes stochastic interval growth. Densities are calculated on the Voronoi domain partition. Figure descriptions are as in Fig. 3.9 in Section 3.4.1. All cells are initialised in the first interval. Transition rates are given by equations* (3.1.9) *and* (3.1.10)*. Parameters are as follows:* $k_0 = 50$, $\Delta x = 1/k_0 = 0.02$, $D = \Delta x^2$, $r = 0.0001$, $\Delta l = 0.1 \times \Delta x = 0.002$, $\Delta x_{split} = 2\Delta x = 0.04$.

B.5 Derivation of a PDE for mean cell numbers divided by mean cell length for the interval-centred domain partition

As noted previously, instead of deriving a PDE for cell density it is possible (and indeed simpler) to derive, as a proxy for cell density: a PDE for average cell numbers divided by average interval length. We begin by assuming average exponential domain growth of each interval, as proved in Section 3.2.3:

$$\langle l_j \rangle (t) = l_j(0) \exp(r\Delta l t) \quad \text{for} \quad j = 1, \ldots, k, \tag{B.5.1}$$

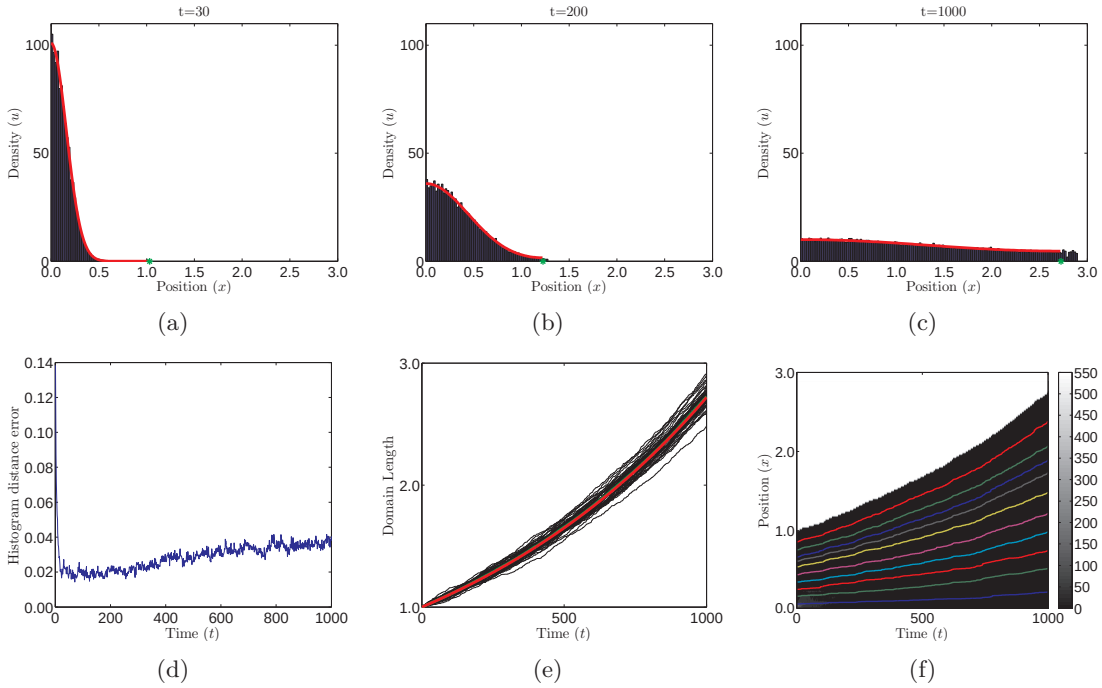


Figure B.5: *Cells undergoing diffusion at several time points. The underlying stochastic model has an initially non-uniform, interval-centred domain partition (different for each repeat) and undergoes stochastic interval growth. Figure descriptions are as in Figs. 3.9 and 3.10. All cells are initialised in the first interval. Transition rates are given by equations* (3.1.9) *and* (3.1.10)*. Parameters are as follows:* $k_0 = 50$, $\Delta x = 1/k_0 = 0.02$, $D = \Delta x^2$, $r = 0.0001$, $\Delta l = 0.1 \times \Delta x = 0.002$, $\Delta x_{split} = 2\Delta x = 0.04$.

where $l_j(0)$ is the initial size of interval j. We define $x_j(t)$ as the mean position of the left-hand edge of interval j:

$$x_j(t) = \sum_{i=1}^{j-1} \langle l_i(t) \rangle = \left(\sum_{i=1}^{j-1} l_i(0) \right) \exp(r\Delta lt) = x_j(0) \exp(r\Delta lt), \tag{B.5.2}$$

where $x_j(0) = \sum_{i=1}^{j-1} l_i(0)$ is the initial position of the left-hand edge of interval j. Consider a continuum level approximation to the density of cells across the domain,

$$n(x_j, t) = \frac{N_j}{\langle l_j(t) \rangle}, \tag{B.5.3}$$

where N_j is the number of cells in interval j at time t which does not change during an interval growth event. Then

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}n(x_j,t) &= \frac{\partial n}{\partial t}+\frac{\partial n}{\partial x}\frac{\mathrm{d}x_j}{\mathrm{d}t}, \\ &= \frac{\partial n}{\partial t}+r\Delta lx\frac{\partial n}{\partial x}. \end{aligned} \tag{B.5.4}$$

However,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{N_j}{l_j}\right) &= -\frac{N_j}{l_j^2}\frac{\mathrm{d}l_j}{\mathrm{d}t}, \\ &= -r\Delta ln. \end{aligned} \tag{B.5.5}$$

Equating equations (B.5.4) and (B.5.5) for the time derivative of cell density leads us to a PDE for cell density:

$$\frac{\partial n}{\partial t}+r\Delta lx\frac{\partial n}{\partial x}=-r\Delta ln, \tag{B.5.6}$$

which can be re-written as

$$\frac{\partial n}{\partial t}+\frac{\partial}{\partial x}\left(r\Delta lxn\right)=0. \tag{B.5.7}$$

In the absence of cell movement between intervals, this is exactly the equation we would expect for the evolution of cell density (see equation (1.3.1) in Chapter 1).

Appendix C

Appendices for Chapter 4

C.1 Properties of the Poisson distribution

C.1.1 Proof of Property 2

The probability that the difference of the two random variables $\mathcal{P}_1 - \mathcal{P}_2$ (which we will now denote by $\mathcal{SK}$ in this derivation) takes the value k is the sum from $j = 0, \ldots, \infty$ of the probability that the first random variable, $\mathcal{P}_1$, takes the value $k + j$ multiplied by the probability that the second random variable, $\mathcal{P}_2$, takes the value j. Using the definition of the probability density function of the Poisson random variable we can rewrite the above more mathematically as:

$$\Pr(\mathcal{SK} = k) = \sum_{j=0}^{\infty} \frac{e^{-\lambda_1} \lambda_1^{(k+j)}}{(k+j)!} \times \frac{e^{-\lambda_2} \lambda_2^{j}}{j!}. \tag{C.1.1}$$

This can be simplified to

$$\begin{aligned} \Pr(\mathcal{SK} = k) &= e^{-[\lambda_1+\lambda_2]} \left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} \sum_{j=0}^{\infty} \frac{(\lambda_1\lambda_2)^{(k/2+j)}}{(k+j)!j!}, \\ &= e^{-[\lambda_1+\lambda_2]} \left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} I_k(2\sqrt{\lambda_1\lambda_2}), \end{aligned} \tag{C.1.2}$$

where $I_k(z)$ is the modified Bessel function of the first kind.

C.1.2 Proof of Property 3

We prove Property 3 using PGFs. The PGF of the sum of a Poisson m-let and a Poisson n-let is given by

$$\begin{aligned} G(z) &= E(z^{\mathcal{P}^m+\mathcal{P}^n}), \\ &= G_{\mathcal{Q}_1}(z^m)G_{\mathcal{Q}_2}(z^n), \\ &= e^{\lambda_1(z^m-1)} \times e^{\lambda_2(z^n-1)}, \\ &= \exp\{\lambda_1 z^m + \lambda_2 z^n - (\lambda_1+\lambda_2)\}, \end{aligned} \tag{C.1.3}$$

where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are Poisson distributions with means λ_1 and λ_2, respectively. The first equality makes use of the assumption that $\mathcal{P}^m$ and $\mathcal{P}^n$ are independent. The last line of equation (C.1.3) is the PGF of a Poisson stopped sum distribution.

Appendix D

Miscellaneous Appendices

D.1 A summary of simulation times

Table D.1 summarises the typical CPU times[1] for the individual-based models in each chapter using the simple, linear search Direct Method (DM) (Gillespie, 1977). In all cases we neglect signal sensing and simulate simple diffusion as a representative example. The initial conditions are such that all cells are initialised in the left-most interval of the domain. It should be noted, however, that the type of initial conditions used can dramatically affect the speed of the simulations. For example, given the linear nature of the search in the original DM, placing all cells in the first interval dramatically reduces the average search depth in comparison to placing all the cells in the final interval. In addition the behaviour of the system itself can influence the simulation speed. Cell migration simulations where cells sense the exponentially decaying (in space) signalling gradient locally, result in a non-homogeneous steady state, in which cell density is higher to the right-hand end of the domain. Such a simulation will have a larger average search depth than a simulation where the cells sense the same signalling gradient in the non-local manner, which also has a non-homogeneous steady state, but a steady state in which cell density is higher at the left-hand end of the domain. As noted in Chapter 4 the SSA can be made more robust to the ordering of the propensity functions (and hence to the initial conditions and behaviour

[1] 'CPU time' refers to the actual time taken to run a simulation, where as 'simulation time', refers to the internal time of the simulation experienced by the cells.

of the system) by either efficient sorting or searching. Fortunately, even with a simple linear searching method, the initial conditions (all cell in the left-most interval and an exponential decay of cells across the domain) and signal sensing mechanisms (diffusion, local, non-local, and average) employed in this thesis do not cause the CPU times to vary significantly.

In these representative simulations, where we have incorporated domain growth it is always exponential. For the results corresponding to the simulations in Chapter 3 growth is implemented stochastically on an initially uniform domain.

In two dimensions results on the non-growing domain are given for a square domain with square tessellation although CPU times for other domains and other tessellations are comparable as are the number of elements of the tessellation. In each case we study the cells diffusing, ignorant of any external signal. We first attempted to carry out the two dimensional cell migration simulations in `Matlab`, but found it prohibitively slow (simulations, even of simple diffusion, never reached steady state, even after a week of simulation). Instead we coded these simulations in `C++`. Both in one and two dimensions we found decreases in CPU times of over two orders of magnitude by using `C++` in comparison to `Matlab`.

	Chapter 2		Chapter 3		Chapter 5	
Growing	N	Y	N	Y	N	Y
Cells	1000		1000		50000	
Initial Compartments	50	50	50	50	2500	1885
Final Compartments	50	~ 136	50	~ 136	2500	~ 8448
Language	`Matlab`		`Matlab`		`C++`	
Simulation time	3000	1000	3000	1000	3000	1500
CPU Time (Hours)	~ 1	~ 1	~ 1	~ 1	~ 45	~ 200

Table D.1: *A comparison of approximate simulation times for the individual-based cell migration algorithms in each chapter. The one-dimensional simulations (Chapter 2 and Chapter 3) have roughly similar simulation times, whereas the two-dimensional simulation times have vastly increased simulations times, even when using a more efficient language such as* `C++`

The times for 40 repeats of the one dimensional simulations appear reasonable, although clearly they could be dramatically reduced by recoding them in `C++`. We found that repeating the simulations of Chapter 2 in `C++` reduced simulation times to ~ 10 seconds. We note that, although we would have expected the CPU times on the growing domains to

take longer than those on non-growing domain, this increase is balanced by the reduced simulation time which we chose in order that the domains would not grow too long.

The results from two dimensions are indicative of the inherent computational complexity of the simulations in higher dimensions. Even using `C++`, our simulations (especially those on the growing domains) are almost prohibitively long. The consideration of yet higher dimensional simulations motivates the requirement for efficient and accelerated stochastic simulations as discussed in Chapter 4.

The simulations presented in this appendix are broadly representative of the simulations carried out in each chapter. However, these results are intended as a guide rather than an exhaustive list.

Bibliography

M. Alber, N. Chen, P.M. Lushnikov and S.A. Newman. Continuous macroscopic limit of a discrete stochastic model for interaction of living cells. *Phys. Rev. Lett.*, 99(16):168102, 2007.

A.R.A. Anderson and M.A.J. Chaplain. Continuous and discrete mathematical models of tumor-induced angiogenesis. *Bull. Math. Biol.*, 60(5):857–899, 1998.

A.R.A. Anderson and V. Quaranta. Integrative mathematical oncology. *Nat. Rev. Cancer*, 8(3):227–234, 2008.

D.F. Anderson. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *J. Chem. Phys.*, 127(21):214107, 2007.

D.F. Anderson. Incorporating postleap checks in tau-leaping. *J. Chem. Phys.*, 128(5):054103, 2008.

K. Anguige and C. Schmeiser. A one-dimensional model of cell diffusion and aggregation, incorporating volume filling and cell-to-cell adhesion. *J. Math. Biol.*, 58(3):395–427, 2009.

A. Arkin, J. Ross and H.H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, 149(4):1633–1648, 1998.

D. Axelrod, D.E. Koppel, J. Schlessinger, E. Elson and W.W. Webb. Mobility measurement by analysis of fluorescence photobleaching recovery kinetics. *Biophys. J.*, 16(9):1055–1069, 1976.

M.S. Bajaj, R.R. Kew, R.O. Webster and T.M. Hyers. Priming of human neutrophil functions by tumor necrosis factor: enhancement of superoxide anion generation, degran-

ulation, and chemotaxis to chemoattractants C5a and F-Met-Leu-Phe. *Inflammation*, 16(3):241–250, 1992.

R.E. Baker, C.A. Yates and R. Erban. From microscopic to macroscopic descriptions of cell migration on growing domains. *Bull. Math. Biol.*, 72(3):719–762, 2010.

R.C. Ball and R.M. Brady. Large scale lattice effect in diffusion-limited aggregation. *J. Phys. A.-Math. Gen.*, 18(13):L809–L813, 1985.

A.L. Barabási and H.E. Stanley. *Fractal Concepts in Surface Growth*. Cambridge University Press, 1995.

M.T. Batchelor and B.I. Henry. Limits to Eden growth in two and three dimensions. *Phys. Lett. A*, 157(4-5):229–236, 1991.

M.T. Batchelor, B.I. Henry and S.D. Watt. Continuum model for radial interface growth. *Physica A*, 260(1-2):11–19, 1998.

H.C. Berg. *Random Walks in Biology*. Princeton University Press, 1993.

H.C. Berg and E.M. Purcell. Physics of chemoreception. *Biophys. J.*, 20(2):193–219, 1977.

D.F. Blair. How bacteria sense and swim. *Annu. Rev. Microbiol.*, 49(1):489–520, 1995.

M. Block, E. Schöll and D. Drasdo. Classifying the expansion kinetics and critical surface dynamics of growing cell populations. *Phys. Rev. Lett.*, 99(24):248101, 2007.

J.L. Blue, I. Beichl and F. Sullivan. Faster Monte Carlo simulations. *Phys. Rev. E*, 51(2):867–868, 1995.

M. Bodnar and J.J.L. Velazquez. Derivation of macroscopic equations for individual cell-based models: a formal approach. *Math. Method. Appl. Sci.*, 28(15):1757–1779, 2005.

M.P. Brenner, L.S. Levitov and E.O. Budrene. Physical mechanisms for chemotactic pattern formation by bacteria. *Biophys. J.*, 74(4):1677–1693, 1998.

A. Brú, S. Albertos, J. Luis Subiza, J.L. García-Asenjo and I. Brú. The universal dynamics of tumor growth. *Biophys. J.*, 85(5):2948–2961, 2003.

A.Q. Cai, K.A. Landman and B.D. Hughes. Modelling directional guidance and motility regulation in cell migration. *Bull. Math. Biol.*, 68(1):25–52, 2006.

T. Callaghan, E. Khain, L.M. Sander and R.M. Ziff. A stochastic model for wound healing. *J. Stat. Phys.*, 122(5):909–924, 2006.

Y. Cao, D.T. Gillespie and L.R. Petzold. Avoiding negative populations in explicit Poisson tau-leaping. *J. Chem. Phys.*, 123(5):054104, 2005.

Y. Cao, D.T. Gillespie and L.R. Petzold. Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.*, 124(4):044109, 2006.

Y. Cao, H. Li and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121(9):4059–4067, 2004.

Y. Cao and L. Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *J. Comput. Phys.*, 212(1):6–24, 2005.

T.A. Carrier and J.D. Keasling. Investigating autocatalytic gene expression systems through mechanistic modeling. *J. Theor. Biol.*, 201(1):25–36, 1999.

M.A.J. Chaplain. Mathematical modelling of angiogenesis. *J. Neuro.-Oncol.*, 50(1):37–51, 2000.

A. Chatterjee, D.G. Vlachos and M.A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *J. Chem. Phys.*, 122(2):024112, 2004.

A. Chevallier, M. Kieny and A. Mauger. Limb-somite relationship: origin of the limb musculature. *Development*, 41(1):245–258, 1977.

H.S.M. Coxeter. *Regular Polytopes*. Dover Publications, 1973.

E.J. Crampin. *Reaction-diffusion patterns on growing domains*. D.Phil. thesis, University of Oxford, 2000.

E.J. Crampin, E.A. Gaffney and P.K. Maini. Reaction and diffusion on growing domains: scenarios for robust pattern formation. *Bull. Math. Biol.*, 61(6):1093–1120, 1999.

M. Deng, W.L. Chen, A. Takatori, Z. Peng, L. Zhang, M. Mongan, R. Parthasarathy, M. Sartor, M. Miller, J. Yang, B. Su, W.W.-Y. Kao and Y. Xia. A role for the mitogen-activated protein kinase kinase kinase 1 in epithelial wound healing. *Mol. Biol. Cell*, 17(8):3446–3455, 2006.

C. Deroulers, M. Aubert, M. Badoual and B. Grammaticos. Modeling tumor cell migration: From microscopic to macroscopic models. *Phys. Rev. E*, 79(3):031917, 2009.

P. Devillard and H.E. Stanley. Scaling properties of Eden clusters in three and four dimensions. *Physica A*, 160(3):298–309, 1989.

D. Drasdo. Coarse graining in simulated cell populations. *Adv. Complex. Syst.*, 8(3):319, 2005.

D. Drasdo and S. Höhme. Individual-based approaches to birth and death in avascular tumors. *Math. Comput. Model.*, 37(11):1163–1175, 2003.

B. Drawert, M.J. Lawson, L. Petzold and M. Khammash. The diffusive finite state projection algorithm for efficient simulation of the stochastic reaction-diffusion master equation. *J. Chem. Phys.*, 132(7):074101, 2010.

M. Eden. *A Probabilistic Model for Morphogenesis*, chapter 4: Information Networks, pages 359–370. Pergamon Press, New York, 1958.

M. Eden. A two-dimensional growth process. In *Proceedings of the Fourth Berkeley Symposium on Mathematics Statistics and Probability*, volume 4, pages 223–239. Universilty of California Press, 1961.

J.C.W. Edwards, G. Cambridge and V.M. Abrahams. Do self-perpetuating B lymphocytes drive human autoimmune disease? *Immunology*, 97(2):188–196, 1999.

S.F. Edwards and D.R. Wilkinson. The surface statistics of a granular aggregate. *P. Roy. Soc. A-Math. Phy.*, 381(1780):17–31, 1982.

J. Elf and M. Ehrenberg. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst. Biol.*, 1(2):230–236, 2004.

S. Engblom, L. Ferm, A. Hellander and P. Lötstedt. Simulation of stochastic reaction-diffusion processes on unstructured meshes. *SIAM J. Sci. Comput.*, 31(3):1774–1797, 2009.

R. Erban and S.J. Chapman. Reactive boundary conditions for stochastic simulations of reaction–diffusion processes. *Phys. Biol.*, 4(1):16–28, 2007.

R. Erban and S.J. Chapman. Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions. *Phys. Biol.*, 6(4):1–18, 2009.

R. Erban and H.G. Othmer. From individual to collective behavior in bacterial chemotaxis. *SIAM J. Appl. Math.*, 65(2):361–391, 2005.

R. Erban and H.G. Othmer. Taxis equations for amoeboid cells. *J. Math. Biol.*, 54(6):847–885, 2007.

F. Family and T. Vicsek. Scaling of the active zone in the Eden process on percolation networks and the ballistic deposition model. *J. Phys. A.-Math. Gen.*, 18(2):L75–L81, 1985.

S.C. Ferreira Jr. and S.G. Alves. Pitfalls in the determination of the universality class of radial clusters. *J. Stat. Mech.-Theory. E.*, 2006(11):P11007, 2006.

B. Flaherty, J.P. McGarry and P.E. McHugh. Mathematical models of cell motility. *Cell Biochem. Biophys.*, 49(1):14–28, 2007.

J. Folkman and M. Hochberg. Self-regulation of growth in three dimensions. *J. Exp. Med.*, 138(4):745–753, 1973.

P. Freche, D. Stauffer and H.E. Stanley. Surface structure and anisotropy of Eden clusters. *J. Phys. A.-Math. Gen.*, 18:L1163–L1168, 1985.

U. Frisch, B. Hasslacher and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56(14):1505–1508, 1986.

M.A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A.*, 104(9):1876–1889, 2000.

D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22(4):403–434, 1976.

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

D.T. Gillespie. The chemical Langevin equation. *J. Chem. Phys.*, 113(1):297–306, 2000.

D.T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115(4):1716–1733, 2001.

D.T. Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58(1):35–55, 2007.

D.T. Gillespie and L.R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.*, 119(16):8229–8234, 2003.

A. Goldbeter, D. Gonze and O. Pourquié. Sharp developmental thresholds defined through bistability by antagonistic gradients of retinoic acid and FGF signaling. *Dev. Dyn.*, 236(6):1495–1508, 2007.

F. Graner and J.A. Glazier. Simulation of biological cell sorting using a two-dimensional extended Potts model. *Phys. Rev. Lett.*, 69(13):2013–2016, 1992.

R. Grima and T.J. Newman. Accurate discretization of advection-diffusion equations. *Phys. Rev. E*, 70(3):36703, 2004.

J. Hall. The study of circulating lymphocytes in vivo: a personal view of artifice and artifact. *Immunol. Today*, 6(5):149–152, 1985.

R.A. Hanahan, D. Weinberg. The hallmarks of cancer. *Cell*, 100(1):57–70, 2000.

L.A. Harris and P. Clancy. A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics. *J. Chem. Phys.*, 125(14):144107, 2006.

T. Hillen and K.J. Painter. Global Existence for a Parabolic Chemotaxis Model with Prevention of Overcrowding. *Adv. Appl. Math*, 26(4):280–301, 2001.

T. Hillen and K.J. Painter. A user's guide to PDE models for chemotaxis. *J. Math. Biol.*, 58(1):183–217, 2009.

P.F. Ho and C.Y. Wang. Cluster growth by mitosis. *Math. Biosci.*, 155(2):139–146, 1999.

N.L. Johnson, S. Kotz and A.W. Kemp. *Univariate Discrete Distributions*. Wiley, New York., 2005.

R. Jullien and R. Botet. Scaling properties of the surface of the Eden model in d= 2, 3, 4. *J. Phys. A.-Math. Gen.*, 18(12):2279–2287, 1985a.

R. Jullien and R. Botet. Surface thickness in the Eden model. *Phys. Rev. Lett.*, 54(18):2055–2055, 1985b.

R. Jullien and R. Botet. *Aggregation and Fractal Aggregates*. World Scientific, Singapore, 1987.

R. Jullien and P. Meakin. Simple three-dimensional models for ballistic deposition with restructuring. *Europhys. Lett.*, 4(12):1385–1390, 1987.

M. Kardar, G. Parisi and Y.C. Zhang. Dynamic scaling of growing interfaces. *Phys. Rev. Lett.*, 56(9):889–892, 1986.

E.F. Keller and L.A. Segel. Initiation of slime mold aggregation viewed as an instability. *J. Theor. Biol.*, 26(3):399–415, 1970.

E.F. Keller and L.A. Segel. Model for chemotaxis. *J. Theor. Biol.*, 30(2):225–234, 1971.

R. Keller. Cell migration during gastrulation. *Curr. Opin. Cell Biol.*, 17(5):533–541, 2005.

J. Kertész and D.E. Wolf. Noise reduction in Eden models: II. Surface structure and intrinsic width. *J. Phys. A.-Math. Gen.*, 21(3):747–761, 1988.

R.J. Keynes and G.M.W. Cook. Repellent cues in axon guidance. *Curr. Opin. Neurobiol.*, 2(1):55–59, 1992.

A. Kicheva, P. Pantazis, T. Bollenbach, Y. Kalaidzidis, T. Bittig, F. Jülicher and M. González-Gaitán. Kinetics of morphogen gradient formation. *Science*, 315(5811):521–525, 2007.

E. Kieri. *Accuracy aspects of the reaction-diffusion master equation on unstructured meshes*. Master's thesis, University of Uppsala, 2011.

A.M. Kierzek. STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm. *Method. Biochem. Anal.*, 18(3):470–481, 2002.

A.M. Kierzek, J. Zaim and P. Zielenkiewicz. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *J. Biol. Chem.*, 276(11):8165–8172, 2001.

S. Kondo and R. Asai. A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*. *Nature*, 376(6543):765–768, 1995.

J.E. Krebs, E.S. Goldstein and S.T. Kilpatrick. *Lewin's Genes X*. Jones & Bartlett Pub, 2010.

T.G. Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *J. Chem. Phys.*, 57(7):2976–2978, 1972.

S. Lampoudi, D.T. Gillespie and L.R. Petzold. The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *J. Chem. Phys.*, 130(9):094104, 2009.

K.A. Landman, G.J. Pettet and D.F. Newgreen. Mathematical models of cell colonization of uniformly growing domains. *Bull. Math. Biol.*, 65(2):235, 2003.

D.A. Lauffenburger and C.R. Kennedy. Localized bacterial infection in a distributed model for tissue inflammation. *J. Math. Biol.*, 16(2):141–163, 1983.

P. Lee and C.W. Wolgemuth. Crawling cells can close wounds without purse strings or signaling. *Plos. Comput. Biol.*, 7(3):e1002007, 2011.

A. Leshem, A. Ar and R.A. Ackerman. Growth, water, and energy metabolism of the soft-shelled turtle (*Trionyx triunguis*) embryo: effects of temperature. *Physiol. Zool.*, 64(2):568–594, 1991.

H. Levine, D.A. Kessler and W.-J. Rappel. Directional sensing in eukaryotic chemotaxis: A balanced inactivation model. *Proc. Natl. Acad. Sci. USA.*, 103(26):9761–9766, 2006.

H. Li and L. Petzold. Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. Technical report, University of California Santa Barbara, 2006.

M.A. Lieberman and L. Glaser. Density-dependent regulation of cell growth: an example of a cell-cell recognition phenomenon. *J. Membrane Biol.*, 63(1):1–11, 1981.

C.C. Lin and L.A. Segel. *Mathematics Applied to Deterministic Problems in the Natural Sciences*, volume 1. Macmillan, New York., 1988.

F. Lin, C.M.-C. Nguyen, S.-J. Wang, W. Saadi, S.P. Gross and N.L. Jeon. Effective neutrophil chemotaxis is strongly influenced by mean IL-8 concentration. *Biochem. Bioph. Res. Co.*, 319(2):576–581, 2004.

F. Lin, C.M.-C. Nguyen, S.-J. Wang, W. Saadi, S.P. Gross and N.L. Jeon. Neutrophil migration in opposing chemoattractant gradients using microfluidic chemotaxis devices. *Ann. Biomed. Eng.*, 33(4):475–482, 2005.

J.A. Madri and D. Graesser. Cell migration in the immune system: the evolving inter-related roles of adhesion molecules and proteinases. *Dev. Immunol.*, 7(2-4):103–116, 2000.

P.A. Maksym. Fast Monte Carlo simulation of MBE growth. *Semicond. Sci. Tech.*, 3(6):594–596, 1988.

F.A. Martín, A. Pérez-Garijo, E. Moreno and G. Morata. The *brinker* gradient controls wing growth in *Drosophila*. *Development*, 131(20):4921–4930, 2004.

J.M. McCollum, G.D. Peterson, C.D. Cox, M.L. Simpson and N.F. Samatova. The sorting

direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Biol. Chem.*, 30(1):39–49, 2006.

P. Meakin. Universality, nonuniversality, and the effects of anisotropy on diffusion-limited aggregation. *Phys. Rev. A*, 33(5):3371–3382, 1986.

P. Meakin and T.A. Witten Jr. Growing interface in diffusion-limited aggregation. *Phys. Rev. A*, 28(5):2985–2989, 1983.

H. Meinhardt. *The algorithmic beauty of sea shells*. Springer-Verlag, New York, 2009.

F.J.R. Meysman, B.P. Boudreau and J.J. Middelburg. Relations between local, nonlocal, discrete and continuous models of bioturbation. *J. Mar. Res.*, 61(3):391–410, 2003.

P.J. Murray, C.M. Edwards, M.J. Tindall and P.K. Maini. From a discrete to a continuum model of cell dynamics in one dimension. *Phys. Rev. E*, 80(3):031912, 2009.

J.M. Osborne, A. Walter, S.K. Kershaw, G.R. Mirams, A.G. Fletcher, P. Pathmanathan, D. Gavaghan, O.E. Jensen, P.K. Maini and H.M. Byrne. A hybrid approach to multi-scale modelling of cancer. *Philos. T. Roy. Soc. A.*, 368(1930):5013–5028, 2010.

H.G. Othmer, S.R. Dunbar and W. Alt. Models of dispersal in biological systems. *J. Math. Biol.*, 26(3):263–298, 1988.

H.G. Othmer and A. Stevens. Aggregation, blowup, and collapse: the ABC's of taxis in reinforced random walks. *SIAM J. Appl. Math.*, 57(4):1044–1081, 1997.

K.J. Painter and T. Hillen. Volume-filling and quorum-sensing in models for chemosensitive movement. *Canad. Appl. Math. Quart.*, 10(4):280–301, 2002.

K.J. Painter, P.K. Maini and H.G. Othmer. Stripe formation in juvenile *Pomacanthus* explained by a generalized Turing mechanism with chemotaxis. *Proc. Natl. Acad. Sci. USA.*, 96(10):5549–5554, 1999.

L.R. Paiva and S.C. Ferreira Jr. Universality class of isotropic on-lattice Eden clusters. *J. Phys. A.-Math. Theor.*, 40(1):F43–F49, 2007.

S. Paku and N. Paweletz. First steps of tumor-related angiogenesis. *Lab. Invest.*, 65(3):334–346, 1991.

C.A. Parent and P.N. Devreotes. A cell's sense of direction. *Science*, 284(5415):765–770, 1999.

N. Paweletz and M. Knierim. Tumor-related angiogenesis. *Crit. Rev. Oncol. Hemat.*, 9(3):197–242, 1989.

T.N. Pettit, G.S. Grant, G.C. Whittow, H. Rahn and C.V. Paganelli. Embryonic oxygen consumption and growth of Laysan and black-footed albatross. *Am. J. Physiol.-Reg. I.*, 242(1):121–128, 1982.

J. Pitt-Francis, P. Pathmanathan, M.O. Bernabeu, R. Bordas, J. Cooper, A.G. Fletcher, G.R. Mirams, P. Murray, J.M. Osborne, A. Walter et al. Chaste: a test-driven approach to software development for biological modelling. *Comput. Phys. Commun.*, 180(12):2452–2471, 2009.

E.W. Raines. The extracellular matrix can regulate vascular cell migration, proliferation, and survival: relationships to vascular disease. *Int. J. Exp. Pathol.*, 81(3):173–182, 2000.

S. Redner. *A Guide to First-Passage Processes*. Cambridge University Press, 2001.

A.J. Ridley, M.A. Schwartz, K. Burridge, R.A. Firtel, M.H. Ginsberg, G. Borisy, J.T. Parsons and A.R. Horwitz. Cell migration: integrating signals from front to back. *Science*, 302(5651):1704–1709, 2003.

D. Rogulja and K.D. Irvine. Regulation of cell proliferation by a morphogen gradient. *Cell*, 123(3):449–461, 2005.

M.A. Shea and G.K. Ackers. The O_R control system of bacteriophage lambda. A physical-chemical model for gene regulation. *J. Mol. Biol.*, 181(2):211–230, 1985.

M.J. Simpson, K.A. Landman and B.D. Hughes. Distinguishing between directed and undirected cell motility within an invading cell population. *Bull. Math. Biol.*, 71(4):781–799, 2009a.

M.J. Simpson, K.A. Landman and B.D. Hughes. Multi-species simple exclusion processes. *Physica A*, 388(4):399–406, 2009b.

M.J. Simpson, K.A. Landman and B.D. Hughes. Cell invasion with proliferation mechanisms motivated by time-lapse data. *Physica A*, 389(18):3779–3790, 2010.

A.M. Smith, R.E. Baker, D. Kay and P.K. Maini. Incorporating chemical signalling factors into cell-based models of growing epithelial tissues. *J. Math. Biol.*, 2011.

F. Spitzer. Interaction of Markov processes. *Adv. Math.*, 5(2):246–290, 1970.

C.A. Staton, S.M. Stribbling, S. Tazzyman, R. Hughes, N.J. Brown and C.E. Lewis. Current methods for assaying angiogenesis *in vitro* and *in vivo*. *Int. J. Exp. Pathol.*, 85(5):233–248, 2004.

C.L. Stokes and D.A. Lauffenburger. Analysis of the roles of microvessel endothelial cell random motility and chemotaxis in angiogenesis. *J. Theor. Biol.*, 152(3):377–403, 1991.

W.A. Strauss. *Partial Differential Equations: An Introduction*. Wiley, New York., 1992.

D.J.T. Sumpter and M. Beekman. From nonlinearity to optimality: pheromone trail foraging by ants. *Anim. Behav.*, 66(2):273–280, 2003.

K.F. Swaney, C.H. Huang and P.N. Devreotes. Eukaryotic chemotaxis: a network of signaling pathways controls motility, directional sensing, and polarity. *Annu. Rev. Biophys.*, 39:265–289, 2010.

P.P.L. Tam. The control of somitogenesis in mouse embryos. *Development*, 65:103–128, 1981.

The Numerical Algorithms Group. *NAG Toolbox for Matlab*, 2009.

T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121(21):10356–10364, 2004.

S. Tolman and P. Meakin. Off-lattice and hypercubic-lattice models for diffusion-limited aggregation in dimensionalities 2–8. *Phys. Rev. A*, 40(1):428–437, 1989.

C.J. Tomlin and J.D. Axelrod. Biology by numbers: mathematical modelling in developmental biology. *Nat. Rev. Genet.*, 8(5):331–340, 2007.

R.T. Tranquillo, D.A. Lauffenburger and S.H. Zigmond. A stochastic model for leukocyte random motility and chemotaxis based on receptor binding fluctuations. *J. Cell. Biol.*, 106(2):303–309, 1988.

S. Turner and J.A. Sherratt. Intercellular adhesion and cancer invasion: a discrete simulation using the extended Potts model. *J. Theor. Biol.*, 216(1):85–100, 2002.

S. Turner, J.A. Sherratt, K.J. Painter and N.J. Savill. From a discrete to a continuous model of biological cell movement. *Phys. Rev. E*, 69(2):021910, 2004.

I.M.M. Van Leeuwen, G.R. Mirams, A. Walter, A. Fletcher, P. Murray, J. Osbourne, S. Varma, S.J. Young, J. Cooper, B. Doyle, J. Pitt-Francis, L. Momtahan, P. Pathmanathan, J. P. Whiteley, S. J. Chapman, D. J. Gavaghan, O. E. Jensen, J. R. King, P. K. Maini, S. L. Waters and H. M. Byrne. An integrative computational model for intestinal tissue renewal. *Cell Proliferat.*, 42(5):617–636, 2009.

T. Vicsek. *Fractal Growth Phenomena.* World Scientific, Singapore, 1992.

G. Wagner, R. Halvorsrud and P. Meakin. Extended Eden model reproduces growth of an acellular slime mold. *Phys. Rev. E*, 60(5):5879–5887, 1999.

C.Y. Wang, P.L. Liu and J.B. Bassingthwaighte. Off-lattice Eden-C cluster growth model. *J. Phys. A.-Math. Gen.*, 28(8):2141–2147, 1995.

M. Ward, C. McCann, M. DeWulf, J.Y. Wu and Y. Rao. Distinguishing between directional guidance and motility regulation in neuronal migration. *J. Neurosci.*, 23(12):5170–5177, 2003.

D. Wessels, J. Murray and D.R. Soll. Behavior of *Dictyostelium* amoebae is regulated primarily by the temporal dynamic of the natural cAMP wave. *Cell Motil. Cytoskel.*, 23(2):145–156, 1992.

J. Westermann, S. Söllner, E.M. Ehlers, K. Nohroudi, M. Blessenohl and K. Kalies. Analyzing the migration of labeled T cells in vivo: an essential approach with challenging features. *Lab. Invest.*, 83(4):459–469, 2003.

S. Wolfram. Cellular automaton fluids 1: Basic theory. *J. Stat. Phys.*, 45(3):471–526, 1986.

L. Wolpert, R. Beddington, T. Jessell, P. Lawrence, E. Meyerowitz and J. Smith. *Principles of Development*. Oxford University Press, New York, 3rd edition, 2006.

T.E. Woolley, R.E. Baker, E.A. Gaffney and P.K. Maini. Power spectra methods for a stochastic description of diffusion on deterministically growing domains. *Phys. Rev. E*, 84(2):021915, 2011.

C.A. Yates, R.E. Baker, R. Erban and P.K. Maini. Going from microscopic to macroscopic on non-uniform growing domains. *Physica A*, Under Review, 2011.

C.A. Yates and K. Burrage. Look before you leap: A confidence-based method for selecting species criticality while avoiding negative populations in τ-leaping. *J. Chem. Phys.*, 134(8):084109, 2011.

J.G. Zabolitzky and D. Stauffer. Simulation of large Eden clusters. *Phys. Rev. A*, 34(2):1523–1530, 1986.