Mathematical methods for the modelling of cell migration and chemotaxis

Michael Nolan

Submitted for
Degree of Doctor of Philosophy,
Department of Mathematics and Statistics,
University of Strathclyde, Glasgow

March 30, 2018

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

Numerical methods are developed for the study of cell migration and chemotaxis with specific focus on reaction-diffusion based models on moving domains. This thesis focuses building on the previous model of cell migration and chemotaxis introduced by Neilson et al [105]. It does this by introducing new and discussing existing numerical methods in a cell migration and chemotaxis context.

A new approach to the solution of forced mean curvature flow is introduced in the form of a moving mesh partial differential equation (MMPDE). This MMPDE is split into two partial differential algebraic equations: one for the normal velocity and one for the tangential velocity. We examine curves moving with a prescribed normal velocity and this decoupling of the velocity components allows mesh adaption to be considered along the tangential component by means of a mesh adaption monitor function. This new method is not restricted to cell migration models and could be used in other contexts where mesh adaption of an evolving curve domain would be desired.

A second-order conservative ALE-FEM scheme is also introduced for the solutions of reaction-diffusion equations on an evolving curve boundary and is subsequently used to derive the concentrations of various chemicals which lay on the model of the cell's membrane. The cell migration model is also extended to higher dimensions using a second ALE-FEM scheme which couples the solutions of reactions taking place on different domains together by means of a flux boundary condition and a two-dimensional mesh movement strategy which complements the new one-dimensional MMPDE approach for curves.

This scheme is used to model diffusion of ligand molecules in the region exterior to the moving cell. Simulations are presented of this new two-dimensional model in-

dicating the effect of cell movement and receptor-ligand binding dynamics on the cell micro-environment and research is currently underway to use this model to investigate biological theories [84, 86].

Acknowledgements

Firstly, I would like to take the time here to thank my advisors Dr John Mackenzie and Prof Robert Insall for their continued support, patience and motivation. For the many times spent explaining concepts, methods and biology to me in great detail. It's been a pleasure getting to know and learn from both of you. Thank you both.

I would like to say thanks also to Dr Matt Neilson who has often made time to answer my numerous questions in some very lengthy detail. I appreciated every word.

Thanks must also be said to the University of Strathclyde and my colleagues at the Department of Mathematics and Statistics as well as my colleagues and friends from the Beatson Insitute, for giving me so many laughs for so many years and all the fun times. With special thanks to Kirsty and Ben, for getting me out of the house regularly and keeping my spirits up.

I would also like to thank CRUK and EPSRC for providing a large part of the financial support for this undertaking.

And last, but by no means least, my parents Brian and Elaine, whom without their continued moral and financial support this thesis would not have come to a closing. Thank you for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cell migration and chemotaxis

1.1.1 Motivation

The study of cell migration and chemotaxis is a vast and active area of biology. Chemotaxis is a sub-branch of cell migration where cells move along or away from a gradient of a chemical concentration in the cell's detectable environment. There are many situations where cells' mechanism to chemotax prove vitally important. In wound healing, white blood cells travel towards damaged tissue by migrating up a gradient of platelet-derived growth factor which is released around the wound [4], [32]. This allows the white blood cells to be poised in a vulnerable area to fight incoming foreign bodies, infection and diseases as they arrive through the damaged tissue. Spermatozoa also migrate using a gradient emitted from the oocyte which provides the cells arriving in the womb with a strong directional path to follow [45].

Cancerous cells also spread the disease by means of chemotaxis [122]. When cancerous cells metastasise, they migrate away from the original tumour and travel to different parts of the body. Melanoma cells are seen to do this in a chamber-based experiment [100], for instance, where the cells migrate up gradient of lysophosphatidic acid which is broken down by the cells as they move off, thus forming a gradient away from the initial tumour which drives the cancerous cells dispersal.

There is thus considerable interest in understanding the mechanisms responsible for

directed cell migration but the exact processes involved are not fully understood and different types of cells also exhibit different forms of locomotion.

1.1.2 Existing biological models

Zebrafish primordial germ cells, for example, migrate using bleb-like protrusions of cytoplasmic flow which extend outward from the cell [21]. These cells migrate by positioning myosin, a protein which has strong ties to cell membrane [150], close to the perceived lowest concentration of background signal thereby loosely defining the rear of the cell. A front is then formed by the combined effects of hydrostatic pressure within the cell, tension and elevated calcium along the leading edge.

Eukaryotic cells like Dictyostelium discoideum, on the other hand, generally move in a manner akin to crawling. They do this by forming pseudopods, a network of filamentous or F-actin formed from polymerisation of monomers of G-actin (globular actin), close to the direction in which the cell wishes to travel [133, 135, 143, 146]. These protrusions form a strong attachment to the substrate the cell is travelling on. The cell is then presumed to make progress by releasing some of the earlier attachments at the trailing edge. This, coupled with hydrostatic pressure [18] and tension forces which maintain the cell's constant volume results in the cell gradually progressing forward.

This process of pseudopod formation and movement occurs naturally in Dictyostelium cells even when there is no obvious food source for the cells to migrate towards. When a food source, such as cyclic-AMP is present in the cells' immediate surrounding however, Dictyostelium can sense which direction the higher concentration is present by means of a number of small receptors which are embedded in the membrane. It is thought that when these receptors bind to free-ligand molecules they bias the position of pseudopod growth such that the pseudopods gradually move closer to the area of highest concentration.

There are many constraints placed on biologists' ability to study cell migration directly. For instance, the inability to observe exactly how all proteins interact with each other and consequently the cell to produce movement leads to assumptions being made on which proteins are most relevant in determining how the cell responds to

chemotactic gradients and other stimuli. Mathematical modelling of cell migration has become a popular tool for observing cell migration as models potentially offer a facility to fill some of these observational gaps with insight which, in many cases, can then be experimentally validated.

1.1.3 Existing mathematical models

In perhaps the first published conceptual model of cell migration, in 1980 Abercombie put forth a model set of circumstances for metazoan cell movement based on intuitive assumptions gained by experimentation and observations [1]. This conceptual model broke the metazon cell migration process into a four component cycle consisting of acting growth protrusion, frontal adhesion formation, rear adhesion release and unilateral contraction of the cytoplasm by means of actin-myosin proteins. This relatively high level model had all the necessary components to generate the net effect of cell movement but the fundamental assumptions could not be verified at the time. It was not until years later when the technology became available that the assumptions made by Abercombie could actually be verified by improved microscopy techniques and empirical data.

More recently, with the increase in computational power, numerical methods have been incorporated into the modelling process as a modern tool for visualisation and deriving quantitative results which can be compared with experimental data. DiMilla et al. [36], for instance, put forth a mathematical description of what the ideal strength characteristics for the cells adhesions should be for optimal cell motility. In doing so they made a number of assumptions about the concentration of integrin receptors, which facilitate the linking between the cell cytoskeleton and the substrate on which it travels. The assumptions were in fact verified years later by [55] and [119] when the technology to do so became available and this helped to strengthen the use of mathematical models in the study of cell motility.

It can often be easier to infer cell movement as a collective and conduct experiments and studies accordingly. As such, many models exist which encapsulate cell migration as a group of cell interactions rather than a single cell model. Models such as [99] and

[138] demonstrate how cells migrating at the front of a steep slope of chemoattractant degrade the field as they move and cause the trailing cells to stagnate while at the same time the leading cells deposit exopodia which act as beacons that the trailing cells may follow. Similar scale approaches have been used to model complex cell proliferation in wound healing sites [70, 110]. Such agent or "rule-based" models are incredibly useful to derive collective cell migration behaviours and often times postulates made with such models are easily verified experimentally. The drawback, however, of this scale of model is that often little can be said about the specific proteins and concentrations within the cells themselves. To gain a more accurate model of cell migration, each cell has to in turn be accurately modelled.

In recent years there have been a number of different approaches to modelling chemotaxis at the single cell level. For instance, [67], [74] and [112] favour a "compass like" approach whereby the cell picks up on the external chemoattractant source and migrates immediately up the steepest path using a biased excitable network. This model treats the activation of the actin protein as an excitable media using Fitzhugh-Nagumo dynamics [53] to model the patterns of spot to wave actin activation which is observed in Dictyostelium cells [147]. By itself, this model has trouble adapting to significant changes in background concentrations but when combined with a local excited, globally inhibited mechanism for the cell receptor activation [151], this model does a good job of predicting cell motility and chemotaxis. In simulated backgrounds of cyclic-AMP, the cells are able to correctly migrate in gradient differences of as little as 1%.

Models such as these are intuitive and would be in line with much of what we would expect any organism to do when presented with more resources in a particular direction. However, there are notable issues handling zero background where we know cells do in fact still migrate, albeit in a somewhat random fashion. There is also a drawback from an experimental observational viewpoint where cells in fact can relocate slowly up the gradient rather than rapidly when responding to a significant gradient change. This type of behaviour is harder to encapsulate within compass-like models but can naturally develop as part of models which use receptors to bias the movement of the

cell rather than to define a front.

At the subcellular level, there have been attempts to model the interactions of specific proteins. In [57, 115], a "ratchet" model describes the motion of the filaments of actin within the cell and how various mechanical forces which occur naturally through energy minimisations within the cell lead to the translocation of actin and the deformation of the cell membrane. Experimentally testable results from this approach have provided a deeper understanding of exactly how all the mechanisms involved in cell migration may interact within the cell and the ways in which various proteins naturally form complex mechanical structures.

This model does a good job of representing the interactions between the actin cytoskeleton and the cell substrate however a model which also takes account of the other proteins involved in cell migration and links the them all together would be more advantageous. Unfortunately a full biological account of exactly what proteins and cell systems are involved in the cell migration process is unknown.

This means that while the protein actin is widely considered a driver protein for cell motility, see for example [69], it is known that other proteins can also affect the cell migration process in different ways. The protein myosin, for example, can also contribute to the overall movement of cells [75]. Thus, although there is high demand for models which are explicit in their description, there is also not enough biological knowledge to develop models with such fidelity as yet. Models which remain slightly more general can be useful to develop assumptions about the various proteins involved in the migration process. These are excellent tools for developing understanding and testing assumptions until more is understood about the biology of cell migration first. For example in [105], the independent proteins aren't strongly identified but the model produces realistic simulations of cell migration. These models could eventually be tied down to various proteins as the technology and knowledge becomes available and in the meantime offer useful insight by speculating on what various concentrations could represent. In some cases, speculations made on models such as these could even be tested to find new results.

More recent models of cell migration incorporate growing domains of different di-

mensionality with the exchange of information between the different domains being mediated through flux boundary conditions building on some of the work conducted by Elliott et al. [46] and Madzvamuse et al. [87, 88, 89, 90, 91, 92, 93]. In [48] and [84] for instance, the receptor occupancy of the cell is modelled using a shared flux boundary term between the cell local environment and the cell membrane. This is used to produces a more physiologically realistic model of the binding and unbinding of free-ligand which takes place to and from receptors on the cell membrane and the cell environment.

In such models, the cell interface must also be tracked using specialist numerical methods. While there exist many cell models which fix the position of the cell membrane [78, 95, 107], a more realistic approach is to allow the position of the membrane to change and evolve over time. A number of techniques exist which facilite such interface tracking of the free boundary model, either by explicitly tracking the evolving interface front or by tracking the front implicitly.

Phase field methods [98, 130] for instance, track the interface implicitly as the level-set of an evolving phase field function, ϕ say, embedded in $\mathbb{R}^2$ or $\mathbb{R}^3$. At the cell boundary there is a phase transition between the cell interior, where $\phi = -1$ for instance, and the micro-environment surrounding the cell, where $\phi = 1$. A smooth transition occurs between these two constants using the solution of a phase field equation with a width parameter, W say, to describe how fast the transition takes place. The membrane is identified with the $\phi = 0$ contour. This technique has been used in [129] to model the effects of intracellular actin-flow, cell adhesions and morphology on cell motility and in [19] and [39] to model vesicle membranes within the cell. However, to predict accurately the membrane position, the parameter W has to be small and for efficiency the computational grid must be refined close to the interface and this requires much additional computational expense.

Level set methods (LSM) [109, 127] work similarly to phase field methods in that they both require calculation of a larger area in which the interface is embedded and both evolve the curve implicitly using a velocity equation to describe the motion of the interface. On the other hand, LSM use a different set of equations to do this from

phase field methods, using instead convection-diffusion to describe the motion of the moving interface as a propagating wavefront in $\mathbb{R}^2$ or $\mathbb{R}^3$ space. This makes them a more intuitive method for handling sharp front problems and discontinuous domains although more care must be taken to solve these equations as the convection-diffusion problem is less stable than the reaction-diffusion problem. LSM have been used in the past to model cell migration; in [105] and [152] for instance, they are used to evolve the boundary and model the shape changes which happen at the cell membrane.

Point clouds [79, 83] have also been used to calculate solutions of reaction-diffusion models with some applications in modelling cell biology [81]. The point cloud solutions to reaction-diffusion systems can also be used to develop implicit interface tracking methods [52, 77, 82] which have certain computational advantages over traditional level set methods and phase field methods because the particles in the point clouds can be rearranged to more optimally capture the solution of the implicit level set function close to the interface boundary and thus less grid nodes or particles are required to properly track the interface front.

In each of the implicit methods above, a mesh triangulation algorithm must be used to resolve the interface and find the level-set. Usually this will be some form of marching cubes [80] but this additional step is not necessary with explicit methods. In explicit front tracking methods, the triangulation for the domain itself is evolved. A notable disadvantage is the additional complexity required to capture topological changes. However, explicit methods tend to have a computational advantage as only the mesh itself need be used and not any addition grids in $\mathbb{R}^2$ or $\mathbb{R}^3$ making them potentially more efficient to calculate.

There exist many explicit methods which move a numerical mesh forward. The MATLAB toolbox DistMesh [113, 114] for instance can evolve a numerical mesh by discretely recomputing new triangulations for the grid as the boundary changes over time. A more efficient way of evolving an explicit mesh however makes use of properties of certain variational problems. In [65], many different variational formulations are looked at for mesh movement. These problems typically take a similar form with differing monitor matrix to produce different effects within the mesh movement, usually

with the aim to enhance the quality of the mesh in specific areas where that may be advantageous. The variational problems are solvable using Galerkin finite element methods which allows the choice of initial grid to be non-regular.

Recently, numerical methods have been developed to evolve meshes on surfaces. In [41], [44], [47] and [49] for instance, a variational moving mesh finite element method is developed which is able to deform and move surface meshes according to a prescribed mesh normal velocity while keeping an overall good quality triangulation through an arbitrary tangential velocity. Some work has already been done to incorporate this technique into cell biology [9] with applications which could potentially include accurately modelling the cell membrane during chemotaxis in future.

1.1.4 New approaches

In this thesis we will look initially at a top-down approach of modelling cell migration and chemotaxis processes based on earlier work conducted by Neilson et al [105]. They have proposed a mathematical model which naturally encapsulates much of the behaviour and characteristics observed by Dictyostelium cells which we wish to emulate. We will then look at extending their model by utilising different numerical methods and approaches, some of which are newly developed as part of this thesis.

Their approach is based on solving a reaction-diffusion equation on a moving boundary domain which represents the cell membrane. They do this using an arbitrary Lagrangian-Eulerian finite element method (ALEFEM) with a semi-implicit IMEX scheme for the temporal integration. In this way, the diffusive and convective term are treated implicitly using a Backward-Euler approach whereas the non-linear reaction terms are treated explicitly using the forward Euler method. They do this because the reaction terms are non-stiff and so do not require an excessively small time-step to maintain stability. In this way an IMEX scheme in [105] provides good accuracy with minimal computation through large time-steps. Such schemes are often used to solve reaction-diffusion equations or ODEs which have a stiff and non-stiff component for the increased computational efficiency (in [5, 16, 28, 54] or [59], for example).

In this thesis we will develop improvements on this scheme by modifying it with a

Crank-Nicolson approach to obtain second order temporal accuracy for use with the cell model later on. In addition to this, Neilson et al have chosen to use a level set method approach in [105]. We will also discuss improvements to the computational efficiency of their model by means of a newly developed explicit moving mesh scheme. The methods that are to be introduced in this thesis however are not restricted solely to cell migration and chemotaxis models. The moving mesh methods can be used to investigate computational fluid dynamics problems or geometry related problems where there is a need for a mesh which evolves in time, for instance in solving problems relating to such as Willmore flow [123, 148] and anisotropic mean curvature flow [134]. The reaction-diffusion methods that are to also be introduced can also be used to investigate other chemical phenomenon non specific to the cell migration model such as pattern formation on domains with growth: in models described in [101, 102] and [103] for example.

1.2 An introduction to current research

1.2.1 A description of the membrane

It is helpful to define certain variables in this section and the cell migration model which will be used extensively throughout the text which we build from the work done in [104], [105] and [106]. The model presented in these publications operates at the single cellular level. The cell is represented as an evolving simple closed curve interface prescribed by the position of a moving domain $\Gamma(t)$ where $\Gamma(t) \subset \mathbb{R}^2$ which is changing with respect to time $t \in I = [0, T],\ (T > 0)$.

Specifically, $\Gamma(t)$ will frequently represent the cell membrane's position in two-dimensional space and by definition shall be a simple closed curve with non-zero encompassing area. This means that any vector $\boldsymbol{x} \in \Gamma(t)$ can be written in terms of a parameterisation at time t where

$$\boldsymbol{x}(s,t) = (x(s,t), y(s,t))^T, \tag{1.1}$$

and where $0 \leq s < 1$.

A natural description of $\Gamma(t)$ then follows from the zero level set,

$$\Gamma(t) = \{\boldsymbol{x} \in \mathbb{R}^2 : d_{\Gamma(t)}(\boldsymbol{x}, t) = 0\}, \tag{1.2}$$

where $d_{\Gamma(t)}$ is the signed distance function which is zero at the cell membrane and positive for $\boldsymbol{x}$ on the exterior of the cell and negative otherwise. This allows us to also quite straightforwardly define the outward facing normal vector

$$\boldsymbol{n}(\boldsymbol{x}, t) = \frac{\nabla d_{\Gamma(t)}(\boldsymbol{x}, t)}{|\nabla d_{\Gamma(t)}(\boldsymbol{x}, t)|}. \tag{1.3}$$

It is important to note that by definition $\Gamma(t)$ is a simple closed curve with non-zero area and no topological changes or discontinuities and so $\boldsymbol{n}(\boldsymbol{x}, t)$ will always exist and be well defined for $\boldsymbol{x} \in \Gamma(t)$. We can now precisiely define some concepts, notation and variables which will be useful throughout this text.

Let $\Gamma(t)$ be a smooth closed curve in $\mathbb{R}^2$ where $\Gamma_0 = \Gamma(0)$ and $\Gamma(t)$ has time-dependent material configurations. A particle P defined on $\Gamma(t)$ has velocity $\dot{\boldsymbol{X}}_p(t)$ therefore, we assume that a velocity field $\boldsymbol{u}$ exists such that points on $\Gamma(t)$ evolve with a velocity field $\dot{\boldsymbol{X}}_p(t) = \boldsymbol{u}(\boldsymbol{X}_p(t), t)$. It follows from (1.1) that the material velocity $\boldsymbol{u}$ can be given by

$$\boldsymbol{u} = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right)\bigg|_s = \frac{\partial \boldsymbol{x}}{\partial t}\bigg|_s, \tag{1.4}$$

where $\cdot|_s$ denotes that we view the derivative of $\boldsymbol{x}$ in this case with respect to the particle's material position. This distinction is important as there are a number of different ways in which the time derivative on $\boldsymbol{x}$ can be derived as will be seen when utilising the arbitrary Lagrangian-Eulerian frame for numerical calculations in later sections.

The arc-length parameterisation of curve $\Gamma(t)$ is also used extensively throughout the text and can be derived from the spatial derivative in $\boldsymbol{x}(s, t)$ where

$$\eta(s, t) = \int_0^s \left|\frac{\partial \boldsymbol{x}}{\partial s}(\bar{s}, t)\right| \mathrm{d}\tilde{s}, \qquad \eta_s = \frac{\partial \eta}{\partial s} = \left|\frac{\partial \boldsymbol{x}}{\partial s}(\bar{s}, t)\right|, \tag{1.5}$$

and where

$$\left|\frac{\partial \boldsymbol{x}}{\partial s}(\bar{s},t)\right| = \sqrt{\left(\frac{\partial x}{\partial s}(\bar{s},t)\right)^2 + \left(\frac{\partial y}{\partial s}(\bar{s},t)\right)^2} \tag{1.6}$$

is the standard discrete L_2 norm. It is also worth noting that the temporal rate of change of the arc-length is

$$\begin{aligned}
\frac{\partial \eta_s}{\partial t}\Big|_s &= \left(\frac{\partial}{\partial t}\sqrt{\left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2}\right)\Bigg|_s \\
&= \left(\frac{\frac{\partial x}{\partial s}\frac{\partial}{\partial t}\frac{\partial x}{\partial s} + \frac{\partial y}{\partial s}\frac{\partial}{\partial t}\frac{\partial y}{\partial s}}{\sqrt{\left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2}}\right)\Bigg|_s \\
&= \frac{\boldsymbol{x}_s}{|\boldsymbol{x}_s|}\cdot\frac{\partial \boldsymbol{x}_s}{\partial t}\Big|_s = \frac{\boldsymbol{x}_s\cdot\boldsymbol{u}_s}{\eta_s},
\end{aligned} \tag{1.7}$$

where $\cdot$ denotes the standard discrete L_2 inner-product which will be used extensively throughout the text. Definition (1.7) will be used in later chapters to derive a numerical procedure for solving reaction-diffusion models on evolving curves.

1.2.2 A description of the cell model

The cell migration model itself is based on a continuous version of the three-equation reaction-diffusion system posed by Hans Meinhardt [94],

$$\begin{aligned}
\dot{a} + \nabla_{\Gamma(t)}\cdot(a\boldsymbol{u}) &= D_a\Delta_{\Gamma(t)}a + \frac{(a^2/b + b_a)\varrho}{(s_c + c)(1 + s_a a^2)} - r_a a, \\
\dot{b} + \nabla_{\Gamma(t)}\cdot(b\boldsymbol{u}) &= D_b\Delta_{\Gamma(t)}b - r_b b + \frac{r_b}{|\Gamma(t)|}\oint_{\Gamma(t)} a\,d\boldsymbol{x}, \\
\dot{c} + \nabla_{\Gamma(t)}\cdot(c\boldsymbol{u}) &= D_c\Delta_{\Gamma(t)}c + b_c a - r_c c,
\end{aligned} \tag{1.8}$$

where

$$\dot{a} = \frac{\partial a}{\partial t}\Big|_s, \qquad \dot{b} = \frac{\partial b}{\partial t}\Big|_s, \qquad \dot{c} = \frac{\partial c}{\partial t}\Big|_s,$$

are the temporal derivatives for the chemical substances with respect to the fixed spatial reference parameter s.

The Laplace-Beltrami operator on $\Gamma(t)$ is defined as

$$\Delta_{\Gamma(t)}\alpha = \nabla_{\Gamma(t)} \cdot (\nabla_{\Gamma(t)}\alpha) \tag{1.9}$$

and is the tangential divergence of the tangential gradient where the tangential gradient

$$\nabla_{\Gamma(t)}\alpha = \nabla\alpha - (\nabla\alpha \cdot \boldsymbol{n})\boldsymbol{n} \tag{1.10}$$

and where $\cdot$ denotes the discrete L_2 inner product and ∇c denotes the Euclidean gradient on $\mathbb{R}^2$ and the normal vector field perpendicular to $\Gamma(t)$ as given in (1.3).

The temporally invariant Laplace-Beltrami operator on $\Gamma(t)$ (1.9) can be written more explicitly as

$$\begin{aligned}
\Delta_{\Gamma(t)}a &= \frac{1}{\eta_s}\frac{\partial}{\partial s}\left(\frac{a}{\eta_s}\right) \\
\Delta_{\Gamma(t)}b &= \frac{1}{\eta_s}\frac{\partial}{\partial s}\left(\frac{b}{\eta_s}\right) \\
\Delta_{\Gamma(t)}c &= \frac{1}{\eta_s}\frac{\partial}{\partial s}\left(\frac{c}{\eta_s}\right)
\end{aligned}$$

when $\Gamma(t)$ takes the form stated above: a simple closed curve with non-zero area defined on $\mathbb{R}^2$ space with a parameter s. It is important to note that this is operator on $\Gamma(t)$ is temporally invariant while the spatial dimension of $\Gamma(t)$ is, in general, changing with respect to time t.

The functions a, b and c represent the local activator, global inhibitor and local inhibitor respectively and hence $D_a < D_c < D_b$. The saturation coefficients are s_a, s_c and r_a, r_b, r_c are the linear decay coefficients. Basal creation coefficients are given by b_a, b_c and finally, ϱ represents a signalling term which biases the auto-catalytic creation of the activator a. The values for these parameters are given in later chapters.

The output from the local activator a is fed into a simple mechanical model which prescribes the motion of the cell in the normal direction and hence also the domain $\Gamma(t)$ movement,

$$\mathcal{V} = \dot{\boldsymbol{x}} \cdot \boldsymbol{n} = K_{\text{prot}}a - f(\kappa), \tag{1.11}$$

where $\mathcal{V}$ is the velocity of the cell in the normal direction, K_{prot} is a constant and $f(\kappa)$ is a function dependent on the local curvature of the cell used to control the area of the cell approximation.

Notice that this model does not explicitly specify which proteins or chemicals the activator and inhibitors correspond to. Instead an underlying assumption is made that the a, b and c concentrations will encompass a variety of different proteins and their interactions. With this generality, the resulting model still produces very natural and realistic looking morphologies [105] which exhibit similar characteristics found in cells such as Dictyostelium. Most notably is the protrusions, known a pseudopods, which form generally in pairs around the cell's leading edge. Generally one of these protrusions will be retracted back into the cell and the net motion from the surviving protrusion coupled by hydrostatic pressure and tension to retract the trailing edge gives the cell model its full motility.

The direction in which the cell chooses to migrate is biased by the signalling term s. Consequently where there is higher degree of signal, there will be a higher chance of the activator rising and generating a pseudopod, although if this is in a significantly different location from where the last pseudopod was formed, then the cell will not usually change its direction immediately. Instead, the pseudopods will gradually relocate to orientate the cell in the direction of the highest concentration of free-ligand, the food source. This gradual movement towards higher concentrations rather than rapid relocation is what is also witnessed in nature [3].

Initially, in Chapter 3, we introduce a procedure for approximating this signal term by using a stochastic modelling process and various assumptions which provide a good approximation for the cell's fractional receptor occupancy. This model is later improved upon after new methods are introduced in Chapter 4, leading to a higher dimensional cell model which takes account of the cell's motion through the environment and how the movement of the cell actually affects the free-ligand concentration.

1.2.3 Common notation

Throughout this text we will often use subscript notation to denote a particular frame of reference or concept. For instance Ω_c will generally denote a reference configuration for the domain Ω. The reference concept will be introduced when looking at the arbitrary Lagrangian-Eulerian approach in the next section. We will also use a subscript h, for instance c_h to denote a linear approximation to a function c, or to a domain: for instance Ω_h is used to denote a piecewise linear approximation to Ω which is composed of a neighbourhood of joined finite elements. These two subscripts may also be combined, where $\Omega_{c,h}$ for instance denotes a triangulation of the domain Ω_c which is a reference domain for Ω.

When viewed as a subscript, the variables ξ, η and t will always refer to a derivative, in other words c_ξ will never refer to anything other than

$$c_\xi \equiv \frac{\partial c}{\partial \xi} \tag{1.12}$$

however this should not be confused with, for instance,

$$\left.\frac{\partial c}{\partial t}\right|_\xi \tag{1.13}$$

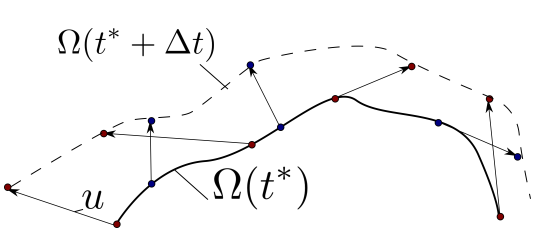
where the evaluation line symbol and subscript of ξ here denote (similar to (1.4)) that the temporal derivative of c is being evaluated under the ξ parameterisation for the domain.

A subscript of $\cdot_0$ will always mean that the temporal variable t will be fixed at $t = 0$, for example

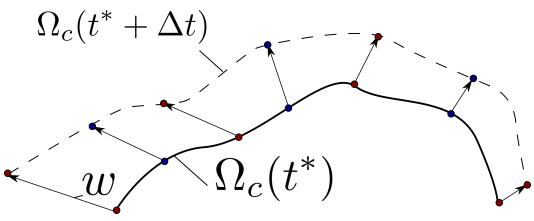
$$c_0 \equiv \left. c(\boldsymbol{x}, t)\right|_{t=0} \equiv c(\boldsymbol{x}, 0). \tag{1.14}$$

1.3 The arbitrary Lagrangian-Eulerian approach

In both instances of curves and general surfaces in high dimensions, there are two velocities that need to be considered to fully model the underlying system, namely the material velocity of the substance or substances which are being modelled and



(a) Movement of the numerical mesh by having grid nodes follow the material velocity.



(b) Movement of the numerical mesh by having grid nodes follow an arbitrary ALE velocity.

Figure 1.1: An illustration depicting the difference between a Lagrangian approach in (a) and the ALE approach in (b). On the left, the mesh points move with the same velocity as the material points which leads to grid entanglement in two places. On the right, a reference domain is used instead to move the mesh points in an arbitrary way so as to avoid the entanglement seen in (a).

the velocity of the underlying numerical mesh upon which the solutions are being approximated. Conventionally, there are three approaches to modelling on surfaces which are themselves evolving.

In the Lagrangian frame, the model is formulated naturally and the particles are the mesh points themselves, for example see Figure 1.1 (a). This approach may be well suited to some applications but in the specific case of the cell problem, because of the nature of the way in which the domain moves to emulate cell morphology and from the corresponding material velocity of the particles, eventually grid entanglement as shown in Figure 1.1 (a) will occur and the underlying numerical methods will then fail. For this reason it would be preferred to not have the mesh move with the same velocity as the particles, or the material velocity $\boldsymbol{u}$.

The Eulerian frame sees the mesh held fixed and the model reformulated so that an additional convective term is introduced. This convective term accounts for the now existing disparity between the mesh node positions and the particle positions. This formulation has the advantage of being more robust in instances where the material velocity could be unpredictable or lead to badly defined meshes. It does however bring in a convective term which may be large as the particles move further from the mesh points. This could mean numerical methods such as up-winding schemes and discontinuous Galerkin FEM for example would have to be employed to ensure stability [7, 27, 58, 76].

The arbitrary Lagrangian-Eulerian (ALE) approach attempts to combine the advantages of both the Lagrangian and Eulerian frame [37, 38, 49, 58]. It does this by reformulating a problem given in the physical domain, defined by the material movement of the particles, to one which is defined on a reference domain whose evolution we may arbitrary control. By doing this we can select a better domain on which to perform our numerical methods: better in the sense that we may control the distribution of the mesh points through this reference domain and avoid any mesh entanglement (such as illustrated in 1.1 (a) where $\Omega_c(t)$ is the corresponding ALE reference domain) that would otherwise arise in a purely Lagrangian setting. In this way the solution can be calculated with the same efficiency of using the Lagrangian frame but also with the same well conditioned meshes of the Eulerian frame.

Much of the understanding and definitions used here are similar in form to [38]. Let us suppose that we are given a problem for which the unknown is $g(\boldsymbol{x}, t)$ and $\boldsymbol{x}$ is the physical coordinate and the solution of $g(\boldsymbol{x}, t)$ is defined in some form of the temporal-derivate in the Lagrangian frame, that is to say that we can express the unknown $g(\boldsymbol{x}, t)$ in a form where

$$\begin{aligned} \frac{\partial g(\boldsymbol{x}, t)}{\partial t} &= f(\boldsymbol{x}, t, g), \qquad && ((\boldsymbol{x}, t) \in \Omega(t) \times I)\,, \\ g(\boldsymbol{x}_0, 0) &= g_0(\boldsymbol{x}), && (\boldsymbol{x}_0 \in \Omega(0))\,. \end{aligned} \tag{1.15}$$

Notice that $\boldsymbol{x} \in \Omega(t)$[1], as in the solution to g is defined in terms of the domain where $\Omega(t)$ is actually also evolving in time. In this frame $\boldsymbol{x}$ follows the material position of the particles. This means to say that given a particular $\boldsymbol{x}_0 \in \Omega(0)$ we can trace the path that the particle makes through physical space by following the velocity of $\Omega(t)$, in other words, we suppose that there exists a mapping, $\mathcal{B}$ say, such that

$$\boldsymbol{x} \in \Omega(t) \iff \exists\, \boldsymbol{x}_0 \in \Omega(0) \;:\; \boldsymbol{x} = \mathcal{B}_t(\boldsymbol{x}_0). \tag{1.16}$$

[1]Note that while $\Omega(t)$ will often be used in this text to denote a domain of two spatial dimensions, the formulation here is actually generalised to no fixed spatial dimensions and $\Gamma(t)$, the cell membrane domain, could also be substituted into the argument in place of $\Omega(t)$. What matters is that there exists bijective mapping functions between the material, physical and referential domains.

In this way we can also define the material velocity of the particles $\boldsymbol{x}$ as

$$\begin{aligned} &\boldsymbol{u} : \Omega(0) \times I \to \mathbb{R}^2, \\ &\boldsymbol{u}(\boldsymbol{x}_0, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{x}_0, t)}{\partial t} = \frac{\partial \mathcal{B}_t(\boldsymbol{x}_0)}{\partial t}. \end{aligned} \tag{1.17}$$

The mapping $\mathcal{B}_t$ is defined by the movement of the physical particles and this may not lead to a smooth or desirable mapping for a numerical procedure. Instead let us suppose that we have a smoother and more desirable mapping which would produce a better numerical mesh and let us define this mapping with $\boldsymbol{\xi}$ as the coordinate system where

$$\begin{aligned} \boldsymbol{\xi} &= \mathcal{A}_t(\boldsymbol{x}_0), \qquad (\boldsymbol{x}_0 \in \Omega(0)), \\ \boldsymbol{\xi} \in \Omega_c(t) &\iff \exists\, \boldsymbol{x}_0 \in \Omega(0) \ : \ \boldsymbol{\xi} = \mathcal{A}_t(\boldsymbol{x}_0). \end{aligned} \tag{1.18}$$

In this sense it would be advantageous to have an expression for g defined on the more desirable mapping function $\mathcal{A}$, in other words under the $\boldsymbol{\xi}$ coordinate system, however we will not in general be provided with such a problem. There is however a way to take the fundamental physical defined in the physical space (1.15) and make it one defined on the reference space $\Omega_c(t)$.

Suppose we have three equivalent forms of g, namely

$$\begin{aligned} g(\boldsymbol{x}, t) &: (\Omega(t), t) \to \mathbb{R}, \\ g^*(\boldsymbol{\xi}, t) &: (\Omega_c(t), t) \to \mathbb{R}, \\ g^{**}(\boldsymbol{X}, t) &: (\Omega(0), t) \to \mathbb{R}, \end{aligned} \tag{1.19}$$

and these are related through the mapping functions above whereby

$$\boldsymbol{X} \in \Omega(0), \quad g^{**}(\boldsymbol{X}, t) = g(\mathcal{B}_t(\boldsymbol{X}), t), \quad g^{**}(\boldsymbol{X}, t) = g^*(\mathcal{A}_t(\boldsymbol{X}), t). \tag{1.20}$$

We may now start to derive the problem (1.15) in terms of g^*, the reference space and hence numerical mesh:

$$\begin{aligned} &\boldsymbol{\xi} \in \Omega_c(t), \\ &\frac{\partial g^*}{\partial t}(\boldsymbol{\xi}, t) = \frac{\partial g^{**}}{\partial t}(\mathcal{A}_t^{-1}(\boldsymbol{\xi}), t) = \frac{\partial g}{\partial t}(\mathcal{B}_t(\mathcal{A}_t^{-1}(\boldsymbol{\xi})), t) \end{aligned} \tag{1.21}$$

or suppose $(\boldsymbol{\xi}, t) = \mathcal{C}_t(\boldsymbol{x}) = \mathcal{B}_t(\mathcal{A}_t^{-1}(\boldsymbol{\xi}))$,

$$\begin{aligned} &\xi \in \Omega_c(t), \\ &\frac{\partial g^*}{\partial(\boldsymbol{\xi}, t)}(\boldsymbol{\xi}, t) = \frac{\partial g}{\partial(\boldsymbol{\xi}, t)}(\mathcal{B}_t(\mathcal{A}_t^{-1}(\boldsymbol{\xi}))t) \\ &\qquad = \frac{\partial g}{\partial(\boldsymbol{x}, t)} \frac{\partial(\boldsymbol{x}, t)}{\partial \mathcal{C}(\boldsymbol{x}, t)} \\ &\qquad = \frac{\partial g}{\partial(\boldsymbol{x}, t)} \frac{\partial(\boldsymbol{x}, t)}{\partial(\boldsymbol{\xi}, t)}, \end{aligned} \tag{1.22}$$

from which we obtain expressions for the spatial and temporal gradient operator on g^*,

$$\frac{\partial g^*}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}, t) = \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{x}, t) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, t) \tag{1.23}$$

$$\frac{\partial g^*}{\partial t}(\boldsymbol{\xi}, t) = \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{x}, t) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, t) + \frac{\partial g}{\partial t}(\boldsymbol{x}, t), \tag{1.24}$$

respectively. The term

$$\frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{x}, t) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, t) \tag{1.25}$$

in (1.24) is a convection-like addition which is brought in through the relative change of velocities between the physical $\Omega(t)$ and reference $\Omega_c(t)$ domains. Notice that we can set $g^* = \boldsymbol{x}(\boldsymbol{\xi}, t)$ in (1.24) so that

$$\frac{\partial \boldsymbol{x}^*}{\partial t}(\boldsymbol{\xi}, t) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{x}}(\boldsymbol{x}, t) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, t) + \frac{\partial \boldsymbol{x}}{\partial t}(\boldsymbol{x}, t) \tag{1.26}$$

from which it follows that the convection-like term can be written as the difference between the reference and physical domain velocity,

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{x}^*}{\partial t}(\boldsymbol{\xi}, t) - \frac{\partial \boldsymbol{x}}{\partial t}(\boldsymbol{x}, t). \tag{1.27}$$

For convenience, throughout this thesis we will often denote concentrations on the physical configuration by an $\boldsymbol{x}$ subscript and concentrations on the reference configuration by a $\boldsymbol{\xi}$ subscript, dropping the asterisks on the function g we will use the forms

$$\frac{\partial g}{\partial t}(\boldsymbol{x}, t) \equiv \frac{\partial g}{\partial t}\Big|_{\boldsymbol{x}} \qquad \text{and} \qquad \frac{\partial g^*}{\partial t}(\boldsymbol{\xi}, t) \equiv \frac{\partial g}{\partial t}\Big|_{\boldsymbol{\xi}} \qquad \text{and} \tag{1.28}$$

as short-hand. It follows that (1.24) can now be written more compactly as

$$\frac{\partial g}{\partial t}\Big|_{\boldsymbol{\xi}} = \nabla g \cdot (\boldsymbol{w} - \boldsymbol{u}) + \frac{\partial g}{\partial t}\Big|_{\boldsymbol{x}}, \tag{1.29}$$

where $\boldsymbol{u}$ is given by (1.4) or, more generally, by

$$\boldsymbol{u} = \frac{\partial \boldsymbol{x}}{\partial t}\Big|_{\boldsymbol{x}}, \tag{1.30}$$

and $\boldsymbol{w}$ is the ALE velocity which comes from the referential domain,

$$\boldsymbol{w} = \frac{\partial \boldsymbol{x}}{\partial t}\Big|_{\boldsymbol{\xi}}. \tag{1.31}$$

Equation (1.29) relates the change of g at the reference configuration (numerical mesh in our case) to the physical configuration as given in (1.15) through the additional convection term and thus problems defined on $\Omega(t)$ can now be defined on $\Omega_c(t)$ through the equivalent mapping (1.29).

1.4 Moving mesh methods for curves

Moving the mesh points in the normal direction will often lead to problems involving grid crossover and will not generally give a good quality mesh (where, for instance, the nodes equidistribute arc-length). To overcome this, the numerical mesh for the cell membrane $\Gamma(t)$ in [104] is evolved using the parameterised finite element method (PFEM) [10] which is a method to evolve parameterised meshes according to functions of curvature or surface diffusion. This procedure has allowed problems involving various growing domains to be investigated [11, 12, 48]. The method works by making use of the geometric identity

$$\Delta_{\Gamma(t)} \boldsymbol{x} = \kappa \boldsymbol{n}, \tag{1.32}$$

where $\boldsymbol{x}$ parameterises the domain $\Gamma(t)$, $\Delta_{\Gamma(t)}$ is the Laplace-Beltrami operator on $\Gamma(t)$ and κ the local curvature of the domain; $\boldsymbol{n}$ is the outward pointing normal. The

geometric identity (1.32) is combined with the normal velocity equation

$$\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = f(\kappa), \tag{1.33}$$

where $\dot{\boldsymbol{x}}$ is the domain velocity and $f(\kappa)$ is a real-valued function which may depend on local curvature.

The PFEM generates a uniform arc-length mesh at each time-step which is useful and often desirable for many applications. There is a caveat which is that the tangential velocity of the mesh is not controllable with PFEM and is calculated intrinsically. A controllable tangential mesh would be desirable because it would allow the relocation of mesh nodes to areas where the underlying numerics would favour more resolution for example. A further caveat is that PFEM's tangential velocity, to maintain the uniform arc-length property of the mesh, can be relatively large when a smaller tangential velocity would suffice. This makes it less favourable for use in applications where the mesh is to be linked to a larger domain as is the case in higher dimensions where the movement of the interface dictates how the higher dimensional meshes should evolve too. A large tangential velocity on the boundary would introduce shearing into the higher dimensional mesh resulting in a need for frequent re-meshing and thus massively increasing computational time.

Pan and Wetton [111] introduce an alternative method which produces meshes similar in quality to PFEM. They make use of a finite difference based scheme for moving the mesh points of a growing domain for which the normal velocity component is the same as (1.33) but the tangential velocity component is prescribed by means of the constraint

$$\boldsymbol{x}_\eta \cdot \boldsymbol{x}_{\eta\eta} = 0, \tag{1.34}$$

where η is the arc-length parameterisation of $\boldsymbol{x}$ and thus ensures a uniform arc-length distribution of mesh points. Equations (1.33) and (1.34) are solved simultaneously for the nodal positions of the evolving mesh.

Following from Pan and Wetton, we look to evolve a curve by mean curvature motion. Given a curve $\boldsymbol{x}$ parameterised by some variable ξ, curvature of $\boldsymbol{x}$ can be related

to the vector given by second derivative of $\boldsymbol{x}$ with respect to arc-length. Namely, if arc-length is denoted as s and mean curvature as κ then $\kappa = \boldsymbol{x}_{\eta\eta} \cdot \boldsymbol{n}$. With this identity in mind, we are interested in specifying the mean curvature motion solely in the normal direction to the interface, the tangential velocity has been set up such that $|\boldsymbol{x}_\eta|_\eta = 0$ which follows from the solution of the system,

$$\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = \kappa, \qquad \boldsymbol{x}_\xi \cdot \boldsymbol{x}_{\xi\xi} = 0.$$

In order to solve this system, Pan and Wetton used both implicit and explicit time integration with a finite difference spatial discretisation.

PFEM and Pan and Wetton's method both generate a smooth evolving uniform mesh which can be used to model reaction-diffusion equations on the cell's membrane. However, the tangential velocity in both these methods are intrinsic and not easily manipulated. We later introduce an adaptive method which allows us to alter the tangential velocity to bring in mesh adaption along the evolving mesh.

1.5 Moving mesh methods for higher dimensions

We so far have been predominantly looking at the cell membrane as a two-dimensional curve with a single spatial dimension. This is in essnse a one-dimensional model but in reality of course cell in life have more dimensions.

In later chapters, a two-dimensional cell model which more accurately models the behaviour of the cell than the single dimensional counterpart will be introduced. Some models exist already in literature for cell modelling in two dimenions such as the surface-based approach in [48] which views the cell membrane as a two-dimensional surface. The to be introduced here however will take the view of extending the current framework, which views the cell membrane $\Gamma(t)$ as a curve, and augmenting it with a more accurate model for the free-ligand molecules which the cells' receptors respond to within its environment. In light of this, we must first discuss procedures for generating the mesh which will be used to model the environment of the cell.

DistMesh [113, 114] is a mesh generating procedure which makes use of Hooke's

law to reposition grid nodes in an ideal fashion such as to satisfy a mesh density specification for the grid. It is used in many applications, for instance in wound healing models [73] and in mathematical modelling of brain tumours [117] . It is robust to different geometries but can be quite slow for moving domains.

The method generates an initial approximation for the mesh and then gradually refines the mesh based on an iterative process of mechanical relaxation for each edge in the mesh until an ideal mesh is generated which has the correct length constraint, that is to say that no edge within the mesh is greater than a specified tolerance $h_{\max}$ in length. It does this in a discrete setting which can often introduce large computational overheads making it a less desirable option for continuous mesh movement. As such, it is an excellent method for generating an initial mesh for other methods.

There exist other mesh generation strategies to evolve the meshes continuously between time-steps which are more desirable. Cao et al.'s [29] moving mesh finite element method (MMPDE) for example is one such method. Their method is derived from a variational problem $I[\boldsymbol{\xi}]$ which they have used to define a coordinate transformation from domain to another. In this way they have a computational domain Ω_c with a coordinate system $\boldsymbol{\xi} = (\xi, \eta) \in \Omega_c$ and a physical domain $\Omega(t)$ with a coordinate system $\boldsymbol{x} = (x, y) \in \Omega(t)$. To obtain the physical mesh which is moving in time using their method, a grid must already exist on Ω_c with known mesh points $\boldsymbol{\xi}_i \in \Omega_c$ say, where i denotes the i^{th} mesh point of the grid. The unknown physical grid is then found as a result of the mapping $\boldsymbol{x}_i(t) = I(\boldsymbol{\xi}_i)$.

In this way, the connectivity of the moving mesh is identical to the connectivity of the mesh on Ω_c and, in addition, the mesh connectivity is maintained at each time-step which reduces computational overheads. Another advantage of this methods is that mesh quality control and solution adaption can straightforwardly be incorporated into an elliptic mesh generation system which is formed from a variational approach. Further, unlike algebraic grid generators, the mesh produced by an elliptic system will have desirable smoothness characteristics.

Exactly how to select the variational $I[\boldsymbol{\xi}]$ remains a subject of ongoing study [61]

but the mapping considered in this thesis though will take the general form

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega(t)} \left((\nabla\xi)^T G^{-1}(\nabla\xi) + (\nabla\eta)^T G^{-1}(\nabla\eta)\right)\, \mathrm{d}\boldsymbol{x}, \tag{1.35}$$

where G is a 2×2 symmetric positive definite matrix. In particular G is often referred to as the monitor matrix for its ability to control the size, shape and orientation of the resulting physical mesh [65].

Having the ability to control the physical mesh through a parameter in the variational form is highly convenient. Generating adapted grids would normally be a computationally more epensive operation but here it comes naturally coupled as part of the moving mesh PDE solution. By selecting an appropriate G monitor matrix, we have the capacity to refine the mesh in particular area, perhaps where there is a numerical shock which requires more mesh points to properly resolve; or if there is an artefact in the solution we wish to look at in closer detail, a vortex for example. Likewise, by selecting a different G we could also chose to speed up computation by relocating mesh points away from areas where we are not interested in but while still maintaining a high level of accuracy in areas where we are.

The choice of G is really problem specific but in general it will be chosen in such a way that the underlying numerical method, for which the mesh is being created for, will have enhanced accuracy in regions of interest. A number of different methods for choosing a monitor matrix which assist the underlying numerical methods have been proposed and, in fact, monitor selection and exactly how to chose an appropriate monitor matrix to minimise a given norm remains an active area of research.

The Hessian as the monitor matrix is a popular choice (see for example [22], [23] and [30]) motivated in large part by D'Azevedo [33] and Simpson [34]. This specific choice has a provable a priori upper bound for the error which is useful in problems where mesh generation is very expensive and it would be prudent to know before calculation if a single time-step of a grid evolving procedure is going to be enough or whether the simulation should re-mesh first.

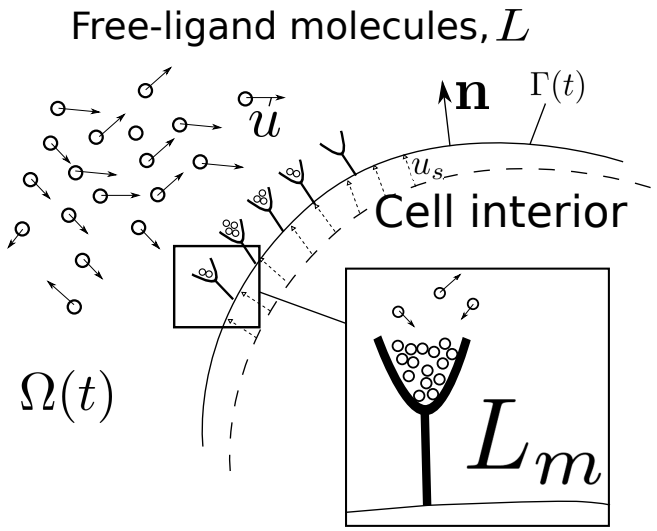


Figure 1.2: An illustration of the free-ligand receptor binding/unbinding process which takes place at the cell membrane. Here, the external free-ligand field L is made up of small free-ligand molecules which exist on $\Omega(t)$ which is the domain on the exterior of the cell $\Gamma(t)$. Some of these molecules may bind with receptors on the cell membrane to form the receptor-bound complex concentration L_m.

1.6 Coupled bulk-surface problems

There are instances where multiple processes are taking place on different domains, and even different dimensions. For instance coupling the process which happens at the membrane between the free-ligand chemoattractant on the exterior of the cell and the receptors which are attached to the membrane. An illustration of this biological process is shown in Figure 1.2. Modelling this process accurately is a crucial step in understanding chemotaxis. The binding and unbinding of the free-ligand molecules onto and off of the receptors can be modelled as a flux boundary condition between the bulk exterior domain $\Omega(t)$ and a surface model for the receptor occupancy. We will see in later chapters that this can be modelled with the advection-diffusion problem

$$\begin{aligned}
\frac{\partial L}{\partial t}\Big|_{\boldsymbol{x}} + \nabla_{\Omega(t)} \cdot (L\boldsymbol{u}) &= D_L \Delta_{\Omega(t)} L, \\
-D_L \frac{\partial L}{\partial \boldsymbol{n}}\Big|_{\Gamma(t)} - [(\boldsymbol{u}\cdot\boldsymbol{n})]L\Big|_{\Gamma(t)} &= k_1(N_r - L_m)L\Big|_{\Gamma(t)} - k_{-1}L_m, \\
\frac{\partial L_m}{\partial t}\Big|_{\boldsymbol{x}} + \nabla_{\Gamma(t)} \cdot (L_m \boldsymbol{u}_s) &= D_s \Delta_{\Gamma(t)} L_m + k_1(N_r - L_m)L\Big|_{\Gamma(t)} - k_{-1}L_m,
\end{aligned} \tag{1.36}$$

where L is the concentration of free-ligand in the environment exterior to the cell; and L_m is the concentration of receptor-ligand complex that exists on the membrane $\Gamma(t)$. The diffusivity coefficient of the free-ligand is D_L, $\boldsymbol{u}$ is the material velocity of the free-ligand in the bulk, $\boldsymbol{u}_s$ and D_s is the diffusivity are the material velocity and diffusivity of the bound ligand-receptors on the membrane respectively; k_1 and k_{-1} are the binding and unbinding rates of the free-ligand to and from the receptors; N_r is the local concentration of receptors.

The $[(\boldsymbol{u}\cdot\boldsymbol{n})]L\Big|_{\Gamma(t)}$ term represents an advective flux and is dependent on the jump in the normal velocity at the boundary. Here $[(\boldsymbol{u}\cdot\boldsymbol{n})] = \boldsymbol{u}^+ - \boldsymbol{u}$, and $\boldsymbol{u}^+ = \lim_{\boldsymbol{x}\to\Gamma(t)^+}\boldsymbol{u}$. Thus at the leading edge of the cell, $[(\boldsymbol{u}\cdot\boldsymbol{n})] < 0$ so the free-ligand will advect onto $\Gamma(t)$, whereas at the trailing edge $[(\boldsymbol{u}\cdot\boldsymbol{n})] > 0$ so the free-ligand will advect off $\Gamma(t)$. This generates what is known as a wind-shield effect and produces a more realistic model of receptor-ligand binding process which is not encapsulated in the original model given in [105].

1.7 Thesis overview

In the next chapter, numerical methods will be introduced which will be used to model cell migration and chemotaxis. In particular, we investigate a conservative arbitrary Lagrangian-Eulerian finite element method for solving reaction-diffusion equations on evolving simple closed curves. We then introduce an adaptive moving mesh method for forced mean curvature flow.

In chapter three, both of these methods will be combined to solve reaction-diffusion equations on evolving domains. Using a continuous model of the Meinhardt reaction-diffusion system, the methods combine to generate convincing simulations of cell migration and chemotaxis. In the fourth chapter, the methods of the previous chapters are extended to two dimensions. These are subsequently used in the fifth chapter to improve upon the original cell migration model. Finally, there is a short discussion of the significant points outlined in this thesis as well as suggestions for further work.

Chapter 2

Finite element solution of reaction-diffusion equations on evolving 1D domains

In the cell migration model to come, we will seek solutions of reaction-diffusion problems which exist on the moving cell membrane. These chemical concentrations will be used to motivate the protrusion mechanisms and therefore to move the cell. To do this we must first derive a method for computing solutions to reaction-diffusion problems on moving curves in general and a method for moving the cell interface correctly.

2.1 Derivation of a reaction-diffusion problem on an evolving curve

Consider a chemical species, c, which is evolving on the domain $\Gamma(t)$, which is the cell membrane domain given in (1.2). We will suppose that $c : \Gamma(t) \times I \to \mathbb{R}$ diffuses according to Fick's law and therefore the total concentration of c adheres to the mass conservation constraint

$$\frac{\partial}{\partial t} \int_{\Gamma(t)} c(\boldsymbol{x}(t), t) \, \mathrm{d}\Gamma(t) = \int_{\Gamma(t)} (\nabla \cdot (\mu \nabla c) + f(c)) \, \mathrm{d}\Gamma(t). \tag{2.1}$$

By noting (1.7), Fick's law and a reaction term $f(c)$ can be expressed in terms of the

s physical coordinate parameterisation as

$$\frac{d}{dt}\int_{s_1}^{s_2} c(\boldsymbol{x}(\tilde{s},t),t)\eta_s(\tilde{s},t)\,\mathrm{d}\tilde{s} = \int_{s_1}^{s_2}\left(\mu\frac{\partial}{\partial s}\left(\frac{c_s(\tilde{s},t)}{\eta_s(\tilde{s},t)}\right)+f(c)\eta_s(\tilde{s},t)\right)\,\mathrm{d}\tilde{s},$$

where μ is the diffusivity of the chemical species and η_s is introduced as a result of the domain movement. Since the limits of integration do not depend on t, the derivative on the left can be taken inside the integral to give

$$\int_{s_1}^{s_2}\left(\frac{\partial c}{\partial t}\Big|_s\eta_s+\left(\frac{\boldsymbol{x}_s\cdot\boldsymbol{u}_s}{\eta_s}\right)c\right)\,\mathrm{d}\tilde{s} = \int_{s_1}^{s_2}\left(\mu\frac{\partial}{\partial s}\left(\frac{c_s}{\eta_s}\right)+f(c)\eta_s\right)\,\mathrm{d}\tilde{s},$$

where the substitution for $\frac{\partial\eta_s}{\partial t}\Big|_s$ has been taken from (1.7). Finally, letting $s_2\to s_1$ and dividing through by η_s we arrive at the point-wise conversion law

$$\frac{\partial c}{\partial t}\Big|_s+\left(\frac{\boldsymbol{x}_s\cdot\boldsymbol{u}_s}{\eta_s^2}\right)c=\frac{\mu}{\eta_s}\frac{\partial}{\partial s}\left(\frac{c_s}{\eta_s}\right)+f(c), \tag{2.2}$$

which is defined for all $\boldsymbol{x}\in\Gamma(t)$. The evolving curve $\Gamma(t)$ is closed so there are no boundary conditions.

2.2 Arbitrary Lagrangian Eulerian formulation

To derive the ALE formulation, we make use of the change of variable discussed in Section 1.3 which allows us to re-write (2.2) in terms of a referential domain, which will be parameterised by ξ, by noting that

$$\frac{\partial c}{\partial t}\Big|_\xi=\nabla c\cdot(\boldsymbol{w}-\boldsymbol{u})+\frac{\partial c}{\partial t}\Big|_s$$

where $\boldsymbol{w}$ is the ALE velocity (1.31).

We may re-write the spatial gradient in terms of the ALE coordinate so that

$$\nabla c=\boldsymbol{J}^{-1}\nabla_\xi c \tag{2.3}$$

where $\boldsymbol{J}$ is the time-dependent Jacobi of the mapping from the physical domain $\Gamma(t)$

to the reference domain Γ_c

$$\boldsymbol{J}(t) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}(t), \tag{2.4}$$

and ∇_ξ simply means the corresponding spatial gradient under the ALE mapping

$$\nabla_\xi c = \frac{\partial c}{\partial \xi}(\xi, t) \equiv c_\xi. \tag{2.5}$$

The inverse of (2.4) in terms of the physical coordinate can be found by noting that

$$\boldsymbol{J}^{-1} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}} = \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} = \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} |\boldsymbol{J}|\xi_x \\ |\boldsymbol{J}|\xi_y \end{bmatrix} = \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} (x_\xi)^2\xi_x + (y_\xi)^2\xi_x \\ (x_\xi)^2\xi_y + (y_\xi)^2\xi_y \end{bmatrix} = \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} x_\xi \\ y_\xi \end{bmatrix} = \frac{1}{|\boldsymbol{J}|} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}, \tag{2.6}$$

where

$$|\boldsymbol{J}| = \det\left(\boldsymbol{J}(t)\right) = x_\xi^2 + y_\xi^2 = \eta_\xi^2. \tag{2.7}$$

It therefore follows that

$$\nabla c = \frac{\boldsymbol{x}_\xi}{|J|} c_\xi = \frac{\boldsymbol{x}_\xi}{\eta_\xi^2} c_\xi, \tag{2.8}$$

and that

$$\frac{\partial c}{\partial t}\Big|_s = \frac{\partial c}{\partial t}\Big|_\xi + \left(\frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi}{\eta_\xi^2} \right) c_\xi,$$

so we may write (2.2) in terms of the ALE coordinate frame as

$$\frac{\partial c}{\partial t}\Big|_\xi + \left(\frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi}{\eta_\xi^2} \right) c_\xi + \left(\frac{\boldsymbol{x}_\xi \cdot \boldsymbol{u}_\xi}{\eta_\xi^2} \right) c = \frac{\mu}{\eta_\xi} \frac{\partial}{\partial \xi} \left(\frac{c_\xi}{\eta_\xi} \right) + f(c), \tag{2.9}$$

with corresponding ALE boundary conditions

$$c(\xi, t)\big|_{\xi=0} = c(\xi, t)\big|_{\xi=1}, \qquad c(\xi, 0) = c_0(\xi), \qquad \xi = \mathcal{A}_t^{-1}(\boldsymbol{x}). \tag{2.10}$$

2.3 Weak ALE formulation

Define the ALE mapping test space $\mathcal{H}^1(\Gamma(t))$ as

$$\mathcal{H}^1(\Gamma(t)) = \{v : \Gamma(t) \to \mathbb{R} : \ v = \hat{v} \circ \mathcal{A}_t^{-1}, \ \hat{v} \in H^1(\Gamma_c)\}, \qquad t \in I,$$

in other words, the set of real value functions v defined on $\Gamma(t)$ which have a corresponding function $\hat{v}$ defined on the ALE reference frame Γ_c. The weak solution for $c(\xi, t)$ is then derived by multiplying (2.9) through by suitable test function $v \in \mathcal{H}^1(\Gamma(t))$, and integrating over $\Gamma(t)$,

$$\int_0^1 \left(\frac{\partial c}{\partial t}\Big|_\xi v + \left(\frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi}{\eta_\xi^2} \right) c_\xi v + \left(\frac{\boldsymbol{x}_\xi \cdot \boldsymbol{u}_\xi}{\eta_\xi^2} \right) cv \right) \, \mathrm{d}\xi = \int_0^1 \left(\frac{\mu}{\eta_\xi} \frac{\partial}{\partial \xi} \left(\frac{c_\xi}{\eta_\xi} \right) v + f(c)v \right) \, \mathrm{d}\xi.$$

Noticing that

$$\frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi \, c_\xi}{\eta_\xi^2} + \frac{\boldsymbol{x}_\xi \cdot \boldsymbol{u}_\xi \, c}{\eta_\xi^2} - \frac{\boldsymbol{w}_\xi \cdot \boldsymbol{x}_\xi \, c}{\eta_\xi^2} \equiv \frac{((\boldsymbol{u} - \boldsymbol{w})c)_\xi \cdot \boldsymbol{x}_\xi}{\eta_\xi^2},$$

we have

$$\int_0^1 \left(\frac{\partial c}{\partial t}\Big|_\xi v + \left(\frac{((\boldsymbol{u} - \boldsymbol{w})c)_\xi \cdot \boldsymbol{x}_\xi}{\eta_\xi^2} \right) v + \left(\frac{\boldsymbol{x}_\xi \cdot \boldsymbol{w}_\xi}{\eta_\xi^2} \right) cv \right) \, \mathrm{d}\xi = \int_0^1 \left(\frac{\mu}{\eta_\xi} \frac{\partial}{\partial \xi} \left(\frac{c_\xi}{\eta_\xi} \right) v + f(c)v \right) \, \mathrm{d}\xi.$$

Since v does not depend on t by definition, then

$$\frac{d}{dt} \int_0^1 v\psi \, \mathrm{d}\xi = \int_0^1 v \left(\frac{\partial \psi}{\partial t}\Big|_\xi + \psi \frac{\boldsymbol{x}_\xi \cdot \boldsymbol{w}_\xi}{\eta_\xi} \right) \mathrm{d}\xi,$$

from which it follows that substituting in $\psi = c$ and integrating the right-hand integral using parts that the weak form is: find $c \in \mathcal{H}^1(\Gamma(t))$ where

$$\frac{d}{dt} \int_0^1 cv\eta_\xi \, \mathrm{d}\xi + \int_0^1 \frac{((\boldsymbol{u} - \boldsymbol{w})c)_\xi \cdot \boldsymbol{x}_\xi}{\eta_\xi} v \, \mathrm{d}\xi + \int_0^1 \mu \frac{c_\xi v_\xi}{\eta_\xi} \, \mathrm{d}\xi = \int_0^1 f(c)v\eta_\xi \, \mathrm{d}\xi \qquad (2.11)$$

for all $v \in \mathcal{H}^1(\Gamma(t))$.

One desirable characteristic of the ALEFEM scheme is conservation. As we will

see later, a discretisation of the formulation given in (2.11) is not guaranteed to be conservative. However, it is possible to modify the weak formulation and obtain a conservative numerical approximation. We first note that

$$\left(\frac{(\boldsymbol{u}-\boldsymbol{w})c\cdot\boldsymbol{x}_\xi}{\eta_\xi}\right)_\xi = \frac{((\boldsymbol{u}-\boldsymbol{w})c)_\xi\cdot\boldsymbol{x}_\xi}{\eta_\xi} + (\boldsymbol{u}-\boldsymbol{w})c\cdot\left(\frac{\boldsymbol{x}_\xi}{\eta_\xi}\right)_\xi.$$

Because the velocity $(\boldsymbol{u}-\boldsymbol{w})$ is tangent to the curve and

$$\begin{aligned}\left(\frac{\boldsymbol{x}_\xi}{\eta_\xi}\right)_\xi &= (\boldsymbol{x}_\eta)_\xi\\ &= \boldsymbol{x}_{\eta\eta}\eta_\xi\\ &= \kappa\eta_\xi\boldsymbol{n},\end{aligned}$$

where η denotes the arc-length parameterisation, it follows that $(\boldsymbol{u}-\boldsymbol{w})\cdot\left(\frac{\boldsymbol{x}_\xi}{\eta_\xi}\right)_\xi = 0$ and hence, on a continuous level, (2.11) and: find $c\in\mathcal{H}^1(\Gamma(t))$ such that

$$\frac{d}{dt}\int_0^1 cv\eta_\xi\,\mathrm{d}\xi + \int_0^1\left(\frac{(\boldsymbol{u}-\boldsymbol{w})\cdot\boldsymbol{x}_\xi c}{\eta_\xi}\right)_\xi v\,\mathrm{d}\xi + \int_0^1\mu\frac{c_\xi v_\xi}{\eta_\xi}\,\mathrm{d}\xi = \int_0^1 f(c)v\eta_\xi\,\mathrm{d}\xi \quad (2.12)$$

for all $v\in\mathcal{H}^1(\Gamma(t))$ are equivalent. We will present numerical experiments later to show that the finite discretisation of (2.12), derived in the next section, is conservative at the discrete level, whereas the discretisation of (2.11) is non-conservative.

2.4 Finite element discretisation

2.4.1 Spatial discretisation

If $\Gamma(t)$ is reasonably smooth, then it can be approximated well using continuous piecewise linear segments. Let $\Gamma_h(t)$ be the piecewise linear curve which approximates $\Gamma(t)$ with nodes defined at positions $\{\boldsymbol{x}_i(t)\}_{i=1}^{N+1}$ at time t so that $\Gamma_h(t)=\cup_{K\in\mathcal{T}_h}K$, where K is the straight edge element connecting each neighbouring nodes in the triangulation $\mathcal{T}_h$. There exists a corresponding approximation for Γ_c, $\Gamma_{c,h}$ say, given by the piecewise linear triangulation $\mathcal{T}_{c,h}$ connecting $\{\boldsymbol{\xi}_i\}_{i=1}^{N+1}$. The nodes $\{\boldsymbol{x}_i(t)\}_{i=1}^{N+1}$ and the

corresponding nodes $\{\boldsymbol{\xi}_i\}_{i=1}^{N+1}$ on $\mathcal{T}_{c,h}$ will come from moving mesh methods which will be introduced in the proceeding sections.

The finite element test space is then defined as

$$\mathcal{L}^1(\Gamma_{c,h}) = \{v_h \in \mathcal{H}^1(\Gamma_{c,h}) : v\Big|_K \in \mathbb{P}_1(K), \forall K \in T_{c,h}\}.$$

It then follows that any function $\psi \in \mathcal{L}^1(\Gamma_{c,h})$ can be written as

$$\psi(\xi) = \sum_{i=1}^{N+1} \psi_i \hat{\phi}_i(\xi),$$

where $\hat{\phi}_i$ is the piecewise linear hat function which satisfies $\hat{\phi}_i(\xi_j) = \delta_{ij}$ (the Kronecker delta). Further, define the test space

$$\mathcal{L}(\Gamma_h(t)) = \left\{ v_h : \Gamma_h(t) \to \mathbb{R} : v_h = \hat{v}_h \circ \mathcal{A}_{h,t}^{-1},\ \hat{v}_h \in \mathcal{L}^1(\Gamma_{c,h}) \right\},$$

the space of real valued piecewise linear functions v_h defined on $\Gamma_h(t)$ which have corresponding piecewise linear functions $\hat{v}_h$ defined on the finite element ALE frame $\Gamma_{c,h}$. The FEM spatial discretisation for the weak formulation (2.12) is then given by: find $c^h \in \mathcal{L}(\Gamma_h(t))$ such that

$$\frac{d}{dt}\int_0^1 c^h v^h \eta_\xi \, \mathrm{d}\xi + \int_0^1 \left(\frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi c^h}{\eta_\xi} \right)_\xi v^h \, \mathrm{d}\xi + \int_0^1 \mu \frac{{c^h}_\xi {v^h}_\xi}{\eta_\xi} \, \mathrm{d}\xi = \int_0^1 f(c^h) v^h \eta_\xi \, \mathrm{d}\xi \tag{2.13}$$

for all $v^h \in \mathcal{L}(\Gamma_h(t))$. Let $\boldsymbol{C}(t) = \{c_i\}_{i=1}^N$ be the vector of unknowns, then (2.13) can be written as

$$\frac{d}{dt}(M(t)\boldsymbol{C}(t)) + \mu H(t)\boldsymbol{C}(t) + A(t)\boldsymbol{C}(t) = M(t)\boldsymbol{F}(\boldsymbol{C}(t)), \tag{2.14}$$

where

$$[M(t)]_{ij} = \int_0^1 \hat{\phi}_i(t)\hat{\phi}_j(t)\eta_\xi \, \mathrm{d}\xi$$

is the time-dependent mass matrix,

$$[H(t)]_{ij} = \int_0^1 \frac{(\hat{\phi}_i)_\xi (\hat{\phi}_j)_\xi}{\eta_\xi} \mathrm{d}\xi$$

is the stiffness matrix and

$$[A(t)]_{ij} = \int_0^1 \frac{(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{x}_\xi \hat{\phi}_i \left(\hat{\phi}_j\right)_\xi}{\eta_\xi} \, \mathrm{d}\xi \tag{2.15}$$

is a convection-like term which is brought in by the difference between the ALE and material velocity. The load vector for the reaction term has entries

$$[\boldsymbol{F}(\boldsymbol{C}(t))]_i = f(C_i(t)).$$

The semi-discretisation of the non-conservative weak formulation gives rise to the same system of equations as (2.14),

$$\frac{d}{dt}(M(t)\boldsymbol{C}(t)) + \mu H(t)\boldsymbol{C}(t) + A(t)\boldsymbol{C}(t) = M(t)\boldsymbol{F}(\boldsymbol{C}(t)), \tag{2.16}$$

with $M(t)$ and $H(t)$ as above but the ALE matrix has entries

$$[A(t)]_{ij} = \int_0^1 \frac{\hat{\phi}_i \left((\boldsymbol{u} - \boldsymbol{w})\hat{\phi}_j\right)_\xi \cdot \boldsymbol{x}_\xi}{\eta_\xi} \, \mathrm{d}\xi. \tag{2.17}$$

2.4.2 Temporal Discretisation

We will use the h suffix to denote that the temporal mapping $\mathcal{A}_t$ corresponds to the piecewise continuous domain $\Gamma_h(t)$ rather than to the continuous domain $\Gamma(t)$. In other words, we have that for any point ξ on Γ_c, the reference domain, there exists a corresponding point on $\Gamma_h(t)$ such that $\boldsymbol{x}(\xi, t) = \mathcal{A}_{h,t}(\xi)$.

A temporal discretisation of (2.14) is obtained by assuming that the moving domain $\Gamma(t)$ evolves smoothly enough that it can be interpolated linearly between time points $\{t^n\}_{i=1}^{N_T}$ where $t^n = n\Delta t$ and $\Delta t = T/N_T$.

That is to say, if $\mathcal{A}_{h,t^n}(\xi)$ and $\mathcal{A}_{h,t^{n+1}}(\xi)$ exist and are well defined then

$$\mathcal{A}_{h,\Delta t}(\xi,t) = \frac{t-t^n}{\Delta t}\mathcal{A}_{h,t^{n+1}}(\xi) + \frac{t^{n+1}-t}{\Delta t}\mathcal{A}_{h,t^n}(\xi), \qquad t \in [t_n, t_{n+1}),$$

from which the grid velocity $\hat{\boldsymbol{w}}^{n+1}_{h,\Delta t}$ in the ALE frame is given by

$$\begin{aligned}
\boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{x},t) &= \hat{\boldsymbol{w}}^{n+1}_{h,\Delta t}(\xi) \circ \mathcal{A}^{-1}_{h,\Delta t}(\boldsymbol{x}), \qquad \text{where} \\
\hat{\boldsymbol{w}}^{n+1}_{h,\Delta t}(\xi,t) &= \frac{\mathcal{A}_{h,t^{n+1}}(\xi,t) - \mathcal{A}_{h,t^n}(\xi,t)}{\Delta t}, \qquad t \in [t^n, t^{n+1}),
\end{aligned}$$

which is piecewise constant with respect to t.

Following from (2.14) we derive a semi-implicit IMEX scheme for the the integration of the derivative. For conviencene, let

$$M^n = M(t^n), \qquad H^n = H(t^n), \qquad A^n = A(t^n)$$

and

$$Z(t) = -\mu H(t)\boldsymbol{C}(t) - A(t)\boldsymbol{C}(t).$$

It follows that (2.14) can be written as

$$\frac{d}{dt}(M(t)\boldsymbol{C}(t)) = Z(t) + M(t)\boldsymbol{F}(\boldsymbol{C}(t)). \tag{2.18}$$

A Crank-Nicolson approach can be used to integrate the derivative and the $Z(t)$ function while the non-linear reaction term is calculated explicitly at the t^n time-step so that

$$\frac{M^{n+1}\boldsymbol{C}^{n+1} - M^n\boldsymbol{C}^n}{\Delta t} = \frac{Z(t^{n+1}) + Z(t^n)}{2} + M^n\boldsymbol{F}(\boldsymbol{C}^n), \tag{2.19}$$

where

$$Z(t^n) = -\mu H^n\boldsymbol{C}^n - A^n\boldsymbol{C}^n \quad \text{and} \quad Z(t^{n+1}) = -\mu H^{n+1}\boldsymbol{C}^{n+1} - A^{n+1}\boldsymbol{C}^{n+1}.$$

A semi-implicit set of linear equations is thus derived for the full discretisation of (2.14), whereby the linear diffusion and mesh movement terms are evaluated at the

time level t^{n+1} with the non-linear reaction term treated explicitly at time level t^n which results in the second order IMEX scheme

$$(M^{n+1} + \frac{1}{2}\Delta t\mu H^{n+1} + \frac{1}{2}\Delta t A^{n+1})\boldsymbol{C}^{n+1} = (M^n - \frac{1}{2}\Delta t\mu H^n - \frac{1}{2}\Delta t A^n)\boldsymbol{C}^n + \Delta t M^n \boldsymbol{F}(\boldsymbol{C}^n). \tag{2.20}$$

Test problems and applications considered here have reasonably smooth non-linear reaction terms so the time step Δt does not have to become too small due to numerical stiffness constraints. The above procedure however can be extended for non-uniform Δt if required when reaction terms are more stiff.

2.5 Numerical experiments

2.5.1 Diffusion around circles

To investigate the order of convergence of (2.13), we first consider the problem

$$\begin{aligned} \frac{\partial c}{\partial t}\Big|_s &= \frac{1}{\eta_s}\frac{\partial}{\partial s}\left(\frac{c_s}{\eta_s}\right), \quad 0 < t \le T = 1, \\ c(s,0) &= \sin(2\pi s), \qquad 0 \le s \le 1, \end{aligned} \tag{2.21}$$

on the unit circle

$$\begin{aligned} x(s,t) &= \cos(2\pi s), \\ y(s,t) &= \sin(2\pi s). \end{aligned} \tag{2.22}$$

The analytical solution for this problem is $c(s,t) = e^{-t}\sin(2\pi s)$. The numerical mesh is selected such that

$$\begin{aligned} \Delta\xi &= 1/N, & \xi_i &= i\Delta\xi, & i &= 0, \ldots, N \\ x(\xi,t) &= \cos(2\pi\xi + \alpha t), & x_i(t) &= x(\xi_i,t), & & \\ y(\xi,t) &= \sin(2\pi\xi + \alpha t), & y_i(t) &= y(\xi_i,t), & & \end{aligned}$$

where α is a constant parameter that indicates a rotation of the mesh. Problem (2.21) is solved using ALEFEM scheme (2.20). The estimated order of convergence (EOC) is

then calculated to be

$$\text{EOC} = \log_2 \left(\frac{||e_N||_{\mathcal{L}_\infty}}{||e_{2N}||_{\mathcal{L}_\infty}} \right),$$

where

$$||e_N||_{\mathcal{L}_\infty} = \max_{0 \le i \le N} \left|\left| c_i(T) - c\left(\xi_i + \frac{\alpha}{2\pi} T, T\right) \right|\right|$$

is the maximum nodal error at the final time T.

	Uniform mesh		Non-uniform mesh	
N	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
32	2.361e-03		7.213e-03	
64	5.886e-04	2.004	1.801e-03	2.002
128	1.454e-04	2.017	4.484e-04	2.006
256	3.460e-05	2.071	1.104e-04	2.023

Table 2.1: Estimated order of spatial convergence results using (2.20) to solve (2.21) on a fixed circular mesh. The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The time-step is fixed with $\Delta t = 3.906 \times 10^{-3}$. Results suggest method (2.20) converges with second order accuracy spatially.

The mesh is also selected such that

$$\begin{aligned} x_i(t) &= \cos(2\pi\xi_i^2), \qquad \xi_i = i/N, \\ y_i(t) &= \sin(2\pi\xi_i^2), \end{aligned} \tag{2.23}$$

for which the grid is no longer equidistributed according to arc-length.

Table 2.1 and Table 2.2 show the results for the test problem for various values of N and time-step dt and with meshes which are uniformly distributed according to arc-length, and with meshes which have nodes distributred according to (2.23). Table

	Uniform mesh		Non-uniform mesh	
Δt	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
3.125e-02	1.469e-04		1.469e-04	
1.563e-02	3.693e-05	1.991	3.694e-05	1.991
7.813e-03	9.168e-06	2.010	9.173e-06	2.010
3.906e-03	2.189e-06	2.066	2.195e-06	2.063

Table 2.2: Estimated order of temporal convergence results using (2.20) to solve (2.21) on a fixed circular mesh. The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The number of elements in the mesh is fixed with $N = 4096$. Results suggest method (2.20) converges with second order accuracy in time.

2.1 uses a fixed time-step and shows the estimated spatial convergences of the problem to be second-order. Similarly, Table 2.2 uses a fixed number of elements to determine the estimated rate of convergence in time. The results in this table suggest this to be also second order.

If the mesh is rotated by setting $\alpha = 0.5$, then a tangential velocity is introduced which is accounted for by the ALE term. Tables 2.3 and 2.4 show that the estimated order of convergence for this experiment does not change and continue to demonstrate that the order of accuracy remains second-order both spatially and temporally, respectively.

Finally, we consider a circular domain which is shrinking at a constant rate β, such that the domain is given by

$$\begin{aligned} x(s,t) &= (1-\beta t)\cos(2\pi s), \qquad 0 \le s \le 1, \\ y(s,t) &= (1-\beta t)\sin(2\pi s). \end{aligned} \tag{2.24}$$

Assuming the initial condition $c(s,0) = \sin(2\pi s)$, the analytical solution for $c(s,t)$ can be derived using separation of variables as follows. Notice that when the domain is

	Uniform mesh		Non-uniform mesh	
N	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
32	2.433e-03		7.723e-03	
64	6.074e-04	2.002	1.931e-03	1.999
128	1.509e-04	2.009	4.814e-04	2.004
256	3.679e-05	2.036	1.190e-04	2.016

Table 2.3: Estimated order of spatial convergence results using (2.20) to solve (2.21) on a circular mesh with a constant angular velocity ($\alpha = 0.5$). The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The time-step is fixed with $\Delta t = 3.906 \times 10^{-3}$. Results suggest method (2.20) converges with second order accuracy spatially.

evolving isotropically the diffusion equation takes the form

$$\left.\frac{\partial c}{\partial t}\right|_s = \frac{c_{ss}}{\eta_s^2} - \frac{\dot{r}}{r}c, \tag{2.25}$$

where $r(t) = 1 - \beta t$ is the radius and $\eta_s = 2\pi(1 - \beta t)$ is the arc-length of the circle domain. Let $c(s,t) = f(s)h(t)$ then substituting into (2.25) we have

$$\begin{aligned} fh_t &= \frac{f_{ss}h}{\eta_s^2} - \frac{\dot{r}}{r}fh \\ \implies \left(\frac{h_t}{h} + \frac{\dot{r}}{r}\right)\eta_s^2 &= \frac{f_{ss}}{f} \equiv -K, \end{aligned}$$

where K is constant, independent of s and t. Re-ordering the left hand term gives

$$h_t + \left(\frac{\dot{r}}{r} + \frac{K}{\eta_s^2}\right)h = 0,$$

	Uniform mesh		Non-uniform mesh	
Δt	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
3.125e-02	1.903e-04		1.903e-04	
1.563e-02	4.792e-05	1.989	4.792e-05	1.989
7.813e-03	1.198e-05	2.000	1.198e-05	2.000
3.906e-03	2.954e-06	2.020	2.958e-06	2.018

Table 2.4: Estimated order of temporal convergence results using (2.20) to solve (2.21) on a circular mesh with a constant angular velocity ($\alpha = 0.5$). The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The number of elements in the mesh is fixed with $N = 4096$. Results suggest method (2.20) converges with second order accuracy in time.

which is a first order differential equation in t for which the solution is

$$\begin{aligned} h(t) &= A\exp\left(-\int\left(\frac{\dot{r}}{r}+\frac{K}{\eta_s^2}\right)dt\right) \\ &= A\exp\left(-\int\left(\frac{-\beta}{1-\beta t}+\frac{K}{(2\pi)^2(1-\beta t)^2}\right)dt\right) \\ &= A\exp\left(-\ln(1-\beta t)-\frac{Kt}{(2\pi)^2(1-\beta t)}\right), \\ &= \frac{A}{1-\beta t}\exp\left(-\frac{Kt}{(2\pi)^2(1-\beta t)}\right). \end{aligned}$$

Furthermore,

$$f_{ss}+Kf=0 \implies f(s)=B\sin(\sqrt{K}s)+C\cos(\sqrt{K}s).$$

Since $c(s,0)=\sin(2\pi s)$, then $C=0$ and $\sqrt{K}=2\pi$ must be used and therefore

$$c(s,t)=\frac{\sin(2\pi s)}{1-\beta t}\exp\left(-\frac{t}{(1-\beta t)}\right).$$

The results for $\beta = 0.5$ using the second-order scheme (2.20) are given in Table 2.5 and

	Uniform mesh		Non-uniform mesh	
N	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
32	3.466e-03		1.264e-02	
64	8.678e-04	1.998	3.169e-03	1.996
128	2.163e-04	2.004	7.921e-04	2.000
256	5.328e-05	2.021	1.973e-04	2.006

Table 2.5: Estimated order of spatial convergence results using (2.20) to solve (2.21) on a circular mesh with a constant shrinkage factor ($\beta = 0.5$). The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The time-step is fixed with $\Delta t = 3.906 \times 10^{-3}$. Results suggest method (2.20) converges with second order accuracy spatially.

Table 2.6. These tables show that the method continues to converge with a second order estimated rate of convergence for the given problem in space and time, respectively.

To demonstrate the conservation property of this scheme with convection-like term (2.15), and lack of conservation in the alternative form for discretizing the convection-like term (2.17), the non-uniform initial unit circle mesh is rotated and shrunk where $\alpha = 1$ and $\beta = 0.5$. The total concentration of $c_h(\xi, t)$ is then given by

$$T_c^h(t) = \frac{1}{2} \sum_{i=1}^{N} ||\boldsymbol{x}_{i+1}(t) - \boldsymbol{x}_i(t)||(c_{i+1}(t) + c_i(t)).$$

Figure 2.1 shows $T_c^h(t)$ for $t \in [0, 1]$. The total amount of c^h can be seen decreasing in the non-conservative scheme by up to as much as $T_c^h(t) = -8.604e-04$ which occurs when $t = 1$. The conservative scheme however satisfies $|T_c^h(t)| < 10^{-14}$ during the same interval. Over even longer time-scales, the non-conservative scheme will introduce large errors due to fictitious concentration which is brought in or lost through domain movement.

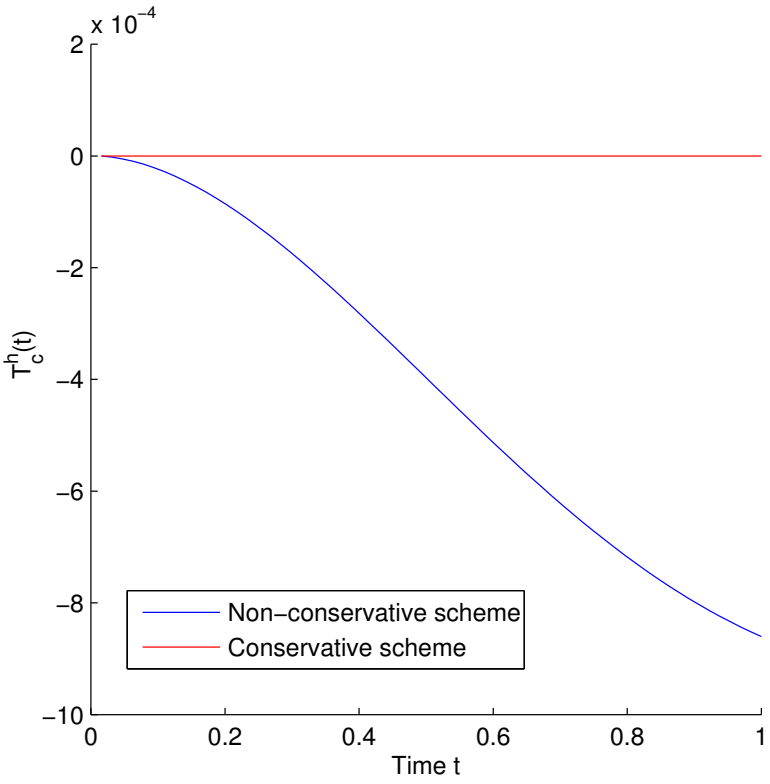


Figure 2.1: The total amount of c_h for the conserved (2.14) and non-conserved (2.16) ALEFEM schemes. Non-uniform circle mesh used with domain shrinking and rotating at constant rate ($\alpha = 1$, $\beta = 0.5$, $N = 64$, $\Delta t = 1.563e - 02$).

	Uniform mesh		Non-uniform mesh	
Δt	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
3.125e-02	6.914e-05		6.928e-05	
1.563e-02	1.699e-05	2.025	1.713e-05	2.016
7.813e-03	4.072e-06	2.061	4.208e-06	2.025
3.906e-03	8.567e-07	2.249	9.938e-07	2.082

Table 2.6: Estimated order of temporal convergence results using (2.20) to solve (2.21) on a circular mesh with a constant shrinkage factor ($\beta = 0.5$). The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The number of elements in the mesh is fixed with $N = 4096$. Results suggest method (2.20) converges with second order accuracy in time.

2.5.2 Diffusion around ellipses

We now consider solving the diffusion equation (2.21) on the stationary ellipse

$$\begin{aligned} x(s,t) &= 2\cos(2\pi s), \quad 0 \leq s \leq 1, \\ y(s,t) &= \sin(2\pi s). \end{aligned} \tag{2.26}$$

The exact analytical solution of (2.21) using this parameterisation for the ellipse domain is non-obvious as $\eta_s = 2\pi\sqrt{4\sin^2(2\pi s) + \cos^2(2\pi s)}$ and hence η_s depends non-trivially on s. Finding a separable solution using this parameterisation is therefore difficult. Instead, a normalised unit arc-length parameterisation, u, is used, where

$$\eta(u) = u\eta(1), \qquad 0 \leq u \leq 1.$$

Using this parameterisation we have $\eta_u = \eta(1)$ and hence we can find a separable solution of (2.21). If the initial condition is $c(u,0) = \sin(2\pi u)$, then the solution is $c(u,t) = e^{-At}\sin(2\pi u)$, where $A = (2\pi/\eta(1))^2$. The equidistributing arc-length mesh points are found by finding the polar parameter values satisfying the non-linear scalar

equations

$$\eta(s_i) = \frac{i}{N}\eta(1),$$

where

$$\eta(s) = \int_0^s 2\pi\sqrt{4\sin^2(2\pi\omega) + \cos^2(2\pi\omega)}\ \mathrm{d}\omega.$$

These are solved using Newton iteration.

	Uniform mesh		Non-uniform mesh	
N	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
16	7.783e-03		2.231e-02	
32	2.058e-03	1.919	5.182e-03	2.106
64	5.155e-04	1.997	1.490e-03	1.798
128	1.291e-04	1.997	3.434e-04	2.117

Table 2.7: Estimated order of spatial convergence results using (2.20) to solve (2.21) on an stationary ellipse mesh. The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The time-step is fixed at $\Delta t = 1 \times 10^4$. Results suggest method (2.20) converges with second order accuracy in space for the test problem.

Table 2.7 and 2.8 show that the estimated order of spatial and temporal convergence continues to be second-order accurate, respectively. This is true when the mesh points are distributed evenly around the curve and when they are re-distributed according to (2.23). This is a good indication that more obscure geometries, which will be looked at later on, will not affect the accuracy of this scheme. In a later chapter, experiments will be done solving reaction-diffusion equations on non-uniform moving meshes with complex geometries using this method. These results give a degree of confidence that the numerical meshes used will not contaminate the PDE solutions.

	Uniform mesh		Non-uniform mesh	
Δt	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC	$\|\|e_N\|\|_{\mathcal{L}_\infty}$	EOC
5.000e-02	1.018e-05		1.018e-05	
2.500e-02	2.546e-06	1.999	2.548e-06	1.998
1.250e-02	6.373e-07	1.998	6.392e-07	1.995
6.250e-03	1.601e-07	1.993	1.621e-07	1.979

Table 2.8: Estimated order of temporal convergence results using (2.20) to solve (2.21) on an stationary ellipse mesh. The mesh is tested both with mesh points distributed evenly around the circle and with mesh points distributed according to (2.23). The number of elements in the mesh is fixed with $N = 4 \times 10^4$. Results suggest method (2.20) converges with second order accuracy in time for the test problem.

2.6 An adapted moving mesh PDE for evolving curves

To evolve the numerical domain a moving mesh PDE (MMPDE) is used. There are instances where a uniform arc-length distribution of grid points is less than ideal and where some adaptivity could be used to improve accuracy. For instance, where the solution to a PDE on the evolving curve is not smooth or has a sharp moving wave front, some degree of solution adaptivity could potentially improve speed and accuracy of the result. Further, geometric adaptivity could be used to maintain features along the domain that can become lost when the mesh is smoothed by equidistributing the nodes; these features (which may amount to a bump or protrusion in the domain) would only be recoverable with a finer grid when no adaptivity is used.

The method of Pan and Wetton [111] generates an arc-length uniformly distributed grid using finite differences to solve the underlying coupled algebraic system of PDEs. We introduce an evolving adaptive moving mesh PDE for curves by building from their framework. Pan and Wetton's method, as proposed for the case where a curve is evolved by mean curvature, makes use of the basic geometric identity $\boldsymbol{x}_{\eta\eta} = \kappa \boldsymbol{n}$, where

η is the arc-length parameter. We will assume that the curve $\boldsymbol{x}$ is simple and closed and that the first and second derivative exist and are continuous at all points along the curve will and make use of the period boundary conditions $\boldsymbol{x}(1) = \boldsymbol{x}(0)$. Therefore, for a general parameterisation ξ, it follows that since

$$\boldsymbol{x}_\xi = \boldsymbol{x}_\eta \eta_\xi \tag{2.27}$$

then

$$\boldsymbol{x}_{\xi\xi} = \boldsymbol{x}_{\eta\xi}\eta_\xi + \boldsymbol{x}_\eta \eta_{\xi\xi} = \boldsymbol{x}_{\eta\eta}\eta_\xi^2 + \boldsymbol{x}_\eta \eta_{\xi\xi}. \tag{2.28}$$

Since $\boldsymbol{x}_\eta = \boldsymbol{t}$, it follows that

$$\boldsymbol{x}_{\xi\xi} \cdot \boldsymbol{n} = (\boldsymbol{x}_{\eta\eta} \cdot \boldsymbol{n})\eta_\xi^2, \tag{2.29}$$

and hence

$$\kappa = \frac{\boldsymbol{x}_{\xi\xi}}{\eta_\xi^2} \cdot \boldsymbol{n} = \frac{\boldsymbol{x}_{\xi\xi}}{||\boldsymbol{x}_\xi||^2} \cdot \boldsymbol{n}. \tag{2.30}$$

For mean curvature flow the normal velocity can therefore be expressed as

$$\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = \kappa = \frac{\boldsymbol{x}_{\xi\xi}}{||\boldsymbol{x}_\xi||^2} \cdot \boldsymbol{n}. \tag{2.31}$$

As we have seen from Section 1.5, to maintain a uniform distribution of mesh points in terms of arc-length we must impose the algebraic constraint $||\boldsymbol{x}_\xi||_\xi = 0$. Equivalently, we have

$$\boldsymbol{x}_\xi \cdot \boldsymbol{x}_{\xi\xi} = 0. \tag{2.32}$$

Pan and Wetton's method therefore amounts to the simultaneous solution to the differential algebraic system (2.31) and (2.32). We augmented their method for our purposes

such that now

$$\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = \alpha \frac{\boldsymbol{x}_{\xi\xi}}{||\boldsymbol{x}_\xi||^2} \cdot \boldsymbol{n} + \beta, \tag{2.33}$$

and

$$||M\boldsymbol{x}_\xi||_\xi = 0 \implies \boldsymbol{x}_\xi \cdot \boldsymbol{x}_{\xi\xi} = -\frac{M_\xi}{M}||\boldsymbol{x}_\xi||^2.$$

Here, M is a positive monitor function which controls the spacing between each node in such a way that, when integrated between any two consecutive nodes, the monitor function multiplied by the arc-length should remain constant. This has the effect of condensing the grid where M is large and expanding it where M is small. The coupled partial differential algebraic equations (PDAEs) are then discretised using finite differences and solved by iterating to convergence. This system works but is very sensitive to changes in M.

To improve robustness, MMPDE5 from [65] is used to derive a new expression for the tangential velocity component

$$\dot{\boldsymbol{x}} \cdot \boldsymbol{t} = \frac{P}{\tau}||M\boldsymbol{x}_\xi||_\xi, \tag{2.34}$$

where $\tau > 0$ is a temporal smoothing parameter which helps to reduce spatial-temporal oscillations and a spatial-balancing function P which helps to ensure that the coefficients of the linear system are balanced evenly over the domain. The choice of P and τ are problem specific but in the cell simulations which follow in this thesis the value of $P = 1$ and $\tau = 10^{-3}$ are chosen which have been seen to have good robustness characteristics for the cell problem in all simulations we've conducted. Ideally P/τ will be small enough to allow a mesh to quickly adapt to the change in the adaptivity function M but large enough to dampen any oscillations which may occur due to the

rapid relocation of mesh points. Notice that (2.34) can be expanded such that

$$\begin{aligned}\frac{\tau}{P}\dot{\boldsymbol{x}}\cdot\boldsymbol{t} &= ||M\boldsymbol{x}_\xi||_\xi \\ &= \frac{MM_\xi\boldsymbol{x}_\xi\cdot\boldsymbol{x}_\xi + M^2\boldsymbol{x}_{\xi\xi}\cdot\boldsymbol{x}_\xi}{||M\boldsymbol{x}_\xi||} \\ &= \frac{M_\xi\boldsymbol{x}_\xi\cdot\boldsymbol{x}_\xi}{||\boldsymbol{x}_\xi||} + \frac{M\boldsymbol{x}_{\xi\xi}\cdot\boldsymbol{x}_\xi}{||\boldsymbol{x}_\xi||} \\ &= (M_\xi\boldsymbol{x}_\xi + M\boldsymbol{x}_{\xi\xi})\cdot\boldsymbol{t},\end{aligned} \tag{2.35}$$

where the last step is obtained by noting that $\boldsymbol{t} = \frac{\boldsymbol{x}_\xi}{||\boldsymbol{x}_\xi||}$. It therefore follows that

$$\left(\frac{\tau}{P}\dot{\boldsymbol{x}} - M_\xi\boldsymbol{x}_\xi - M\boldsymbol{x}_{\xi\xi}\right)\cdot\boldsymbol{x}_\xi = 0 \tag{2.36}$$

or

$$\boldsymbol{x}_{\xi\xi}\cdot\boldsymbol{x}_\xi = -\frac{M_\xi}{M}||\boldsymbol{x}_\xi||^2 + \frac{\tau}{MP}\dot{\boldsymbol{x}}\cdot\boldsymbol{x}_\xi. \tag{2.37}$$

It is interesting to note that as $\tau \to 0$ the equidistribution condition $||M\boldsymbol{x}_\xi||_\xi = 0$ is recovered.

While much of this thesis makes use of finite element methods, Pan and Wetton's approach for grid generation makes use of finite differences and since we intend to augment their method with a mesh adaption quality, we will attempt to do this in finite differences. It is worth noting however that attempt were initially made to generate an adaptive finite element method for grid generation based on extensions to the Parameterised Finite Element Method [11, 12, 48] spoken of earlier but there were issues with robustness of this method. The non-uniqueness of the grid generated by PFEM was thought to be the cause for these problems and this approach was therefore abandoned in favour of the finite difference method presented here now.

A numerical approximation for (2.33) can be found using central differences

$$\boldsymbol{x}_{\xi\xi}(\xi_i, t^n) \approx \frac{\boldsymbol{x}_{i-1}^n - 2\boldsymbol{x}_i^n + \boldsymbol{x}_{i+1}^n}{(\Delta\xi)^2}, \tag{2.38}$$

$$\boldsymbol{x}_{\xi}(\xi_i, t^n) \approx \frac{\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n}{2\Delta\xi}, \tag{2.39}$$

$$M_{\xi}^n \approx \frac{M_{i+1}^n - M_{i-1}^n}{2\Delta\xi}, \tag{2.40}$$

where $M_i^n = M(\xi_i, t^n)$. In most instances the monitor function M will depend on the underlying PDE being solved or the geometry of the domain itself. In some situations M_ξ may already be known in which case the last approximation will not be required and the exact function M_ξ can be substituted instead to give a more accurate result.

Finally, as $\boldsymbol{t}_i^n = \boldsymbol{x}_{\xi i}^n / ||\boldsymbol{x}_{\xi i}^n|| = (t_{(1)i}^n, t_{(2)i}^n)$ then we set $\boldsymbol{n}_i^n = (-t_{(2)i}^n, t_{(1)i}^n)$ and a numerical approximation of (2.33) is as follows

$$\left(-\alpha_i \boldsymbol{x}_{i-1}^{n+1} + \left(\frac{\gamma_i^n}{\Delta t} + 2\alpha_i\right)\boldsymbol{x}_i^{n+1} - \alpha_i \boldsymbol{x}_{i+1}^{n+1}\right) \cdot \boldsymbol{n}_i^n = \gamma_i^n \left(\frac{\boldsymbol{x}_i^n \cdot \boldsymbol{n}_i^n}{\Delta t} + \beta_i\right), \qquad i = 1, \ldots, N-1, \tag{2.41}$$

where $\gamma_i^n = ||\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n||^2/4$ and the periodic boundary condition is enforced by $\boldsymbol{x}_N^n = \boldsymbol{x}_1^n$. Similarly, using central differences again, a numerical approximation for (2.37) is as follows

$$\begin{aligned}\left(-\boldsymbol{x}_{i-1}^{n+1} + \left(-\frac{\tau\Delta\xi^2}{P_i\Delta t} + 2\right)\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_{i+1}^{n+1}\right) \cdot \left(\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n\right) \\ = -\frac{M_{i+1}^n - M_{i-1}^n}{M_i^n}\gamma_i^n - \boldsymbol{x}_i^n \cdot \left(\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n\right)\frac{\tau\Delta\xi^2}{M_i^n P_i \Delta t}.\end{aligned} \tag{2.42}$$

Because this scheme uses a coupled non-linear system to find both x and y components of the curve, certain care must be taken. For instance, a single solve may not be close enough to the desired solution. It has been seen that a sequence of iterations will usually converge to the desired curve. The final iterative numerical approximation is

as follows

$$\begin{aligned}
&\left(-\alpha_i \boldsymbol{x}_{i-1}^{[n+1,m+1]} + \left(\frac{\gamma_i^{[n,m]}}{\Delta t} + 2\alpha_i\right)\boldsymbol{x}_i^{[n+1,m+1]} - \alpha_i \boldsymbol{x}_{i+1}^{[n+1,m+1]}\right)\cdot \boldsymbol{n}_i^{[n,m]} \\
&\qquad = \gamma_i^{[n,m]}\left(\frac{\boldsymbol{x}_i^{[n,0]}\cdot \boldsymbol{n}_i^n}{\Delta t} + \beta_i\right), \\
&\left(-\boldsymbol{x}_{i-1}^{[n+1,m+1]} + \left(-\frac{\tau\Delta\xi^2}{P_i M_i^n \Delta t} + 2\right)\boldsymbol{x}_i^{[n+1,m+1]} - \boldsymbol{x}_{i+1}^{[n+1,m+1]}\right)\cdot\left(\boldsymbol{x}_{i+1}^{[n,m]} - \boldsymbol{x}_{i-1}^{[n,m]}\right) \\
&\qquad = -\frac{M_{i+1}^n - M_{i-1}^n}{M_i^n}\gamma_i^{[n,m]} + \boldsymbol{x}_i^{[n,0]}\cdot\left(\boldsymbol{x}_{i+1}^{[n,m]} - \boldsymbol{x}_{i-1}^{[n,m]}\right)\frac{\tau\Delta\xi^2}{M_i^n P_i \Delta t},
\end{aligned} \tag{2.43}$$

where $\boldsymbol{x}_i^{[n,m]}$ denotes the approximation to $\boldsymbol{x}_i^n$ after m iterations. Further, a degree of under relaxing has been seen to improve robustness of the method and indeed lead to a faster convergence over fewer iterations. Therefore, the numerical procedure is as follows:

1. Select appropriate τ and P for the problem. We will always select $\tau = 10^{-3}$ and $P = 1/M$ unless specified otherwise. We take $P = 1/M$ because this slows down the mesh point movement and has desirable improved stability characteristics.

2. Let x_i^n and y_i^n be the current node positions at time n. Set $u_i = x_i^n$, $v_i = y_i^n$ and set TOL $= 10^{-8}$.

3. Do until convergence criteria given in (j) is met

   (a) Set $D_i^x = u_{i+1} - u_{i-1}$, $\quad D_i^y = v_{i+1} - v_{i-1}$ and $S_i = \sqrt{D_i^{x2} + D_i^{y2}}$.

   (b) Set $N_i^x = -D_i^y/S_i$ and $N_i^y = D_i^x/S_i$.

   (c) Set $\mu_i = \tau/(M_i^n P_i^n \Delta t)$.

   (d) Set up linear system for normal velocity approximation (both x and y) according to

$$\begin{aligned}
\mathrm{A}_{i,i-1}^x &= \alpha_i N_i^x, & \mathrm{A}_{i,i-1}^y &= \alpha_i N_i^y, \\
\mathrm{A}_{i,i}^x &= \left(-2\alpha_i + S_i^2/(4\Delta t)\right) N_i^x, & \mathrm{A}_{i,i}^y &= \left(-2\alpha_i + S_i^2/(4\Delta t)\right) N_i^y, \\
\mathrm{A}_{i,i+1}^x &= \alpha_i N_i^x, & \mathrm{A}_{i,i+1}^y &= \alpha_i N_i^y.
\end{aligned}$$

(e) Set up linear system for algebraic constraint/tangential velocity (both x and y) according to

$$\begin{aligned} \mathrm{B}^x_{i,i-1} &= D^x_i, & \mathrm{B}^y_{i,i-1} &= D^y_i, \\ \mathrm{B}^x_{i,i} &= -\left(2+\mu_i\Delta\xi^2\right) D^x_i, & \mathrm{B}^y_{i,i} &= -\left(2+\mu_i\Delta\xi^2\right) D^y_i, \\ \mathrm{B}^x_{i,i+1} &= D^x_i, & \mathrm{B}^y_{i,i+1} &= D^y_i. \end{aligned}$$

(f) Set up RHS for normal velocity approximation according to

$$F_i = S_i^2\left(\beta + (N_i^x x_i^n + N_i^y y_i^n)/\Delta t\right)/4.$$

(g) Set up RHS for tangential velocity approximation according to

$$G_i = -((M_{i+1}^n - M_{i-1}^n)/(4M_i^n))S_i^2 - \mu_i(x_i^n D_i^x + y_i^n D_i^y)\Delta\xi^2.$$

(h) Set $\boldsymbol{u}^o = \boldsymbol{u}$ and $\boldsymbol{v}^o = \boldsymbol{v}$.

(i) Solve the system

$$\begin{bmatrix} A^x & A^y \\ B^x & B^y \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{G} \end{bmatrix}.$$[1]

(j) If $\max_{1\le i\le N+1} ||(u_i, v_i) - (u_i^o, v_i^o)||_2 < \mathrm{TOL}$, then go to 4.

(k) Under relaxing step: set $\boldsymbol{u} = \boldsymbol{u}/2 + \boldsymbol{u}^o/2$ and $\boldsymbol{v} = \boldsymbol{v}/2 + \boldsymbol{v}^o/2$.

(l) Go to step (a).

4. End: new nodal positions at time $n+1$ are now given by $x_i^{n+1} = u_i$ and $y_i^{n+1} = v_i$.

The first order discretisation given above is simple and easier to implement for researchers looking to use the MMPDE1D method quickly and without stringent mesh accuracy requirements. From a mathematical perspective however it is more desirable to derive a higher order method for increased accuracy.

[1]The resulting system of (i) is sparse and each of the sub-matricies A^x, A^y, B^x and B^y have a tridiagonal structure so a direct solve is carried out using the Matlab *backslash* operator which is optimized to find solutions for sparse matrices efficiently and accurately using variant methods of Gaussian elimination.

An attempt initially was made to increase the order of the scheme by more accurately calculating the temporal integration of both components: the tangential movement along the curve; and the normal movement perpendicular to the curve. A Crank-Nicolson integration was used simultaneously by both components for this, however, many stability issues were observed during the testing of this method so it was abandoned in favour of a hybrid approach which appeared to offer the same level of accuracy without the stability problems.

The alternative discretisation which offers second-order temporal accuracy is similar to the first order method outlined above but derived using a Crank-Nicolson temporal discretisation for the normal component only. The tangential component is discretised in time using the backward Euler scheme similar to before however the time-step is now halved. Doing this appears to provide considerable increases in stability as when the tangential component is disretised using Crank-Nicolson τ must be chosen several orders of magnitude larger to prevent instabilities from developing. This is not ideal because the added adaption will then be slowed down several orders of magnitude.

Care must be taken to ensure that the normal component is taken at the correct half-way temporal point. The finite difference scheme for MMPDE1D is now given as:

$$\begin{aligned}
&\left(-\frac{1}{2}\alpha_i \boldsymbol{x}_{i-1}^{[n+1,m+1]} + \left(\frac{\gamma_i^{[n,m]}}{4\Delta t} + \alpha_i\right)\boldsymbol{x}_i^{[n+1,m+1]} - \alpha_i \boldsymbol{x}_{i+1}^{[n+1,m+1]}\right)\cdot \hat{\boldsymbol{n}}_i^n \\
&\qquad = \gamma_i^{[n,m]}\left(\frac{\boldsymbol{x}_i^{[n,0]}\cdot\hat{\boldsymbol{n}}^n}{\Delta t} + \beta_i\right) - \frac{\gamma_i^{[n,m]}}{\gamma_i^{[n,0]}}\left(-\frac{1}{2}\alpha_i \boldsymbol{x}_{i-1}^{[n,0]} + \alpha_i \boldsymbol{x}_i^{[n,0]} - \frac{1}{2}\alpha_i \boldsymbol{x}_{i+1}^{[n,0]}\right)\cdot \hat{\boldsymbol{n}}_i^n, \\
&\left(\frac{1}{2}\boldsymbol{x}_{i-1}^{[n+1,m+1]} - \left(\frac{\tau\Delta\xi^2}{P_i M_i^n \Delta t} + 1\right)\boldsymbol{x}_i^{[n+1,m+1]} + \frac{1}{2}\boldsymbol{x}_{i+1}^{[n+1,m+1]}\right)\cdot \hat{\boldsymbol{t}}_i^n \\
&\qquad = -\frac{M_{i+1}^n - M_{i-1}^n}{M_i^n}\gamma_i^{[n,m]} + \boldsymbol{x}_i^{[n,0]}\cdot\left(\boldsymbol{x}_{i+1}^{[n,m]} - \boldsymbol{x}_{i-1}^{[n,m]}\right)\frac{\tau\Delta\xi^2}{M_i^n P_i \Delta t}.
\end{aligned} \tag{2.44}$$

where $\hat{\boldsymbol{n}}_i^n = (\boldsymbol{n}_i^{[n+1,m]} + \boldsymbol{n}_i^n)/2$, $\hat{\boldsymbol{t}}_i^n = (\boldsymbol{t}_i^{[n+1,m]} + \boldsymbol{t}_i^n)/2$ and $\boldsymbol{t}_i^{[n,m]} = \boldsymbol{x}_{i+1}^{[n,m]} - \boldsymbol{x}_{i-1}^{[n,m]}$. In this form, the algorithm is similar to before.

1. Select appropriate τ and P for the problem.

2. Let x_i^n and y_i^n be the current node positions at time n. Set $u_i = x_i^n$, $v_i = y_i^n$ and set TOL $= 1e-8$.

3. Set $\bar{D}_i^x = u_{i+1} - u_{i-1}, \quad \bar{D}_i^y = v_{i+1} - v_{i-1}$ and $\bar{S}_i = \sqrt{(\bar{D}_i^x)^2 + (\bar{D}_i^y)^2}$.

4. Set $\bar{N}_i^x = -\bar{D}_i^y / \bar{S}_i$ and $\bar{N}_i^y = \bar{D}_i^x / \bar{S}_i$.

5. Do until convergence criteria given in (l) is met

   (a) Set $D_i^x = u_{i+1} - u_{i-1}, \quad D_i^y = v_{i+1} - v_{i-1}$ and $S_i = \sqrt{D_i^{x2} + D_i^{y2}}$.

   (b) Set $\hat{S}_i = (S_i + \bar{S}_i)/2$.

   (c) Set $N_i^x = -D_i^y / S_i$ and $N_i^y = D_i^x / S_i$.

   (d) Set $\hat{N}_i^x = (N_i^x + \bar{N}_i^x)/2$ and $\hat{N}_i^y = (N_i^y + \bar{N}_i^y)/2$.

   (e) Set $\mu_i = \tau / (M_i^n P_i^n \Delta t)$.

   (f) Set up linear system for normal velocity approximation (both x and y) according to

$$\begin{aligned}
\mathrm{A}_{i,i-1}^x &= \alpha_i \hat{N}_i^x/2, & \mathrm{A}_{i,i-1}^y &= \alpha_i \hat{N}_i^y/2, \\
\mathrm{A}_{i,i}^x &= \left(-\alpha_i + S_i^2/(4\Delta t)\right) \hat{N}_i^x, & \mathrm{A}_{i,i}^y &= \left(-\alpha_i + S_i^2/(4\Delta t)\right) \hat{N}_i^y, \\
\mathrm{A}_{i,i+1}^x &= \alpha_i \hat{N}_i^x/2, & \mathrm{A}_{i,i+1}^y &= \alpha_i \hat{N}_i^y/2.
\end{aligned}$$

   (g) Set up linear system for algebraic constraint/tangential velocity (both x and y) according to

$$\begin{aligned}
\mathrm{B}_{i,i-1}^x &= D_i^x, & \mathrm{B}_{i,i-1}^y &= D_i^y, \\
\mathrm{B}_{i,i}^x &= -\left(2 + \mu_i \Delta \xi^2\right) D_i^x, & \mathrm{B}_{i,i}^y &= -\left(2 + \mu_i \Delta \xi^2\right) D_i^y, \\
\mathrm{B}_{i,i+1}^x &= D_i^x, & \mathrm{B}_{i,i+1}^y &= D_i^y.
\end{aligned}$$

   (h) Set up RHS for normal velocity approximation according to

$$\begin{aligned}
F_i = S_i^2 \left(\beta + (\hat{N}_i^x x_i^n + \hat{N}_i^y y_i^n)/\Delta t\right)/4 - \left(S_i^2/\bar{S}_i^2\right) \Big(&\alpha_i \hat{N}_i^x x_{i-1}^n/2 - \alpha_i \hat{N}_i^x x_i^n + \alpha_i \hat{N}_i^x x_{i+1}^n/2 \\
&+ \alpha_i \hat{N}_i^y y_{i-1}^n/2 - \alpha_i \hat{N}_i^y y_i^n + \alpha_i \hat{N}_i^y y_{i+1}^n/2\Big).
\end{aligned}$$

(i) Set up RHS for tangential velocity approximation according to

$$G_i = -((M_{i+1}^n - M_{i-1}^n)/(4M_i^n))S_i^2 - \mu_i(x_i^n D_i^x + y_i^n D_i^y)\Delta\xi^2.$$

(j) Set $\boldsymbol{u}^o = \boldsymbol{u}$ and $\boldsymbol{v}^o = \boldsymbol{v}$.

(k) Solve the system

$$\begin{bmatrix} A^x & A^y \\ B^x & B^y \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{G} \end{bmatrix}.$$

(l) If $\max_{1\le i\le N+1} ||(u_i, v_i) - (u_i^o, v_i^o)||_2 <$ TOL then go to 6.

(m) Under relaxing step: set $\boldsymbol{u} = \boldsymbol{u}/2 + \boldsymbol{u}^o/2$ and $\boldsymbol{v} = \boldsymbol{v}/2 + \boldsymbol{v}^o/2$.

(n) Go to step (a).

6. End: new nodal positions at time $n+1$ are now given by $x_i^{n+1} = u_i$ and $y_i^{n+1} = v_i$.

2.7 The de Boor algorithm for parametric curves

To initiate the moving mesh method, it is important to be able to generate a starting mesh that equidistributes the monitor function. This ensures a smooth initial evolution of the mesh points and improves solution accuracy and stability. If we assume the initial curve is given parametrically in the form $\boldsymbol{x}(s) = [x(s), y(s)]$, we need to generate a partition $\{s_i\}_{i=1}^N$ such that

$$\int_0^{s_i} M(t)\left|\frac{d\eta}{ds}\right| ds = \frac{i}{N}\int_0^1 M(t)\left|\frac{d\eta}{ds}\right| ds, \qquad i = 1, \ldots, N.$$

To generate an approximation of the equidistribution mesh we will use an adaption of the so-called de Boor algorithm [35]. We assume M and $|d\eta/ds|$ can be evaluated on an arbitrary background partition $\{s_i^{old}\}_{i=1}^N$ and the function $M(t)|d\eta/ds|$ approximated

by the piecewise constant function

$$\rho(s) = \begin{cases} M(s_{1/2})|d\eta/ds|_{s_{1/2}}, & s \in [s_1, s_2] \\ M(s_{3/2})|d\eta/ds|_{s_{3/2}}, & s \in (s_2, s_3] \\ \vdots & \vdots \\ M(s_{N-1/2})|d\eta/ds|_{s_{N-1/2}}, & s \in (s_{N-1}, s_N], \end{cases}$$

where $s_{i+1/2} = (s_i + s_{i+1})/2$, $i = 1, \ldots, N-1$. A new partition $\{s_i^{new}\}_{i=1}^N$, which exactly equidistributes $\rho(s)$, can be found using inverse linear interpolation (algorithmic details can be found in [6] and [62]). The new partition of course only equidistributes ρ over the old partition and hence iteration is used to update the partition further by simply setting the old partition to be the new partition and repeating the de Boor step. A sequence of partitions can then be generated which eventually converges to the final approximately equidistributed partition of the parameterised domain. The physical mesh point locations $\{x(s_i), y(s_i)\}_{i=1}^N$ are obtained from the parametric map of the final converged partition $\{s_i\}_{i=1}^N$.

2.8 Numerical experiments

In the experiments that follow, given the analytical description of the curve, we can measure the error in the approximation of the evolving area enclosed by the curve and its radius. Similarly to what is done in [8], the error in the area is measured as

$$||e_{N(A)}|| = \sqrt{\sum_{n=1}^{N_T} \left(A_N^n - A(n\Delta t)\right)^2 \Delta t},$$

where the area for a polygon is

$$A_N^n = \frac{1}{2} \sum_{i=1}^{N} \left(x_i^n y_{i+1}^n - x_{i+1}^n y_i^n\right).$$

The error for the radius is given as

$$||e_{N(r)}|| = \sqrt{\sum_{n=1}^{N_T} \Delta t \sum_{i=1}^{N} \left(r_i^n - r(n\Delta t)\right)^2 \frac{h_i^n + h_{i+1}^n}{2}},$$

where $r_i^n = ||\boldsymbol{x}_i^n||$ is the radius of node i at time step n, $r(n\Delta t)$ is the exact radius, and $h_i^n = ||\boldsymbol{x}_i^n - \boldsymbol{x}_{i-1}^n||$ is the arc-length between node $i-1$ and i.

2.8.1 Uniform arc-length experiments

We first consider the initial domain given by the circle centred at the origin

$$\begin{aligned} x(s,0) &= \cos(2\pi s), \qquad 0 \leq s \leq 1, \\ y(s,0) &= \sin(2\pi s), \end{aligned}$$

evolving according to the evolution equation $\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = -1$ up to the finial time $T = 0.95$. The exact solution of this model problem, at time t, is a circle centred on the origin of radius $r(t) = 1 - t$. The experiment is conducted with a uniform arc-length distribution of mesh nodes, $M = 1$, and the estimated orders of convergence using (2.43) are given in Table 2.9. The method produces results which are tending towards second-order accuracy in the area error norm as expected. The error results using the second-order temporal scheme (2.44) are shown in Table 2.10; they show that this scheme is tending towards second-order accuracy in time. Interestingly, the norms of the error in the radius are machine zero because in this particular case the scheme gives the exact solution for the radius. To see why this is the case, take

$$\boldsymbol{x}_i^n = (1 + \beta t^n) \left(\cos(2\pi\xi_i), \sin(2\pi\xi_i)\right)^T.$$

It follows that

$$\boldsymbol{n}_i^n = \left(\cos(2\pi\xi_i), \sin(2\pi\xi_i)\right)^T,$$

and therefore $\boldsymbol{x}_i^n \cdot \boldsymbol{n}_i^n = 1 + \beta t^n$ and

$$\boldsymbol{x}_i^{n+1} \cdot \boldsymbol{n}_i^n = 1 + \beta t^{n+1} = 1 + \beta t^n + \beta \Delta t.$$

It follows that

$$\boldsymbol{x}_i^{n+1} \cdot \boldsymbol{n}_i^n = \boldsymbol{x}_i^n + \beta \Delta t.$$

Notice that, when $\alpha = 0$, the normal components of (2.43) and (2.44) reduce simply to

$$\boldsymbol{x}_i^{[n+1,m+1]} \cdot \boldsymbol{n}_i^{[n,m]} = \boldsymbol{x}_i^{[n,0]} \cdot \boldsymbol{n}_i^{[n,m]} + \Delta t \beta.$$

Therefore, because the initial mesh is selected to be

$$\boldsymbol{x}_i^0 = \left(\cos(2\pi\xi_i), \sin(2\pi\xi_i)\right)^T,$$

it follows that the normal velocity is exact using (2.43) and (2.44).

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$
16	5.000e-03	3.561e-02		7.378e-16
32	1.250e-03	8.996e-03	1.985	1.868e-15
64	3.125e-04	2.255e-03	1.996	3.656e-15
128	7.813e-05	5.641e-04	1.999	8.644e-15
256	1.953e-05	1.410e-04	2.000	5.315e-14

Table 2.9: Estimated order of convergence and error results for uniform arc-length circle mesh shrinking at a constant rate using (2.43).

A second experiment is conducted whereby the domain evolves now according to mean curvature flow $\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = -\kappa$. If the circle has initial radius r_0, centred at the origin, the exact solution is a circle centred on the origin of radius $r(t) = \sqrt{r_0^2 - 2t}$.

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$
16	5.000e-03	3.561e-02		8.145e-16
32	2.500e-03	8.982e-03	1.987	1.259e-15
64	1.250e-03	2.252e-03	1.996	1.564e-15
128	6.250e-04	5.637e-04	1.998	3.034e-15
256	3.125e-04	1.410e-04	1.999	3.507e-15

Table 2.10: Estimated order of convergence and error results for uniform arc-length circle mesh shrinking at a constant rate using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.9 showing that the estimated temporal convergence rate for (2.44) is second order.

We take $r_0 = 1.67332005307$ so that $r(T) = 1$ with $T = 0.9$. The estimated orders of convergence for a uniform arc-length mesh evolved with (2.43) and (2.44) are given in Tables 2.11 and 2.12, respectively. The results demonstrate MMPDE1D converging with second-order accuracy in both area and radius error norms now. Further, Table 2.12 shows the scheme (2.44) tending towards second-order accuracy in time.

The final experiment of this section uses the initial ellipse domain given by

$$\begin{aligned} x(s,0) &= 2\cos(2\pi s), \qquad 0 \le s \le 1, \\ y(s,0) &= \sin(2\pi s). \end{aligned} \tag{2.45}$$

The domain evolves according to $\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = -\kappa$ to the final time $T = 0.9$. Unfortunately it is not clear what the extreme radius values are for the ellipse case as it evolves, so the $||e_{N(r)}||$ error norm will not be used here. Instead we make use of the fact that, from the divergence theorem, it can be shown that any simple closed curve which shrinks according to mean curvature has exact area given by

$$A(t) = A(0) - 2\pi t,$$

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	2.534e-01		1.773e-02	
32	1.250e-03	6.330e-02	2.001	4.242e-03	2.063
64	3.125e-04	1.582e-02	2.000	1.049e-03	2.015
128	7.813e-05	3.955e-03	2.000	2.616e-04	2.004
256	1.953e-05	9.887e-04	2.000	6.535e-05	2.001

Table 2.11: Estimated order of convergence and error results for uniform arc-length circle mesh shrinking according to mean curvature using (2.43).

which allows us to calculate $||e_{N(A)}||$ for the ellipse domain. The results of the estimated order of convergence for $||e_{N(A)}||$ are given in Table 2.13 and 2.14 and again show second-order spatial accuracy as well as second-order temporal accuracy for (2.44).

2.8.2 Experiments with mesh adaption

We now consider some experiments using a non-constant monitor function. We introduce an arbitrary adaption for the evolving meshes based on the time dependent monitor function

$$M(x, y, t) = 1 + \operatorname{sech}((x + 4t - 2.5)/0.2). \tag{2.46}$$

This monitor function should refine mesh points around the $x = 2.5 - 4t$ axis.

The first experiment from the uniform arc-length section, whereby the initially unit circle centred at the origin shrinks according to $\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = -1$ is now repeated using the monitor function (2.46). The results of the estimated order of convergence for the first-order temporal scheme are shown in Table 2.15. Furthermore, the results of the second-order temporal scheme are shown in Table 2.16. Both tables demonstrate that even with a non uniform arc-length distribution of nodes, MMPDE1D continues to

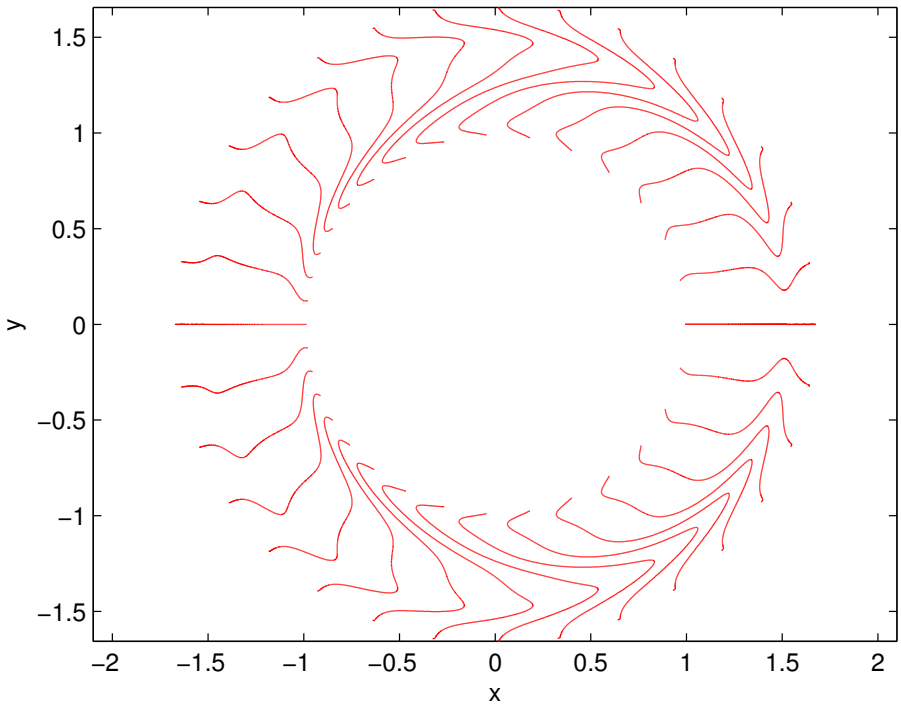


Figure 2.2: Adaptive mesh trajectories for an initial circle according to the evolution law $\dot{\boldsymbol{x}} \cdot \boldsymbol{n} = -\kappa$. Mesh has $N = 32$ points and the monitor function used is (2.46).

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	2.498e-01		1.711e-02	
32	2.500e-03	6.251e-02	1.998	4.123e-03	2.053
64	1.250e-03	1.562e-02	2.000	1.019e-03	2.016
128	6.250e-04	3.904e-03	2.000	2.538e-04	2.006
256	3.125e-04	9.758e-04	2.000	6.335e-05	2.002

Table 2.12: Estimated order of convergence and error results for uniform arc-length circle mesh shrinking according to mean curvature using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.11 showing that the estimated temporal convergence rate for (2.44) is second order.

converge with second-order accuracy in the area and radius error norms.

Likewise, the second experiment whereby an initial circle evolves according to mean curvature now with the new monitor function (2.46) shows a similar result in the estimated order of convergence. With the temporally first-order scheme, the second-order spatial accuracy is verified by Table 2.17. The second-order temporal scheme shows similar results and confirms second-order accuracy in time; results are given in Table 2.18. The trajectories of each mesh point can be seen in Figure 2.2. It is easy to trace out the adaption wave-front in the figure; as the circle shrinks, the wave-front gradually moves from right to left through the domain. This can be seen in the plot of the normalised arc-length, Figure 2.3, by the clustering together of the nodes as the domain is evolved.

The evolving ellipse experiment (2.45) is reproduced using the adapting monitor function (2.46). The results, are given in Table 2.19, also indicate second-order spatial convergence. Results for the temporally second-order scheme are presented in Table 2.20. The mesh generated for the evolving ellipse domain is shown in Figure 2.4 at times $t = 0.31$ and $t = 0.62$. The black line shows the position of the adapting wave-front moving through the domain around which the mesh points are clustered. The node

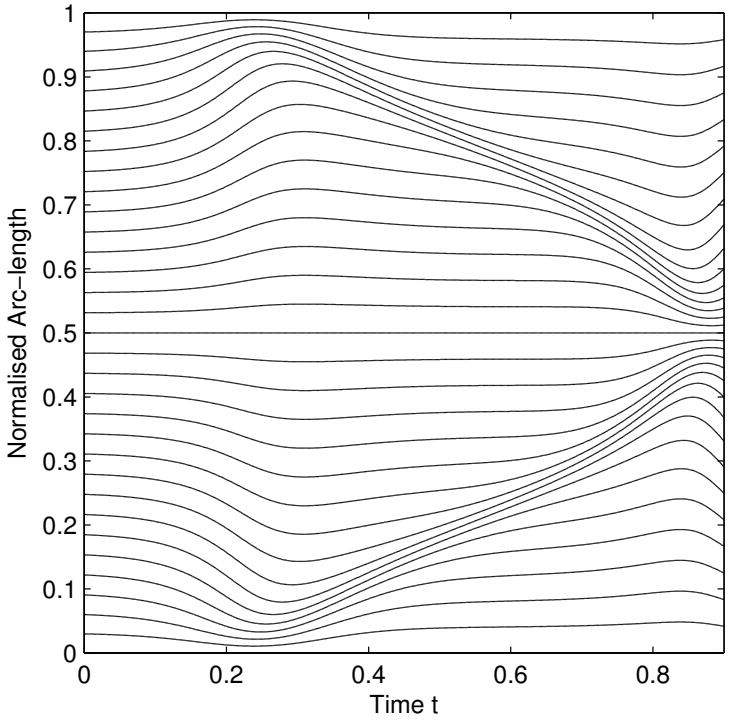


Figure 2.3: Evolution of the normalised arc-length with respect to time as adapted wave-front advances through shrinking circle domain. The trajectories here show how the mesh points move on the circular boundary to contract and relax as the simulation progresses. The mesh points start uniformly spaced at $t = 0$ and bunching can be seen as the wave begins to pass through the interface from $0.2 < t < 0.8$ where it exits and the mesh begins to relax again as witnessed by the mesh points beginning to distribute uniformly for $t > 0.8$.

N	Δt	$\|\|e_{N(A)}\|\|$	EOC
16	5.000e-03	2.388e-01	
32	1.250e-03	6.266e-02	1.930
64	3.125e-04	1.597e-02	1.972
128	7.813e-05	4.019e-03	1.991
256	1.953e-05	1.007e-03	1.997

Table 2.13: Estimated order of convergence and error results for uniform arc-length ellipse mesh shrinking by mean curvature using (2.43).

clustering is more apparent in Figure 2.5, where the normalised arc-length trajectories of the nodes can be seen evolving with respect to time. As the wave-front passes through, the trajectories bunch closer together.

Note that the value of $||M\boldsymbol{x}_\xi||_\xi$ approaches zero as the accuracy of the numerical mesh in approximating the given adaptivity constraint imposed by the monitor function M improves. Therefore, to check how well the moving mesh method equidistributes the monitor function, Figure 2.6 shows a plot of

$$E(t^n) = \max_{1\leq i\leq N} \left| \frac{M_{i+1}^n + M_i^n}{2N^2} ||x_{i+1}^n - x_i^n|| - \frac{M_i^n + M_{i-1}^n}{2N^2} ||x_i^n - x_{i-1}^n|| \right|, \tag{2.47}$$

which is a measure of $||M\boldsymbol{x}_\xi||_\xi$. It is interesting to note that there are oscillations in $E(t)$ as the wave-front passes through. It may be possible to improve the results here further by decreasing the temporal smoothing parameter τ, but this has the potential drawback of decreasing the stability of the method.

For our purposes though, we have demonstrated that MMPDE1D does a good job of not only accurately approximating the position of simple evolving domains but also accurately allowing grid points to relocate along the tangent to the domain according an arbitrary monitor function.

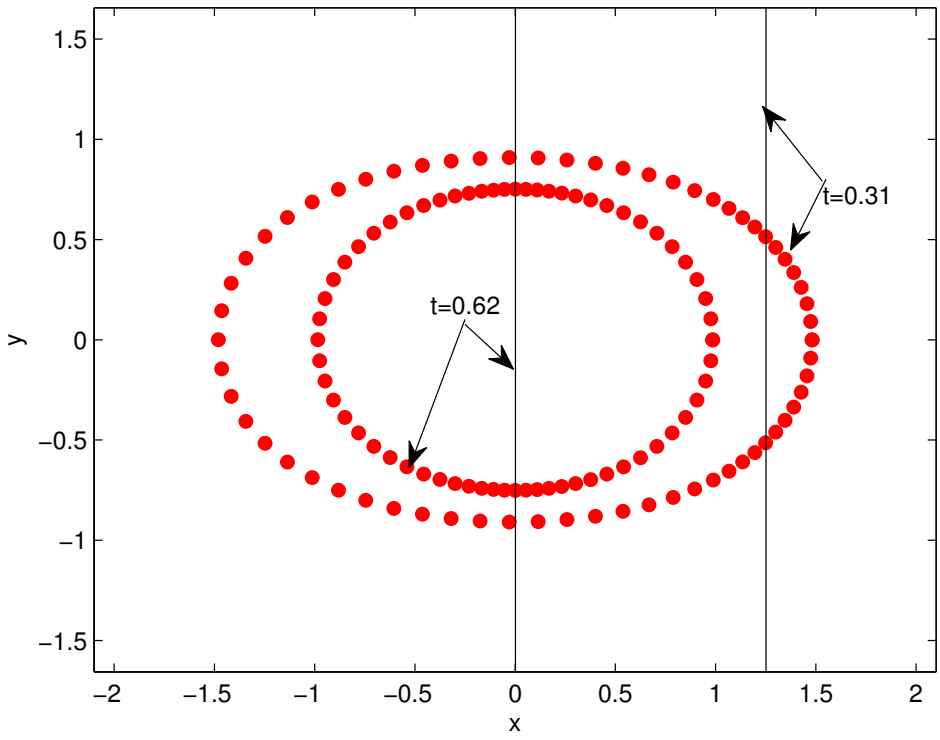


Figure 2.4: The ellipse mesh being used at $t = 0.31$ and $t = 0.62$; black lines show where the peak in M is for given time.

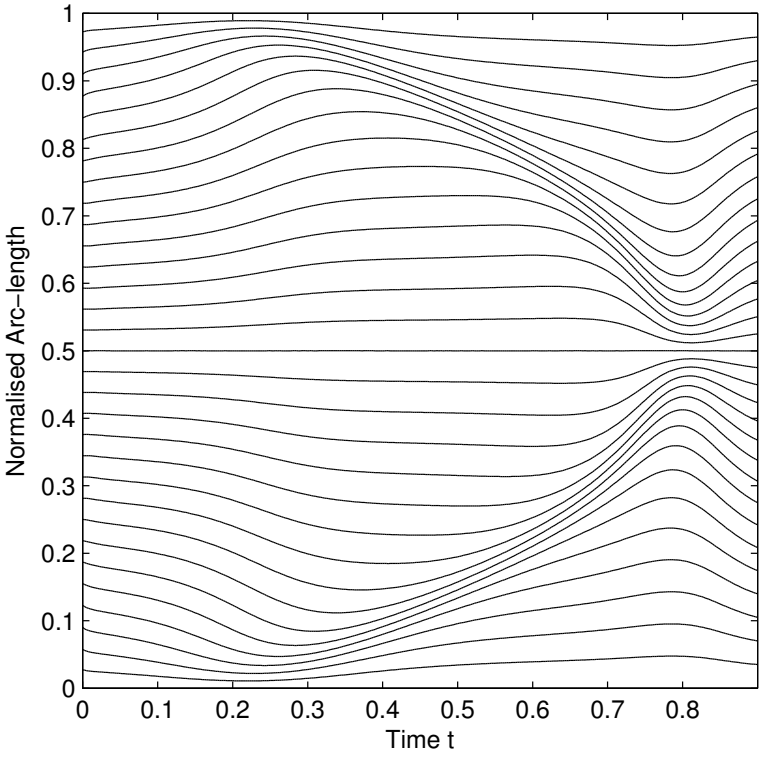


Figure 2.5: Normalised arc-length changing with respect to time as adapted wave-front advances through shrinking ellipse domain.

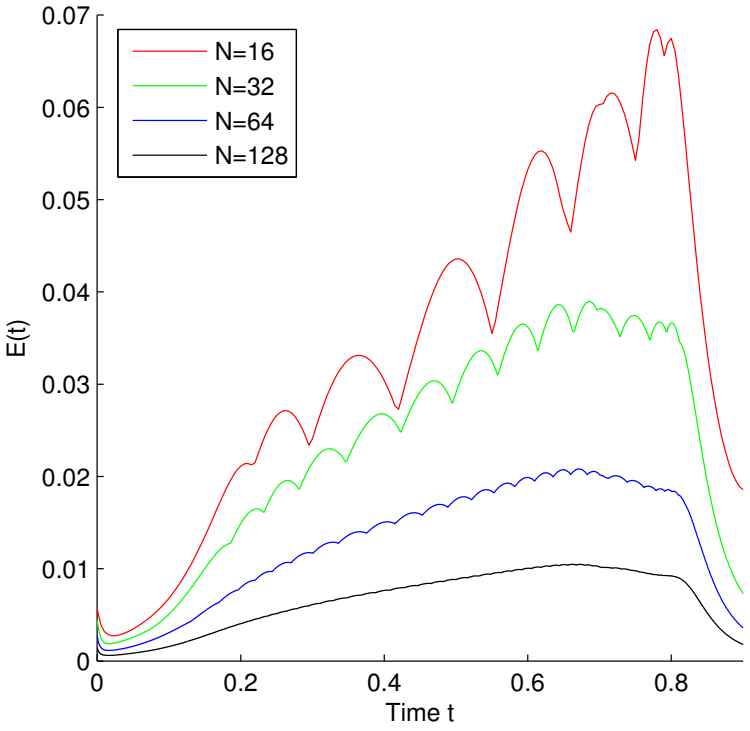


Figure 2.6: Plot of $E(t)$, a measure of how well the constraint $||M\boldsymbol{x}_\xi||_\xi = 0$ is satisfied, on the evolving ellipse domains with an adapting wave front progressing through the evolving domains using (2.43). If $||M\boldsymbol{x}_\xi|| = 0$ then the mesh points will equidistribute the monitor function (2.46) exactly.

N	Δt	$\|\|e_{N(A)}\|\|$	EOC
16	5.000e-03	2.476e-01	
32	2.500e-03	6.340e-02	1.965
64	1.250e-03	1.594e-02	1.992
128	6.250e-04	3.990e-03	1.998
256	3.125e-04	9.977e-04	2.000

Table 2.14: Estimated order of convergence and error results for uniform arc-length ellipse mesh shrinking by mean curvature using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.13 showing that the estimated temporal convergence rate for (2.44) is second order.

2.8.3 Geometry-based adaption

As well as being able to adapt the mesh to areas of numerical difficulty in relation to the underlying PDEs being solved on the domain, the use of an adaptive moving mesh approach presents us with the opportunity to redistribute mesh points so that the domain geometry is well described. This may assist greatly in problems where the domain evolves in time in terms of a geometric evolution law and where the domain has localised regions of high curvature.

In the absence of a reliable error estimate for the approximation $\Gamma_h(t)$ of an evolving curve $\Gamma(t)$, we will base our analysis of a suitable monitor function on a study of interpolation error. Therefore, the aim is to find a monitor function which, when equidistributed, results in a distribution of mesh points that minimises an appropriate measure of the difference between a smooth curve Γ and its linear polygonal interpolant Γ_h. Here we focus on the minimisation of the maximal distance between Γ and Γ_h. In Figure 2.7 we show the segment Γ_i of Γ with end points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$. Also shown is the linear approximation $\Gamma_{h,i}$ of Γ_i. For each $\boldsymbol{x} \in \Gamma_i$, we define the distance, $d(\boldsymbol{x})$, from $\boldsymbol{x}$ to $\Gamma_{h,i}$ as the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_* \in \Gamma_{h,i}$, where the line through $\boldsymbol{x}$ and $\boldsymbol{x}_*$ is

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	3.582e-02		1.926e-03	
32	1.250e-03	9.059e-03	1.983	5.468e-04	1.816
64	3.125e-04	2.271e-03	1.996	1.411e-04	1.954
128	7.813e-05	5.683e-04	1.999	3.554e-05	1.989
256	1.953e-05	1.421e-04	2.000	8.899e-06	1.998

Table 2.15: Estimated order of convergence and error results for non-uniform circle mesh shrinking at a constant rate using (2.43).

perpendicular to $\Gamma_{h,i}$. To simplify the analysis, we note that the distance between Γ_i and $\Gamma_{h,i}$ is invariant to a coordinate rotation and translation. We therefore translate coordinates so that $\boldsymbol{x}_i$ maps to the origin and rotate coordinates such that the line segment between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ is parallel to the positive $\bar{x}$ axis, as shown in Fig. 2.7. Finding the maximal distance between Γ_i and $\Gamma_{h,i}$ is therefore equivalent to finding the maximum absolute value of the transformed graph $\bar{\Gamma}(\bar{x})$ for $0 \leq \bar{x} \leq |\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|$, and this can be estimated using a standard argument from linear interpolation theory.

Without loss of generality, let's assume that the maximum of $\bar{\Gamma}$ occurs at $\bar{x}_*$ and assume $\bar{x}_*$ is closer to $\bar{x} = 0$ than $\bar{x} = h_i \equiv |\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|$. Using a Taylor series expansion of $\bar{\Gamma}$ about $\bar{x} = 0$, and noting that $\bar{\Gamma}(0) = 0$ and $\bar{\Gamma}'(\bar{x}_*) = 0$, we have

$$\bar{\Gamma}(\bar{x}_*) = \frac{\bar{x}_*^2}{2}\bar{\Gamma}''(\bar{x}_*) + O(\bar{x}_*)^3.$$

The curvature κ of $\bar{\Gamma}$ is

$$|\bar{\kappa}| = \frac{|\bar{\Gamma}''|}{(1 + (\bar{\Gamma}')^2)^{3/2}},$$

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	3.553e-02		1.912e-03	
32	2.500e-03	8.959e-03	1.988	5.419e-04	1.819
64	1.250e-03	2.246e-03	1.996	1.398e-04	1.954
128	6.250e-04	5.622e-04	1.998	3.524e-05	1.989
256	3.125e-04	1.406e-04	1.999	8.828e-06	1.997

Table 2.16: Estimated order of convergence and error results for non-uniform circle mesh shrinking at a constant rate using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.15 showing that the estimated temporal convergence rate for (2.44) is second order.

and since $\bar{\Gamma}'(\bar{x}_*) = 0$, it follows that $|\bar{\Gamma}''(\bar{x}_*)| = |\bar{\kappa}(\bar{x}_*)|$. Therefore, we find that

$$\begin{aligned} \max_{\bar{x}\in(0,h_i)} |\bar{\Gamma}(\bar{x})| &= \frac{\bar{x}_*^2}{2}|\bar{\Gamma}''(\bar{x}_*)| + O(\bar{x}_*)^3 \\ &\leq \frac{h_i^2}{8}|\bar{\kappa}(\bar{x}_*)| + O(h_i)^3. \end{aligned} \tag{2.48}$$

The curvature of Γ_i is clearly invariant to the translation and rotation mapping above and hence an approximately optimal distribution of mesh points $\{\boldsymbol{x}_i\}_{i=1}^N$, which minimises the maximal error over all segments, is obtained when

$$h_i^2|\kappa_i| = h_{i+1}^2|\kappa_{i+1}|, \qquad i = 1, \ldots, N-1, \tag{2.49}$$

where $\kappa_i = \max_{\boldsymbol{x}\in\Gamma_i} |\kappa(\boldsymbol{x})|$. It therefore follows that the quantity $h_i|\kappa_i|^{1/2}$ is constant in each segment and this suggests that a suitable monitor function for curve approximation should be based on equidistribution of $|\kappa|^{1/2}$.

Alternative monitor functions can be derived to minimise different error measures between Γ and Γ_h. For example, it has been shown that equidistribution of $|\kappa|^{1/3}$ leads to an interpolatory linear polygonal curve that minimises the discrepancy in the

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	2.583e-01		1.761e-02	
32	1.250e-03	6.408e-02	2.011	4.100e-03	2.103
64	3.125e-04	1.616e-02	1.988	1.018e-03	2.010
128	7.813e-05	4.186e-03	1.948	2.653e-04	1.940
256	1.953e-05	1.111e-03	1.913	7.194e-05	1.883

Table 2.17: Estimated order of convergence and error results for non-uniform circle mesh shrinking according to mean curvature using (2.43).

enclosed area of the closed curves Γ and Γ_h. Equidistribution of $|\kappa|^{2/3}$ minimises the total length discrepancy [128].

Since κ can potentially be zero along a curve, it is important to include a positive floor on the monitor function to ensure that no areas of the curve become starved of mesh points and this is clearly essential if a PDE is being approximated on the curve using the same mesh. A simple curvature-based monitor function therefore takes the form

$$M = M_{\text{floor}} + |\kappa|^{1/2}, \tag{2.50}$$

where M_{floor} is a positive parameter which we can use to modulate the intensity of the $|\kappa|^{1/2}$ adaption. For example, we consider the monitor function

$$M = 1 + |\kappa|^{1/2} \tag{2.51}$$

in results which follow. For evolutionary curves however, it may be difficult to a priori choose an appropriate value for M_{floor}. We will therefore consider the time-dependent

N	Δt	$\|\|e_{N(A)}\|\|$	EOC	$\|\|e_{N(r)}\|\|$	EOC
16	5.000e-03	2.416e-01		1.543e-02	
32	2.500e-03	5.947e-02	2.022	3.590e-03	2.104
64	1.250e-03	1.477e-02	2.010	8.757e-04	2.035
128	6.250e-04	3.684e-03	2.003	2.173e-04	2.011
256	3.125e-04	9.204e-04	2.001	5.419e-05	2.003

Table 2.18: Estimated order of convergence and error results for non-uniform circle mesh shrinking according to mean curvature using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.17 showing that the estimated temporal convergence rate for (2.44) is second order.

floor

$$M_{\text{floor}}(t) = \frac{1}{|\Gamma(t)|}\int_{\Gamma(t)} |\kappa|^{1/2} d\eta. \tag{2.52}$$

A similar floor on the monitor function has been used to great effect in [14] and [15] for the adaptive solution of PDEs in one dimension. A major potential advantage of the floor (2.52) is that it does not require any a priori choice of parameters and adapts to the length of the evolving curve.

To implement the monitor functions (2.51) and (2.50) in the MMPDE1D method (2.34) requires the evaluation of the monitor function at (ξ_i, t^n). To this end, we use a central difference approximation of the curvature given by

$$\kappa_i^n = \frac{(\boldsymbol{x}_{i-1}^n - 2\boldsymbol{x}_i^n + \boldsymbol{x}_{i+1}^n)\cdot \boldsymbol{n}_i^n}{||\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n||^2/4}, \tag{2.53}$$

and $\boldsymbol{n}_i^n$ given by

$$\boldsymbol{n}_i^n = \left(-\frac{y_{i+1}^n - y_{i-1}^n}{||\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n||}, \frac{x_{i+1}^n - x_{i-1}^n}{||\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n||}\right), \tag{2.54}$$

N	Δt	$\|\|e_{N(A)}\|\|$	EOC
16	5.000e-03	2.315e-01	
32	1.250e-03	5.942e-02	1.962
64	3.125e-04	1.502e-02	1.984
128	7.813e-05	3.769e-03	1.994
256	1.953e-05	9.437e-04	1.998

Table 2.19: Estimated order of convergence and error results for non-uniform ellipse mesh shrinking according to mean curvature using (2.43).

where

$$S_i^n = ||\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_i^n||.$$

For the time dependent floor (2.52) we use a simple quadrature approximation

$$M_i^n = \frac{1}{|\Gamma_h(t^n)|} \sum_{j=1}^{N} \left(\frac{|\kappa_{j+1}^n|^{1/2} + |\kappa_j^n|^{1/2}}{2} \right) S_j^n + |\kappa_i^n|^{1/2},$$

where

$$|\Gamma_h(t^n)| = \sum_{j=1}^{N} S_j^n.$$

To enhance the robustness of the adaptive grid procedure, the monitor function can be smoothed using a spatial averaging technique [**?**, 65, 118]

$$\tilde{M}_{i+1/2} = \frac{\sum_{k=i-p}^{i+p} M_{k+1/2}(q/(q+1))^{|k-i|}}{\sum_{k=i-p}^{i+p}(q/(q+1))^{|k-i|}}, \tag{2.55}$$

where q is a positive real number and p is a non-negative integer. For the simulations presented next we fix $p = 2$ and $q = 3$.

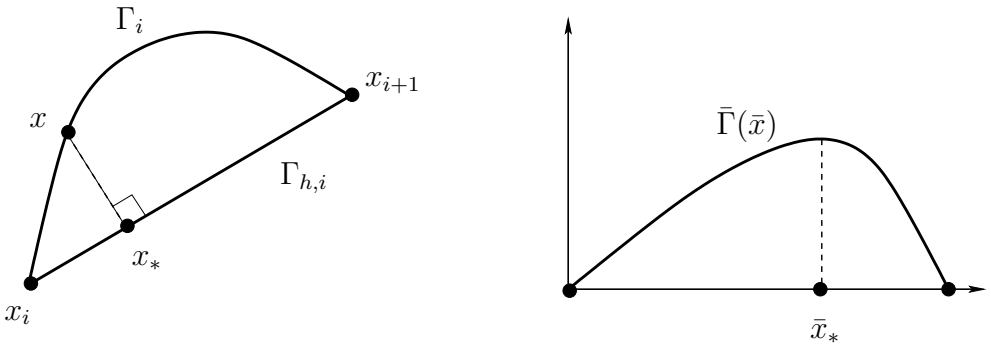


Figure 2.7: (Left) Segment of a smooth curve Γ_i and interpolating linear approximation $\Gamma_{h,i}$ between mesh points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$. The distance between the curves at point $\boldsymbol{x}$ is the distance from $\boldsymbol{x}$ to $\boldsymbol{x}_*$. (Right) Translated and rotated segment is transformed to the graph $\bar{\Gamma}(\bar{x})$. The maximal distance between Γ_i and $\Gamma_{h,i}$ is equal to the absolute maximum value $\bar{\Gamma}(\bar{x}_*)$.

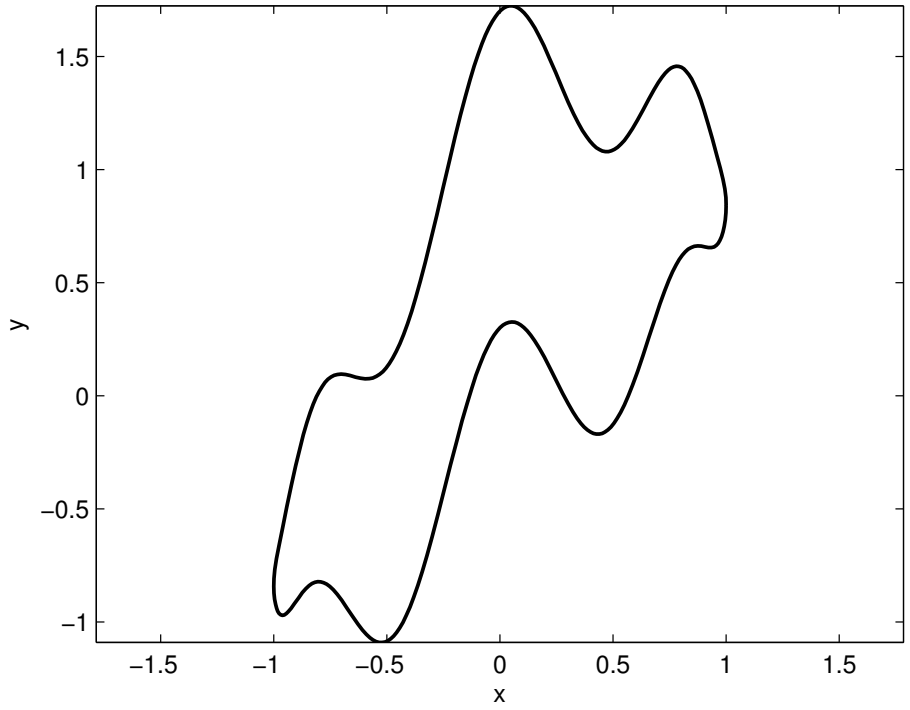


Figure 2.8: The non-convex initial domain used to test geometry-based mesh adaptation.

N	Δt	$\|\|e_{N(A)}\|\|$	EOC
16	5.000e-03	2.377e-01	
32	2.500e-03	5.975e-02	1.992
64	1.250e-03	1.493e-02	2.001
128	6.250e-04	3.745e-03	1.995
256	3.125e-04	9.497e-04	1.979

Table 2.20: Estimated order of convergence and error results for non-uniform ellipse mesh shrinking according to mean curvature using (2.44). Here the time-steps are reduced only by a factor of two in comparison with Table 2.19 showing that the estimated temporal convergence rate for (2.44) is second order.

To compare the performance of the uniform arc-length monitor function $M = 1$, and the curvature-based monitor functions (2.51) and (2.50), we consider mean-curvature flow ($\alpha = 1$, $\beta = 0$ in (2.33)) of the non-convex initial curve

$$\begin{aligned} &x(s,0) = \cos(2\pi s), \qquad (0 \le s \le 1), \\ &y(s,0) = 0.5\sin(2\pi s) + \sin(\cos(2\pi s)) + \sin(2\pi s)\left(0.2 + \sin(2\pi s)\sin^2(6\pi s)\right), \end{aligned} \tag{2.56}$$

which is shown in Figure 2.8. This example has also been used in [96] and [128] to test alternative tangentially stabilised curve evolution algorithms.

The mesh obtained using the uniform partition $s_i = i/N$, $i = 0, \ldots, N$ is shown in Figure 2.9. We can see that the distribution of points is far from ideal in representing the initial curve with some areas of high curvature not having sufficient resolution. The initial meshes produced using the de Boor algorithm and the monitor functions $M = 1$ and $M = M_{\text{floor}} + |\kappa|^{1/2}$ are also shown in Figure 2.9. We can see that the uniform arc-length mesh obtained using $M = 1$ is also poor at resolving the areas of high curvature. The best initial mesh is obtained using the curvature-based monitor function (2.50), and we can observe a nice balance of mesh points towards high-curvature regions and

areas of low curvature. The mesh trajectories using each of the three monitor functions are shown in Figure 2.10. We can see that, in all cases, the grid trajectories evolve smoothly in time, and those obtained with the curvature-based monitors are clustered towards areas of high curvature. This can be seen more easily in Figure 2.11, where the grid nodes obtained with $M = M_{\text{floor}} + |\kappa|^{1/2}$ at various time intervals are compared to a fine grid solution. To plotting accuracy we can see very good agreement. Area error results and estimated orders of convergence using the three monitor functions are given in Table 2.21. We can see that we achieve second-order convergence using all three monitor functions. The same data is plotted in Figure 2.12 and it is clear that the best monitor function from the three is (2.50). The evolution of the area error using the three monitor functions is shown in Figure 2.13. Again we can see that the improvement in the resolution of the initial curve is maintained over time using the curvature-based monitor functions and that (2.50) results in a lower overall error. All of the simulations of the initial meshes are evolved using MMPDE1D using the three monitor functions considered with temporal smoothing parameter $\tau = 1$ to account for the greater difficulty evolving this geometry than the ellipse and circle cases.

		$M = 1$		$M = 1 + \sqrt{\lvert\kappa\rvert}$		$M = M_{\text{floor}} + \sqrt{\lvert\kappa\rvert}$	
N	Δt	$\lvert\lvert e_{N(A)} \rvert\rvert$	EOC	$\lvert\lvert e_{N(A)} \rvert\rvert$	EOC	$\lvert\lvert e_{N(A)} \rvert\rvert$	EOC
16	1.000e-04	8.497e-02		8.369e-02		8.204e-02	
32	5.000e-05	4.397e-02	0.951	4.267e-02	0.972	4.113e-02	0.996
64	2.500e-05	1.301e-02	1.757	1.227e-02	1.798	1.140e-02	1.851
128	1.250e-05	2.932e-03	2.150	2.481e-03	2.306	2.045e-03	2.479
256	6.250e-06	7.629e-04	1.942	6.154e-04	2.011	4.653e-04	2.136

Table 2.21: Error in the approximated area and estimated orders of convergence for a non-convex initial geometry (2.56), evolved according to mean curvature flow using MMPDE1D with three different monitor functions.

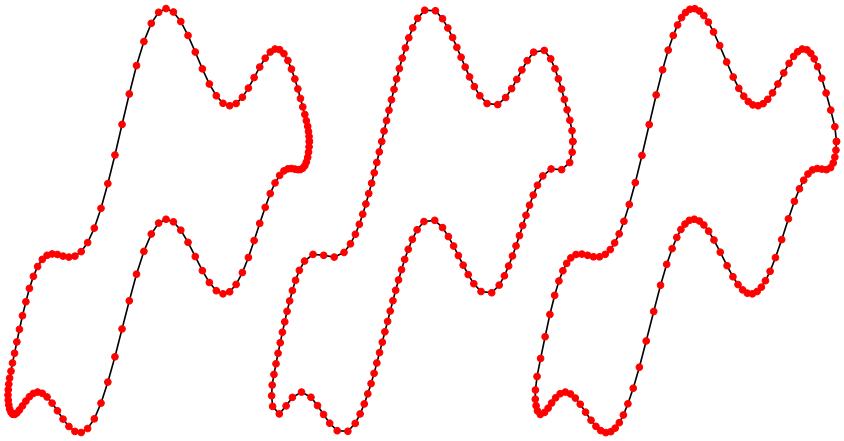


Figure 2.9: Initial meshes for the non-convex curve (2.56) using $N = 128$ points. On the left is the mesh generated by the uniform partitioning of the parameter s. In the middle is the mesh generated by uniformly partitioning the arc-length. On the right, the mesh is generated using the adapting monitor function (2.50) with (2.52).

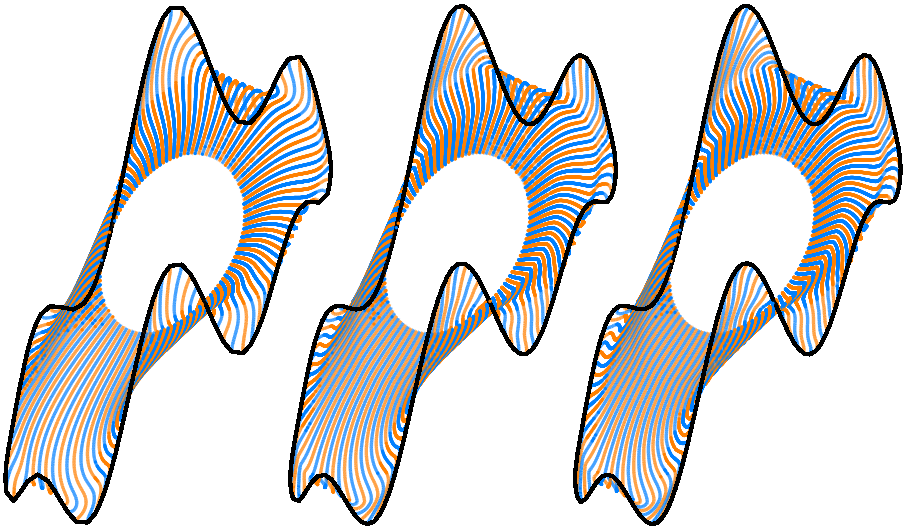
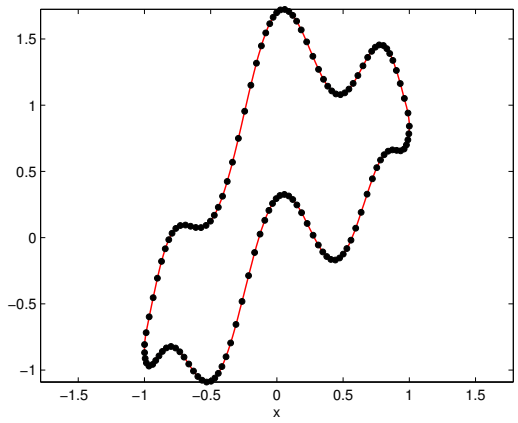
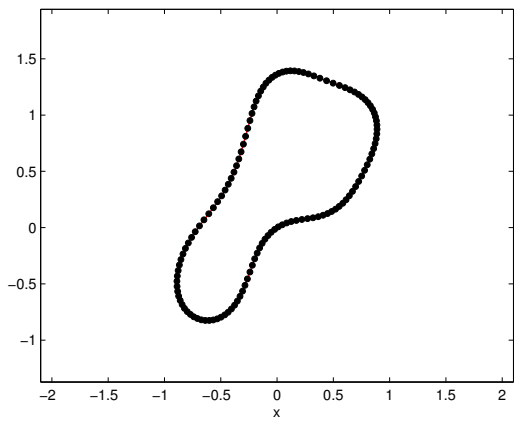


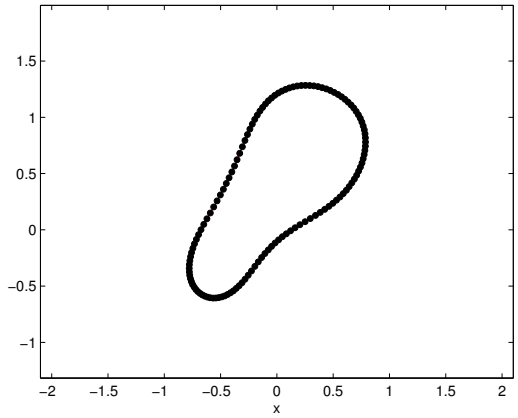
Figure 2.10: Mesh trajectories for mean curvature flow of a non-convex initial curve with monitor function $M = 1$ on the left, $M = 1 + |\kappa|^{1/2}$ in the middle and $M = M_{\text{floor}} + |\kappa|^{1/2}$ on the right. ($N = 128$, $\Delta t = 10^{-4}$)



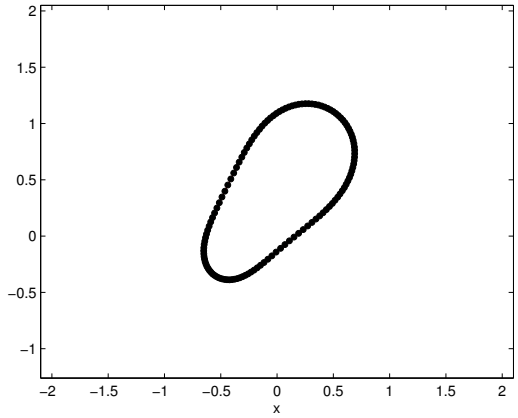
(a) $t = 0$



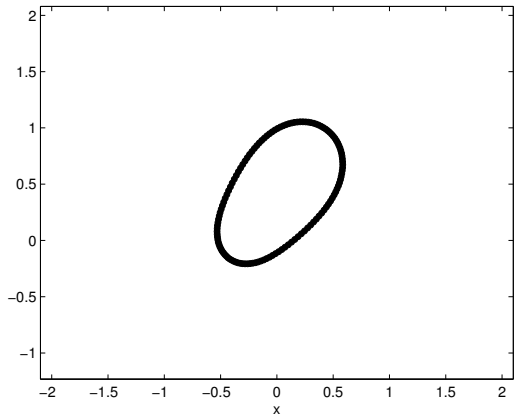
(b) $t = 0.05$



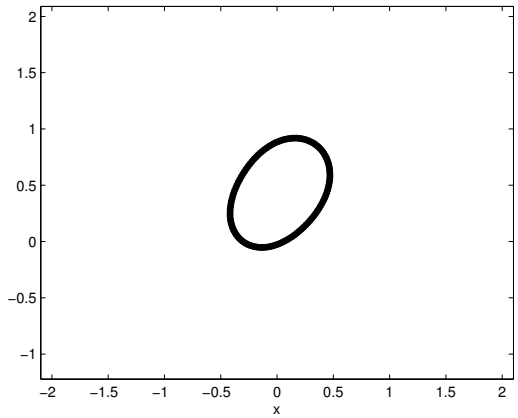
(c) $t = 0.1$



(d) $t = 0.15$



(e) $t = 0.2$



(f) $t = 0.25$

Figure 2.11: Snapshots of non-convex initial domain evolving by mean curvature flow at various times. Red line plots fine mesh solution ($N = 512$, $\Delta t = 1.9525e - 7$). Nodes for coarse solution ($N = 128$, $\Delta t = 1.562e - 06$) are shown in black. The monitor function $M = M_{\text{floor}} + |\kappa|^{1/2}$ is used. Mesh nodes cluster around areas of high curvature initially and gradually become more equidistributed as the domain, and hence curvature, becomes smoother.

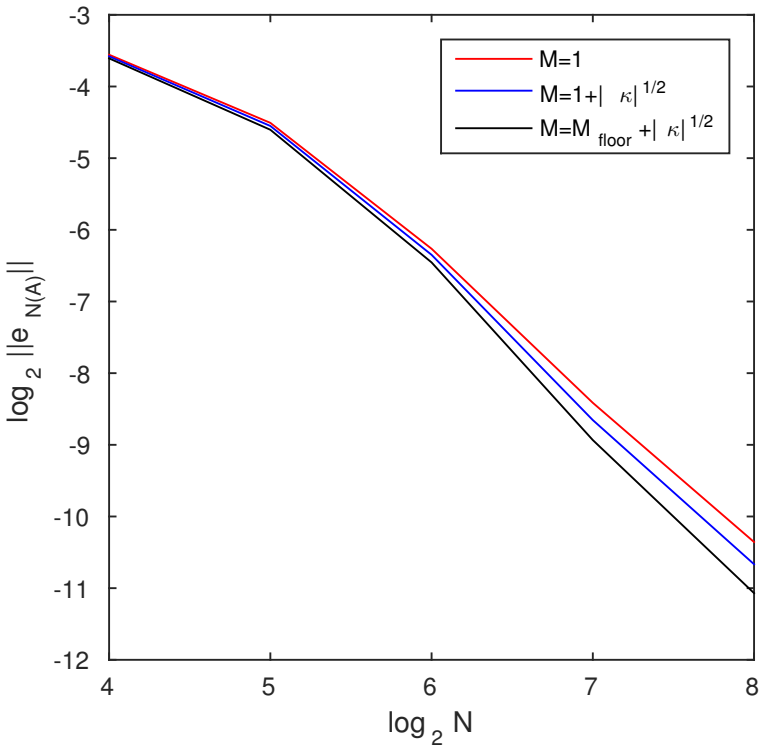


Figure 2.12: The error in the approximate area decreases with respect to the increasing number of nodes at a rate of second-order accuracy. The approximations generated with adaptivity (blue and black) have smaller error than the uniform arc-length case (red).

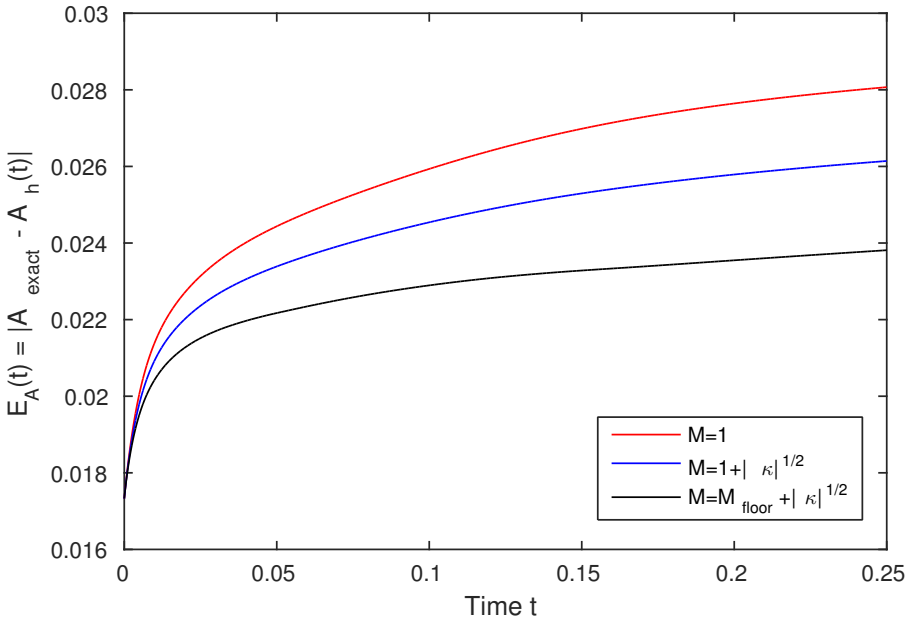


Figure 2.13: The error between the approximate area and the exact area of the non-convex initial curve (2.56) evolved by mean curvature flow; mesh points adapted by monitor functions $M = 1$ (red), $M = 1 + |\kappa|^{1/2}$ (blue) and $M = M_{\text{floor}} + |\kappa|^{1/2}$ (black). Here $\tau = 1$, $N = 64$ and $\Delta t = 2.5e - 5$.

2.9 Conclusions

In conclusion:

- This chapter has introduced an ALEFEM scheme which can be used to solve reaction-diffusion equations on evolving curves with numerical experiments suggesting that the method is second-order accurate. This method can now be used to conduct simulations with the cell model introduced in Section 1.2.2. This will be the subject of study at in the following chapter.

- A modification of the parameterised form of the governing equations of the scheme was shown to be necessary to ensure a conservative numerical approximation. An example was presented showing this conservation property. However a proof has not yet been formalised to show that the ALEFEM scheme is conserved for every case.

- A novel adaptive moving mesh PDE method for numerically evolving a curve under forced mean-curvature flow has been introduced. This method features a novel approach to control mesh points movement in the tangential direction, while simultaneously evolving a curve with the correct normal velocity making it highly desirable in applications where a high level of control over the mesh is warranted; for example as boundary conditions in higher dimension mesh generation schemes as we shall use it for in later chapters.

- A novel curvature-based monitor function was developed to drive tangential mesh movement to resolve high curvature regions for evolving curves. This new monitor function has been shown to dramatically improve solution accuracy compared to a uniform arc-length mesh in the examples presented.

The methods introduced here are not restricted to cell migration and chemotaxis models and can in fact be used in a great variety of research where there is a need to calculate some form of reaction-diffusion on an evolving curve domain or a need to

resolve a numerical curve mesh with adaption. In the next Chapter however, the work presented here will be looked at in the cell migration and chemotaxis context which is the original purpose for the development of these methods.

Chapter 3

Cell migration using a reaction-diffusion model on an evolving curve

3.1 A modified Meinhardt reaction-diffusion system for cell migration

Cell migration can be modelled using reaction-diffusion equations and a model of the mechanical mechanism which links chemical concentrations to membrane growth. The three equation reaction-diffusion model of Meinhardt [94] is selected for this purpose because the behaviour of the solutions of the system has desirable cell-like characteristics. For instance, travelling waves and out of phase oscillations of chemical concentrations are known to happen along cell membranes and in the solutions to the Meinhardt system [72]. Also, the solution is very sensitive to perturbations of nutrients in the environment. This means that, unlike many other reaction-diffusion systems, the solution for this model does not go towards a steady state solution easily. It is clearly apparent that this is a desirable characteristic for cells wishing to migrate.

The model posed by Meinhardt was augmented by Neilson et al. [105] to be spatially continuous with non-zero diffusive terms. This augmented model is calculated on a time-dependent evolving simple closed curve $\Gamma(t)$, which henceforth represents the

membrane domain. A non-dimensional mathematical description of the model is

$$\begin{aligned}
\dot{a} + \nabla_{\Gamma(t)} \cdot (a\boldsymbol{u}) &= D_a \Delta_{\Gamma(t)} a + \frac{(a^2/b + b_a)\varrho}{(s_c + c)(1 + s_a a^2)} - r_a a, \\
\dot{b} + \nabla_{\Gamma(t)} \cdot (b\boldsymbol{u}) &= D_b \Delta_{\Gamma(t)} b - r_b b + \frac{r_b}{|\Gamma(t)|} \oint_{\Gamma(t)} a d\boldsymbol{x}, \\
\dot{c} + \nabla_{\Gamma(t)} \cdot (c\boldsymbol{u}) &= D_c \Delta_{\Gamma(t)} c + b_c a - r_c c,
\end{aligned} \tag{3.1}$$

where a, b and c are the concentrations of the activator, global inhibitor and local inhibitor, respectively. The diffusivities of each concentration are D_a, D_b and D_c, where $D_a < D_c < D_b$, $\Delta_{\Gamma(t)}$ is the diffusion operator along the interface $\Gamma(t)$ defined in (1.9) and $\nabla_{\Gamma(t)}$ is the tangential gradient operator given in (1.10).

The position of the interface is given by $\boldsymbol{x}$ with the material velocity being $\boldsymbol{u}$. The signal term, which is related to the concentration of nutrients that the cell can sense in its environment, is ϱ. The $\nabla_{\Gamma(t)} \cdot \boldsymbol{u}$ term represents a focusing/dilution of the concentration brought about by the local changes in the geometry of $\Gamma(t)$. The constants b_a and b_c are the basal rate of production of a and c, while r_a, r_b and r_c are the corresponding linear decay rates. The last two constants s_a and s_c control the saturation of the activator with respect to a and c. The parameters used in the simulations which follow are based on values taken from [94, 105] and are given in Table 3.1.

The signal term is a measure of fractional receptor occupancy. Receptor occupancy is found by a binding-unbinding process of the ligand concentration on the cell exterior to the receptors. The receptors are assumed to be able to sense a gradient of chemoattractant surrounding the membrane. This gradient is obtained through a diffusive chemoattractant source exterior to the cell which we will represent as

$$\frac{\partial L}{\partial t}\Big|_{\boldsymbol{x}} = D_L \Delta_{\Omega(t)} L, \tag{3.2}$$

$$-D_L \frac{\partial L}{\partial \boldsymbol{n}}\Big|_{\Gamma(t)} - [(\boldsymbol{u} \cdot \boldsymbol{n})] L\Big|_{\Gamma(t)} = k_1 \left(N_r - L_m\right) L|_{\Gamma(t)} - k_{-1} L_m, \tag{3.3}$$

where L is the free-ligand concentration and D_L is the diffusivity of the chemoattractant in the cell's environment and $\Omega(t)$ is the domain exterior to the cell. The

Quantity	Symbol	Value
Decay rate of activator	r_a	2×10^{-2}
Basic production rate of activator	b_a	1×10^{-1}
Saturation of activator autocatalysis	s_a	5×10^{-4}
Diffusion coefficient of activator	D_a	4×10^{-7}
Production & decay rate of global inhibitor	r_b	3×10^{-2}
Diffusion coefficient of global inhibitor	D_b	4×10^{-5}
Production rate of local inhibitor	b_c	7×10^{-3}
Decay rate of local inhibitor	r_c	1.3×10^{-2}
Diffusion coefficient of local inhibitor	D_c	2.8×10^{-6}
Michaelis-Menten constant	s_c	2×10^{-1}

Table 3.1: Non-dimensional parameter values for cell migration simulations model (3.1). These values taken from existing research conducted by Neilson et al. [94, 105].

interaction between the membrane and the extracellular environment is modelled using the boundary flux condition (3.3) where $[(\boldsymbol{u} \cdot \boldsymbol{n})]L\Big|_{\Gamma(t)}$ is the advective flux term, $\boldsymbol{u}$ is the material velocity for the receptors, k_1 is the free-ligand to receptor binding rate and k_{-1} the corresponding unbinding rate. Equation (3.2) is not explicitly solved here and is instead approximated by means of a stochastic differential equation and a number of assumptions to give an approximation for the fractional receptor occupancy on the cell membrane and the ligand concentration on the exterior of the cell. However methods will be introduced in proceeding chapters to calculate (3.2) explicitly as part of the two dimensional cell model which will also be introduced in later chapters.

A law of mass action equation is used to model the kinetics of the receptor-ligand complex. The membrane is assumed to have R_{tot} receptors which are assumed to be

uniformly distributed along the membrane. As such, the receptor-ligand complex is assumed to evolve according to

$$\frac{\partial L_m}{\partial t} + \nabla_{\Gamma(t)} \cdot (L_m \boldsymbol{u}) = D_s \Delta_{\Gamma(t)} L_m + k_1 \left(N_r - L_m\right) L|_{\Gamma(t)} - k_{-1} L_m, \tag{3.4}$$

where N_r is the local concentration of receptors and L_m is the local concentration of receptor-ligand complex on the membrane.

If we assume that there is little or negligible diffusion of the receptors-bound complex along the membrane and also that the receptors are in a fixed position relative to the membrane movement so that $\boldsymbol{u} = 0$, then

$$\frac{dL_m}{dt} = k_1 \left(N_r - L_m\right) L\Big|_{\Gamma(t)} - k_{-1} L_m, \tag{3.5}$$

where $N_r - L_m$ is the local concentration of unoccupied receptors. Note that (3.5) has no boundary condition. Also notice that the local concentration of receptor bound ligand L_m varies between zero, when no receptors are bound and N_r, when all receptors are bound. It follows that the local fractional receptor occupancy is

$$R^o = \frac{L_m}{N_r}. \tag{3.6}$$

Cells move randomly in the absence of any external directional information [24, 25, 68, 141]. As such the signal term is a combination of an intrinsic noise component η and the fractional receptor occupancy. This will also make the signal to noise ratio non-zero and more realistic and will more faithfully emulate what the cell will sense in a background with little to no background gradient. The combined effects of the noisy external signal and random intrinsic noise is thus modelled by the signal term

$$\varrho = r_a \left(\eta + R^o\right),$$

which feeds in multiplicatively to the autocatalytic reaction term of the local activator equation of (3.1).

The local activator a is coupled with a simple mechanical model for the relocation

of the cell membrane,

$$\mathcal{V} = \dot{\boldsymbol{x}} \cdot \boldsymbol{n} = K_{\text{prot}} a(\boldsymbol{x}) - \lambda \kappa(\boldsymbol{x}), \tag{3.7}$$

where λ is a cortical tension coefficient and $\kappa(\boldsymbol{x})$ is local surface curvature. It is expected that actin polymerization creates a protrusive pressure at the cell membrane which develops into pseudopodia and that the local activator a will represent or relate to the local concentration of actin in some way [105]. The coefficient K_{prot} is a protrusive velocity scaling parameter and is taken to be

$$K_{\text{prot}} = 1 \times 10^{-5}. \tag{3.8}$$

The cortical tension factor λ is calculated using a shell-liquid drop model [31, 42, 43, 105] such that the cell volume is conserved and the cell will maintain a spherical shape when not migrating which is observed in nature. Specifically λ evolves temporally according to

$$\frac{d\lambda}{dt} = \frac{\lambda^0 \lambda (A - A^0 + dA/dt)}{A^0(\lambda + \lambda^0)} - \beta\lambda, \tag{3.9}$$

where λ^0 and β are positive parameters and $A^0 = |\Gamma(0)|$ is the initial prescribed area of the cell. The net effect of both the protrusive and retractive components of the normal velocity creates realistic cell morphologies and migration. Subsequent simulations and numerical experiments use $\beta = 2 \times 10^{-2}$.

3.2 Simulating cell migration

The cell membrane domain $\Gamma(t)$ is approximated as a series of $N = 200$ connecting line segments which is initially a circle of radius $r_0 = 0.1$ centred at the origin. This choice for the cell radius is also used by Neilson et al. [105] and was originally chosen such that the Meinhardt model gives a fair representation of cell motility and morphology found in Dictyostelium cells.

The fractional receptor occupancy will have the same rates of change as L_m (3.5), and hence the change in fractional receptor occupancy with respect to time can be

written as

$$\begin{aligned}\frac{dR^o}{dt} &= k_1\left(1-R^o\right)L\Big|_{\Gamma(t)} - k_{-1}R^o \\ &= k_1L\Big|_{\Gamma(t)} - \left(k_1L\Big|_{\Gamma(t)} + k_{-1}\right)R^o \\ &= \left(k_1L\Big|_{\Gamma(t)} + k_{-1}\right)\left(\frac{k_1L\Big|_{\Gamma(t)}}{k_1L\Big|_{\Gamma(t)} + k_{-1}} - R^o\right),\end{aligned} \tag{3.10}$$

which is a Michaelis-Menten equation which has a maximum saturation concentration of $R^o = 1$.

A temporal noise model based on an Ornstein-Uhlenbeck process [20] is used to model the fractional receptor occupancy (3.10) whereby

$$dR^o = \theta\left(\mu - R^o\right)dt + \sigma dW_m^t, \tag{3.11}$$

where W_m^t denotes the Weiner process and $\theta > 0$, μ, and $\sigma > 0$ are the parameters for the Ornstein-Uhlenbeck process. Here $\theta = k_1L\Big|_{\Gamma(t)} + k_{-1}$ and $\mu = \frac{k_1L\Big|_{\Gamma(t)}}{k_{-1}+k_1L\Big|_{\Gamma(t)}}$. The Ornstein-Uhlenbeck process parameter for variance is asymptotically related to the variance of the probability distribution in that

$$\sigma = \sqrt{2\theta \mathrm{Var}(R^o)}. \tag{3.12}$$

The binding of individual receptors by ligands is reasonably assumed to be stochastically independent and so the binding process can be modelled with a binomial distribution. At each time-step an equilibrium is assumed to have been reached between the binding and unbinding processes of the free-ligand and receptors at the membrane, thus we use the fractional receptor occupancy at steady state,

$$\bar{R}^o = \frac{k_1L\Big|_{\Gamma(t)}}{k_{-1} + k_1L\Big|_{\Gamma(t)}} = \frac{L\Big|_{\Gamma(t)}}{K_d + L\Big|_{\Gamma(t)}}, \tag{3.13}$$

where $K_d = k_{-1}/k_1$. Now at every time-step each of the N_r receptors will have a

$p = 1 - \bar{R}^o$ chance of binding with a free-ligand in the cell exterior and so the variance of the receptor-ligand binding process is

$$\mathrm{Var}(L_m) = N_r p(1-p) = N_r(1-\bar{R}^o)\bar{R}^o = \frac{N_r K_d L\Big|_{\Gamma(t)}}{\left(K_d + L\Big|_{\Gamma(t)}\right)^2}, \tag{3.14}$$

from which we get the variance of the fractional receptor-ligand binding process as

$$\mathrm{Var}(R^o) = \mathrm{Var}\left(\frac{L_m}{N_r}\right) = \frac{1}{N_r^2}\mathrm{Var}(L_m) = \frac{K_d L\Big|_{\Gamma(t)}}{N_r\left(K_d + L\Big|_{\Gamma(t)}\right)^2} = \frac{K_d L\Big|_{\Gamma(t)}}{N_r(\theta/k_1)^2}. \tag{3.15}$$

It follows that (3.11) will model the fractional receptor occupancy with realistic stochasticity if

$$\sigma = \sqrt{2\theta\mathrm{Var}(R^o)} = \sqrt{\frac{2\theta K_d L\Big|_{\Gamma(t)}}{N_r(\theta/k_1)^2}} = \sqrt{\frac{2k_{-1}\bar{R}^o}{N_r}}, \tag{3.16}$$

and hence the complete Ornstein-Uhlenbeck process for modelling fractional local receptor occupancy is

$$dR^o = \left(k_1 L\Big|_{\Gamma(t)} + k_{-1}\right)\left(\bar{R}^o - R^o\right)dt + \sqrt{\frac{2k_{-1}\bar{R}^o}{N_r}}dW_m^t. \tag{3.17}$$

If the receptors are uniformly distributed around the membrane then the natural choice for the local number of receptors is

$$N_r = \frac{R_{\mathrm{tot}}}{N}, \tag{3.18}$$

where the total number of receptors in Dictyostelium is found to be of the order $R_{\mathrm{tot}} = 7 \times 10^4$ [71]. The binding-unbinding rates of receptor-ligand complex is also measured [142] from which we chose $k_1 = 1/30$, $k_{-1} = 1$. It must be noted that while the values for R_{tot}, k_1 and k_{-1} are dimensionally accurate, the rest of the model is not but a proper non-dimensionalisation of the entire model will be looked at in Chapter 5 using

these parameters as a starting point.

The intrinsic noise component, which emulates what the cell will sense in a background with little to no discernible gradient, is assumed to satisfy a similar process of the form (3.11), where the parameters μ, θ and σ are user-chosen. Here we use

$$d\eta = (1-\eta)\,dt + \frac{0.1}{N_m} dW^t. \tag{3.19}$$

The combined effects of the noisy external signal and random intrinsic noise is therefore modelled by

$$\varrho(\boldsymbol{x}, t) = r_a\,(\eta + R^o)\,,$$

which feeds in multiplicatively to the autocatalytic reaction term of the local activator equation of (3.1).

The approximate area conservation differential equation (3.9) is integrated using a backward-Euler scheme where

$$\frac{dA}{dt} = \frac{A^n - A^{n-1}}{\Delta t}, \tag{3.20}$$

and

$$\lambda^{n+1} = \lambda^n + \frac{\lambda^n \lambda^0 (\Delta t (A^n - A^0) + (A^n - A^{n-1}))}{A^0(\lambda^n + \lambda^0)} - \Delta t \beta \lambda^n. \tag{3.21}$$

The initial conditions for local activator a, global inhibitor b and local inhibitor c are derived from small perturbations about their respective homogeneous steady states in (3.1). By noticing that the term $\frac{1}{|\Gamma(t)|}\oint_{\Gamma(t)} a d\boldsymbol{x} = a$ for a homogeneous, it follows that the homogeneous steady state is such that

$$\begin{aligned} \frac{s(a^{*2}/b^* + b_a)}{(s_c + c^*)(1 + s_a a^{*2})} - r_a a^* &= 0, \\ -r_b b^* + r_b a^* &= 0, \\ b_c a^* - r_c c^* &= 0. \end{aligned} \tag{3.22}$$

The equation for the steady state local activator a^* can be found by re-arranging to

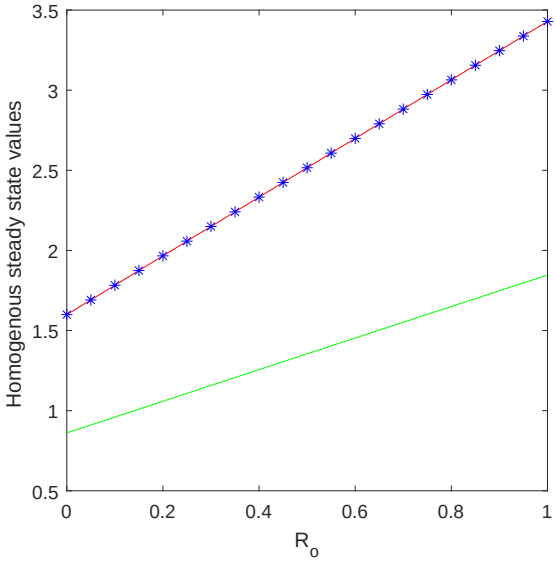


Figure 3.1: The homogeneous steady state solution of the Meinhardt system (3.1) as the fractional receptor occupancy, R_o, is varied. The homogeneous steady state for the local activator a^* is given in red, the global inhibitor b^* is given in blue and the local inhibitor c^* in green.

obtain a quartic equation

$$b_c s_a a^{*4} + r_c s_a s_c a^{*3} + b_c a^{*2} + r_c \left(s_c - (1 + R_o)\right) a^* - b_a r_c (1 + R_o) = 0, \tag{3.23}$$

and $b^* = a^*$, $c^* = \frac{b_c}{r_c} a^*$. The real positive solution for a^* in (3.23) will always exist and be unique for any choice of positive reaction kinetics parameters and fractional receptor occupancy R_o. This follows from Descartes' rule of signs [103] whereby because the polynomial (3.23) has exactly one sign change in the coefficients between the a^{*2} and a^* terms, then there exists exactly one positive root for the polynomial. Figure 3.1 shows the variation in the value of a^* as a function of the fractional receptor occupancy R_o, where the other parameters in the model are given in Table 3.1. Using the kinetic parameters in Table 3.1, it follows that $a^* \simeq 1.599$.

The initial conditions for the cortical tension equation (3.9) are given by

$$A^0 = \pi (r_0)^2, \quad \lambda^0 = K_{\text{prot}} r_0 a^*. \tag{3.24}$$

The mechanical model depends on the solution of the activator but the solution

of the reaction-diffusion system depends on the domain position. Rather than solving one large coupled non-linear system, the reaction-diffusion and mechanical systems are solved separately, linearised by the results of the previous time-step. A small time-step, $\Delta t = 0.1$, is taken to ensure that this linearisation still gives an accurate representation of the solution. The Meinhardt system (3.1) is solved using the ALEFEM1D scheme and the computational mesh is evolved using MMPDE1D both introduced in the previous chapter.

The complete cell migration algorithm is thus as follows:

1. Set initial parameters.
2. Do until end of simulation.
   (a) Calculate receptor occupancy R^o from free-ligand concentration $L\Big|_{\Gamma(t)}$ using Ornstein-Uhlenbeck process (3.11).
   (b) Calculate intrinsic noise term η using Ornstein-Uhlenbeck process (3.19)
   (c) Solve Meinhardt system (3.1) for local activator, global and local inhibitor using ALEFEM1D.
   (d) Use MMPDE1D to move membrane interface using the local activator a and λ (3.21).
   (e) Save and render result as needed.
3. Repeat until completion.

Algorithm 3.1: The complete algorithm for the one dimensional cell migration model.

To model the effect of a chemotactic background on the cell, a gradient varying parameter ρ is used whereby

$$\rho = \frac{\bar{R}_f^o - \bar{R}_b^o}{2r_0}, \tag{3.25}$$

where $\bar{R}_f^o$ denotes the expected receptor occupancy for receptors in the highest concentration of gradient (in some sense the front or leading edge of the cell) and $\bar{R}_b^o$ the lowest concentration (the back or trailing edge). In the examples which follow, lack of

gradient ($\rho = 0$) is modelled by having fractional receptor occupancy around the cell being $\bar{R}^o = 0.15$, whereas a strong chemotactic gradient of 20% (for which $\rho = 0.2$) is modelled with $\bar{R}^o_f = 0.19$ and $\bar{R}^o_b = 0.15$. To achieve such a gradient, we align the peak concentration vertically so that the chemotactic source of free-ligands is given as

$$L(x, y) = 5.2941, \tag{3.26}$$

for the zero gradient case and

$$L(x, y) = 8.7146y + 6.1656, \tag{3.27}$$

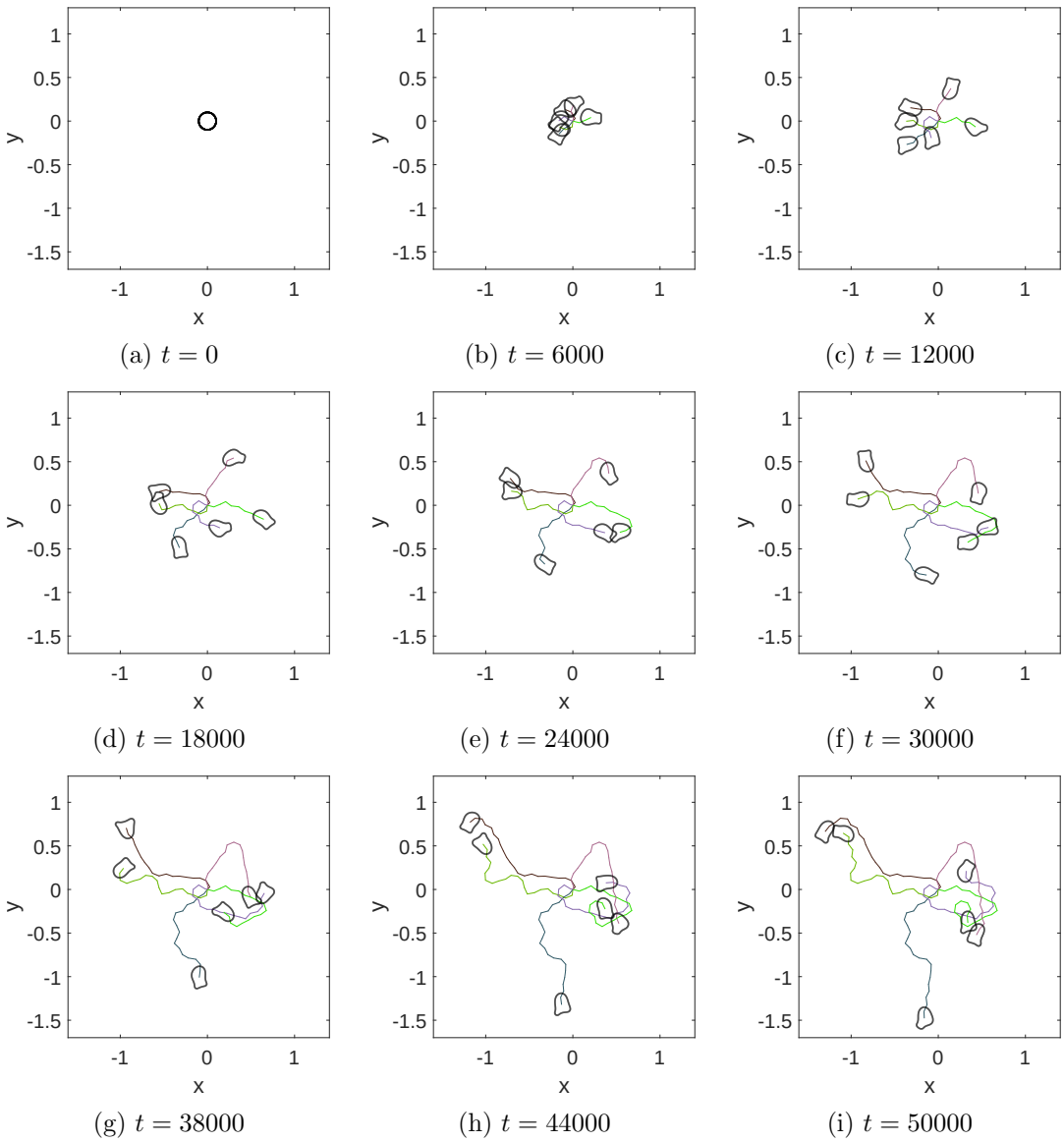
for the non-zero case $\rho = 0.2$.

In the case of no gradient, the cells do not migrate in any one particular direction as shown in Figure 3.2. Each cell does though select a direction of its own choosing and persist well with migrating in that direction until it decides to select a new direction. Whereas by contrast in the 20% chemotactant gradient case, which is shown in Figure 3.3, the cells are directed strongly to go in the $+y$ direction by the perceived increase in free-ligands along this path.

Trajectories for the centroids of cells generated using various other values for ρ are shown in Figure 3.4. This shows a clear relationship between the free-ligand gradient and the average direction of chemotaxis for the group of cells. As the gradient is increased, the cells become more polarised in the direction of increasing gradient. Whereas by contrast, as the gradient is decreased, the cells become less polarised and begin to follow a random migration pattern. This mimics what real cells do to seek out alternative sources of food and not become trapped when the gradient is small.

A single cell simulation is shown at various time-steps in Figure 3.5 with corresponding solutions of the Meinhardt system given in Figure 3.6. For the zero gradient test case (3.26), the fractional receptor occupancy on the cell quickly reaches a quasi steady state where $R^o \approx 0.15$ with very minute perturbations to R^o being introduced through the noise term in the Ornstein-Uhlenbeck process. The corresponding figures for the linear gradient case (3.27) are given in Figure 3.7 and Figure 3.9. The results

for the Meinhardt system are given in Figure 3.8 showing the peak splitting behaviour which eventually leads to the pseudopods forming. It is interesting to note that in general two peaks will take form initially but one of them will usually die out and a pseudopod will begin to form near the location of the dominant peak. Despite the larger gradient in the cells' environment, the cells' morphology remains invariant which is ideal as the higher gradient concentration should not alter the cells' ability to migrate and chemotax.

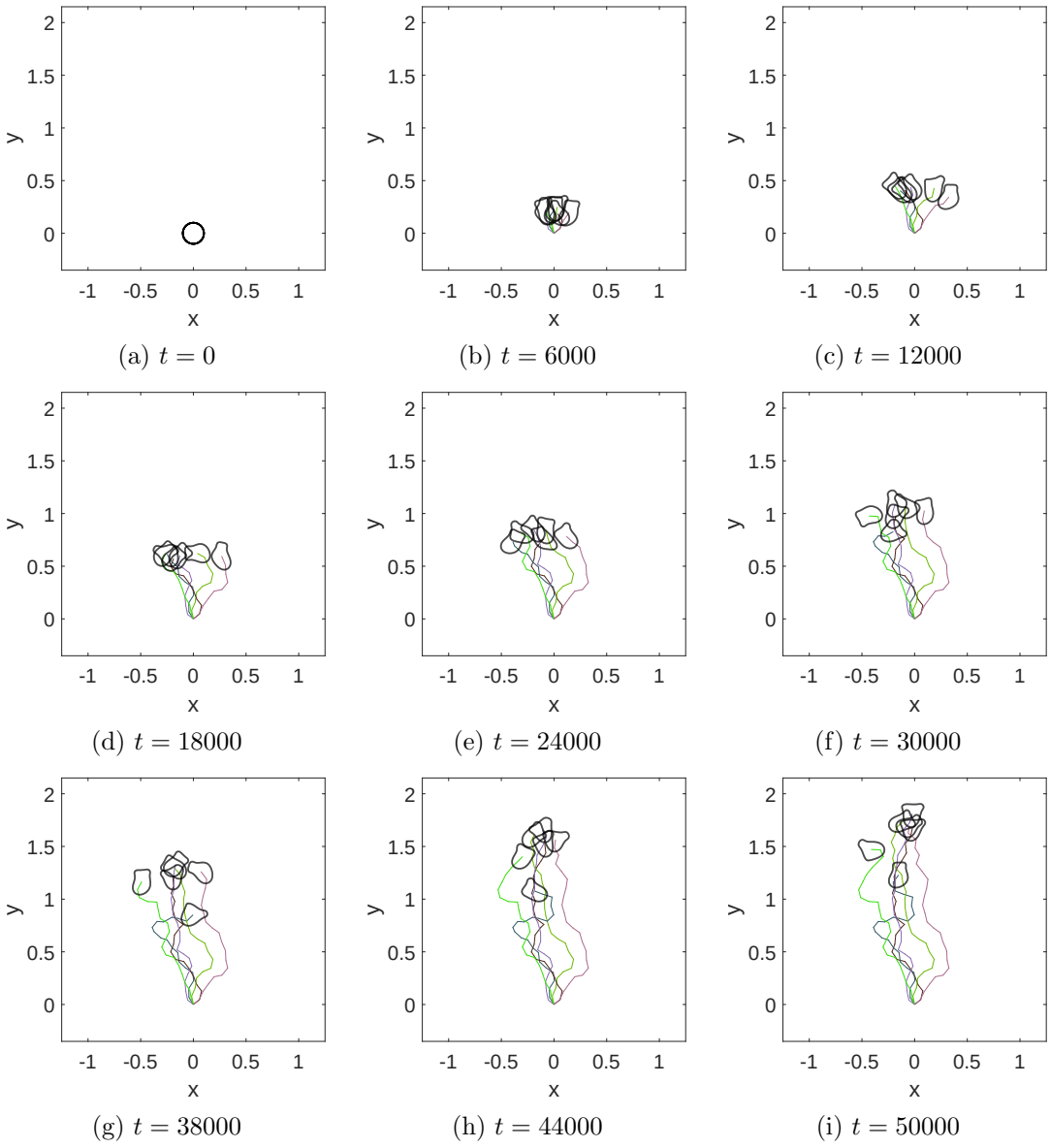


(a) $t = 0$ (b) $t = 6000$ (c) $t = 12000$

(d) $t = 18000$ (e) $t = 24000$ (f) $t = 30000$

(g) $t = 38000$ (h) $t = 44000$ (i) $t = 50000$

Figure 3.2: Snapshots of evolving cells and corresponding trajectory plots at various times. In the figures there is one cell, with the set of parameters as defined in Table 3.1, which has been simulated six times with various starting conditions. The cells are generated independently and do not interact with each other. The plots show the superimposed trajectories from these different simulations with $\rho = 0$.

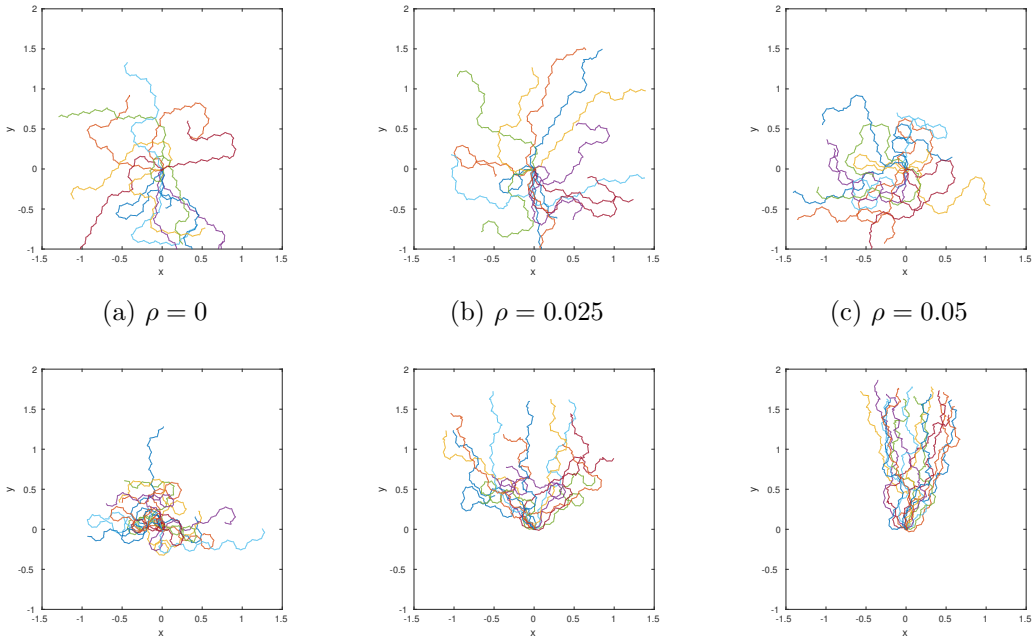


(a) $t = 0$ (b) $t = 6000$ (c) $t = 12000$

(d) $t = 18000$ (e) $t = 24000$ (f) $t = 30000$

(g) $t = 38000$ (h) $t = 44000$ (i) $t = 50000$

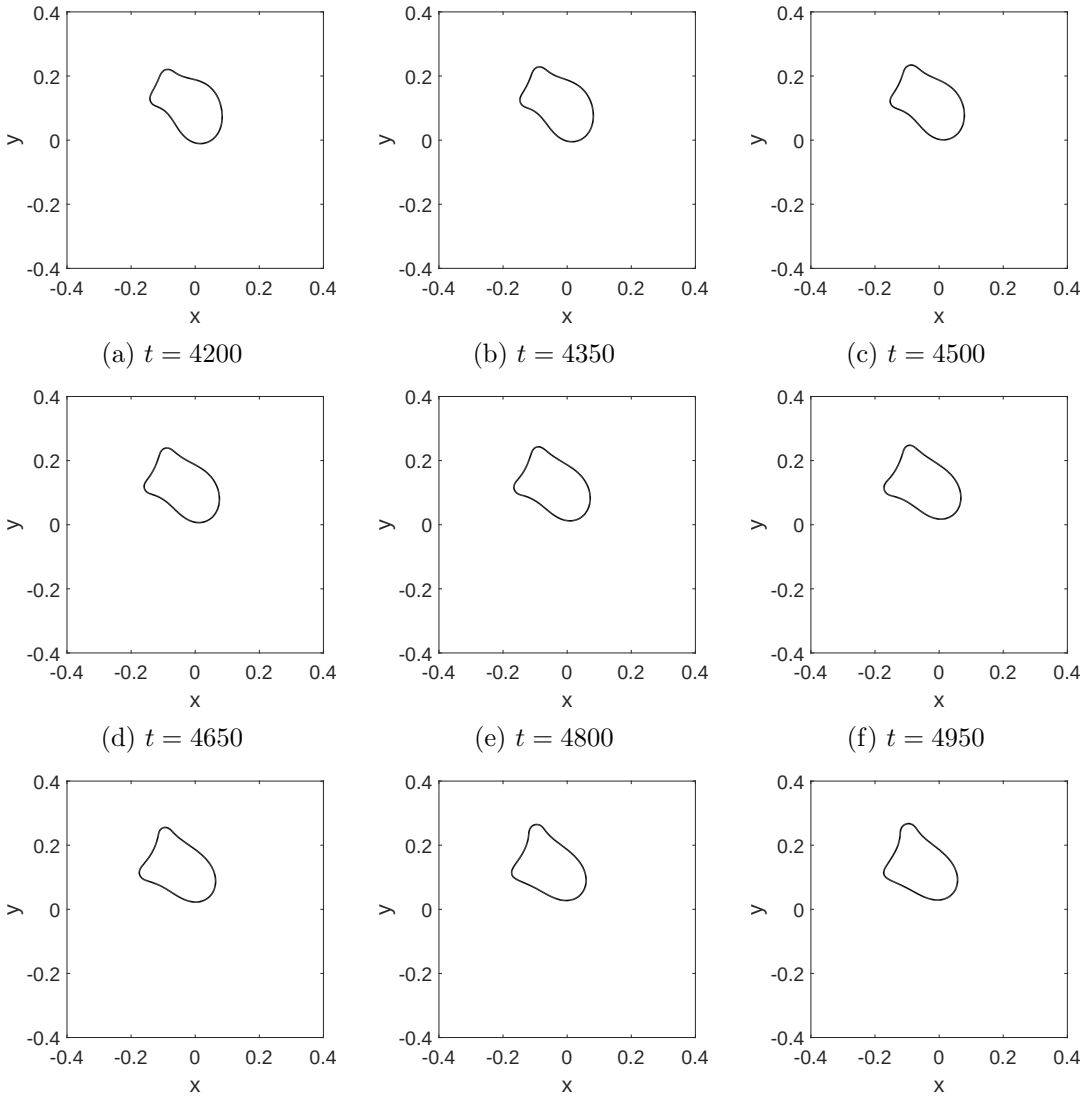
Figure 3.3: Snapshots of evolving cells and corresponding trajectory plots at various times. ($\rho = 0.2$)



(a) $\rho = 0$ (b) $\rho = 0.025$ (c) $\rho = 0.05$

(d) $\rho = 0.1$ (e) $\rho = 0.15$ (f) $\rho = 0.2$

Figure 3.4: Trajectory plots of a small sample of cells moving towards a high concentration of chemoattractant source which illustrates the chemotactic response of the simulated cells to changes in background chemotactic gradient. Notice the gradual inclination to move towards the higher concentration (along the $+y$ axis). Cells in lower gradients are more reluctant to move in the direction of perceived attractant. As the gradient is increased that reluctance is reduced but does not disappear completely as witnessed by the spread of the cells in (e) and (f).



(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 3.5: An example of a cell migrating in a homogeneous background at various time-steps. The protrusions coincide with regions of high activator as shown in Figure 3.6.

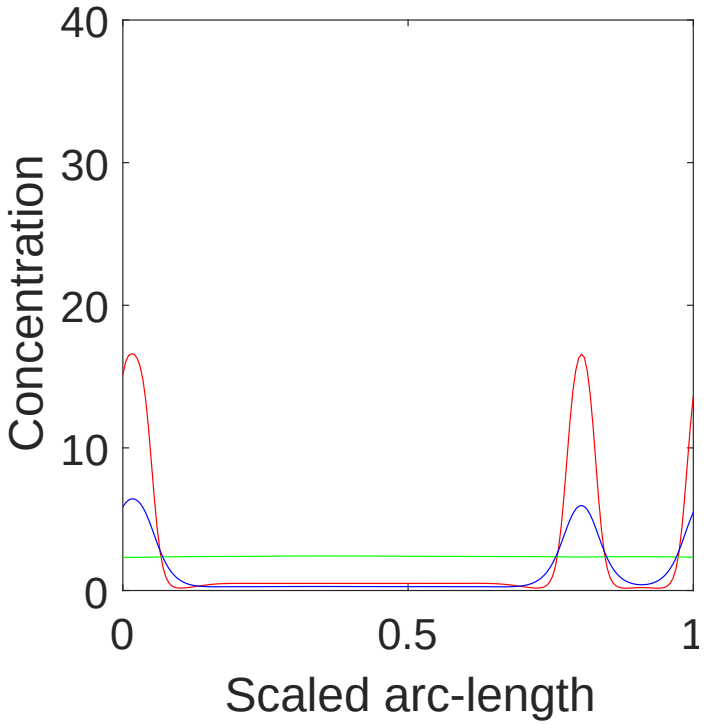
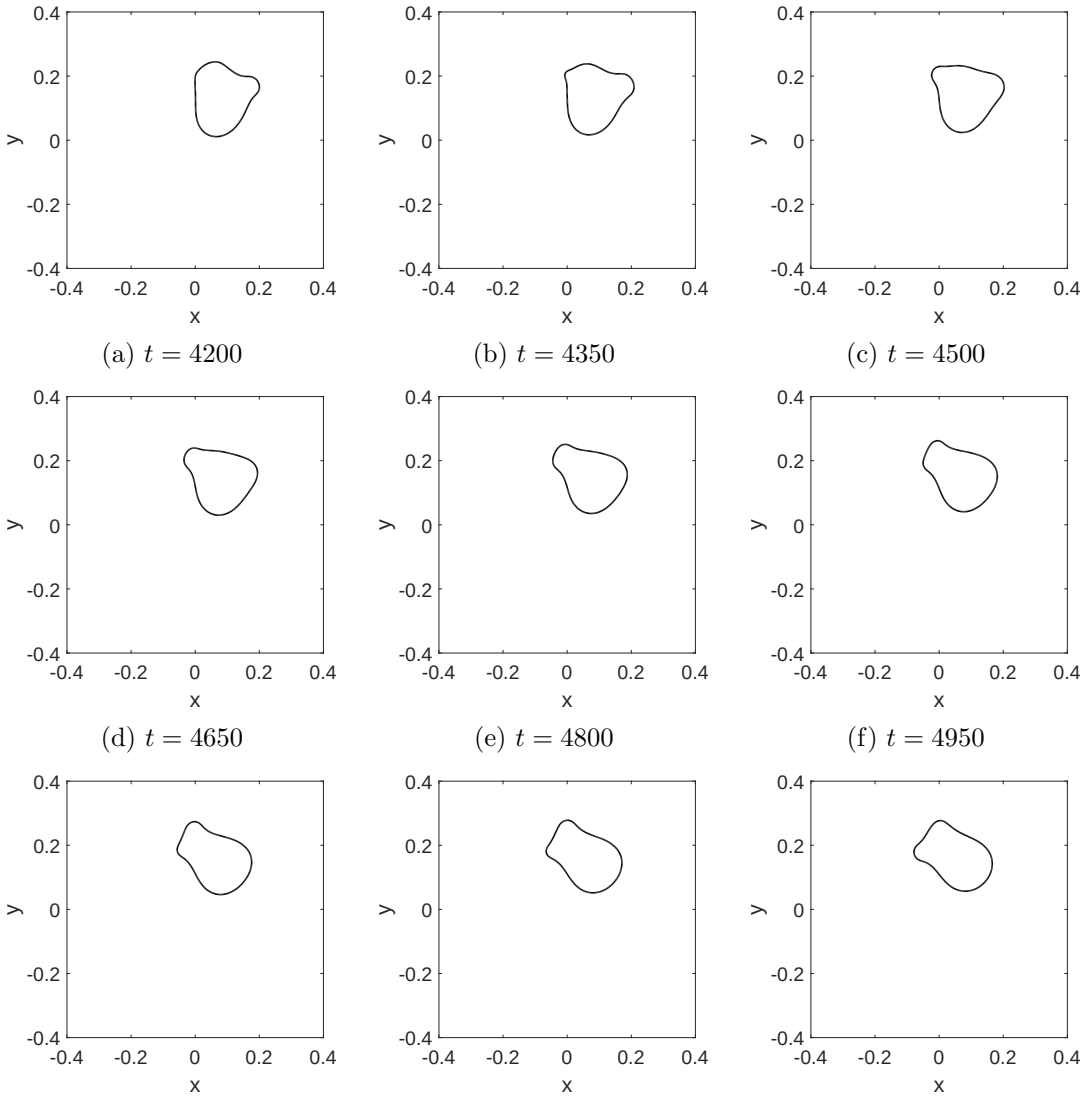


Figure 3.6: The corresponding solutions to the Meinhardt system for the cell simulation shown in Figure 3.5(e) where $t = 4800$. Here the activator a is in red, the global inhibitor b green and the local inhibitor c blue. The solutions of the Meinhardt system between $t = 4200$ and $t = 5400$ are of a very similar form.

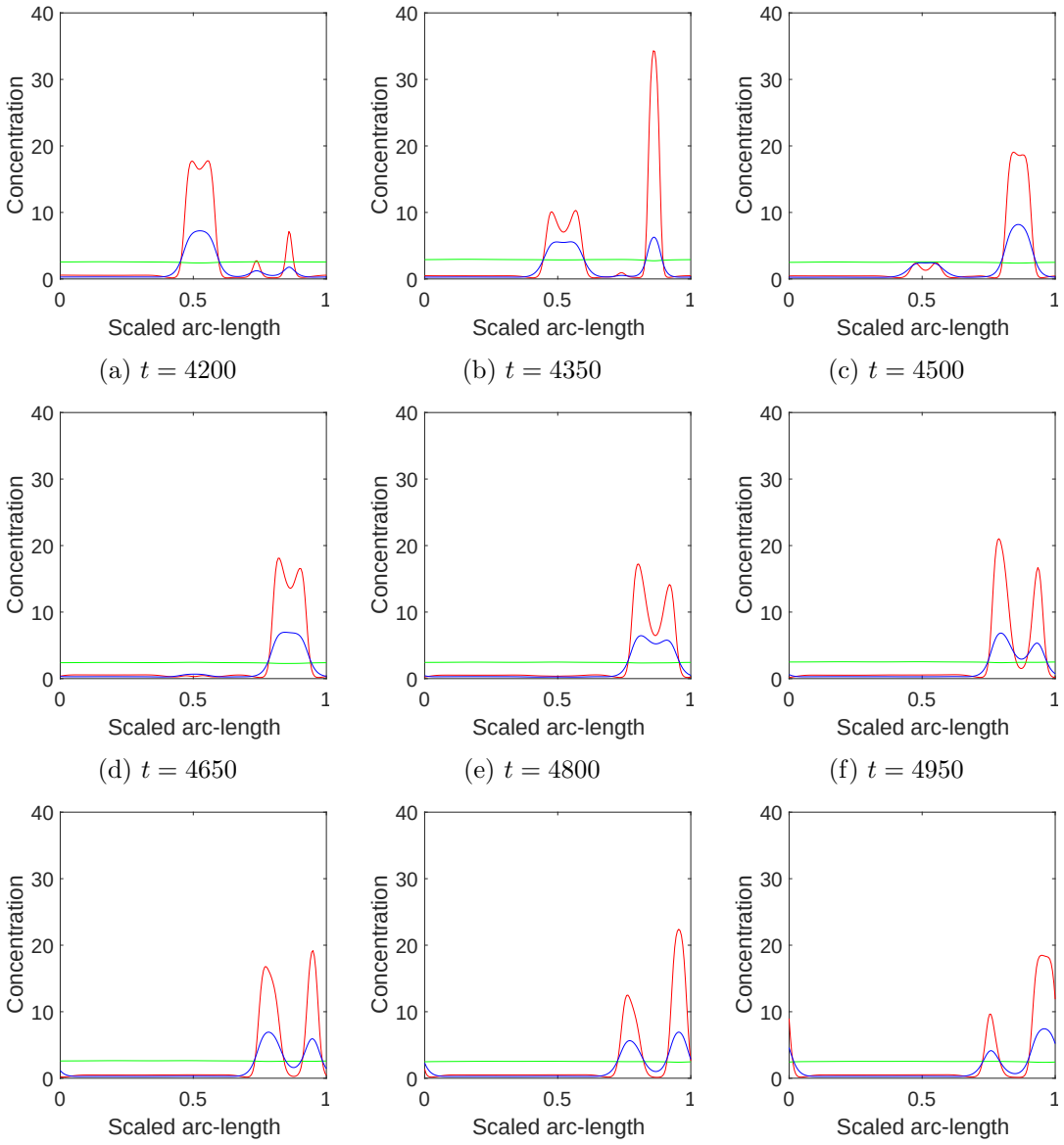


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 3.7: An example of a cell migrating in an inhomogeneous background at various time-steps. ($\rho = 0.2$)

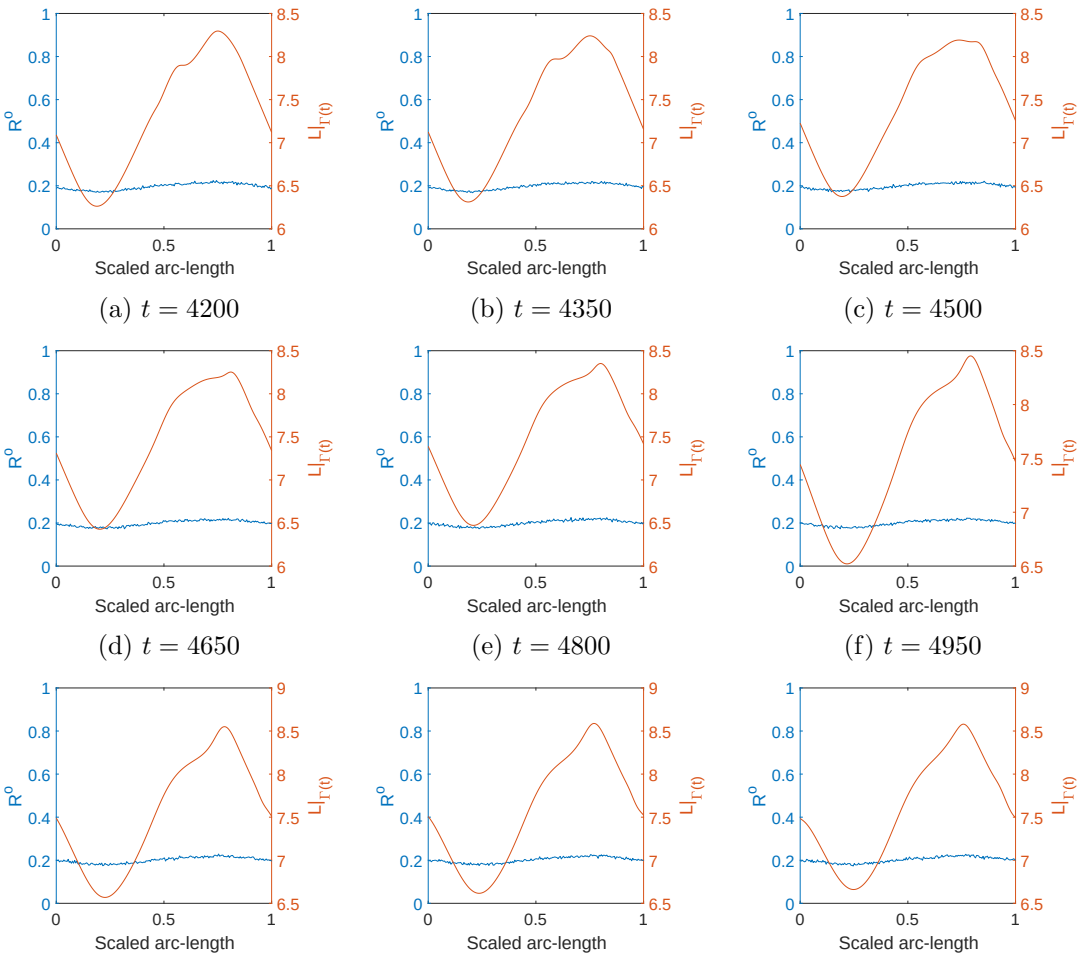


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 3.8: The corresponding solutions to the Meinhardt system for the cell simulation shown in Figure 3.7. Here the activator a is in red, the global inhibitor b green and the local inhibitor c blue. ($\rho = 0.2$)



(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 3.9: The corresponding fractional receptor occupancy and free-ligand concentration for the cell simulation shown in Figure 3.7. With a chemotactic gradient of $\rho = 0.2$, in the red plot we can see the range of the free-ligand concentration at the cell membrane and in the blue, the fractional receptor occupancy on the membrane. It is interesting to note that the peaks in free-ligand do not necessarily always coincide with the pseudopods in Figure 3.7 however, we can see with reference to Figure 3.8 that generally the peaks in activator which form near to the peak of the free-ligand field will be more likely to generate a psuedopod.

3.3 Comparison of adapted moving meshes with the cell migration model

Although the MMPDE1D method was originally developed to help the two-dimensional methods with the use of τ to slow the rate of tangential movement which led to skewing of the bulk mesh, it can also be used to adapt the mesh points along the cell membrane.

Often uniform arc-length distributed mesh points are ideal. In the cell migration model presented here however this is not the case. As evident by Figure 3.6 and Figure 3.8, the solution of the activator and local inhibitor of the Meinhardt system is mostly zero except in small intervals where there are steep peaks. This suggests that a mesh with nodes refined around the activator peaks and coarsened where the solution is constant may give more accurate results; possibly with fewer mesh points, leading to a computationally more efficient simulation. However, care must also be taken to ensure the domain geometry itself is also preserved. If too many nodes are removed where the solution is constant, then the mesh may not approximate the correct domain if the curvature of the domain is high.

It is hard to know what monitor function would provide an optimal combination for both preserving the underlying curve domain and providing the most efficient mesh on which to solve the Meinhardt system. Three monitor functions will be looked at here. One refines the mesh on the basis of geometric features of the cell membrane, the second refines the mesh based on the solution to the activator equation and the third is a linear combination of the first two.

To quantify the differences between each mesh, a closest point metric is used. Specifically the Hausdorff distance [121] given as

$$||e\left(\Omega_h(t), \Omega^*(t)\right)|| = \max\left\{ \max_{\boldsymbol{y}\,\in\,\Omega^*(t)} \left\{ \min_{\boldsymbol{x}\in\Omega_h(t)} ||\boldsymbol{x}-\boldsymbol{y}|| \right\}, \max_{\boldsymbol{x}\,\in\,\Omega_h(t)} \left\{ \min_{\boldsymbol{y}\in\Omega^*(t)} ||\boldsymbol{x}-\boldsymbol{y}|| \right\} \right\} \tag{3.28}$$

which is the maximum closest Euclidean distance between what will be the computationally generated mesh $\Omega_h(t)$, and $\Omega^*(t)$, which will act as the the model solution and

be calculated using $N_{\text{model}} = 1000$ nodes with no adaption.

As was seen in the previous chapter, using a monitor function which varies the mesh node density depending on the curvature can improve the approximation of the numerical domain. Because of the nature of the cell migration model, improving the accuracy of the numerical domain is tantamount to improving the accuracy of the model itself. Using MMPDE1D in the cell migration model can produce mesh refinement specifically in high curvature areas which correspond to areas where cell pseudopods form. This will also be where the the activator of the Meinhardt equations forms, or has formed, a peak which suggests that a geometry based feedback for mesh adaption may also improve the accuracy of the underlying PDE solver. A monitor matrix of the form

$$M = |\kappa|^{1/2} + \frac{1}{|\Gamma(t)|}\int_{\Gamma(t)} |\kappa|^{1/2} \ \mathrm{d}l, \tag{3.29}$$

refines the mesh in areas of high curvature. A demonstration of a cell fixed at time $t = 4100$ is shown in Figure 3.10 (a), (d) and (g) with no adaption. By contrast, Figure 3.10[1] (b), (e) and (h) show the same cell using the monitor matrix (3.29). There is a noticeable improvement in accuracy using the curvature-based adaption over the cell generated with the uniform mesh when compared to the model solution given as the blue lines in Figure 3.10 (d) and (e). The model solution is taken from the cell model generated with $N_{\text{model}} = 1000$ uniformly distributed mesh points whereas the adaption examples have $N = 200$.

If the objective were to improve the accuracy of the solutions to the Meinhardt system (3.1) rather than to better resolve the numerical domain, an alternative monitor function might involve solutions of the PDEs themselves. Consider the monitor function

$$M = |\nabla_{\Gamma(t)} a|^{1/2} + \frac{1}{|\Gamma(t)|}\int_{\Gamma(t)} |\nabla_{\Gamma(t)} a|^{1/2} \, \mathrm{d}l, \tag{3.30}$$

where the activator solution is now used to refine the numerical mesh. This choice of

[1]Note that there is an appearance that the solutions have been shifted although this is not the case. The peaks in the solutions of the Meinhardt system are in almost identical positions with respect to the physical coordinate $\boldsymbol{x}$ and it is only because the nodes themselves are moving that, when plotting with respect to scaled arc-length ξ, the physical coordinate for $\xi = 0$, $\boldsymbol{x}(\xi, t)$, is not guranteed to be identical in each curve.

monitor function which depends only on the activator, and not on the global or local inhibitor, is justified by noting that the global inhibitor is roughly constant for the Meinhardt system due to its high diffusivity; and because the local inhibitor closely follows the activator, therefore changes in the activator dictate changes to the local inhibitor and global inhibitor equations by design. In areas where a is small, there will be little movement of the mesh points, however the mesh points will clump together close to areas where the solution is rapidly changing, specifically close to where the peaks in the activator form. This is seen in Figure 3.10 (c), (f) and (i), where there is a definite clumping of the mesh points taking place along the areas where the cell has formed pseudopods, and hence areas of high activity in a. However, it is easily noticed by Figure 3.10 (i) that (3.30) has reduced the accuracy of the solution to this cell and this becomes progressively worse as the simulation continues. This is because the mesh points away from the pseudopods are being coarsened and this is affecting the solution on the domain as a whole. For instance, we can see that while there is very good definition made by curvature based adaption in Figure 3.10 (b) at the trailing edge ($+y$) of the cell and even in Figure 3.10 (a), the no adaption case, there is a loss of resolution at the rear of the cell in Figure 3.10 (c) caused by the mesh points moving away from this area. One solution might be to reduce the range of the (3.30) using some sort of modulation coefficient or to combine (3.30) and (3.29) to obtain an adaption method which tries to encapsulate the behaviours of both monitor functions.

To account for both the complexity of the geometry and the roughness of the solution of underlying PDEs, it is not immediately obvious how to combine both forms of adaptivity in a way which is optimal. One way is easily understood and modified is a linear combination of the two,

$$M = \mu_1|\kappa|^{1/2} + \mu_2|\nabla_{\Gamma(t)}a|^{1/2} + \frac{1}{|\Gamma(t)|}\int_{\Gamma(t)}\left(\mu_1|\kappa|^{1/2} + \mu_2|\nabla_{\Gamma(t)}a|^{1/2}\right)\ \mathrm{d}l, \tag{3.31}$$

where μ_1 and μ_2 are user specified scaling parameters.

Let

$$||e\left(\Omega_h, \Omega^*\right)|| = \frac{1}{T}\int_0^T ||e\left(\Omega_h(t), \Omega^*(t)\right)||\ \mathrm{d}t \tag{3.32}$$

be the total average error for the entire run, where $T = 5000$. Let $\{\boldsymbol{x}_i(t)\}_{i=1}^{N_{\text{model}}}$ be the mesh points which make up $\Omega^*(t)$ and $\{\boldsymbol{x}_{hj}(t)\}_{j=1}^{N}$ be the mesh points which make up $\Omega_h(t)$. Due to the dense nature of the $\Omega^*(t)$ mesh, (3.28) and (3.32) are approximately

$$||e\left(\Omega_h(t), \Omega^*(t)\right)|| \simeq \max\left\{ \max_{1\leq j\leq N}\left\{ \min_{1\leq i\leq N_{\text{model}}} ||\boldsymbol{x}_i(t) - \boldsymbol{x}_{hj}(t)|| \right\}, \max_{1\leq i\leq N_{\text{model}}}\left\{ \min_{1\leq j\leq N} ||\boldsymbol{x}_i(t) - \boldsymbol{x}_{hj}(t)|| \right\} \right\}, \tag{3.33}$$

and

$$||e\left(\Omega_h, \Omega^*\right)|| \simeq \frac{1}{N_T}\sum_{n=1}^{N_T} ||e\left(\Omega_h(t^n), \Omega^*(t^n)\right)||, \tag{3.34}$$

where $N_T = 50000$ since $\Delta t = 0.1$.

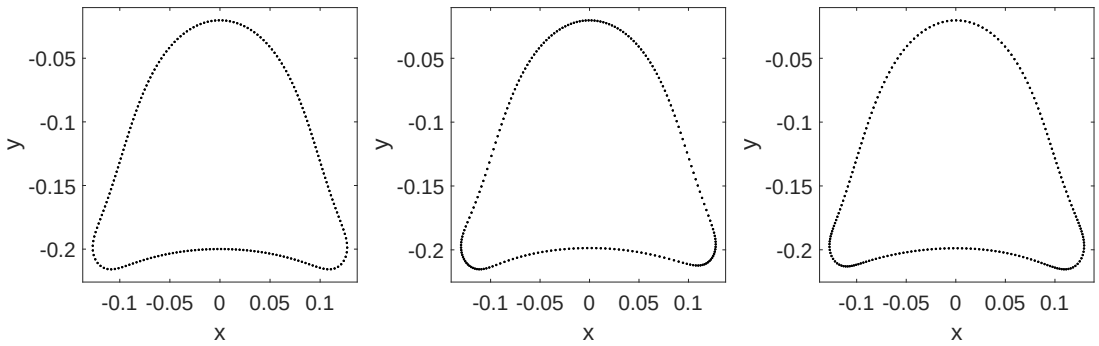
The results for each of the cases in Figure 3.10 are given in Table 3.2 and show the activator based monitor function (3.30) as being a better alternative to the mesh generation computed without any adaption by the measure of the distance function $||e\left(\Omega_h, \Omega^*\right)||$. Table 3.3 shows the results using monitor function (3.31) with the various values of μ_1 and μ_2 corresponding to Figure 3.11. The improvement using the monitor functions (3.29) and (3.30) are apparent if only slight. It is therefore expected that a proper investigation of the choice of monitor function for the cell problem could produce greater numerical improvements however that is beyond the scope of this text. The main reason that MMPDE1D was developed is to allow a natural and robust means of extending the cell model to higher dimensions without the shearing of mesh points along the boundary which happens in some other methods such as PFEM and to that end MMPDE1D does an excellent job as will be seen in the proceeding chapters.

Description	$\|\|e\left(\Omega_h(t), \Omega^*(t)\right)\|\|\Big\|_{t=4100}$	$\|\|e\left(\Omega_h, \Omega^*\right)\|\|$
No adaption ($M = 1$)	4.286e-03	2.824e-03
Curvature based adaption (3.29)	2.843e-03	2.346e-03
Activator based adaption (3.30)	2.765e-03	2.829e-03

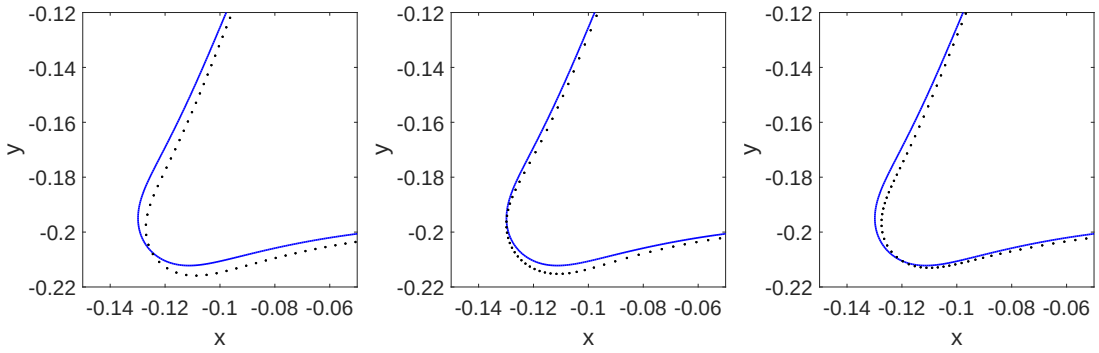
Table 3.2: The error between the numerical approximation $\Omega_h(t)$ for the cell membrane using various different monitor functions to relocate the $N = 200$ mesh points and the exact solution as calculated by a model result $\Omega^*(t)$ made from $N_{\text{model}} = 1000$ mesh points without adaption.

μ_1	μ_2	$\|\|e\left(\Omega_h, \Omega^*\right)\|\|$
1.000e-01	2.000e-01	2.520e-03
5.000e-02	1.000e-01	2.607e-03
8.000e-02	6.000e-02	2.538e-03

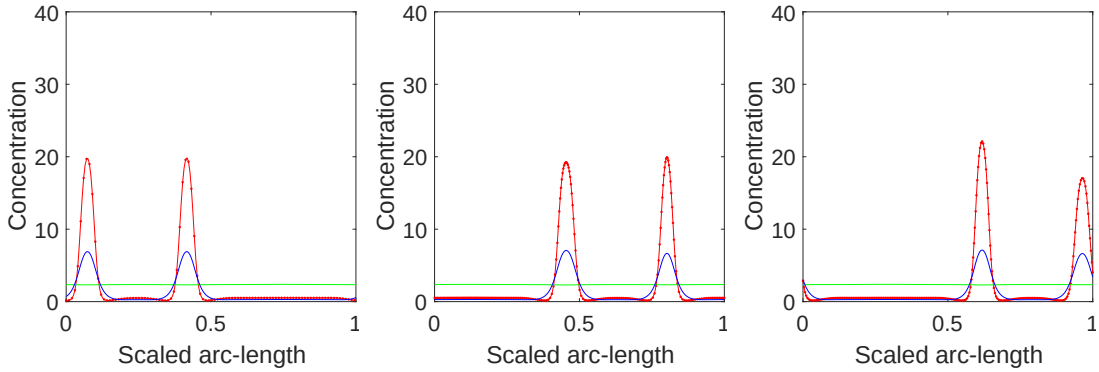
Table 3.3: The error between the numerical approximation $\Omega_h(t)$ for the cell membrane using the monitor function (3.31) with different values of μ_1 and μ_2 to relocate the $N = 200$ mesh points and the exact solution as calculated by a model result $\Omega^*(t)$ made from $N_{\text{model}} = 1000$ mesh points without adaption.



(a) No adaption (b) Curvature based adaption (c) Activator based adaption

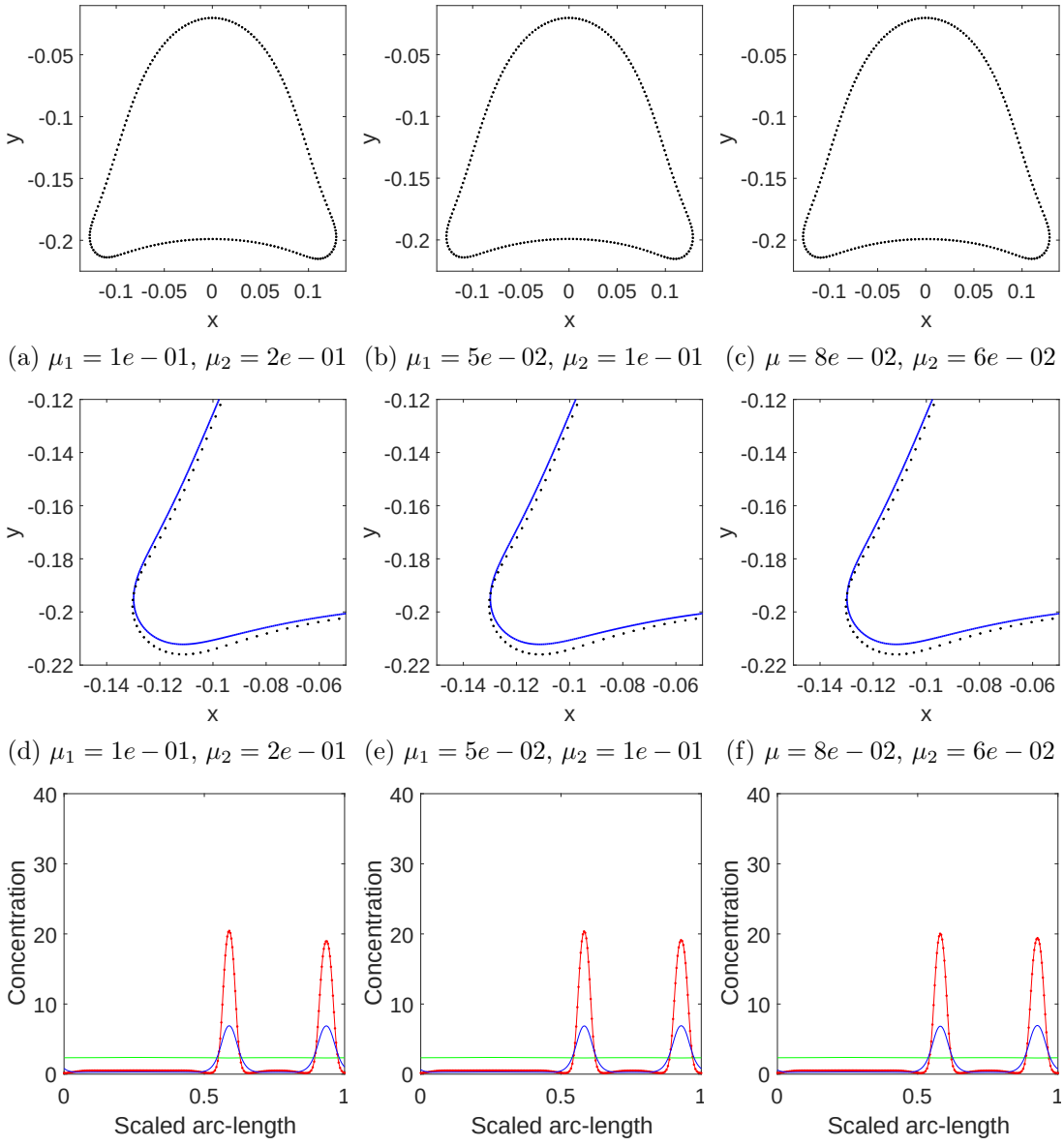


(d) No adaption (e) Curvature based adaption (f) Activator based adaption



(g) No adaption (h) Curvature based adaption (i) Activator based adaption

Figure 3.10: An example of adaption using MMPDE1D applied to the cell model problem. Here the cell profiles, as shown in (a), (b) and (c) for the particular case of $t = 4100$, have $N = 200$ mesh points. Figures (a), (d) and (g) show the case without adaption. Figures (b), (e) and (h) use curvature based monitor matrix (3.29). Figures (c), (f) and (i) use a monitor matrix based on the solution of the activator (3.30). Figures (d), (e) and (f) are an enlargement of (a), (b) and (c), respectively with the blue line representing the model solution $\Omega^*(t)$. Figures (g), (h) and (i) show the solutions to the Meinhardt equations for this particular time-step with the activator a in red, global inhibitor b in green and local inhibitor c in blue.



(a) $\mu_1 = 1e-01$, $\mu_2 = 2e-01$ (b) $\mu_1 = 5e-02$, $\mu_2 = 1e-01$ (c) $\mu = 8e-02$, $\mu_2 = 6e-02$

(d) $\mu_1 = 1e-01$, $\mu_2 = 2e-01$ (e) $\mu_1 = 5e-02$, $\mu_2 = 1e-01$ (f) $\mu = 8e-02$, $\mu_2 = 6e-02$

(g) $\mu_1 = 1e-01$, $\mu_2 = 2e-01$ (h) $\mu_1 = 5e-02$, $\mu_2 = 1e-01$ (i) $\mu = 8e-02$, $\mu_2 = 6e-02$

Figure 3.11: An example of adaption using MMPDE1D with (3.31) applied to the cell model problem for various values of μ_1 and μ_2. Here the cell profiles, as shown in (a), (b) and (c) for the particular case of $t = 4100$, have $N = 200$ mesh points. Figures (d), (e) and (f) are an enlargement of (a), (b) and (c) respectively with the blue line representing the model solution $\Omega^*(t)$. Figures (g), (h) and (i) show the solutions to the Meinhardt equations for this particular time-step with the activator a in red, global inhibitor b in green and local inhibitor c in blue.

3.4 Conclusions

In conclusion:

- The methods which were introduced in Chapter 2 have been, in this chapter, applied to an existing model of cell migration and chemotaxis which uses a reaction-diffusion based model which is solved on an evolving domain. The reaction-diffusion model is based on an earlier system by Hans Meinhardt for modelling patterns that appear in biology. This is the first time the MMPDE1D method introduced in Chapter 2 has been used in an application of cell migration and chemotaxis research and the results presented show the method to produce cell simulations with results comparable to the PFEM and level-set method originally used by Neilson et al [104, 105].

- A receptor-ligand model is used to simulate the binding and unbinding of the free-ligand molecules in the cell environment to the membrane. A quasi-steady state solution is discussed from which the fractional receptor occupancy is then fed into the Meinhardt system by means of the signalling term coupled with an intrinsic noise term. The activator solution develops peaks near to the high concentration of bound receptors. The activator is then used to drive the cell migration and is fed into a simple mechanical model as a protrusive pressure term combined with an area conserving term. The net effect gives the domain growth cell-like behaviours.

- A brief study was conducted to test the effectiveness of various adaption monitor functions applied to the cell mesh using the novel method MMPDE1D introduced in the previous Chapter. There were suggestions that different monitor functions could improve the overall accuracy of the simulation in different ways and that many of the choices could potentially increase overall computational efficiency.

The model of cell migration in this chapter does a fair job but is limited. In one way, it cannot fully interact with the environment and certain assumptions are made

about the free-ligand background field as being approximated by quasi-steady state solution of receptor-ligand binding kinetics in a background concentration which is linear and unaffected by the presence of the migrating cell when in truth the cell will be perturbing the field and altering the concentration such that the limitations on this model may prove to be inaccurate. To properly model such interactions, new methods must first be introduced to extend the model and cope with problems introduced in higher dimensions.

Chapter 4

Finite element solution of reaction-diffusion equations on evolving 2D domains

4.1 Introduction

We have focused so far on the generation of a model which simulates the evolving membrane of a cell. However, many mechanisms which contribute to cell migration take place in the intracellular and extracellular environment which interact with the membrane. Processes which are of particular interest include the treatment of focal adhesions, actin polymerization close to the leading edge and actin waves throughout the cell. To model these processes higher dimensional techniques have to be introduced.

4.2 Reaction-diffusion on an evolving 2D planar domain

This section introduces an ALEFEM scheme for the solution of reaction-diffusion PDEs on evolving domains in $\mathbb{R}^2$. Let $c(\boldsymbol{x}, t)$ be a chemical substance which exists on an evolving domain $\Omega(t) \subset \mathbb{R}^2$, where $(\boldsymbol{x}, t) = ((x, y), t) \in \mathbb{R}^2 \times I,\ I = [0, T],\ T > 0$; and let the particles which make up the substance be defined by the position vector field $\boldsymbol{X}(\boldsymbol{x}_0, t) = (X(\boldsymbol{x}_0, t),\, Y(\boldsymbol{x}_0, t)) : \Omega(0) \times I \to \Omega(t)$ and evolve according to a material

velocity field $\boldsymbol{u} : \Omega(t) \times I \to \mathbb{R}^2$, where

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}(\boldsymbol{X}(\boldsymbol{x}_0, t), t) = \left(\frac{\partial X}{\partial t}\Big|_{\boldsymbol{x}_0}, \frac{\partial Y}{\partial t}\Big|_{\boldsymbol{x}_0} \right). \tag{4.1}$$

The general reaction-diffusion equation is then

$$\frac{\partial c}{\partial t}\Big|_{\boldsymbol{x}} - \mu \Delta c + \nabla \cdot (\boldsymbol{u} c) = f(c), \tag{4.2}$$

where $\frac{\partial c}{\partial t}\Big|_{\boldsymbol{x}}$ is the time derivative of $c(\boldsymbol{x}, t)$ at time t and at the physical coordinate $\boldsymbol{x} \in \Omega(t)$ in the Eulerian frame, the diffusivity of the chemical is μ and the reaction term is $f(c)$.

To model information transference between bulk and surface domains, non-zero flux flow boundary conditions are defined

$$-\mu \nabla c\Big|_{\partial\Omega(t)} \cdot \boldsymbol{n} + (\boldsymbol{u} \cdot \boldsymbol{n}) c\Big|_{\partial\Omega(t)} = g\left(c\Big|_{\partial\Omega(t)}, \, c_s \right), \tag{4.3}$$

where the second term accounts for the advective component of flux on the moving boundary; g is the reaction term on the boundary which also depends on a second surface-bound chemical species c_s[1], and $\boldsymbol{n}$ is the outward facing unit normal vector to the domain. Let the boundary chemical c_s be modelled similarly to the one-dimensional reaction-diffusion equations introduced in earlier chapters,

$$\frac{\partial c_s}{\partial t}\Big|_{\boldsymbol{x}} + \nabla_\Gamma \cdot (\boldsymbol{u} c_s) = D_s \Delta_\Gamma c_s + g\left(c\Big|_{\partial\Omega(t)}, \, c_s \right) + h(c_s), \tag{4.4}$$

where Δ_Γ is the Laplace-Beltrami operator on $\partial\Omega(t)$ defined in (1.9), D_s is the diffusivity of c_s and $h(c_s)$ is the surface reaction term.

Recall the terminology and definitions introduced in Section 1.3 which allow us to change the problem (4.2), expressed in the physical domain, to one which is expressed in a reference ALE domain. It follows from there that (4.2) can be re-written in the

[1]Note that s is a subscript to represent that c_s is a surface-bound species and is not short hand partial derivative notation.

ALE frame as

$$\frac{\partial c}{\partial t}\Big|_{\boldsymbol{\xi}} - \boldsymbol{w}\cdot\nabla c - \mu\Delta c + \nabla\cdot(\boldsymbol{u}c) = f(c), \tag{4.5}$$

or, as $\nabla\cdot(\boldsymbol{u}c) = c(\nabla\cdot\boldsymbol{u}) + \boldsymbol{u}\cdot\nabla c$,

$$\frac{\partial c}{\partial t}\Big|_{\boldsymbol{\xi}} - \mu\Delta c + (\boldsymbol{u}-\boldsymbol{w})\cdot\nabla c + (\nabla\cdot\boldsymbol{u})c = f(c). \tag{4.6}$$

Since

$$\begin{aligned} c\nabla\cdot\boldsymbol{w} + \nabla\cdot((\boldsymbol{u}-\boldsymbol{w})c) &= c\nabla\cdot\boldsymbol{w} + \nabla\cdot(\boldsymbol{u}c) - \nabla\cdot(\boldsymbol{w}c) \\ &= c\nabla\cdot\boldsymbol{w} + \nabla\cdot(\boldsymbol{u}c) - \boldsymbol{w}\cdot\nabla c - c\nabla\cdot\boldsymbol{w} \\ &= \nabla\cdot(\boldsymbol{u}c) - \boldsymbol{w}\cdot\nabla c, \end{aligned}$$

then (4.5) can also be re-written as

$$\frac{\partial c}{\partial t}\Big|_{\boldsymbol{\xi}} - \mu\Delta c + c\nabla\cdot\boldsymbol{w} + \nabla\cdot((\boldsymbol{u}-\boldsymbol{w})c) = f(c). \tag{4.7}$$

In an analogous manner to the bulk, the evolution of surface-based species can also be rewritten as

$$\frac{\partial c_s}{\partial t}\Big|_{\boldsymbol{\xi}} + \nabla_\Gamma\cdot(\boldsymbol{u}c_s) - \boldsymbol{w}\cdot\nabla_\Gamma c = D_s\Delta_\Gamma + g\left(c\Big|_{\partial\Omega(t)},\ c_s\right) + h(c_s). \tag{4.8}$$

4.2.1 Weak ALE formulation

To define a weak formulation for (4.7) we consider test functions $\hat{v} \in H^1(\Omega_c)$ and the corresponding test function space on the ALE frame is then

$$\mathcal{H}(\Omega(t)) = \left\{v : \Omega(t) \to \mathbb{R} : v = \hat{v}\circ\mathcal{A}_t^{-1},\ \hat{v}\in H^1(\Omega_c)\right\}$$

for the bulk species and

$$\mathcal{H}_s(\partial\Omega(t)) = \left\{v_s : \partial\Omega(t) \to \mathbb{R} : v_s = \hat{v}_s\circ\mathcal{A}_t^{-1},\ \hat{v}_s\in H^1(\partial\Omega_c)\right\}$$

for the surface-bound species.

Reynolds transport formula states that if $\psi(x,t)$ is a function defined on $\Omega(t)$, and $V_t \subseteq \Omega(t)$ such that $V_t = \mathcal{A}_t(V_c)$ with $V_c \subseteq \Omega_c$, then

$$\frac{d}{dt}\int_{\Omega(t)} v\psi \,\mathrm{d}\boldsymbol{x} = \int_{\Omega(t)} \left(\frac{\partial \psi}{\partial t}\Big|_{\boldsymbol{\xi}} + \psi \nabla \cdot \boldsymbol{w} \right) v \,\mathrm{d}\boldsymbol{x}. \tag{4.9}$$

If (4.7) is multiplied by test function $v \in \mathcal{H}(\Omega(t))$ and integrated over the current physical domain $\Omega(t)$ then

$$\int_{\Omega(t)} \left(\frac{\partial c}{\partial t}\Big|_{\boldsymbol{\xi}} + c\nabla \cdot \boldsymbol{w} + \nabla \cdot ((\boldsymbol{u} - \boldsymbol{w})c) - \mu \Delta c \right) v \,\mathrm{d}\boldsymbol{x} = \int_{\Omega(t)} f(c) v \,\mathrm{d}\boldsymbol{x},$$

from which, using (4.9) and integrating by parts, it follows that

$$\begin{aligned} \frac{d}{dt}\int_{\Omega(t)} cv \,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} (\nabla \cdot ((\boldsymbol{u} - \boldsymbol{w})c))v \,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} \mu \nabla c \cdot \nabla v \,\mathrm{d}\boldsymbol{x} \\ = \int_{\Omega(t)} f(c) v \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega(t)} \left(\mu \nabla c\Big|_{\partial\Omega_t} \cdot \boldsymbol{n} \right) \,\mathrm{d}s. \end{aligned} \tag{4.10}$$

Inserting condition (4.3) and collecting terms, (4.10) becomes

$$\begin{aligned} \frac{d}{dt}\int_{\Omega(t)} cv \,\mathrm{d}\boldsymbol{x} - \int_{\Omega(t)} ((\boldsymbol{u} - \boldsymbol{w})c) \cdot \nabla v \,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} \mu \nabla c \cdot \nabla v \,\mathrm{d}\boldsymbol{x} \\ = \int_{\Omega(t)} f(c) v \,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega(t)} (((\boldsymbol{u} - \boldsymbol{w})c) \cdot \boldsymbol{n}) v \,\mathrm{d}s \\ + \int_{\partial\Omega(t)} \left(g\left(c\Big|_{\partial\Omega(t)},\, c_s \right) - (\boldsymbol{u} \cdot \boldsymbol{n}) c\Big|_{\partial\Omega(t)} \right) v \,\mathrm{d}s, \qquad \forall\, v \in \mathcal{H}(\Omega(t)). \end{aligned} \tag{4.11}$$

The boundary species has a similar weak ALE form where

$$\begin{aligned} \frac{d}{dt}\int_{\partial\Omega(t)} c_s v_s \,\mathrm{d}s + \int_{\partial\Omega(t)} \left(\nabla_\Gamma \cdot ((\boldsymbol{u} - \boldsymbol{w})c_s) \right) v_s \,\mathrm{d}s \\ + D_s \int_{\partial\Omega(t)} \nabla_\Gamma c_s \cdot \nabla_\Gamma v_s \,\mathrm{d}s = \int_{\partial\Omega(t)} (g + h) v_s \,\mathrm{d}s, \qquad \forall\, v_s \in \mathcal{H}_s(\partial\Omega(t)). \end{aligned} \tag{4.12}$$

4.2.2 Finite element discretisation

We will assume that, for each $t \in [0,T]$, the physical and reference domains $\Omega(t)$ and Ω_c are approximated by polygonal domains $\Omega_h(t)$ and $\Omega_{c,h}$, respectively. We will

assume that $\Omega_{c,h}$ is covered by a fixed triangulation $\mathcal{T}_{c,h}$ with straight edges, so that $\Omega_{c,h} = \cup_{K\in\mathcal{T}_{c,h}} K$. The approximation of the boundary domain $\partial\Omega_h(t)$ is chosen to be simply the boundary of $\Omega_h(t)$. The total number of vertices in $\mathcal{T}_{c,h}$ will be denoted N. The number of vertices on the boundary will be denoted by N_s. The Lagrangian finite element spaces for the bulk and surface species respectively are thus defined as

$$\begin{aligned}\mathcal{L}^1(\Omega_{c,h}) &= \left\{\hat{v}_h \in H^1(\Omega_{c,h}) :\ \hat{v}_h|_K \in \mathbb{P}_1(K),\ \forall K \in \mathcal{T}_{c,h}\right\},\\ \mathcal{L}^1_0(\Omega_{c,h}) &= \left\{\hat{v}_h \in \mathcal{L}^1(\Omega_{c,h}) :\ \hat{v}_h = 0,\ \boldsymbol{\xi} \in \partial\Omega_{c,h}\right\},\end{aligned} \tag{4.13}$$

and

$$\mathcal{L}^1(\partial\Omega_{c,h}) = \left\{\hat{v}_{s,h} \in H^1(\partial\Omega_{c,h}) :\ \hat{v}_{s,h}|_K \in \mathbb{P}_1(e),\ \forall e \in \mathbb{E}(\partial\Omega_{c,h})\right\}; \tag{4.14}$$

where $\mathbb{P}_1(K)$ denotes the space of piecewise linear real-valued functions defined on the triangular element K and $\mathbb{P}_1(e)$ denotes the piecewise linear real-valued functions defined on the edge $e \in \mathbb{E}(\partial\Omega_{c,h})$ with

$$\mathbb{E}(\partial\Omega_{c,h}) = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \ :\ \text{ there exists an edge between } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ on } \partial\Omega_{c,h}\ \} \tag{4.15}$$

being the set of straight line segments which make up the polygonal boundary of $\Omega_{c,h}$.

A temporal discretisation of (4.12) is first obtained by subdividing $[0, T]$ into N_t equal time intervals of size $\Delta t = T/N_t$ and denoting $t^n = n\Delta t$, $n = 0, 1, \ldots, N_t$. The ALE mapping is discretised using linear interpolation between time levels such that

$$\mathcal{A}_{h,\Delta t}(\boldsymbol{\xi}, t) = \frac{t - t^n}{\Delta t}\mathcal{A}_{h,t^{n+1}}(\boldsymbol{\xi}) + \frac{t^{n+1} - t}{\Delta t}\mathcal{A}_{h,t^n}(\boldsymbol{\xi}), \qquad t \in [t^n,\ t^{n+1}), \tag{4.16}$$

where $\mathcal{A}_{h,t}$ is the piecewise linear map at time t. The mesh velocity is therefore piecewise constant in time and is given by

$$\begin{aligned}\boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{\xi}) &= \frac{\mathcal{A}_{h,t^{n+1}} - \mathcal{A}_{h,t^n}}{\Delta t}, \qquad t \in [t^n,\ t^{n+1}),\\ \boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{x}) &= \boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{\xi}) \circ \mathcal{A}^{-1}_{h,\Delta t}(\boldsymbol{x}).\end{aligned} \tag{4.17}$$

Let the test space of admissible piecewise linear functions on $\Omega_h(t)$ be defined as

$$\mathcal{H}_h\left(\Omega_h(t)\right) = \left\{ v_h : \Omega_h(t) \to \mathbb{R} : v_h = \hat{v}_h \circ \mathcal{A}_{h,t}^{-1},\ \hat{v} \in \mathcal{L}^1\left(\Omega_{c,h}\right) \right\}, \tag{4.18}$$

and

$$\mathcal{H}_{s,h}\left(\partial\Omega_h(t)\right) = \left\{ v_h : \Omega_h(t) \to \mathbb{R} : v_h = \hat{v}_h \circ \mathcal{A}_{h,t}^{-1},\ \hat{v} \in \mathcal{L}^1\left(\partial\Omega_{c,h}\right) \right\}, \tag{4.19}$$

for the bulk and surface species respectively. The discrete problem is then: find $c_h \in \mathcal{H}_h\left(\Omega_h(t)\right)$ such that

$$\begin{aligned} \frac{d}{dt}\int_{\Omega_h(t)} c_h v_h \,\mathrm{d}\boldsymbol{x} - \int_{\Omega_h(t)} ((\boldsymbol{u}-\boldsymbol{w}_h)c_h)\cdot\nabla v_h \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_h(t)} \mu\nabla c_h\cdot\nabla v_h \,\mathrm{d}\boldsymbol{x} \\ = \int_{\Omega_h(t)} f(c_h) v_h \,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega_h(t)} (((\boldsymbol{u}-\boldsymbol{w}_h)c_h)\cdot\boldsymbol{n}) v_h \,\mathrm{d}s \\ + \int_{\partial\Omega_h(t)} \left(g\left(c_h\Big|_{\partial\Omega_h(t)},\ c_s \right) - (\boldsymbol{u}\cdot\boldsymbol{n}) c_h\Big|_{\partial\Omega_h(t)} \right) v_h \,\mathrm{d}s, \end{aligned} \tag{4.20}$$

for all $v_h \in \mathcal{H}_h\left(\Omega_h(t)\right)$, for the bulk species, and simultaneously find $c_{s,h} \in \mathcal{H}_{s,h}(\partial\Omega(t))$ such that

$$\begin{aligned} \frac{d}{dt}\int_{\partial\Omega_h(t)} c_{s,h} v_{s,h} \,\mathrm{d}s + \int_{\partial\Omega_h(t)} (\nabla_\Gamma\cdot((\boldsymbol{u}-\boldsymbol{w}_h)c_{s,h})) v_{s,h} \,\mathrm{d}s + D_s\int_{\Omega_h(t)} \nabla_\Gamma c_{s,h}\cdot\nabla_\Gamma v_{s,h} \,\mathrm{d}s \\ = \int_{\partial\Omega_h(t)} (g+h) v_{s,h} \,\mathrm{d}s \end{aligned} \tag{4.21}$$

for all $v_{s,h} \in \mathcal{H}_{s,h}(\partial\Omega(t))$, for the surface-bound species.

If $C(t) = \{c_i(t)\}_{i=1}^{N}$ and $C_s(t) = \{c_{s,i}(t)\}_{i=1}^{N_s}$ denote the vectors of nodal unknowns for the approximations c_h and $c_{s,h}$ respectively, where

$$c_h(\boldsymbol{x},t) = \sum_{j=1}^{N} c_j(t)\phi_j(\boldsymbol{x},t), \qquad c_{s,h}(\boldsymbol{x},t) = \sum_{j=1}^{N_s} c_{s,i}(t)\phi_j(\boldsymbol{x},t), \tag{4.22}$$

and $\boldsymbol{w}_h$ denotes the nodal mesh velocities, where

$$\boldsymbol{w}_h(\boldsymbol{x},t) = \sum_{j=1}^{N} \dot{\boldsymbol{x}}_j(t)\phi_j(\boldsymbol{x},t), \tag{4.23}$$

then (4.20) is discretised spatially into the system of ODEs

$$\frac{d}{dt}(M(t)C(t)) + [K(t) + A(t, \boldsymbol{w}_h) - B(t, \boldsymbol{w}_h)]C(t) + D(t, C(t)) = F(C(t)). \quad (4.24)$$

Here

$$[M(t)]_{ij} = \int_{\Omega_h(t)} \phi_i(t)\phi_j(t)\ \mathrm{d}\boldsymbol{x} \quad (4.25)$$

is the time-dependent mass matrix, while

$$\begin{aligned}
[K(t)]_{ij} &= \mu \int_{\Omega_h(t)} (\nabla\phi_j(t) \cdot \nabla\phi_i(t))\ \mathrm{d}\boldsymbol{x}, \\
[A(t, \boldsymbol{w}_h)]_{ij} &= \int_{\partial\Omega_h(t)} [(\boldsymbol{u} - \boldsymbol{w}_h) \cdot \boldsymbol{n}]\phi_i(t)\phi_j(t)\ \mathrm{d}s, \\
[B(t, \boldsymbol{w}_h)]_{ij} &= \int_{\Omega_h(t)} [(\boldsymbol{u} - \boldsymbol{w}_h) \cdot \nabla\phi_j(t)]\phi_i\ \mathrm{d}\boldsymbol{x}, \\
[D(t, C(t), C_s(t))]_i &= \int_{\partial\Omega_h(t)} [g(c_h(t), c_{s,h}(t)) - (\boldsymbol{u} \cdot \boldsymbol{n})c_h(t))]\phi_i(t)\ \mathrm{d}s,
\end{aligned}$$

and the load vector

$$[F(C(t))]_i = \int_{\Omega_h(t)} f(c_h(t))\phi_i(t)\ \mathrm{d}\boldsymbol{x}.$$

On the boundary, the spatial discretisation of (4.21) results in a similar system of ODEs

$$\frac{d}{dt}\left(M_s(t)C_s(t)\right) + \left[K_s(t) + A_s(t, \boldsymbol{w}_h(t))\right] C_s(t) = D_s\left(C(t), C_s(t)\right) + H(C_s(t)), \quad (4.26)$$

where

$$\begin{aligned}
[M_s(t)]_{ij} &= \int_{\partial\Omega_h(t)} \phi_j(t)\phi_i(t)\ \mathrm{d}s, \\
[K_s(t)]_{ij} &= \mu_s \int_{\partial\Omega_h(t)} \nabla_\Gamma\phi_j(t) \cdot \nabla_\Gamma\phi_i(t)\ \mathrm{d}s, \\
[A_s(t)]_{ij} &= \int_{\partial\Omega_h(t)} (\boldsymbol{u} \cdot \boldsymbol{n})\ \phi_i(t)\phi_j(t)\ \mathrm{d}s,
\end{aligned}$$

and

$$[H(C_s^n)]_i = \int_{\partial\Omega_h(t)} h(c_{s,h}(t))\phi_i(t)\ \mathrm{d}s. \quad (4.27)$$

In order to solve the coupled system (4.24) and (4.26), the boundary concentration

$\tilde{C}_s^{n+1}$ is first predicted using a semi-implicit backward Euler method, where the linear diffusion and mesh movement terms are treated implicitly, and the non-linear reaction coupling terms are treated explicitly. The predicted boundary solution therefore satisfies the linear system

$$[M_s^{n+1} + \Delta t(K_s^{n+1} + A_s^{n+1})]\tilde{C}_s^{n+1} = M_s^n C_s^n + \Delta t[D(C^n, C_s^n) + H(C_s^n)], \tag{4.28}$$

To achieve a second order accuracy and minimise any error that will be introduced in the coupling of the two systems, the bulk approximation is then updated using a Crank-Nicolson approximation for the time derivative in (4.24) in a manner similar to (2.19):

$$\begin{aligned}[M^{n+1} + \frac{1}{2}\Delta t(K^{n+1} + A^{n+1} + B^{n+1})]C^{n+1} = [M^n - \frac{1}{2}\Delta t(K^n + A^n + B^n)]C^n \\ + \frac{1}{2}\Delta t[F(C^{n+1}) + F(C^n) - D(C^{n+1}, \tilde{C}_s^{n+1}) - D(C^n, C_s^n)].\end{aligned} \tag{4.29}$$

Also, to address the non-linear terms, note that (4.29) can be written as

$$\omega(C^{n+1}) = 0 \qquad \text{where} \qquad \omega(c) = E(c) - \frac{1}{2}\Delta t F(c) - \frac{1}{2}\Delta t D(c, \tilde{C}_s^{n+1}) - \sigma. \tag{4.30}$$

Here

$$E(c) = \left[M^{n+1} + \frac{1}{2}\Delta t(K^{n+1} + A^{n+1} + B^{n+1})\right] c \tag{4.31}$$

and

$$\sigma = \left[M^n - \frac{1}{2}\Delta t(K^n + A^n + B^n)\right] C^n + \frac{1}{2}\Delta t\left(F(C^n) - D(C^n, C_s^n)\right). \tag{4.32}$$

Using Newton's method, C^{n+1} may be found by iterating to sufficient convergence,

$$J\big|_{c=C^{[n+1,i]}}(C^{[n+1,i+1]} - C^{[n+1,i]}) = -\omega(C^{[n+1,i]}), \qquad i = 1,\ 2,\ \ldots,\ N_{\text{its}}, \tag{4.33}$$

where $C^{[n+1,i]}$ denotes the approximation for C^{n+1} at the i^{th} iteration, N_{its} is the

number of iterations and J is the Jacobian of $\omega(c)$

$$J = \frac{d\omega}{dc} = M^{n+1} + \frac{1}{2}\Delta t(K^{n+1} + A^{n+1} + B^{n+1}) - \frac{1}{2}\Delta t\frac{\partial F}{\partial c} - \frac{1}{2}\Delta t\frac{\partial D}{\partial c}, \tag{4.34}$$

where

$$\left[\frac{\partial F}{\partial c}\right]_{ij} = \frac{\partial F_i}{\partial c_j} \qquad \text{and} \qquad \left[\frac{\partial D}{\partial c}\right]_{ij} = \frac{\partial D_i}{\partial c_j} \tag{4.35}$$

depends on the definition of g. It is worth noting that when $\frac{dg}{dc} = g_0$ is constant then

$$\begin{aligned} \left[\frac{\partial D}{\partial c}\right]_{ij} &= g_0 \int_{\partial\Omega_h(t)} \phi_i\phi_j \,\mathrm{d}s - \int_{\partial\Omega_h(t)} \phi_i\phi_j(\boldsymbol{u}\cdot\boldsymbol{n})_b\phi_b \,\mathrm{d}s \\ \implies \frac{\partial D}{\partial c} &= g_0 M_s^{n+1} - B_s^{n+1}, \end{aligned} \tag{4.36}$$

where

$$\left[B_s^{n+1}\right]_{ij} = \int_{\partial\hat{\Omega}(t)} \phi_i\phi_j(\boldsymbol{u}\cdot\boldsymbol{n})_b\phi_b \,\mathrm{d}s \tag{4.37}$$

and

$$J = M^{n+1} + \frac{1}{2}\Delta t(K^{n+1} + A^{n+1} + B^{n+1}) - \frac{1}{2}\Delta t\frac{\partial F}{\partial c} - \frac{1}{2}\Delta t\left(g_0 M_s^{n+1} - B_s^{n+1}\right). \tag{4.38}$$

Many physical problems take $g(c, c_s)$ and $f(c)$ to be linear with respect to c and c_s. In these instances C^{n+1} can be computed directly as the solution of

$$\begin{aligned} &\left(M^{n+1} + \frac{1}{2}\Delta t(K^{n+1} + A^{n+1} - B^{n+1})\right) C^{n+1} + \frac{1}{2}\Delta t\left(M_s^{n+1} g(C^{n+1}, C_s^{n+1}) - B_s^{n+1}C^{n+1}\right) \\ &\quad = M^n C^n - \frac{1}{2}\Delta t\left(K^n + A^n - B^n\right)C^n + \frac{1}{2}\Delta t\left(F(C^{n+1}) + F(C^n)\right) - \frac{1}{2}\Delta t D(C^n, C_s^n) \end{aligned} \tag{4.39}$$

without the need for Newton iteration.

Finally, a Crank-Nicolson correction step ensures that the boundary solution is second-order in time whereby

$$\begin{aligned} [M_s^{n+1} + \frac{1}{2}\Delta t(K_s^{n+1} + A_s^{n+1})]C_s^{n+1} &= [M_s^n - \frac{1}{2}\Delta t(K_s^n + A_s^n)]C_s^n \\ &+ \frac{1}{2}\Delta t[D(C^{n+1}, \tilde{C}_s^{n+1}) + D(C^n, C_s^n) + H(\tilde{C}_s^{n+1}) + H(C_s^n)]. \end{aligned} \tag{4.40}$$

Note that the correction step only requires the solution of a linear system of equations even if the reaction terms on the surface, c and c_s are non-linearly coupled together. The linear systems arising above are solved using the iterative method BiCGSTAB [56, 140] and an incomplete LU (ILU) factorization as a pre-conditioner.

4.3 Numerical examples

4.3.1 Diffusion on an isotropically growing 2D disc with zero flux boundary conditions

To test the spatial and temporal order of convergence of the ALEFEM scheme (4.29), we first consider the solution of the diffusion equation on an isotropically growing disc domain,

$$\boldsymbol{x}(\boldsymbol{\xi}, t) = (1 + \beta t)\boldsymbol{\xi}, \tag{4.41}$$

where $\boldsymbol{\xi} \in \Omega_c$, the unit circle centred on the origin, and $\beta > 0$ is a constant growth rate. For this case we will assume that the material velocity

$$\boldsymbol{u} = \frac{d\boldsymbol{x}}{dt}\Big|_{\boldsymbol{\xi}} = \beta\boldsymbol{\xi}. \tag{4.42}$$

On this growing domain we consider the diffusion problem

$$\begin{aligned} \frac{\partial c}{\partial t}\Big|_{\boldsymbol{x}} - \mu\Delta c + \nabla\cdot(\boldsymbol{u}c) &= 0, \\ \frac{\partial c}{\partial \boldsymbol{n}}\Big|_{\partial\Omega(t)} &= 0, \end{aligned} \tag{4.43}$$

where $\Omega(t) = \{\boldsymbol{x} \in \mathbb{R}^2 \ : \ ||\boldsymbol{x}||_2 < 1 + \beta t, \ t > 0, \ \beta \in \mathbb{R}^+\}$. We can make use of results by Plaza et al. [116] to transform the isotropically growing disc diffusion problem (4.43) on the time-dependent physical domain into a problem using the stationary coordinate system $\boldsymbol{\xi}$. The standard formulation for the Laplacian in polar coordinates can then

be used to rewrite (4.43) in terms of an equivalent function $c^*(\theta, r, t)$, where

$$\begin{aligned} \frac{\partial c^*}{\partial t}\Big|_{\boldsymbol{\xi}} &= \frac{\mu}{\rho^2}\left(c^*_{rr} + \frac{1}{r^2}c^*_{\theta\theta} + \frac{1}{r}c^*_r\right) - \frac{2\dot{\rho}}{\rho}c^*, \\ \frac{\partial c^*}{\partial r}\Big|_{r=1} &= 0, \end{aligned} \tag{4.44}$$

and where $\rho(t) = 1 + \beta t$ is the linear growth function. The $\frac{2\dot{\rho}}{\rho}c^*$ term in (4.44) accounts for the effect that the change in size of the domain has on the total concentration. We will only consider initial conditions that give rise to radially symmetric solutions where $c^*_{\theta\theta} = 0$. Separable solutions are found by assuming that $c^*(r, t) = f(r)h(t)$, from which (4.44) gives us

$$\begin{aligned} \dot{h}f &= \frac{\mu}{\rho^2}\left(hf'' + \frac{1}{r}hf'\right) - \frac{2\dot{\rho}}{\rho}hf \\ \implies \frac{\dot{h}}{h} &= \frac{\mu}{\rho^2}\left(\frac{f''}{f} + \frac{1}{r}\frac{f'}{f}\right) - \frac{2\dot{\rho}}{\rho} \\ \implies \left(\frac{\dot{h}}{h} + \frac{2\dot{\rho}}{\rho}\right)\frac{\rho^2}{\mu} &= \frac{f''}{f} + \frac{1}{r}\frac{f'}{f}. \end{aligned}$$

Since the left hand side is independent of r and the right hand side is independent of t, it follows that

$$\left(\frac{\dot{h}}{h} + \frac{2\dot{\rho}}{\rho}\right)\frac{\rho^2}{\mu} = K, \tag{4.45}$$

and

$$\frac{f''}{f} + \frac{1}{r}\frac{f'}{f} = K, \tag{4.46}$$

for a fixed constant K. Re-writing (4.45), we note that

$$\dot{h} + \left(\frac{2\dot{\rho}}{\rho} - \frac{\mu K}{\rho^2}\right)h = 0 \tag{4.47}$$

is a linear differential equation in t and hence

$$h(t) = A \exp\left(\int^t \left(-\frac{2\dot{\rho}}{\rho} + \frac{\mu K}{\rho^2}\right)\, \mathrm{d}t^*\right) \tag{4.48}$$

$$= \frac{A}{\rho^2} \exp\left(\int^t \left(\frac{\mu K}{\rho^2}\right)\, \mathrm{d}t^*\right). \tag{4.49}$$

Since $\rho(t) = 1 + \beta t$ then

$$h(t) = \frac{A}{(1+\beta t)^2} \exp\left(\int^t \left(\frac{\mu K}{(1+\beta t^*)^2}\right)\, \mathrm{d}t^*\right) \tag{4.50}$$

$$= \frac{A}{(1+\beta t)^2} \exp\left(\frac{-\mu K}{\beta(1+\beta t)}\right). \tag{4.51}$$

Now, from (4.46),

$$r^2 f'' + r f' - K r^2 f = 0. \tag{4.52}$$

Making the change of variable $s = \sqrt{-K}r$ $(K < 0)$, it then follows that

$$f' = f_s s_r = \sqrt{-K} f_s \qquad \text{and} \qquad f'' = f_{ss} s_r^2 + f_s s_{rr} = -K f_{ss}. \tag{4.53}$$

Therefore we can rewrite (4.52) as

$$r^2 \left(-K f_{ss}\right) + r\left(\sqrt{-K} f_s\right) - K r^2 f = 0 \tag{4.54}$$

and hence

$$s^2 f_{ss} + s f_s + s^2 f = 0. \tag{4.55}$$

Equation (4.55) is a Bessel equation of order zero which has the general solution

$$f(s) = \tilde{A} J_0(s) + \tilde{B} Y_0(s), \tag{4.56}$$

where J_0 and Y_0 are the zero order Bessel functions of the first and second kind, respectively. Using the homogeneous Neumann boundary condition we have $f'(1) = 0$

and hence $\sqrt{-K}f_s(\sqrt{-K}) = 0$. Therefore

$$\tilde{A}\frac{dJ_0}{ds}\Big|_{s=\sqrt{-K}} + \tilde{B}\frac{dY_0}{ds}\Big|_{s=\sqrt{-K}} = 0, \tag{4.57}$$

and, by noting that

$$\frac{dJ_0(s)}{ds} = -J_1(s) \qquad \text{and} \qquad \frac{dY_0(s)}{ds} = -Y_1(s), \tag{4.58}$$

(4.57) can be simplified to

$$\tilde{A}J_1(\sqrt{-K}) + \tilde{B}Y_1(\sqrt{-K}) = 0, \tag{4.59}$$

where J_1 and Y_1 are the first-order Bessel functions of the first and second kind respectively. The general solution is therefore

$$c^*(\theta, r, t) \equiv c^*(r, t) = \frac{A\tilde{A}J_0(\sqrt{-K}r) + A\tilde{B}Y_0(\sqrt{-K}r)}{(1+\beta t)^2} \exp\left(\frac{-\mu K}{\beta(1+\beta t)}\right), \tag{4.60}$$

such that K satisfies (4.59). It follows that the solution in physical variables is

$$c(\boldsymbol{x}, t) = \frac{A\tilde{A}J_0(\sqrt{-K}||\boldsymbol{x}||_2) + A\tilde{B}Y_0(\sqrt{-K}||\boldsymbol{x}||_2)}{(1+\beta t)^2} \exp\left(\frac{-\mu K}{\beta(1+\beta t)}\right). \tag{4.61}$$

The specific solution used in the numerical experiments is obtained by setting $\tilde{B} = 0$ and $A\tilde{A} = \exp(\mu K/\beta)$ so that

$$c(\boldsymbol{x}, t) = \frac{J_0(\sqrt{-K}||\boldsymbol{x}||_2)}{(1+\beta t)^2} \exp\left(\frac{-\mu K}{\beta(1+\beta t)} + \frac{\mu K}{\beta}\right), \tag{4.62}$$

where $K = -14.681971$ and $\beta = 1$.

A numerical mesh is generated using DistMesh [113, 114] such that all edges in the triangulation are no longer than $h_{\max}$. The subsequent domain growth is simulated by isotropically rescaling the initial DistMesh mesh to match the linear growth function, in other words

$$\boldsymbol{x}_i(t^n) = \rho(t^n)\boldsymbol{x}_i(0), \qquad i = 1, 2 \cdots, N. \tag{4.63}$$

In doing so, the connectivity between mesh nodes is unchanged at each time-step and the ALE velocity is equal to the material velocity. Problem (4.43) can be integrated up to $T = 1$ using (4.29) without a surface species (taking $F = D = 0$). The error in the numerical solution is calculated using an approximation for the L_2 norm, namely

$$\begin{aligned} ||e_h(T)||_{L_2} &= \left(\int_0^T \int_{\Omega_h(t)} |c_h(\boldsymbol{x}, t) - c(\boldsymbol{x}, t)|^2 \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \right)^{1/2} \\ &\approx \left(\sum_{m=0}^{N^t-1} \sum_{\chi \in \mathcal{T}_h(t^{m+1/2})} |c_h(\boldsymbol{x}_\chi, t^{m+1/2}) - c(\boldsymbol{x}_\chi, t^{m+1/2})|^2 \, |\chi| \, \Delta t \right)^{1/2}, \end{aligned} \tag{4.64}$$

where $N^t = T/\Delta t$, $\mathcal{T}_h(t^{m+1/2})$ is the set of elements belonging to the triangulation of $\Omega_h(t^{m+1/2})$, the midpoint domain between $\Omega_h(t^m)$ and $\Omega_h(t^{m+1})$, and $\boldsymbol{x}_\chi$ is the centre of mass for element χ. The initial approximation is given exactly at $t = 0$, where $c_h(\boldsymbol{x}, 0) = c(\boldsymbol{x}, 0)$ using (4.62).

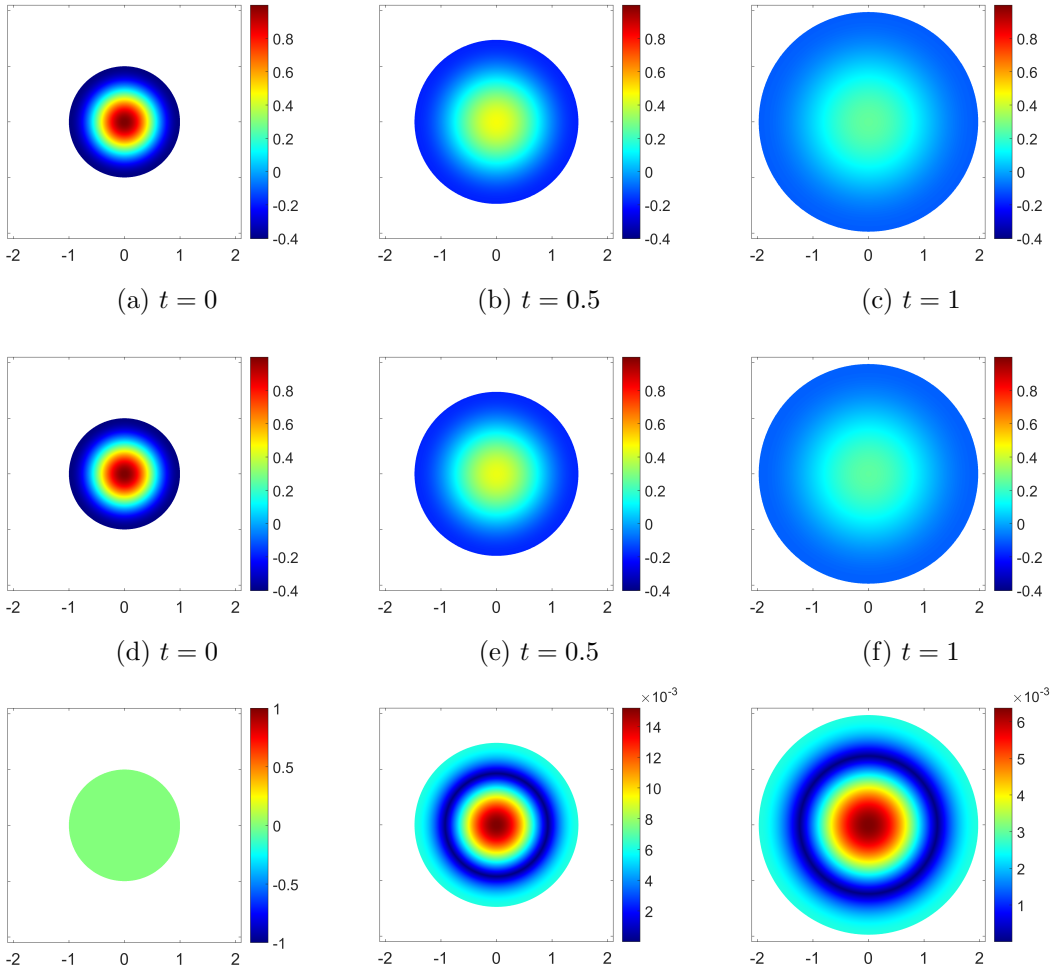
Snapshots at three time points of the approximate solution, the analytical solution and the error are shown in Figure 4.1. We can see that the rapid growth of the domain has resulted in the global redistribution of the initial profile. Results for this experiment demonstrate second-order temporal and spatial convergence which are shown in Tables 4.1 and 4.2. Here N is the number of mesh points or degrees of freedom for the numerical approximation c_h; N_E is the number of elements and $h_{\max}$ is the longest edge length between two adjacent mesh points.

Δt	$\|\|e_h(T)\|\|_{L_2}$	EOC
2.000e-1	2.522e-02	
1.000e-1	6.986e-03	1.852
5.000e-2	1.833e-03	1.930
2.500e-2	4.690e-04	1.967

Table 4.1: Estimated order of temporal convergence and error results for a disc growing at a constant rate where $h_{\max}$ is the longest possible edge within each mesh. ($\mu = 10^{-6}$, $\beta = 1$, $N = 145106$, $N_E = 288944$ and $h_{\max} = 5 \times 10^{-3}$)

N	N_E	$h_{\max}$	$\|\|e_h(T)\|\|_{L_2}$	EOC
2270	4382	4.000e-02	2.519e-04	
9065	17818	2.000e-02	6.497e-05	1.955
36271	71908	1.000e-02	1.562e-05	2.056
145106	288944	5.000e-03	3.805e-06	2.037

Table 4.2: Estimated order of spatial convergence and error results for a disc growing at a constant rate where $h_{\max}$ is the longest possible edge within each mesh. ($\Delta t = 1 \times 10^{-5}$)



(a) $t = 0$ (b) $t = 0.5$ (c) $t = 1$

(d) $t = 0$ (e) $t = 0.5$ (f) $t = 1$

(g) $t = 0$ (h) $t = 0.5$ (i) $t = 1$

Figure 4.1: Solution of the diffusion problem (4.43) on an expanding disc domain. The first row shows the approximation $c_h(\boldsymbol{x}, t)$. The second row shows the analytical solution $c(\boldsymbol{x}, t)$. The third row shows the difference between approximation and analytical solution $|e_h(\boldsymbol{x}, t)| = |c_h(\boldsymbol{x}, t) - c(\boldsymbol{x}, t)|$. Here $\mu = 10^{-6}$, $\beta = 1$, $N = 145106$, $N_E = 288944$, $h_{\max} = 5 \times 10^{-3}$ and $\Delta t = 2.5 \times 10^{-2}$.

4.3.2 A coupled bulk-surface problem

We next consider the coupled diffusion problem

$$\begin{aligned} \frac{\partial c}{\partial t}\Big|_{\boldsymbol{x}} &= \Delta c, \qquad 0 < t \leq T = 1, \qquad && \boldsymbol{x} \in \Omega, \\ \frac{\partial c}{\partial \boldsymbol{n}}\Big|_{\partial\Omega} &= c_s - c, && \boldsymbol{x} \in \partial\Omega, \\ \frac{\partial c_s}{\partial t} &= \Delta_\Gamma c_s - c_s + c, && \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{4.65}$$

where $\Omega = \{\boldsymbol{x} \in \mathbb{R}^2 \ : \ ||\boldsymbol{x}||_2 < 1\}$. On this stationary circular domain the bulk and surface Laplacian operators can be re-written in terms of polar coordinates (θ, r) like before giving

$$\begin{aligned} \frac{\partial c*}{\partial t} &= c_{rr} + \frac{1}{r^2} c_{\theta\theta} + \frac{1}{r} c_r, \\ \frac{\partial c*}{\partial r}\Big|_{r=1} &= c_s^* - c^*\Big|_{r=1}, \\ \frac{\partial c_s^*}{\partial t} &= \frac{\partial^2 c_s^*}{\partial \theta^2} - c_s^* + c^*\Big|_{r=1}. \end{aligned} \tag{4.66}$$

Similarly to Novak et al. [108], we assume a solution of the form

$$\begin{aligned} c_s^*(\theta, t) &= A e^{-k^2 t} \cos\theta, \\ c^*(r, \theta, t) &= f(r) c_s^*(\theta, t). \end{aligned} \tag{4.67}$$

Since

$$c^*\Big|_{r=1} - c_s^* = f(1) c_s^*(\theta, t) - c_s^*(\theta, t) = (f(1) - 1) c_s^*(\theta, t), \tag{4.68}$$

then

$$\frac{\partial c_s^*}{\partial t} = \frac{\partial^2 c_s^*}{\partial \theta^2} + (f(1) - 1) c_s^*(\theta, t). \tag{4.69}$$

Now

$$\begin{aligned} \frac{\partial c_s^*}{\partial t} &= -k^2 A e^{-k^2 t} \cos\theta, \\ \frac{\partial^2 c_s^*}{\partial \theta^2} &= -A e^{-k^2 t} \cos\theta = -c_s^*. \end{aligned} \tag{4.70}$$

Therefore

$$\begin{aligned} -k^2 A e^{-k^2 t}\cos\theta &= -A e^{-k^2 t}\cos\theta + (f(1)-1)A e^{-k^2 t}\cos\theta \\ \implies -k^2 &= -1 + (f(1)-1) \\ \implies f(1) &= 2 - k^2. \end{aligned} \tag{4.71}$$

Subsituting $c^* = f(r)c_s^*$ into (4.70) we get

$$\begin{aligned} f(r)\frac{\partial c_s^*}{\partial t} &= f''(c_s^*) + \frac{1}{r}f'(r)c_s^* + \frac{1}{r^2}f(c_s^*)_{\theta\theta} \\ \implies -k^2 f &= f'' + \frac{1}{r}f' - \frac{1}{r^2}f \\ \implies f'' + \frac{1}{r}f' + \left(k^2 - \frac{1}{r^2}\right)f &= 0 \\ \implies r^2 f'' + rf' + \left((rk)^2 - 1\right)f &= 0. \end{aligned} \tag{4.72}$$

We now make the change of variable $s = rk$ which substituting into (4.72) gives

$$\begin{aligned} \frac{s^2}{k^2}(k^2\ddot{f}) + \frac{s}{k}k\dot{f} + (s^2-1)f &= 0 \\ \implies s^2\ddot{f} + s\dot{f} + (s^2-1)f &= 0. \end{aligned}$$

This is a Bessel equation of order one. The general solution is

$$f(s) = \hat{A}J_1(s) + \hat{B}Y_1(s), \tag{4.73}$$

where J_1 and Y_1 are Bessel functions of order one of the first kind and second kind, respectively and $\hat{A}$ and $\hat{B}$ are constants to be determined. As $Y_1(s)$ is not finite at $s = 0$, then $\hat{B} = 0$. From (4.71) we have

$$\begin{aligned} f(k) + k^2 - 2 &= 0 \\ \implies \hat{A}J_1(k) + k^2 - 2 &= 0. \end{aligned} \tag{4.74}$$

From the flux boundary condition

$$f'c_s^* = (1 - f(1))c_s^*$$
$$\implies f'(1) + f(1) = 1.$$

As $f' = k\dot{f}$ then

$$k\hat{A}\dot{J}_1(k) + \hat{A}J_1(k) = 1 \tag{4.75}$$
$$\implies k\hat{A}\dot{J}_1(k) = k^2 - 1 \tag{4.76}$$

by using (4.74). Equations (4.74) and (4.76) form a system of two equations for the two unknowns k and $\hat{A}$. We can eliminate $\hat{A}$ by taking $(k^2 - 2)$ to the right hand side of (4.74) and then dividing into (4.76) so that

$$k\frac{\dot{J}_1(k)}{J_1(k)} = \frac{k^2 - 1}{2 - k^2} \tag{4.77}$$
$$\implies (2 - k^2)k\dot{J}_1(k) + (1 - k^2)J_1(k) = 0. \tag{4.78}$$

Using the identity $\dot{J}_1 = \frac{1}{2}(J_0 - J_2)$, k can be determined by solving the non-linear algebraic equation

$$(2 - k^2)k\left(\frac{1}{2}(J_0(k) - J_2(k)) + (1 - k^2)J_1(k)\right) = 0. \tag{4.79}$$

The Matlab function *fsolve* can be used to find numerical solutions for (4.79). Once a particular value for k is calculated then

$$\hat{A} = \frac{2 - k^2}{J_1(k)}, \tag{4.80}$$

and so the full solution is as follows,

$$c^*(r, \theta, t) = \hat{A}J_1(rk)Ae^{-k^2 t}\cos\theta. \tag{4.81}$$

For simplicity we can take

$$A = \frac{1}{\hat{A}} = \frac{J_1(k)}{2 - k^2} \tag{4.82}$$

so that

$$\begin{aligned} c^*(r, \theta, t) &= J_1(rk)e^{-k^2 t} \cos\theta, \\ c_s^*(\theta, t) &= \frac{J_1(k)}{2 - k^2} e^{-k^2 t} \cos\theta, \end{aligned} \tag{4.83}$$

and hence

$$\begin{aligned} c(\boldsymbol{x}, t) &= J_1(||\boldsymbol{x}||_2 k)e^{-k^2 t} \cos\theta, \\ c_s(\boldsymbol{x}, t) &= \frac{J_1(k)}{2 - k^2} e^{-k^2 t} \cos\theta. \end{aligned} \tag{4.84}$$

The numerical experiments are conducted using the exact solution (4.84) with $k = 1.177706027$ with the L_2 error on the surface solution being defined similarly as to the bulk, where

$$\begin{aligned} ||e_{s,h}(T)||_{L_2} &= \sqrt{\frac{\Delta t}{2} \sum_{n=0}^{N_t - 1} \left(||e_{s,h}(t^n)||_{L_2}^2 + ||e_{s,h}(t^{n+1})||_{L_2}^2 \right)}, \\ ||e_{s,h}(t^n)||_{L_2} &= \sqrt{\sum_{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in \mathbb{E}(\partial\Omega_h(t^n))} \frac{1}{2} ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \left(||e_{s,h}(\boldsymbol{x}_i, t^n)||_{L_2}^2 + ||e_{s,h}(\boldsymbol{x}_j, t^n)||_{L_2}^2 \right)}, \\ e_{s,h}(\boldsymbol{x}_i, t^n) &= c_{s,h}(\boldsymbol{x}_i, t^n) - c_s(\boldsymbol{x}_i, t^n), \end{aligned} \tag{4.85}$$

and

$$\mathbb{E}\left(\partial\Omega_h(t^n)\right) = \left\{ \{\boldsymbol{x}_i, \boldsymbol{x}_j\} \ : \ \text{there exists an edge between } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ on } \partial\Omega_h(t^n) \right\}, \tag{4.86}$$

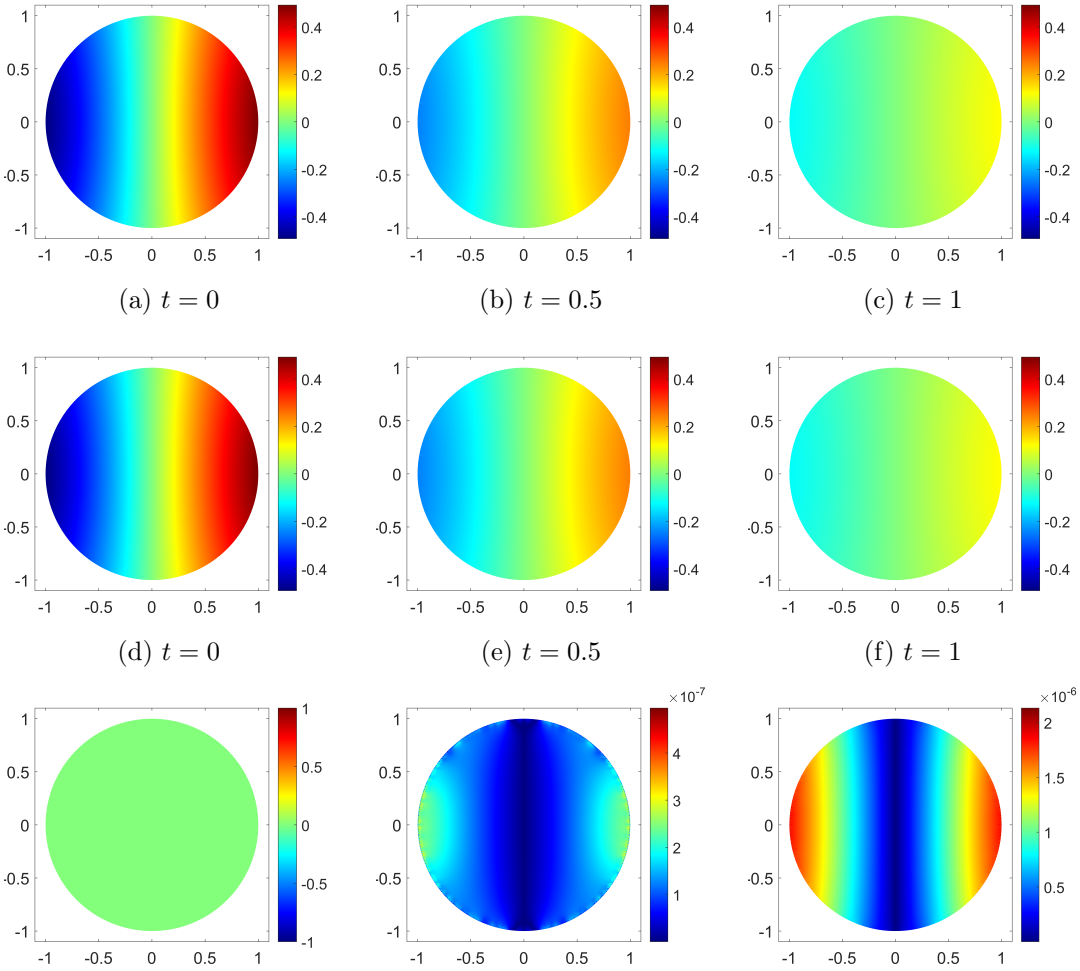
is the set of edges which make up the mesh boundary. Plots of the numerical solution, exact solution and corresponding errors are given in Figure 4.2 and Figure 4.3 at various times on the bulk $\Omega(t)$ and boundary $\partial\Omega(t)$, respectively. The results shown in Table 4.3 and 4.4 demonstrate the estimated order of convergence of the method to be second-order accurate both spatially and temporally, respectively, for bulk and surface species.

N	N_E	$h_{\max}$	$\|\|e_h(T)\|\|_{L_2}$	EOC	$\|\|e_{s,h}(T)\|\|_{L_2}$	EOC
2270	4382	4.000e-02	4.637e-05		1.011e-04	
9065	17818	2.000e-02	1.085e-05	2.095	2.459e-05	2.040
36271	71908	1.000e-02	2.543e-06	2.093	5.864e-06	2.068
145106	288944	5.000e-03	6.299e-07	2.013	1.458e-06	2.008

Table 4.3: Estimated order of spatial convergence and error results for coupled bulk-surface reaction-diffusion systems. ($\Delta t = 1 \times 10^{-5}$)

Δt	$\|\|e_h(T)\|\|_{L_2}$	EOC	$\|\|e_{s,h}(T)\|\|_{L_2}$	EOC
2.000e-01	6.169e-04		4.133e-03	
1.000e-01	1.460e-04	2.079	1.042e-03	1.988
5.000e-02	3.609e-05	2.016	2.631e-04	1.986
2.500e-02	9.000e-06	2.004	6.710e-05	1.971

Table 4.4: Estimated order of temporal convergence and error results for coupled bulk-surface reaction-diffusion systems. ($N = 145106$, $N_E = 288944$, $h_{\max} = 5 \times 10^{-3}$)

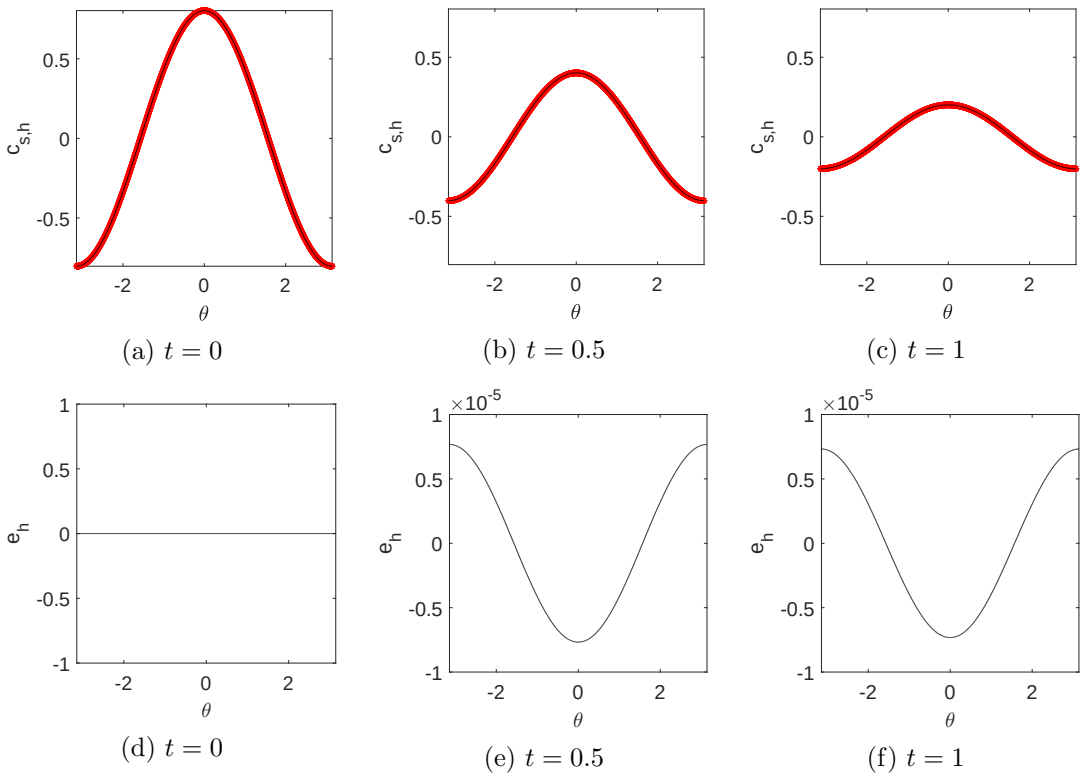


(a) $t = 0$ (b) $t = 0.5$ (c) $t = 1$

(d) $t = 0$ (e) $t = 0.5$ (f) $t = 1$

(g) $t = 0$ (h) $t = 0.5$ (i) $t = 1$

Figure 4.2: Solution of the bulk in the coupled bulk-surface model problem. The first row shows the approximation $c_h(\boldsymbol{x}, t)$. The second row shows analytical solution $c(\boldsymbol{x}, t)$. The third row shows the difference between approximation and analytical solution $|e_h(\boldsymbol{x}, t)| = |c_h(\boldsymbol{x}, t) - c(\boldsymbol{x}, t)|$. Here $N = 145106$, $N_E = 288944$, $h_{\max} = 5 \times 10^{-3}$ and $\Delta t = 1 \times 10^{-2}$.



(a) $t = 0$ (b) $t = 0.5$ (c) $t = 1$

(d) $t = 0$ (e) $t = 0.5$ (f) $t = 1$

Figure 4.3: Solution on the surface in the coupled bulk-surface model problem. The first row shows the approximation $c_{s,h}(\boldsymbol{x}, t)$ in red and the analytical solution $c_s(\boldsymbol{x}, t)$ in black. The second row shows the difference between the two $e_h(\boldsymbol{x}, t) = c_{s,h}(\boldsymbol{x}, t) - c_h(\boldsymbol{x}, t)$. Here $N = 145106$, $N_E = 288944$, $h_{\max} = 5 \times 10^{-3}$ and $\Delta t = 1 \times 10^{-2}$.

4.4 Two-dimensional grid generation

It is possible to extend the numerical schemes developed in this and previous chapters to higher dimensions but before doing so a method for generating robust computational meshes which evolve with time must be developed. There currently exist a number of well known mesh generation strategies for two-dimensional domains (for example in [51, 65, 136, 137, 144, 145]). The DistMesh algorithm ([113] and [114]) is one such procedure for generating static meshes for domains defined implicitly in two or three dimensions. DistMesh generates an initial approximation for the mesh and then gradually refines the mesh based on an iterative process of mechanical relaxation for each edge in the mesh until an ideal mesh is generated which has the correct length constraint, that is to say that no edge within the mesh is greater than a specified tolerance $h_{\max}$ in length. However, DistMesh is a comparatively slow algorithm and there exist other mesh generation strategies to evolve the meshes continuously between time steps.

Another strategy for mesh generation is to use a variational formulation. One advantage over other methods is that mesh quality control and solution adaption can straightforwardly be incorporated into an elliptic mesh generation system which is formed from a variational approach. Further, unlike algebraic grid generators, the mesh produced by an elliptic system will have desirable smoothness characteristics. One disadvantage of variational mappings however is that a solution may not exist or be unique and the domains over which they are applied to need to have should be continuous in order to make use of transformation relations key to the mapping processes. This limits their use in, for instance, problems with topology changes which would introduce discontinuities in the mapping.

The moving mesh FEM approach of Cao et al. [29] is a variational mapping technique and can be seen in Figure 4.4 to generate a mesh for the evolving domain of the moving cell. The mapping is found which maps from a reference domain $\Omega_c \subset \mathbb{R}^2$ to the physical domain $\Omega(t) \subset \mathbb{R}^2$ at time t as demonstrated in Figure 4.4. A natural choice for the reference domain is often $\Omega_c = \Omega(0)$.

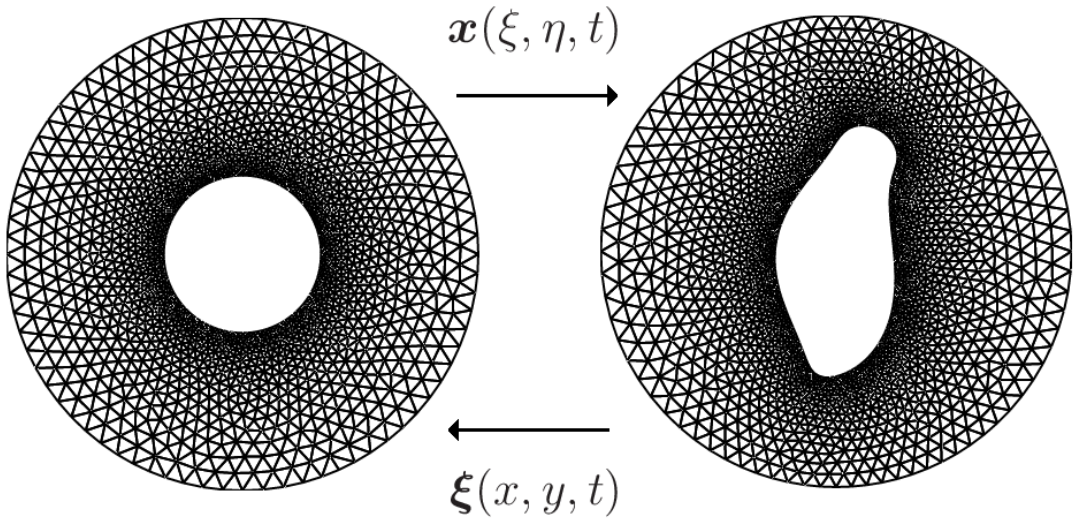


Figure 4.4: On the left hand side, the domain $\Omega_c = \{\boldsymbol{\xi} \in \mathbb{R}^2 \times \mathbb{R} \ : \ 0.1 < ||\boldsymbol{\xi}|| < 0.3\}$ is triangulated using DistMesh. The resulting mesh is mapped onto a physical domain in the environment of the cell using (4.113). The interior boundary points of the physical domain are found from the mechanical model of the cell membrane introduced in the previous chapter where $\Omega(t) = \{\boldsymbol{x} \in \mathbb{R}^2 \times \mathbb{R} \ : \ 0 < d_{\Gamma(t)}(\boldsymbol{x}) \text{ and } ||\boldsymbol{x} - \boldsymbol{c}_m|| < 0.3\}$, where $\boldsymbol{c}_m$ is the centre of mass for the cell.

Meshes from MMPDEs can however become inaccurate when integrated over large timescales. To address this, usually a re-meshing step is computed to remove any degenerated triangles and smooth the mesh using the same technique which was employed to develop the initial computational mesh. From the point in time where a re-mesh has taken place, $t = t^*$ say, the new computational mesh should be such that $\Omega_c = \Omega(t^*)$.

Given a coordinate system $\boldsymbol{\xi} = (\xi, \eta)$ in the reference domain Ω_c, there exists a corresponding coordinate system $\boldsymbol{x} = (x, y)$ in the physical domain $\Omega(t)$ which is found as the mapping of the mesh on the reference domain through the functional

$$I[\boldsymbol{\xi}] = \frac{1}{2} \int_{\Omega(t)} \left((\nabla \xi)^T G^{-1} (\nabla \xi) + (\nabla \eta)^T G^{-1} (\nabla \eta) \right) \, \mathrm{d}\boldsymbol{x}, \tag{4.87}$$

where G is a 2×2 symmetric positive definite monitor matrix which controls the density of the mesh points allowing areas of high interest to be more accurately resolved or to address numerical difficulties (for example, areas where there are large solution gradients). When Dirichlet boundary conditions are specified, the solution $\boldsymbol{\xi}(\boldsymbol{x})$ which

minimizes the functional $I[\boldsymbol{\xi}]$ must also solve the corresponding Euler-Lagrange equation

$$-\nabla \cdot (G^{-1}\nabla \boldsymbol{\xi}) = 0. \tag{4.88}$$

For an arbitrary differentiable real-valued function Ψ on $\Omega(t)$, it follows that

$$\nabla_{\boldsymbol{\xi}} \Psi = J \nabla \Psi, \qquad \nabla \Psi = J^{-1} \nabla_{\boldsymbol{\xi}} \Psi, \tag{4.89}$$

where J is the Jacobian of the mapping $\boldsymbol{x}(\boldsymbol{\xi})$,

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}, \quad J^{-1} = \frac{1}{|J|} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix}, \quad \text{and} \quad \nabla_{\boldsymbol{\xi}} = \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix}. \tag{4.90}$$

Coordinate transformation relations found by substituting $\Psi = \xi$ and $\Psi = \eta$ into (4.89) can be used to compute the inverse mapping of $\boldsymbol{\xi}(\boldsymbol{x})$ for the physical coordinate system $\boldsymbol{x}(\boldsymbol{\xi})$, namely

$$\nabla \xi = \frac{1}{|J|} \begin{bmatrix} y_\eta \\ -x_\eta \end{bmatrix} = \frac{S\boldsymbol{x}_\eta}{|J|} \quad \text{and} \quad \nabla \eta = \frac{1}{|J|} \begin{bmatrix} -y_\xi \\ x_\xi \end{bmatrix} = -\frac{S\boldsymbol{x}_\xi}{|J|}, \tag{4.91}$$

where $|J| = x_\xi y_\eta - y_\xi x_\eta = \boldsymbol{x}_\xi^T S \boldsymbol{x}_\eta$ and

$$S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{4.92}$$

Furthermore,

$$\begin{aligned} (\nabla \xi)^T G^{-1} \nabla \xi &= \left(\frac{S\boldsymbol{x}_\eta}{|J|} \right)^T G^{-1} \left(\frac{S\boldsymbol{x}_\eta}{|J|} \right) \\ &= \frac{1}{|J|^2} \boldsymbol{x}_\eta^T \left(S^T G^{-1} S \right) \boldsymbol{x}_\eta \\ &= \frac{1}{|J|^2 g} \boldsymbol{x}_\eta^T G \boldsymbol{x}_\eta, \end{aligned} \tag{4.93}$$

where $g = \det(G)$ and the last step has been derived by noting that

$$S^T G^{-1} S = \frac{1}{|G|} G^T = \frac{1}{|G|} G,$$

since G is symmetric. Similarily

$$(\nabla\eta)^T G^{-1}(\nabla\eta) = \frac{1}{|J|^2 g}\boldsymbol{x}_\xi^T G \boldsymbol{x}_\xi. \tag{4.94}$$

Thus we make the substitution

$$I[\boldsymbol{x}] = \frac{1}{2}\int_{\Omega_c}\left(\frac{1}{|J|^2 g}\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta + \frac{1}{|J|^2 g}\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi\right)|J|\,\mathrm{d}\boldsymbol{\xi} \tag{4.95}$$

for which it can be shown that the function $\boldsymbol{x}(\boldsymbol{\xi})$ which minimises $I[\boldsymbol{x}]$ subject to Dirichlet boundary conditions must also solve the corresponding Euler-Lagrange equation [149]

$$\left(\frac{\partial}{\partial\xi}\frac{\partial}{\partial x_\xi} + \frac{\partial}{\partial\xi}\frac{\partial}{\partial y_\xi} + \frac{\partial}{\partial\eta}\frac{\partial}{\partial x_\eta} + \frac{\partial}{\partial\eta}\frac{\partial}{\partial y_\eta}\right)\left(\frac{\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta}{2\boldsymbol{x}_\xi^T S\boldsymbol{x}_\eta g} + \frac{\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi}{2\boldsymbol{x}_\xi^T S\boldsymbol{x}_\eta g}\right) = 0. \tag{4.96}$$

Russell and Huang demonstrate in [65] and [60] (see also [64], [26] and [149]) that (4.96) can be further simplified to

$$\begin{aligned}\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta}{Jg}\right) - \frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G\boldsymbol{x}_\eta}{Jg}\right) = 0,\\ -\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G\boldsymbol{x}_\xi}{Jg}\right) + \frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi}{Jg}\right) = 0.\end{aligned} \tag{4.97}$$

To facilitate the smooth movement of the evolving mapping an augmented gradient flow equation of the functional $I[\boldsymbol{x}]$ is used whereby

$$\frac{\partial\boldsymbol{\xi}}{\partial t} = \frac{P}{\tau\sqrt{g}}\nabla\cdot(G^{-1}\nabla\boldsymbol{\xi}), \tag{4.98}$$

where $\tau > 0$ is a user defined temporal smoothing parameter and $P : \Omega(t) \times I \to \mathbb{R}^+$ is a spatial balancing function chosen such that the mesh movement has a spatially uniform scale [60, 15]. Throughout the following chapters we use $P = J^2$ which has desirable smoothness characteristics for the underlying numerical system. In general, $\tau = 1 \times 10^{-3}$ will be used which is also used for other applications in [13] and [85].

In actuality, independent and dependent variables are interchanged so that the

MMPDE (4.98) is given explicitly in $\boldsymbol{x}$,

$$\begin{aligned}\frac{\partial \boldsymbol{x}}{\partial t} =& -\frac{P\boldsymbol{x}_\xi}{\tau|J|\sqrt{g}}\left\{\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta}{|J|g}\right)-\frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G\boldsymbol{x}_\eta}{|J|g}\right)\right\}\\ &-\frac{P\boldsymbol{x}_\eta}{\tau|J|\sqrt{g}}\left\{-\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G\boldsymbol{x}_\xi}{|J|g}\right)+\frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi}{|J|g}\right)\right\}.\end{aligned} \tag{4.99}$$

Further, to make the problem of finding the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ easier to solve with current numerical methods and techniques, (4.99) can actually be expanded and re-written in parabolic form whereby

$$\frac{\partial \boldsymbol{x}}{\partial t} = \frac{P}{\tau}\left(A\boldsymbol{x}_{\xi\xi} + B\boldsymbol{x}_{\xi\eta} + C\boldsymbol{x}_{\eta\eta} + D\boldsymbol{x}_\xi + E\boldsymbol{x}_\eta\right), \tag{4.100}$$

with

$$\begin{aligned}A &= \frac{1}{J^3 g^{3/2}}\left(-\left(\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta\right)\boldsymbol{x}_\xi\boldsymbol{x}_\eta^T S + J\boldsymbol{x}_\eta\boldsymbol{x}_\eta^T G + \left(\boldsymbol{x}_\eta^T G\boldsymbol{x}_\xi\right)\boldsymbol{x}_\eta\boldsymbol{x}_\eta^T S\right),\\ B &= \frac{1}{J^3 g^{3/2}}\left(\left(\boldsymbol{x}_\eta^T G\boldsymbol{x}_\eta\right)\boldsymbol{x}_\xi\boldsymbol{x}_\xi^T S - J\boldsymbol{x}_\xi\boldsymbol{x}_\eta^T G + \left(\boldsymbol{x}_\xi^T G\boldsymbol{x}_\eta\right)\boldsymbol{x}_\xi\boldsymbol{x}_\eta^T S\right) +\\ &\qquad \frac{1}{J^3 g^{3/2}}\left(-\left(\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi\right)\boldsymbol{x}_\eta\boldsymbol{x}_\eta^T S - J\boldsymbol{x}_\eta\boldsymbol{x}_\xi^T G - \left(\boldsymbol{x}_\eta^T G\boldsymbol{x}_\xi\right)\boldsymbol{x}_\eta\boldsymbol{x}_\xi^T S\right),\\ C &= \frac{1}{J^3 g^{3/2}}\left(\left(\boldsymbol{x}_\xi^T G\boldsymbol{x}_\xi\right)\boldsymbol{x}_\eta\boldsymbol{x}_\xi^T S + J\boldsymbol{x}_\xi\boldsymbol{x}_\xi^T G + \left(\boldsymbol{x}_\xi^T G\boldsymbol{x}_\eta\right)\boldsymbol{x}_\xi\boldsymbol{x}_\xi^T S\right),\\ D &= \frac{1}{J^3 g^{1/2}}\left(-\boldsymbol{x}_\eta^T\frac{\partial}{\partial\xi}\left(\frac{G}{g}\right)\boldsymbol{x}_\eta + \boldsymbol{x}_\xi^T\frac{\partial}{\partial\eta}\left(\frac{G}{g}\right)\boldsymbol{x}_\eta\right),\\ E &= \frac{1}{J^3 g^{1/2}}\left(\boldsymbol{x}_\eta^T\frac{\partial}{\partial\xi}\left(\frac{G}{g}\right)\boldsymbol{x}_\xi - \boldsymbol{x}_\xi^T\frac{\partial}{\partial\eta}\left(\frac{G}{g}\right)\boldsymbol{x}_\xi\right) \quad .\end{aligned} \tag{4.101}$$

Equation (4.100) can be further simplified depending on the selection of the monitor matrix G. For our purposes only the monitor matrix proposed by Winslow [149] will be used, namely

$$G = \omega(\boldsymbol{x}, t) I_{2\times 2} \tag{4.102}$$

where $\omega(\boldsymbol{x}, t)$ is a positive weight monitor function and $I_{2\times 2}$ is the 2×2 identity matrix.

The coefficients in (4.101) then simplify to

$$\begin{aligned} A &= \frac{\boldsymbol{x}_\eta^T \boldsymbol{x}_\eta}{\omega J^2}, \\ B &= -\frac{2(\boldsymbol{x}_\eta^T \boldsymbol{x}_\xi)}{\omega J^2}, \\ C &= \frac{\boldsymbol{x}_\xi^T \boldsymbol{x}_\xi}{\omega J^2}, \\ D &= \frac{\omega_\xi \boldsymbol{x}_\eta^T \boldsymbol{x}_\eta - \omega_\eta \boldsymbol{x}_\eta^T \boldsymbol{x}_\xi}{(\omega J)^2}, \\ E &= \frac{-\omega_\xi \boldsymbol{x}_\eta^T \boldsymbol{x}_\xi + \omega_\eta \boldsymbol{x}_\xi^T \boldsymbol{x}_\xi}{(\omega J)^2}. \end{aligned} \tag{4.103}$$

4.4.1 Finite element discretisation

Similar to before, let $\Omega_{c,h} = \bigcup_{K \in \mathcal{T}_{c,h}} K$ denote a polygonal approximation of the domain Ω_c with nodes $\boldsymbol{\xi}_i,\ i = 1, 2, \ldots, N$, where N is the number of mesh points and $\mathcal{T}_{c,h}$ the set of elements which make up the approximation. The corresponding approximation for the physical domain $\Omega(t)$ is given by the mapping $\Omega_h(t) = \boldsymbol{x}(\Omega_{c,h}, t)$ or, equivalently, $\boldsymbol{x}_i(t) = \boldsymbol{x}(\boldsymbol{\xi}_i, t)$, where $\boldsymbol{x}$ satisfies (4.100) with coefficients (4.103) and the corresponding triangulation is denoted by $\mathcal{T}_h(t)$.

An approximate solution of $\boldsymbol{x}(\xi, t)$ is found using the finite element method. In terms of linear basis functions, function $\boldsymbol{v}(\boldsymbol{\xi}) \in \left(\mathcal{L}^1(\Omega_{c,h})\right)^2$ can be written as

$$\boldsymbol{v}(\boldsymbol{\xi}) = \sum_{i=1}^{N} \boldsymbol{v}_i \hat{\phi}_i(\boldsymbol{\xi}), \quad \boldsymbol{v}_i \in \mathbb{R}^{2\times 1}, \quad i = 1, 2, \cdots, N. \tag{4.104}$$

The L_2 inner product on $\Omega_{c,h}$ is denoted

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega_{c,h}} (\boldsymbol{u} \cdot \boldsymbol{v})\, \mathrm{d}\boldsymbol{\xi}. \tag{4.105}$$

To address the non-linearity in (4.100) a semi-implicit backward Euler temporal discretisation is used where the coefficients (A, B, C, D, E) are evaluated at the time-step

$t = t^n$ giving the linear parabolic semi-discretised form

$$\frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{t^{n+1} - t^n} = \frac{P}{\tau}\left(A^n \boldsymbol{x}_{\xi\xi}^{n+1} + B^n \boldsymbol{x}_{\xi\eta}^{n+1} + C^n \boldsymbol{x}_{\eta\eta}^{n+1} + D^n \boldsymbol{x}_{\xi}^{n+1} + E^n \boldsymbol{x}_{\eta}^{n+1}\right). \tag{4.106}$$

Equation (4.106) is multiplied by test function $\boldsymbol{v} \in \left(\mathcal{L}_0^1(\Omega_{c,h})\right)^2$ and integrated over $\Omega_{c,h}$ to obtain the fully discretised weak formulation: find $\hat{\boldsymbol{x}} \in \left(\mathcal{L}^1(\Omega_{c,h})\right)^2$ such that

$$\begin{aligned}\frac{\tau}{P}\left(\frac{\hat{\boldsymbol{x}}^{n+1} - \hat{\boldsymbol{x}}^n}{t^{n+1} - t^n},\ \boldsymbol{v}\right) = &-(A^n \hat{\boldsymbol{x}}_{\xi}^{n+1}, \boldsymbol{v}_{\xi}) - \frac{1}{2}(B^n \hat{\boldsymbol{x}}_{\xi}^{n+1}, \boldsymbol{v}_{\eta}) - \frac{1}{2}(B^n \hat{\boldsymbol{x}}_{\eta}^{n+1}, \boldsymbol{v}_{\xi}) \\ &- (C^n \hat{\boldsymbol{x}}_{\eta}^{n+1}, \boldsymbol{v}_{\eta}) + (D^n \hat{\boldsymbol{x}}_{\xi}^{n+1}, \boldsymbol{v}) + (E^n \hat{\boldsymbol{x}}_{\eta}, \boldsymbol{v}),\end{aligned} \tag{4.107}$$

for all $\boldsymbol{v} \in \left(\mathcal{L}_0^1(\Omega_{c,h})\right)^2$. Let $\hat{\boldsymbol{X}}(t^n) = \{\hat{\boldsymbol{X}}_i(t^n)\}_{i=1}^N = \{x_{h,i}(t^n), y_{h,i}(t^n)\}_{i=1}^N$ be the set of nodal unkowns for $\hat{\boldsymbol{x}}^n$, in other words

$$\hat{\boldsymbol{x}}^n = \sum_{i=1}^{N} \hat{\boldsymbol{X}}_i(t^n)\hat{\phi}_i(\boldsymbol{\xi}), \tag{4.108}$$

and

$$\boldsymbol{x}_h^n = \sum_{i=1}^{N} x_{h,i}(t^n)\hat{\phi}_i(\boldsymbol{\xi}), \qquad \boldsymbol{y}_h^n = \sum_{i=1}^{N} y_{h,i}(t^n)\hat{\phi}_i(\boldsymbol{\xi}). \tag{4.109}$$

Equation (4.107) can be expressed as

$$\begin{bmatrix} M\left(\boldsymbol{x}_h^{n+1} - \boldsymbol{x}_h^n\right) \\ M\left(\boldsymbol{y}_h^{n+1} - \boldsymbol{y}_h^n\right) \end{bmatrix} = \frac{\left(t^{n+1} - t^n\right) P}{\tau} \begin{bmatrix} K(\hat{\boldsymbol{X}}(t^n))\boldsymbol{x}_h^{n+1} \\ K(\hat{\boldsymbol{X}}(t^n))\boldsymbol{y}_h^{n+1} \end{bmatrix}, \tag{4.110}$$

where

$$[M]_{ij} = (\hat{\phi}_i, \hat{\phi}_j) \tag{4.111}$$

and

$$\begin{aligned}[K(\boldsymbol{x})]_{ij} = &-(A(\boldsymbol{x})\hat{\phi}_{i\xi}, \hat{\phi}_{j\xi}) - \frac{1}{2}(B(\boldsymbol{x})\hat{\phi}_{i\xi}, \hat{\phi}_{j\eta}) - \frac{1}{2}(B(\boldsymbol{x})\hat{\phi}_{i\eta}, \hat{\phi}_{j\xi}) \\ &- (C(\boldsymbol{x})\hat{\phi}_{i\eta}, \hat{\phi}_{j\eta}) + (D(\boldsymbol{x})\hat{\phi}_{i\xi}, \hat{\phi}_j) + (E(\boldsymbol{x})\hat{\phi}_{i\eta}, \hat{\phi}_j).\end{aligned} \tag{4.112}$$

In this thesis uniform time-steps are used, $t^{n+1} - t^n = \Delta t$, for which the fully discretised

system is

$$\begin{bmatrix} M\boldsymbol{x}_h^{n+1} - \frac{\Delta t P}{\tau} K(\hat{\boldsymbol{X}}^n)\boldsymbol{x}_h^{n+1} \\ M\boldsymbol{x}_h^{n+1} - \frac{\Delta t P}{\tau} K(\hat{\boldsymbol{X}}^n)\boldsymbol{y}_h^{n+1} \end{bmatrix} = \begin{bmatrix} M\boldsymbol{x}_h^n \\ M\boldsymbol{y}_h^n \end{bmatrix}, \tag{4.113}$$

where $\hat{\boldsymbol{X}}_i^n = \hat{\boldsymbol{X}}_i(t^n)$ and $\hat{\boldsymbol{X}}_i^{n+1} = \hat{\boldsymbol{X}}_i(t^{n+1})$. Let the set of boundary nodes indices be denoted

$$\mathcal{R}_{\partial\Omega_{c,h}} = \{i \in 1, 2, \cdots, N : \boldsymbol{\xi}_i \in \partial\Omega_{c,h}\}. \tag{4.114}$$

The bulk mesh moves according to Dirichlet boundary conditions where at the boundary nodes of the physical domain $\partial\Omega(t)$, it is assumed the mesh position is known. The complete two-dimensional moving mesh method (MMPDE2D) is thus obtained by solving (4.113) in conjunction with the Dirichlet boundary condition

$$\hat{\boldsymbol{X}}_i^{n+1} = \hat{\boldsymbol{x}}(\boldsymbol{\xi}_i, t^{n+1}), \qquad (i \in \mathcal{R}_{\partial\Omega_{c,h}}). \tag{4.115}$$

System (4.113) is a $2N \times 2N$ set of linear equations but can be decoupled into two separate $N \times N$ systems for the $\boldsymbol{x}$ and $\boldsymbol{y}$ components separately whereby

$$Q(\hat{\boldsymbol{X}}^n)\boldsymbol{x}_h^{n+1} = M\boldsymbol{x}_h^n \qquad \text{and} \qquad Q(\hat{\boldsymbol{X}}^n)\boldsymbol{y}_h^{n+1} = M\boldsymbol{y}_h^n, \tag{4.116}$$

where

$$Q(\hat{\boldsymbol{X}}^n) = M - \frac{P\Delta t}{\tau} K(\hat{\boldsymbol{X}}^n). \tag{4.117}$$

To compute solutions to (4.116), an incomplete LU factorization [124],

$$Q(\hat{\boldsymbol{X}}^n) \approx L(\hat{\boldsymbol{X}}^n)U(\hat{\boldsymbol{X}}^n), \tag{4.118}$$

is calculated at each time-step and is used to precondition a BiCGSTAB [56, 140] iterative solver for both components.

A mesh generated for modelling cell movement is shown in Figure 4.4. Here, the physical boundary is found using the cell model introduced earlier by Alorithm 3.1, more specifically by the $\boldsymbol{x}_i$ mesh points which are taken directly from the MMPDE1D method. The reference grid was generated using DistMesh such that the number of

nodes along the inner boundary matched the number of nodes, 200, being used to model the cell membrane. Doing this provides a natural extension of the existing curve mesh being generated from MMPDE1D into the higher dimension mesh without overlap or need to interpolate from one mesh to the other. The mesh in Figure 4.4 does not use adaptivity through the monitor matrix term; instead adaptivity is generated in the initial triangulation and this ensures that the mesh is denser close to the cell membrane to capture activity going on in that area with a higher resolution. In the next chapter we look at how a cell may interact with the environment and chemical background which it perceives. Keeping the mesh density sparse further away from the cell membrane allows us a more efficient computation for this kind of modelling.

An example of MMPDE2D (4.113) to evolve a mesh which adaptively resolves the travelling wave profile $g(\boldsymbol{x}, t) = 1+2\text{sech}^2((1.1+x-5t)/0.04)$ without any re-meshing is shown in Figure 4.5. In Figures 4.5 (a), (b), (c) the boundary conditions are fixed and only the interior mesh points are being relocated. This leads to the triangles close to the boundaries becoming skewed and undesirable. To mitigate this problem, MMPDE1D introduced in Chapter 2, can be used to relocate the mesh points along the boundary without changing the curve which the mesh points are sitting on. The MMPDE1D method is used to derive the mesh points of the ellipse on the interior and makes use of the same adaptivity function for which the bulk mesh is being adapted to. The mesh points generated are at each time-step initially calculated along the boundary and then for the interior with the boundary points found by MMPDE1D feeding then into the MMPDE2D boundary conditions. This is seen in Figures 4.5 (d), (e), (f), where in Figure 4.5 (f) there is noticibly less skewing occuring than in 4.5 (c). A comparison using MMPDE1D when the boundary conditions are simultaneously evolving as well is shown in Figures 4.5 (g), (h), (i).

The quality of a triangle can be measured using the area-length ratio [51]

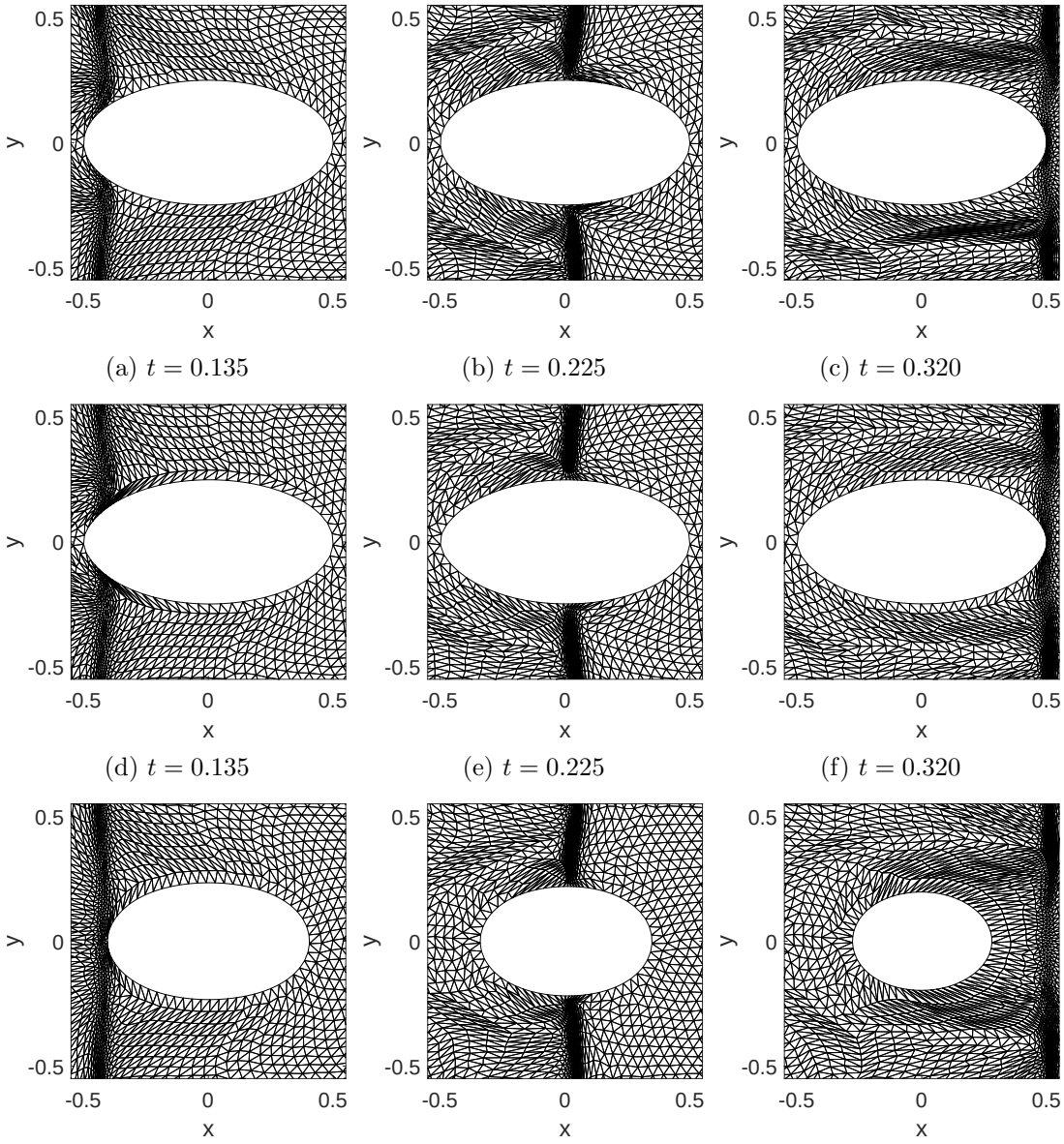
$$q(\rho) = \frac{4}{\sqrt{3}} \frac{A_\rho}{\sum_{k=1}^{3} l_{\rho k}^2/3}, \tag{4.119}$$

where ρ denotes the triangle, with A_ρ the corresponding area of the triangle and l_{ρ_k} the lengths of each of the triangle's edges. The quality measure q_ρ will be one when

the triangle ρ is equilateral and reduces to zero when ρ is degenerate. Thus the mean and minimum quality of $\mathcal{T}_h(t)$, the triangulation of $\Omega_h(t)$, is calculated with

$$q_a(\mathcal{T}_h(t)) = \frac{1}{N_E} \sum_{\rho \in \mathcal{T}_h(t)} q(\rho) \qquad \text{and} \qquad q_m(\mathcal{T}_h(t)) = \min_{\rho \in \mathcal{T}_h(t)} q(\rho), \tag{4.120}$$

respectively and where N_E denotes the number of elements in the triangulation $\mathcal{T}_h(t)$. Table 4.5 shows the mean and minimum triangle quality for the meshes shown in Figure 4.5. Here, it is easy to notice that the average quality for the mesh remains the same between the two test cases; however, the worst triangles of the mesh are improved markedly by including adaption along the boundary using MMPDE1D. This is of course because the most degenerate triangles from the mesh are those close to and on the boundary itself. By using MMPDE1D we can reduce the skewing that occurs between the bulk and boundary meshes. This consequently reduces the number of re-meshing operations required throughout the cell simulations in the next chapter and improves the accuracy of the underlying numerics by having triangles which are more isotropic. In cell simulations where the parameterised finite element method was used to evolve the mesh for the cell membrane, often a re-mesh operation would be required within no more than 100 time-steps due to the skewing that the PFEM introduced. With MMPDE1D being used to evolve the cell mesh this is reduced now to a re-mesh on average every 1000 steps which is significantly faster.



(a) $t = 0.135$ (b) $t = 0.225$ (c) $t = 0.320$

(d) $t = 0.135$ (e) $t = 0.225$ (f) $t = 0.320$

(g) $t = 0.135$ (h) $t = 0.225$ (i) $t = 0.320$

Figure 4.5: Example of a mesh with interior ellipse boundary evolved using MMPDE2D and adapting wave-front function $g(\boldsymbol{x}, t) = 1 + 2\text{sech}^2((1.1 + x - 5t)/0.04)$. Figures (a), (b) and (c) demonstrate MMPDE2D adaption on the interior with fixed boundary conditions. Figures (d), (e) and (f) show MMPDE2D adaption coupled with MMPDE1D being used to simultaneously adapt the mesh points along the boundary. In Figures (g), (h) and (i), the boundary conditions are along evolved so that the ellipse shrinks according to mean curvature flow. ($N = 2260$, $N_E = 4294$, $h_{\max} = 1.5 \times 10^{-2}$)

	Without MMPDE1D adaption		With MMPDE1D adaption	
t	$q_a(\mathcal{T}_h(t))$	$q_m(\mathcal{T}_h(t))$	$q_a(\mathcal{T}_h(t))$	$q_m(\mathcal{T}_h(t))$
0.135	4.414e-01	1.742e-02	4.578e-01	2.691e-02
0.225	4.460e-01	3.741e-02	4.603e-01	5.561e-02
0.320	4.115e-01	3.166e-02	4.178e-01	6.323e-02

Table 4.5: A comparison of triangle quality between the two meshes generated in Figure 4.5 (a), (b), (c) and (d), (e), (f) using MMPDE1D with adaption to move the mesh nodes on the boundary. ($N = 2260$, $N_E = 4294$, $h_{\max} = 1.5 \times 10^{-2}$)

4.5 Picard iteration

When applied to cell migration problems, MMPDE2D (4.106) will only go so far and eventually the computational and physical mesh will need to be regenerated from points where the current physical mesh has been integrated too far from the original reference mesh. In these instances a re-gridding step is needed and can be implemented using DistMesh again to derive a fresh reference and physical mesh. It is possible to extend the period over which re-gridding is not necessary by more accurately solving (4.100).

We can use a simple Picard iteration scheme to greatly improve the accuracy of the mesh at each time-step, whereby

$$\begin{aligned} \hat{\boldsymbol{X}}^{[n+1,0]} &= \hat{\boldsymbol{X}}^{n}, \\ \begin{bmatrix} Q(\hat{\boldsymbol{X}}^{[n+1,m]}) & 0 \\ 0 & Q(\hat{\boldsymbol{X}}^{[n+1,m]}) \end{bmatrix} \hat{\boldsymbol{X}}^{[n+1,m+1]} &= \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \hat{\boldsymbol{X}}^{n}, \\ m &= 0, \ldots, N_{\text{its}} - 1, \end{aligned} \tag{4.121}$$

where $\hat{\boldsymbol{X}}^{[n+1,m]}$ denotes the m^{th} iteration for $\hat{\boldsymbol{X}}^{n+1}$ and $\hat{\boldsymbol{X}}^{n+1} = \hat{\boldsymbol{X}}^{[n+1,N_{\text{its}}]}$.

We can see how this iteration scheme improves accuracy in a simple example. The domain $\Omega(0) = \{\boldsymbol{x} \in \mathbb{R}^2 : ||\boldsymbol{x}||_{L_2} < 1\}$ is triangulated using DistMesh and translated

such that $\Omega(t) = \{(x, y) \in \mathbb{R}^2 : (x - t)^2 + y^2 < 1\},\ 0 \leq t \leq T,\ T = 1$. The error is measured using an the average and maximum displacement from the exact solution for each of the mesh nodes,

$$
\begin{aligned}
||\boldsymbol{e}(t)||_a &= \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_i(t) - x_i^{\text{exact}}(t))^2 + (y_i(t) - y_i^{\text{exact}}(t))^2}, \\
||\boldsymbol{e}(t)||_m &= \max_{1 \leq i \leq N} \sqrt{(x_i(t) - x_i^{\text{exact}}(t))^2 + (y_i(t) - y_i^{\text{exact}}(t))^2}.
\end{aligned}
\tag{4.122}
$$

The error $||e(t)||_a$ gives a measure of the average Euclidean distance between each mesh point calculated using MMPDE2D with Picard iterations (4.121) and the known exact position, the $||e(t)||_m$ error is then the measure of the maximum of these distances. Results, shown in Table 4.6, demonstrate the improved accuracy by using the Picard iteration scheme with N_{its} iterations. As the number of iterations increases, the average and maximum displacement error is reduced and the mesh quality is improved.

N_{its}	$\|\|\boldsymbol{e}(T)\|\|_m$	$\|\|\boldsymbol{e}(T)\|\|_a$	$\lvert q_a(\mathcal{T}_h(T)) - q_a(\mathcal{T}_h(0))\rvert$	$\lvert q_m(\mathcal{T}_h(T)) - q_m(\mathcal{T}_h(0))\rvert$
1	4.996e-04	2.381e-04	8.252e-05	8.121e-05
2	1.669e-04	7.955e-05	2.756e-05	2.713e-05
4	3.358e-05	1.599e-05	5.528e-06	5.451e-06
8	2.016e-06	9.529e-07	3.161e-07	3.172e-07
16	5.684e-08	1.963e-08	1.022e-08	7.413e-09

Table 4.6: Comparison of Picard iteration for MMPDE2D using a different number of iterations. The model solution is a translating disc mesh generated with DistMesh and with other parameters $T = 1$, $N = 1455$, $N_E = 2782$, $h_{\max} = 5 \times 10^{-2}$. Here $\Delta t = 5 \times 10^{-2}$. $q_a(\mathcal{T}_h(0))$ and $q_m(\mathcal{T}_h(0))$ denote the quality of the initial DistMesh generated meshes. It is worth noting that $q_a(\mathcal{T}_h(0)) = 3.982 \times 10^{-1}$ and $q_m(\mathcal{T}_h(0)) = 3.830 \times 10^{-1}$.

The final gridding algorithm in the cell problem, *MMPDE2D-Picard*, is thus as follows:

1. Set $m = 0, \quad \hat{\boldsymbol{X}}^{[n+1,m]} = \hat{\boldsymbol{X}}^{n}$.

2. Do until convergence criteria is met:

   (a) Solve

$$\begin{bmatrix} Q(\hat{\boldsymbol{X}}^{[n+1,m]}) & 0 \\ 0 & Q(\hat{\boldsymbol{X}}^{[n+1,m]}) \end{bmatrix} \hat{\boldsymbol{X}}^{[n+1,m+1]} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \hat{\boldsymbol{X}}^{n}.$$

   This can be done in a number of ways but for efficiency the system $Q(\hat{\boldsymbol{X}}^{[n+1,m]})$ can be decomposed using an incomplete LU decomposition which can then be used as a pre-conditioner to speed up two separate BiCGSTAB [56, 140] calculations to obtain solutions for x and y components of the system separately.

   (b) If $q_m(\mathcal{T}_h^{[m+1]}(t^{n+1})) > 0.15$ then set $\hat{\boldsymbol{X}}^{n+1} = \hat{\boldsymbol{X}}^{[n+1,m+1]}$ and go to 4, otherwise set $m = m + 1$.

   (c) If $m = N_{\text{its}}$ then perform a re-mesh using DistMesh, re-interpolate any numerical variables which rely on the mesh and go to 4.

3. Go to 2.

4. Finish, return $\boldsymbol{X}^{n+1}$ as new grid.

The *MMPDE2D-Picard* method iterates until the mesh quality is good enough, as defined by $q_m(\mathcal{T}_h^{[m+1]}(t^{n+1}))$, where $\mathcal{T}_h^{[m+1]}(t^{n+1})$ is the intermediate mesh for $\mathcal{T}_h(t^{n+1})$ at the $m + 1^{\text{th}}$ iteration. The value of $N_{\text{its}} = 10$ has been found to work well experimentally within the context of the cell problems given in the next chapter. However, if a grid of sufficient quality is not found within N_{its} iterations, then DistMesh is used to generate a new grid.

The mesh points along the boundary $\boldsymbol{x}^{n+1} \in \partial\Omega_h(t^{n+1})$ form a polygon which is used by the DistMesh function *dpoly* to generate an implicit signed distance function which can then be used as the input into *distmesh2d* to generate the new grid $\mathcal{T}_h(t^{n+1})$. In order to prevent any unintended movement of the boundary nodes by the implicit

grid generation algorithm, the new grid has its boundary defined explicitly by the polygon $\boldsymbol{x}^{n+1} \in \partial\Omega_h(t^{n+1})$ using the *pfix* parameter of *distmesh2d* and ensuring the DistMesh edge length constraint parameter *h0* is consistent near the boundary between time-steps.

Once DistMesh has been used to generate a mesh, the number of elements and position of the mesh points will change so any variables which are defined on the old mesh $\mathcal{T}_h(t^n)$ must be interpolated onto the new mesh. On the interior this is straightforwardly done by evaluating the function on the newly generated mesh, in other words for a given quantity $u_h^{\Omega_h(t^n)}(\boldsymbol{x})$ which is defined on $\Omega_h(t^n)$, the corresponding quantity $u_h^{\Omega_h(t^{n+1})}(\boldsymbol{x})$ on $\Omega(t^{n+1})$ is

$$u_h^{\Omega(t^{n+1})}(\boldsymbol{x}^{n+1}) = u_h^{\Omega(t^n)}(\boldsymbol{x}^{n+1}), \qquad \boldsymbol{x}^{n+1} \in \Omega_h(t^{n+1}). \tag{4.123}$$

This will work so long as no newly generated interior mesh points move outside of $\Omega_h(t^n)$. For nodes on the boundary $\partial\Omega_h(t^{n+1})$ which will have been positioned outside of the old domain $\Omega_h(t^n)$, a linear projection is used to determine new values for quantities at these points such that

$$u_h^{\Omega(t^{n+1})}(\boldsymbol{x}^{n+1}) = u_h^{\Omega(t^n)}(\boldsymbol{x}^*), \qquad \boldsymbol{x}^{n+1} \in \partial\Omega_h(t^{n+1}), \tag{4.124}$$

where

$$||\boldsymbol{x}^{n+1} - \boldsymbol{x}^*||_{L_2} \leq ||\boldsymbol{x}^{n+1} - \boldsymbol{\chi}||_{L_2}, \qquad \forall\, \boldsymbol{\chi} \in \partial\Omega_h(t^n). \tag{4.125}$$

This introduces a small interpolation error which is dependent on the mesh velocity at the boundary. In the cell migration problem, the membrane velocity is sufficiently slow that this error is $O(\Delta t)$ and since the underlying PDEs of the Meinhardt system are being solved using a Backward-Euler FEM solver, the interpolation error introduced is minor by comparison to the error in the numerical methods. This being said, there have been recent developments by Anderson et al. [2] using an intermediate reference domain and ALE based convection problem formulation to more accurately reduce interpolation errors for similar re-meshing problems.

4.6 Conclusions

In conclusion:

- A second-order accurate arbitrary Lagrangian-Eulerian finite element method which can be used to solve reaction-diffusion problems on growing domains of different dimensionality (curves in $\mathbb{R}$ and surfaces in $\mathbb{R}^2$) while simultaneously coupling the solutions of each together through a flux boundary condition term is discussed.

- Numerical experiments are presented which show the method converging with second order accuracy both temporally and spatially. This is a non-trivial realisation as one of the experiments showing second order accuracy is actually composed of two systems being simultaneously solved on domains with different dimensionality.

- The moving mesh approach of Cao, Huang and Russell [29] is discussed. This is a variational approach to mesh generation which has some advantages making it a natural choice for mesh generation in the cell problem. Their approach is improved upon for the modelling of cell migration and chemotaxis by generalising the method using an iterative scheme. A method is created which uses a Picard-style iterative scheme for the preposed movement of the mesh in the cell problem.

We now look to use these new methods and the methods of the other chapters to develope the cell migration and chemotaxis model.

Chapter 5

A two-dimensional model for cell migration and chemotaxis

Finally, we get to the crux of the matter. It is time now to bring all of the methods built in the previous chapters together to do what we have wanted to achieve from the beginning: to develop a better model of cell migration and chemotaxis. In this chapter we will seek to do just that. We will make use of both the one-dimensional and two-dimensional ALEFEM solvers to find the concentrations of chemical species both in the cells' membrane and in the cells' immediate environment.

We will do this by extending the model of receptor occupancy given in Chapter 3 into two dimensional whereby using methods for coupling systems of reactions-diffusion equations introduced in Chapter 4 to properly resolve the free-ligand field (3.4) to obtain a more accurate model.

We will then use both MMPDE methods to drive the underlying mesh movement forward, making use of the tangential control MMPDE1D gives us to smooth out the motion of the cell membrane and allow the MMPDE2D method to accurately generate the corresponding mesh for the exterior of the cell without skewing or introducing unwanted mesh degeneration.

To begin with, we derive physiologically correct parameters that we may use in the higher dimensional model.

5.1 Developing a cell migration model in higher dimensions

In the earlier chapter, certain assumptions are made about how the cell interacts with the surrounding environment, as well as what the cell's receptors perceive the free-ligand concentration in the environment to be. The movement of the cell itself through the environment has previously been presumed to have no effect on the ligand concentration and this of course is not strictly true as the cell itself can perturb the environment and affect the concentration of surrounding chemoattractant. We will thus look to use the ALEFEM method discussed in the previous chapter (4.29) to compute the solution to (3.4). In doing so we will improve the model we currently have by improving the accuracy of binding and unbinding process which takes place between the free-ligand and cells' receptors.

We begin by doing a non-dimensionalisation of the parameters from the original cell migration model (given by Algorithm 3.1) so that we may now find the correct values to use for the diffusivity of free and membrane-bound ligand in a two dimensional model.

To more faithfully describe the cell-environment interaction and the receptor-free-ligand binding-unbinding process we revisit (3.2), (3.4) and (3.3), where the membrane receptor occupancy is modelled by the binding-unbinding cytosol-membrane coupled problem

$$\begin{aligned} \frac{\partial L}{\partial t}\Big|_{\boldsymbol{x}} + \nabla_{\Omega(t)} \cdot (L\boldsymbol{u}) &= D\Delta_{\Omega(t)} L, \\ -D_L \frac{\partial L}{\partial \boldsymbol{n}}\Big|_{\Gamma(t)} - [(\boldsymbol{u}\cdot\boldsymbol{n})]L\Big|_{\Gamma(t)} &= k_1(N_r - L_m)L\Big|_{\Gamma(t)} - k_{-1}L_m, \\ \frac{\partial L_m}{\partial t}\Big|_{\boldsymbol{x}} + \nabla_{\Gamma(t)} \cdot (L_m \boldsymbol{u}_s) &= D_s\Delta_{\Gamma(t)} L_m + k_1(N_r - L_m)L\Big|_{\Gamma(t)} - k_{-1}L_m, \end{aligned} \tag{5.1}$$

where L is the concentration of free-ligand in the environment exterior to the cell and L_m is the concentration of receptor-ligand complex that exists on the membrane $\Gamma(t)$. The diffusivity coefficient of the free-ligand is D_L, $\boldsymbol{u}$ is the material velocity of the free-ligand in the bulk, $\boldsymbol{u}_s$ and D_s are the material velocity and diffusivity of the bound ligand-receptors on the membrane, respectively. In the one dimensional model, we made the assumption that there was no lateral diffusion of bound receptor along the

membrane but empirical evidence suggests that, while this value may be small, it is non-zero (see [40], for example). The binding and unbinding rates of the free-ligand to and from the receptors on the membrane are k_1 and k_{-1}, respectively. We will assume no lateral movement of the receptors using a material velocity of $\boldsymbol{u}_s = 0$. Simulations here will be conducted without any flow in the bulk and so the material velocity for the free-ligand is $\boldsymbol{u}\Big|_{\Omega(t)\setminus\Gamma(t)} = 0$. On the cell membrane, however, the material velocity should match the velocity of the interface so that $\boldsymbol{u}\Big|_{\Gamma(t)} = \dot{\boldsymbol{x}}\Big|_{\Gamma(t)}$.

As is done in [86], the coupled bulk-surface system problem (5.1) is non-dimensionalised for computational reasons. Let

$$\bar{\boldsymbol{x}} = \frac{\boldsymbol{x}}{L_*}, \qquad \bar{L} = \frac{L}{l_*}, \qquad \bar{t} = \frac{t}{t_*}, \qquad \bar{D} = \frac{D}{L_*^2/t_*} \tag{5.2}$$

define the non-dimensional variables where L_* is a characteristic length scale, l_* is a characteristic ligand concentration, and t_* a characteristic time scale. Since $\boldsymbol{u}\Big|_{\Omega(t)\setminus\Gamma(t)} = 0$ in the simulations which follow, problem (5.1) can be rescaled to fit the non-dimensional variables (5.2)

$$\frac{\partial \bar{L}}{\partial \bar{t}} = \bar{D}\bar{\nabla}\bar{L}, \tag{5.3}$$

where $\bar{\nabla}$ denotes the Laplace operator with respect to the non-dimensional spatial variables.

Similarly, let

$$\begin{aligned} \bar{L}_m &= \frac{L_m}{(L_m)_*}, \qquad \bar{\boldsymbol{u}} = \frac{\boldsymbol{u}}{L_*/t_*}, \qquad \bar{D}_s = \frac{D_s}{L_*^2/t_*}, \\ \bar{R}_{\text{tot}} &= \frac{R_{\text{tot}}}{(L_m)_*}, \qquad \bar{k}_1 = \frac{k_1}{1/(t_* l_*)}, \qquad \bar{k}_{-1} = \frac{k_{-1}}{1/t_*}, \end{aligned} \tag{5.4}$$

be the non-dimensional variables which exist on the membrane $\Gamma(t)$. Here $(L_m)_*$ is a characteristic concentration of the ligand-receptor complex. It follows that the concentration of reaction-bound ligand in (5.1) is specified in non-dimensional form as

$$\frac{\partial \bar{l}_s}{\partial \bar{t}} + \bar{\nabla}_{\bar{\Gamma}(t)} \cdot (\bar{\boldsymbol{u}}\bar{l}_s) + \bar{k}_1(\bar{R}_{\text{tot}} - \bar{l}_s)\bar{l}\Big|_{\bar{\boldsymbol{x}}\in\bar{\Gamma}(t)} - \bar{k}_{-1}\bar{l}_s. \tag{5.5}$$

Finally, if $(l_s)_* = l_* L_*$, the normal flux condition can be written as

$$-\bar{D}\frac{\partial \bar{l}}{\partial \boldsymbol{n}}\Big|_{\bar{\Gamma}(t)} - [(\bar{\boldsymbol{u}}\cdot\boldsymbol{n})]\bar{l}\Big|_{\bar{\Gamma}(t)} = \bar{k}_1(\bar{R}_{\text{tot}} - \bar{l}_s)\bar{l}\Big|_{\bar{\boldsymbol{x}}\in\bar{\Gamma}(t)} - \bar{k}_{-1}\bar{l}_s. \tag{5.6}$$

The non-dimensional equations are therefore similar in form to the original dimensional equations (5.1) provided $(L_m)_* = l_* L_*$.

As the radius of the cell in the simulations is initially $r_0 = 0.1$ and since a typical Dictyostelium cell is of the order $10\mu m$ in diameter [131], it follows that the approximate characteristic length scale is $L_* = 50\mu m$.

The reference time scale t_* is chosen such that the cell speed obtained from the numerical simulations is $10\mu m \text{min}^{-1}$ which is approximately the speed of a migrating Dictyostelium cell. In the numerical experiments a reference time scale of $t_* = 1/80s$ is used.

The diffusivity of cyclic-AMP, which will be our basis for the free-ligand source, is measured to be around $\bar{D} = 444\mu m^2/s$ [40] (see also [66], [120], [125] and [126]) and so it follows that with the characteristic length and time scales as chosen,

$$\bar{D} = \frac{D}{L_*^2/t_*} = 14.208. \tag{5.7}$$

Similarly, using the physical estimate of the receptor-ligand complex $\bar{D}_s = 0.1\mu m^2/s$ [139], the corresponding non-dimensional coefficient is

$$\bar{D}_s = \frac{D_s}{L_*^2/t_*} = 3.2 \times 10^{-4}. \tag{5.8}$$

The reference ligand concentration is taken to be $l_* = 1nM$. For the non-dimensional and the dimensional flux conditions to be equivalent we therefore set $(L_m)_* = l_* L_* = 50nM\mu m$.

As the surface area of a sphere of radius r is $A(r) = 4\pi r^2$, then with $r = 5\mu m$ we have $A = 100\pi\mu m^2$. An estimate of the total number of receptors per cell is around

7×10^4 [71] and hence the number concentration of receptors is

$$R_{\text{tot}} = \frac{7 \times 10^4}{100\pi} \approx 223\mu m^{-2}. \tag{5.9}$$

An estimate of the non-dimensional value for the total receptor concentration is therefore

$$\bar{R}_{\text{tot}} = \frac{R_{\text{tot}}}{(l_s)_*} = 7.433. \tag{5.10}$$

The far-field boundary of $\Omega(t)$ is moved such that it remains a disc of radius R with its origin in the centre of the moving cell,

$$\Omega(t) = \{\boldsymbol{x} \in \mathbb{R}^2 : 0 \leq d_{\Gamma(t)}(\boldsymbol{x}),\ ||\boldsymbol{x} - \bar{\boldsymbol{x}}|| \leq R\}, \tag{5.11}$$

where $\bar{\boldsymbol{x}}$ is the centre of the cell, R is the radius of the far-field boundary and $d_{\Gamma(t)}(\boldsymbol{x})$ is the signed distance function from the cell membrane $\Gamma(t)$ such that $d_{\Gamma(t)}(\boldsymbol{x}) > 0$ for $\boldsymbol{x}$ exterior to the cell. Because of the relatively low rate of diffusion of the free-ligand molecules and the slow kinetics of the cell, a value of $R = 3r_0 = 0.3$ was selected so that the computational window is approximately three cell-lengths in diameter. This prevents unnecessary calculation of the free-ligand concentration $L(\boldsymbol{x})$ in areas far away from the cell where there would be little or no perceived benefit in doing so. Further, by using a smaller computational window, the solutions of the free-ligand and kinetic processes happening close to the cell are calculated in less time.

The boundary conditions on the far-field boundary are such that the free-ligand is assumed to be in its original unperturbed state, as such

$$L\big|_{||\boldsymbol{x}-\bar{\boldsymbol{x}}||=R} = L_g(\boldsymbol{x}). \tag{5.12}$$

The free-ligand gradient functions from the earlier chapter are reused for a fair comparison whereby, in the case of zero gradient $\rho = 0$,

$$L_g(\boldsymbol{x}) = 5.2941, \tag{5.13}$$

and

$$L_g((x,y)) = 8.7146y + 6.1656, \tag{5.14}$$

for the linear gradient $\rho = 0.2$. The initial condition is taken also to be $L(\boldsymbol{x})\Big|_{t=0} = L_g(\boldsymbol{x})$.

5.2 The cell migration algorithm

The complete algorithm which makes use of methods introduced in Chapter 4 to obtain a more precise model of cell migration is thus as follows:

1. Set initial parameters.
2. Do until end of simulation:
   (a) Use ALEFEM2D with non-zero flux conditions to calculate bulk-surface reaction terms for free-ligand and receptor-bound ligand concentrations.
   (b) Calculate receptor occupancy from free-ligand concentration.
   (c) Solve Meinhardt system for local activator, global and local inhibitor.
   (d) Calculate membrane (normal) velocity.
   (e) Use MMPDE1D to move membrane interface.
   (f) Use MMPDE2D-Picard to move bulk (exterior) mesh. Adjust the far-field boundary condition as appropriate to ensure cell is in the centre of the moving mesh.
   (g) Save and render result as needed.
3. Repeat until completion.

In the simulations which follow, grids are generated initially using DistMesh with $N = 5889$ mesh points and $N_E = 11432$ mesh elements and the simulations evolve with a time-step of $\Delta t = 0.1$. The intermediate movement of the mesh is done using MMPDE1D on the boundary and MMPDE2D-Picard in the bulk. In both instances the temporal smoothing parameter of $\tau = 10^{-3}$ is used to smooth out the tangential

velocities of the membrane mesh points and thus reduce skewing and grid deformation close to the boundary. This reduces the number of re-mesh operations done throughout each simulation using DistMesh down to an average of approximately 1 per 1000 time-steps. At the initial DistMesh grid generation and re-meshing steps a density function is used such that, in the grid generated, the number of mesh points increase closer to the cell membrane, the area of interest. Specifically the number of mesh points go from 146 on the far field boundary to 200 along $\Gamma(t)$. This adaptivity from the reference meshes is maintained by the MMPDEs in the intermediate steps using a monitor matrix of $M = I_{2\times 2}$ where I is the identity matrix. All of the simulations to follow are run to $T = 50,000$ (625 sec).

5.3 Numerical experiments

5.3.1 Cell migration in zero gradient background

A series of snapshots at various times of the cell migration model are shown in Figure 5.1 with the zero gradient free-ligand field (5.13). There is no cell-cell interactions taking place and each of the cells represent a separate single cell simulation with different initial perturbation from the homogeneous steady state. These cells can be seen to migrate randomly in a similar fashion to the original model in Figure 3.2 however with a slightly reduced speed.

In Figures 5.2 and 5.3, the exterior free-ligand field for two cells are shown. Here we can see that the field is not constant close to the cell and that the cell itself is actually influencing the concentration of free-ligand close to the membrane. This type of non-linear behaviour is more in line with real cells but is not captured in the original one-dimensional model. In Figure 5.6 the free-ligand field is shown in differently with the values for L which lie along the $x = 0$ axis being plotted separately. This helps to show the local depletion and local focusing of the concentration close to the cell membrane which is brought about by the cell and may be useful to biologists looking to conduct research into the phenomenon.

Despite this subtle but definite change in the free-ligand field, Figure 5.4 shows

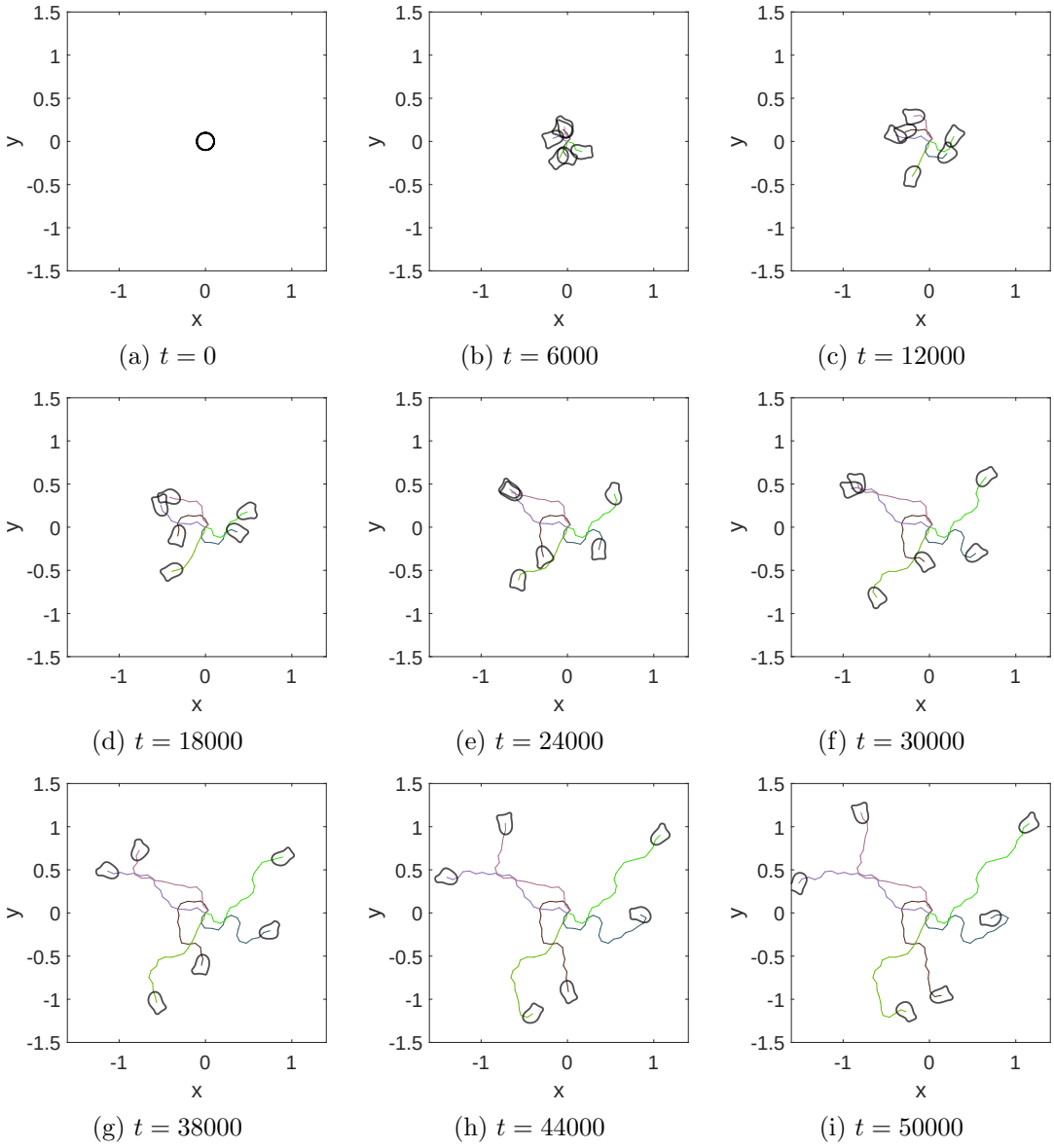
that solutions to the Meinhardt system are roughly the same as in the original model. The concentration of receptor-bound ligand fluctuates slightly as shown in Figure 5.5, unlike in the original model which was constant everywhere. Notice that the formation of pseudopodia happens in much the same way that it did previously and that the change to the receptor concentration has not impeded the cells ability to migrate or changed the morphology of the cells. The free-ligand field close to the cell is perturbed by the cell's movements; this has the knock on effect of making the field non-linear in nature. Here there can be seen a small depletion of free-ligand in the direction which the cell is migrating and a slight increase close to the trailing edge.

This self-generated perturbation in the free-ligand background caused by the cells own movement through the environment is generally known as the "wind-shield" effect and is also seen in real cells and can affect the the ability of highly motile cells to chemotax efficiently in shallow gradients unless some addition mechanisms are employed by the cell such as surface ligand degradation or receptor internalisation [50].

To compare the original and newer model, the mean square displacement, a measure of directionality and persistence can be used where

$$MSD(t) = \frac{1}{N_c} \sum_{n=1}^{N_c} ||\bar{\boldsymbol{x}}_n(t) - \bar{\boldsymbol{x}}_n(0)||^2, \tag{5.15}$$

and where N_c is the number of cells in the simulation and $\bar{\boldsymbol{x}}_n(t)$ is the cell centre at time t. Plots of the trajectories for $N_c = 56$ cell simulations are shown in Figure 5.7. At the end of the simulation, $T = 5 \times 10^4$, the cells from the original model move an average distance of 1.085 from the origin whereas the cells from the newer model move an average distance of 0.746 so in general we see that the cells in the newer model are not going as far as the original model. This is also seen in the corresponding mean square displacements shown in Figure 5.8 where the mean square displacement can be seen to be less persistent for the newer model with the zero gradient. As this is a shallow (zero) gradient and there has been no surface ligand degradation or receptor internalisation modelled then this is not unlike the behaviour described in [50].

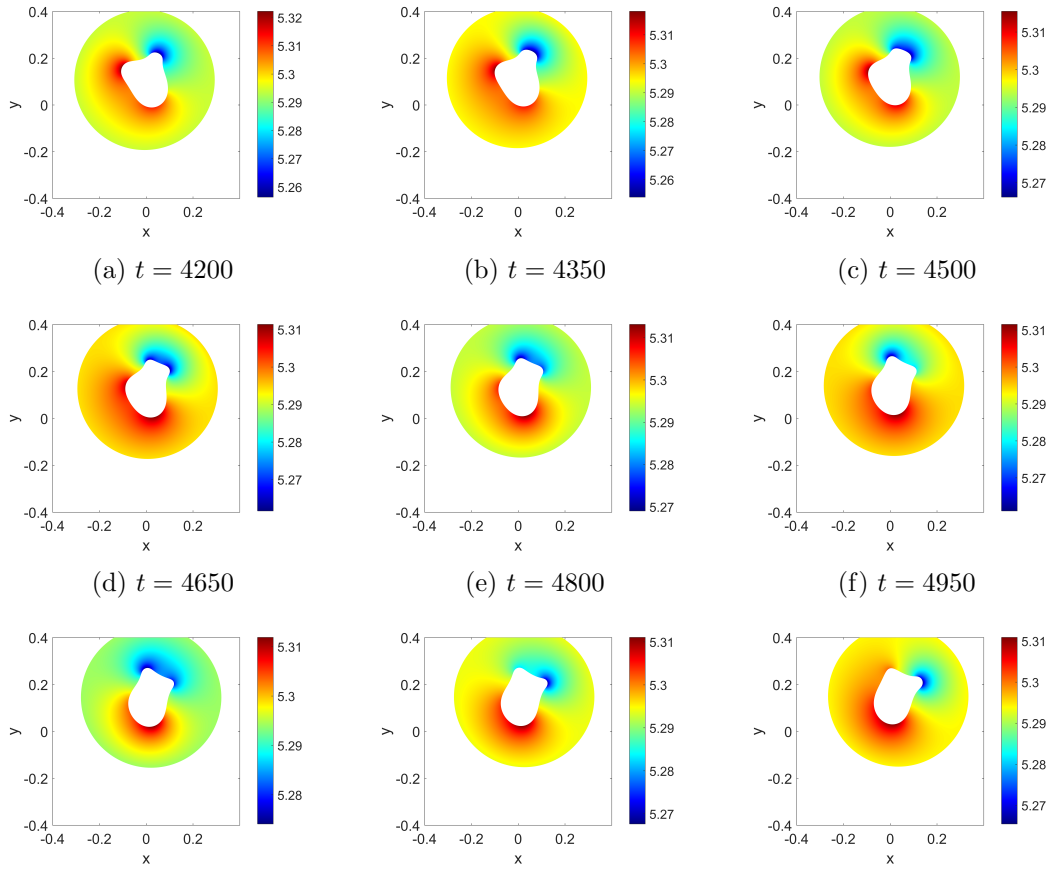


(a) $t = 0$ (b) $t = 6000$ (c) $t = 12000$

(d) $t = 18000$ (e) $t = 24000$ (f) $t = 30000$

(g) $t = 38000$ (h) $t = 44000$ (i) $t = 50000$

Figure 5.1: Snapshots of evolving cells and corresponding trajectory plots at various times. Initial ligand field is homogeneous.

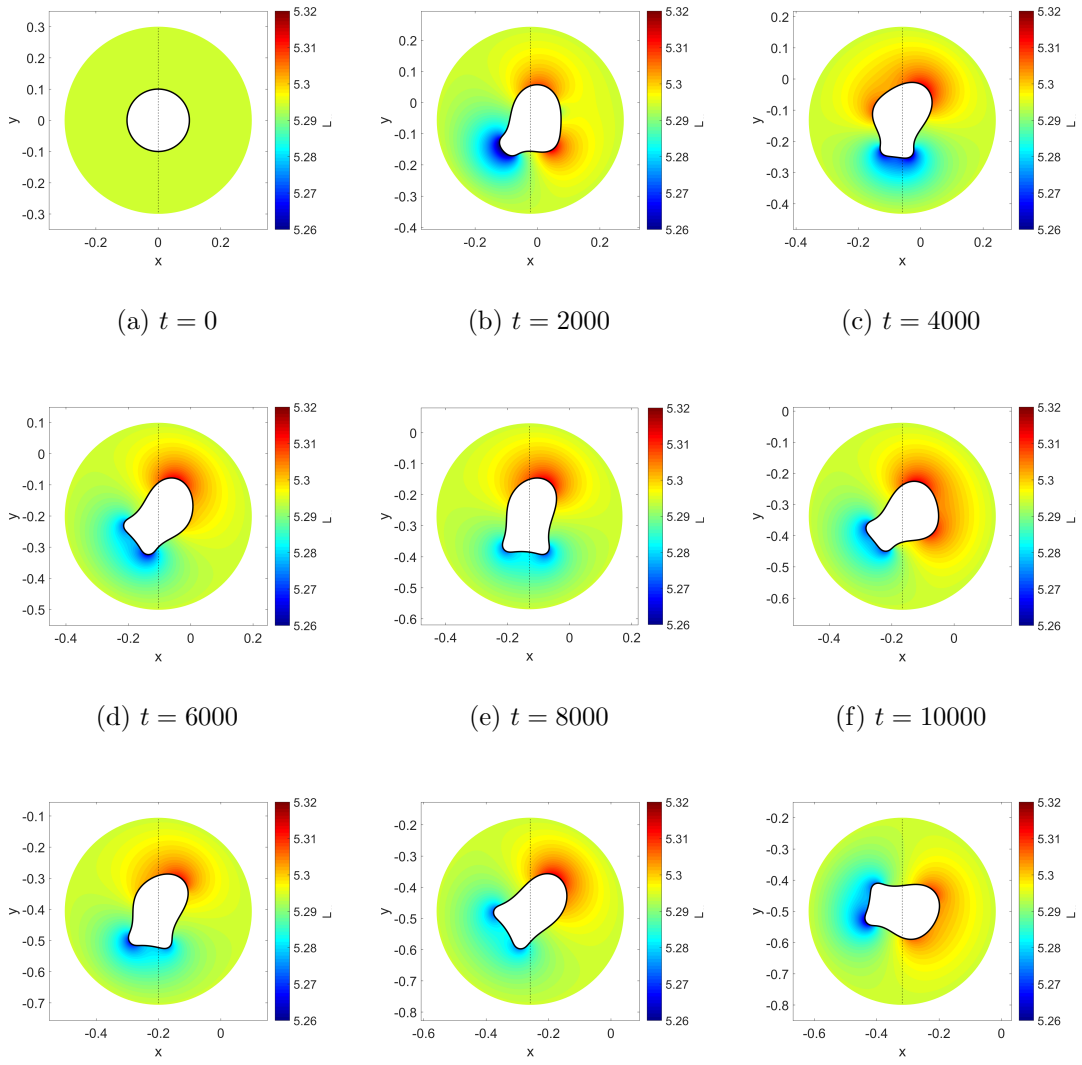


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 5.2: An example of a cell migrating in a zero gradient of free-ligand at various times. The protrusions coincide with regions of high activator as shown in Figure 5.4. The colour corresponds to the level of the free ligand concentration in the cell's exterior $L(\boldsymbol{x}, t)$.

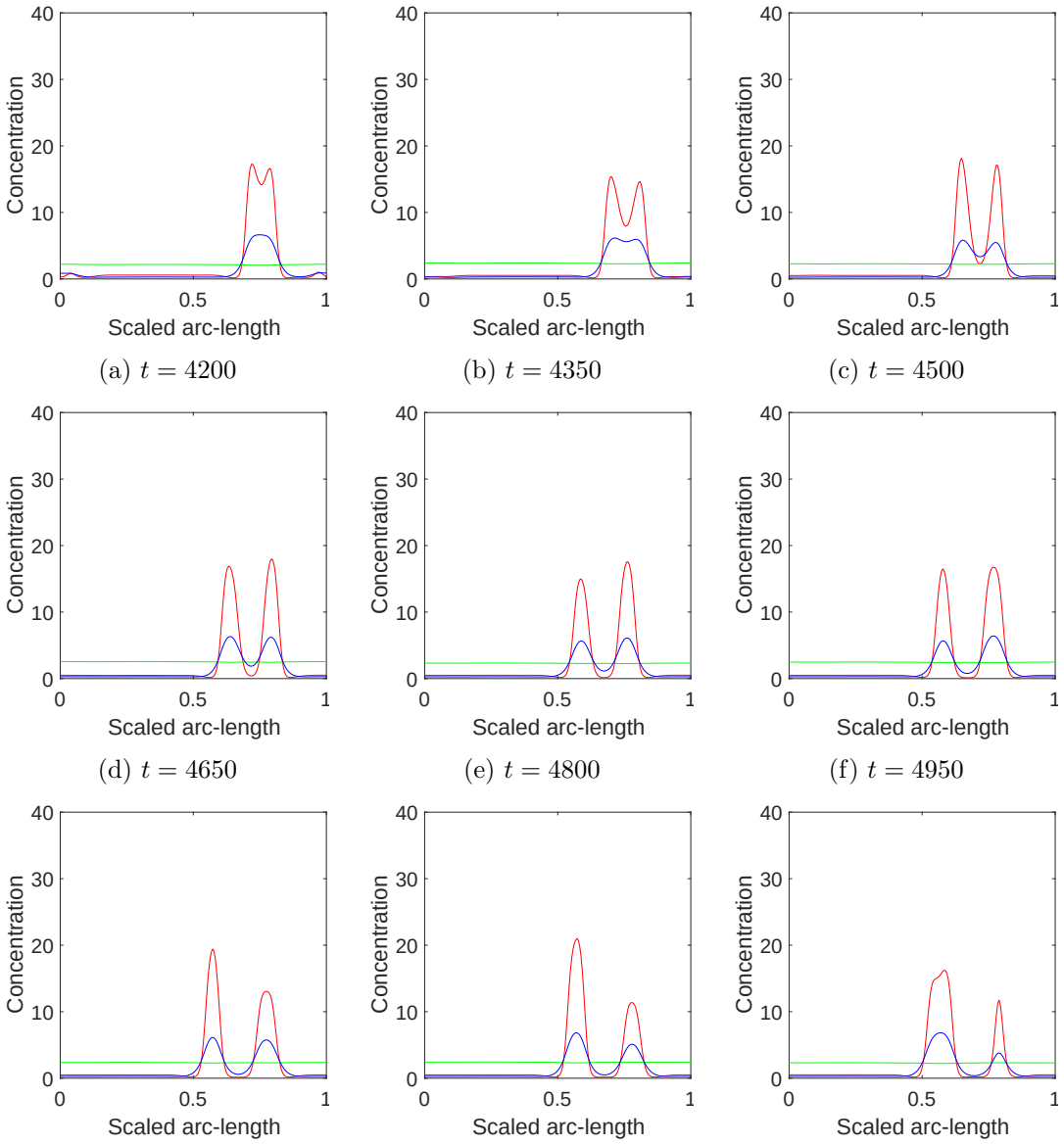


(a) $t = 0$ (b) $t = 2000$ (c) $t = 4000$

(d) $t = 6000$ (e) $t = 8000$ (f) $t = 10000$

(g) $t = 12000$ (h) $t = 14000$ (i) $t = 16000$

Figure 5.3: Example cell evolving with $\rho = 0$. The black line indicates the data used for the plot in Figure 5.6.

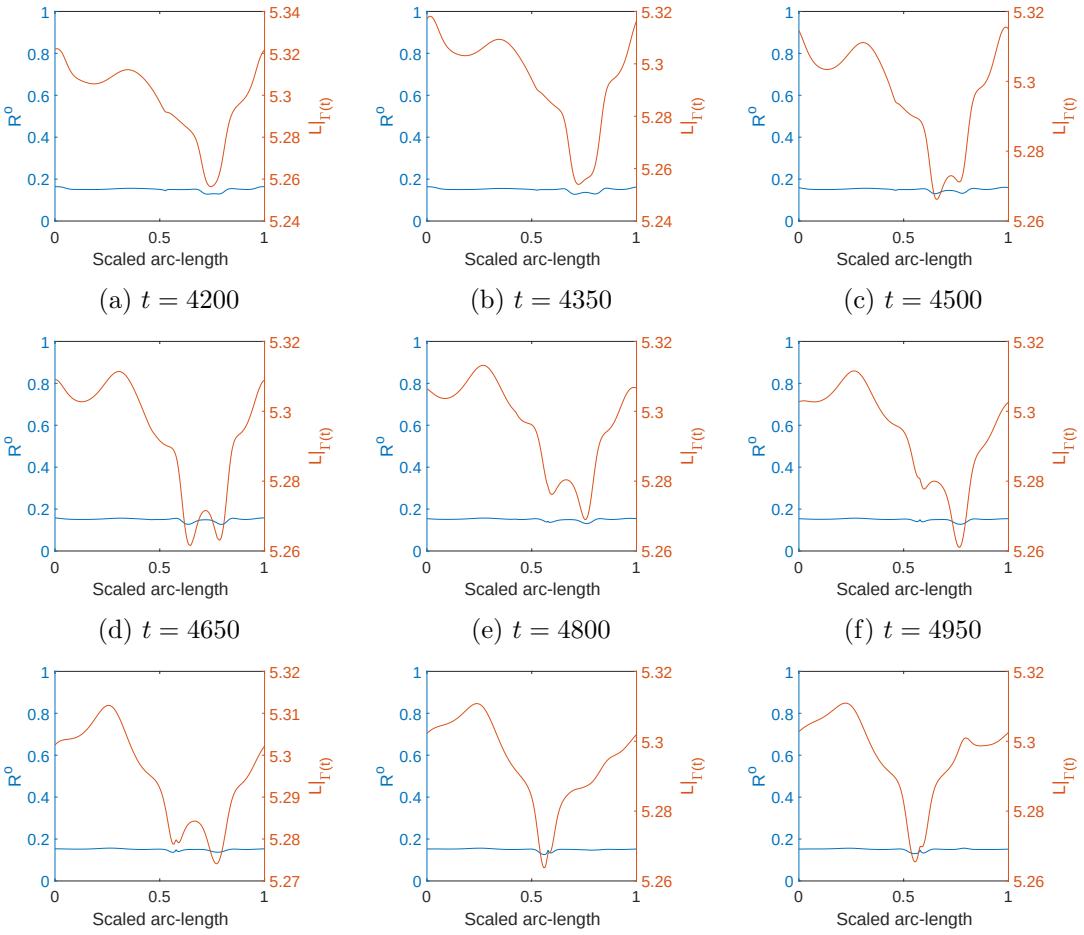


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 5.4: The corresponding solutions to the Meinhardt system for the cell simulation shown in Figure 5.2. Here the activator a is in red, the global inhibitor b green and the local inhibitor c blue.



(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 5.5: The corresponding fractional receptor occupancy and free-ligand concentration for the cell simulation shown in Figure 5.2. Here the free-ligand concentration in the bulk at the immediate boundary of the membrane is coloured in red, with the corresponding fractional receptor occupancy at the same point along the membrane given in blue.

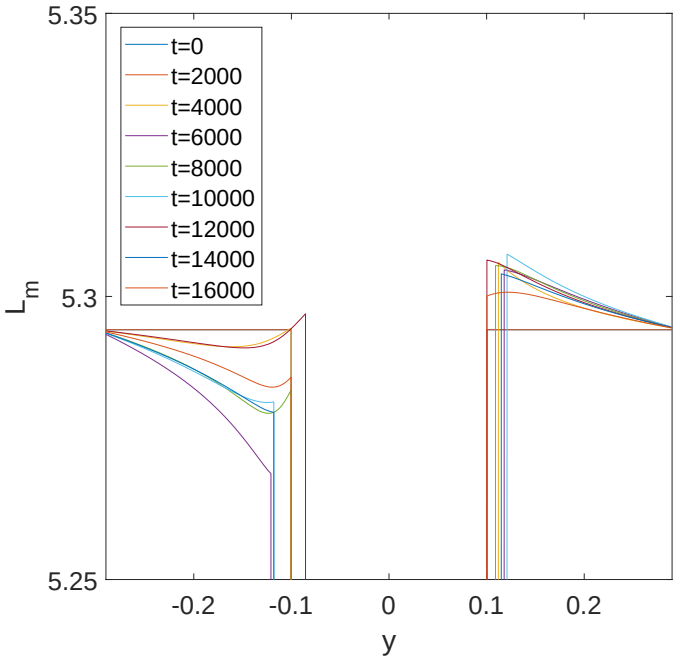
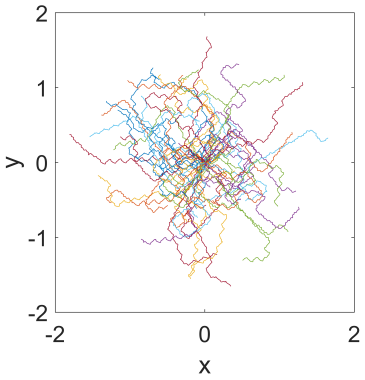
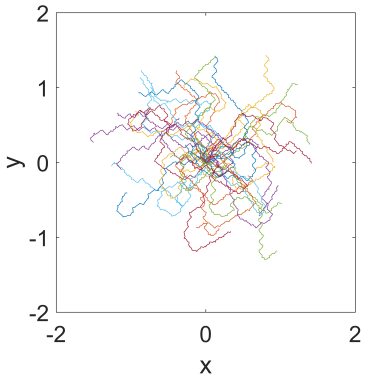


Figure 5.6: A slice through of L along the vertical line passing through the cell centre for the cell in Figure 5.3 which shows the slight wind-shield effect induced by the propagation of the cell through the environment.

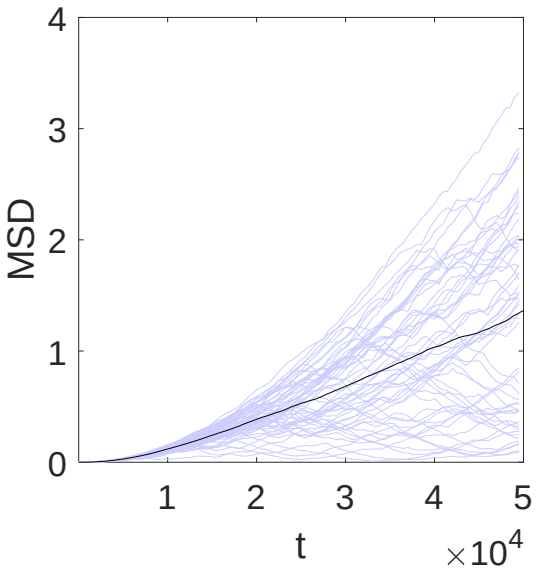


(a) Trajectory plot of the centroids of cells generated using the one-dimensional model with $\rho = 0$.

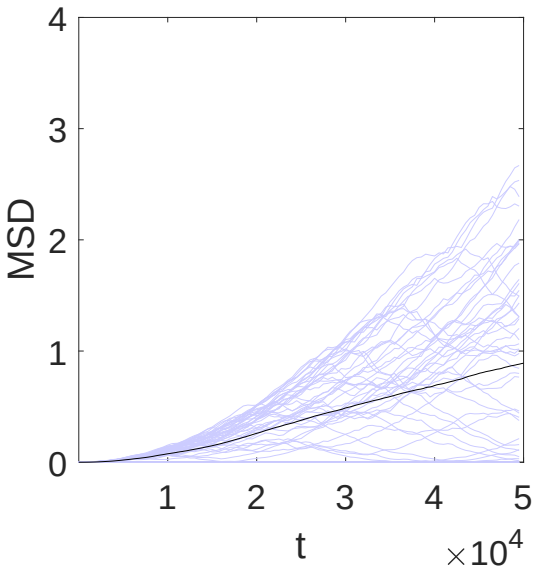


(b) Trajectory plot of the centroids of cells generated using the two-dimensional model with $\rho = 0$.

Figure 5.7: Trajectory plots for $N_c = 56$ cell simulations with the zero gradient free-ligand field with both the original one-dimensional cell model of Chapter 3 in (a) and the new two-dimensional cell model in (b).



(a) Mean square displacement plot for one-dimensional cell model with $\rho = 0$.



(b) Mean square displacement plot for two-dimensional model with $\rho = 0$.

Figure 5.8: A comparison of the mean square displacement for $N_c = 56$ cell simulations using the original one-dimensional cell model of Chapter 3 in (a) and newer two-dimensional cell model in (b).

5.3.2 Cell migration in linear background

We now examine what happens when the free-ligand concentration follows a linear gradient (5.14). Figure 5.9 shows an example of six cells migrating in the $\rho = 0.2$ gradient (5.14). As with the earlier model, cells modelled using the newer model chemotax quite well up the free-ligand gradient which has its highest concentration along the $+y$ axis. The morphology and general behaviour of the cells appears similar to the earlier model and we can see by Figure 5.12 that the solutions of the Meinhardt equations exhibit the same type of peak-splitting behaviour and same order of magnitude which drove pseudopod formation and ultimately the movement of the cell as in the original model. In Figure 5.13 we can also see that the receptor occupancy is smoothed out slightly now due to the modelling of the diffusive processes taking place on and near to the receptors however it is also much more varied than the original model owing to the fact that the cell membrane geometry and position now affects the solution.

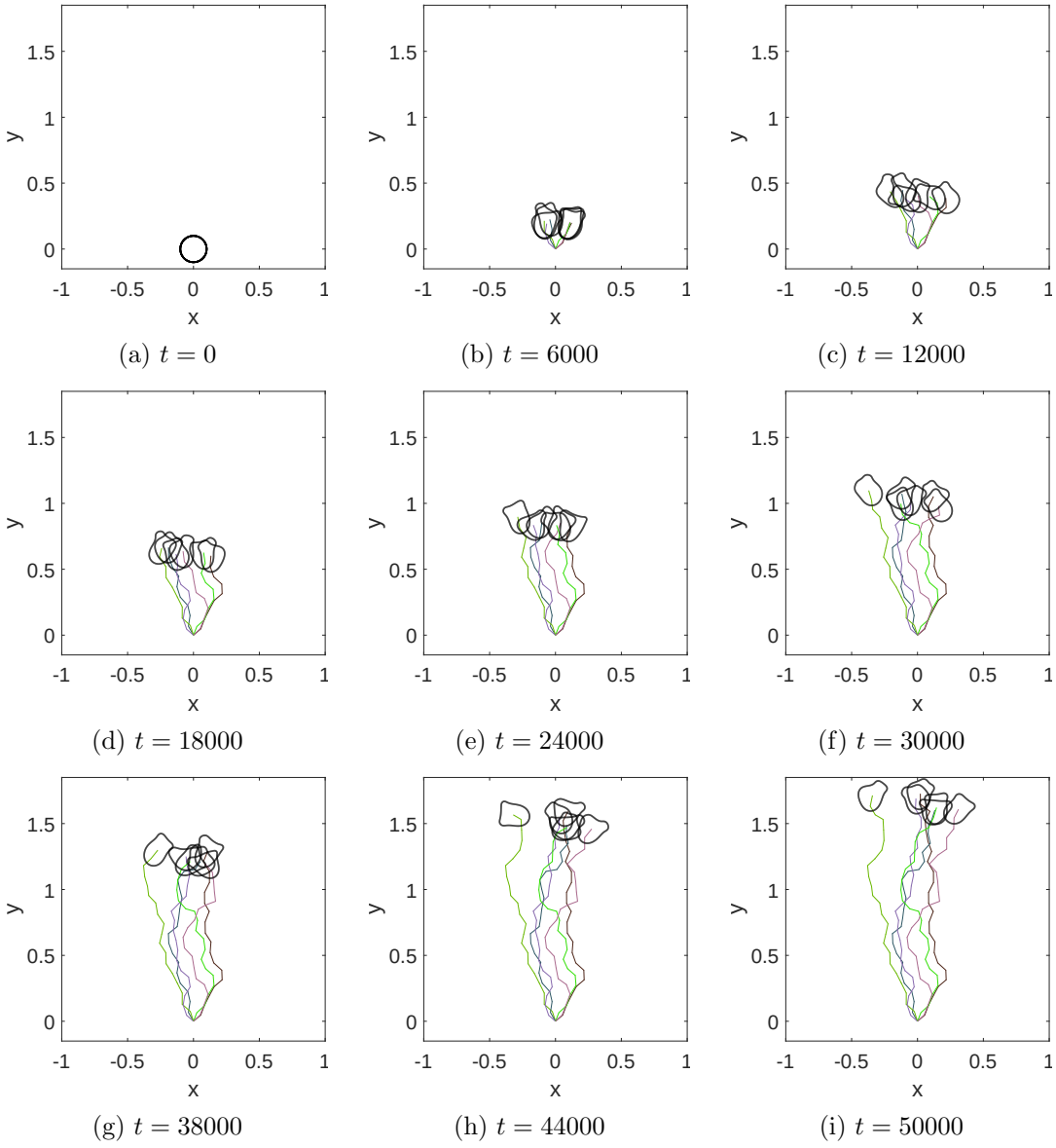
Figures 5.10 and 5.11 show the free-ligand field L on the exterior of the cell. In the original model L was a linear gradient in y, here, however, there is subtle but noticeable differences in L close to the membrane where the cell is perturbing the ligand around it both by the movement of the cell and by the binding/unbinding reactions taking place along the membrane. In Figure 5.14, the ligand L can be seen along the y axis, where we notice that the ligand tappers off smoothly from the trailing edge of the cell but is concentrated along the protruding edge by the effect of the cell pulling the ligand from the environment and onto unbound receptors. A similar phenomenon is analysed in the modelling of zebrafish [132], where the Doppler effect is used as an analogy to describe the relative changes of the wavelength of the ligand in the immediate vicinity of the cell. This same form of compression and retraction of the ligand field close to the cell is witnessed in Figure 5.14. This net effect creates a larger difference in receptor occupancy between the protruding edge and the trailing edge and thus steepings the perceived cell gradient.

Despite this however, cell simulations done using the newer model show these cells

to go slower and chemotax less. Plots of the trajectories for $N_c = 56$ cell simulations are shown in Figure 5.15 for both the original and newer model. The mean direction of the cells using the new model with a linear gradient of free-ligand (5.14) can be calculated using the Matlab Toolbox CircStat [17]. In the case of the original one dimensional model, the mean angle of the cells in Figure 5.15 (a) is 89.543° with an angular variance of 1.803° whereas the two-dimensional model the mean angle of the cells in 5.15 (b) at $t = 50000$ is 87.854° with an angular variance of 17.996°. Looking closely at Figure 5.15 we can actually see that the one-dimensional cell cases are more spread out compared to the two-dimensional cases but all of the cells for this case are generally heading in the correct direction towards the higher concentration of gradient. In the two-dimensional case, many of the cells start migrating in the correct direction towards the higher concentration and persist along that path with narrower focus than the one-dimensional equivalent. However, it can also be seen that there is a higher chance of cells choosing to migrate in other directions or that take longer to get started heading in the expected direction.

Figure 5.8 shows the mean square displacements for each case. The persistence of the cells which set off in the expected $+y$ direction towards the higher concentration of background do so with slightly greater persistence in the two-dimensional model than in the original model however the cells in this model which go in an unexpected direction do so with a great deal of reluctance and bring the mean square displacement of the overall simulation in the newer model to a level slightly below the original model.

This demonstrates that the larger variations and less defined peaks in the receptor occupancy are actually having a detrimental effect on the cells' ability to resolve the gradient here. This newer model however does not take account of the other biological interactions which are taking place on the membrane such as receptor internalization and relocation and it is possible a lot of features such as persistence and chemotaxis would be improved upon by incorporating these processes. For instance, work has taken place to extend this model to account for enzyme degradation of extracellular ligand fields in [86] which suggest improvements to account for enzyme degradation improve the ability of the cell to sense the surrounding gradient and thus improve the cells'



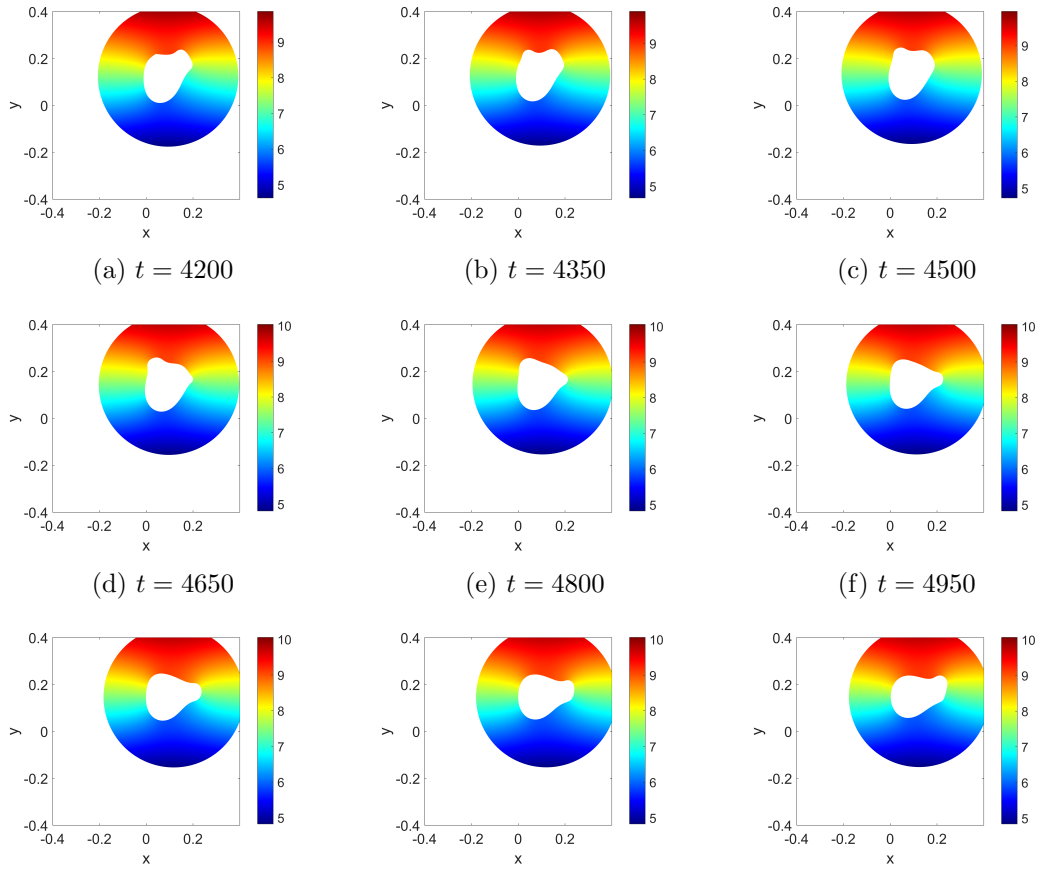
(a) $t = 0$ (b) $t = 6000$ (c) $t = 12000$

(d) $t = 18000$ (e) $t = 24000$ (f) $t = 30000$

(g) $t = 38000$ (h) $t = 44000$ (i) $t = 50000$

Figure 5.9: Snapshots of evolving cells and corresponding trajectory plots at various times. ($\rho = 0.2$).

chemotactic efficiency.

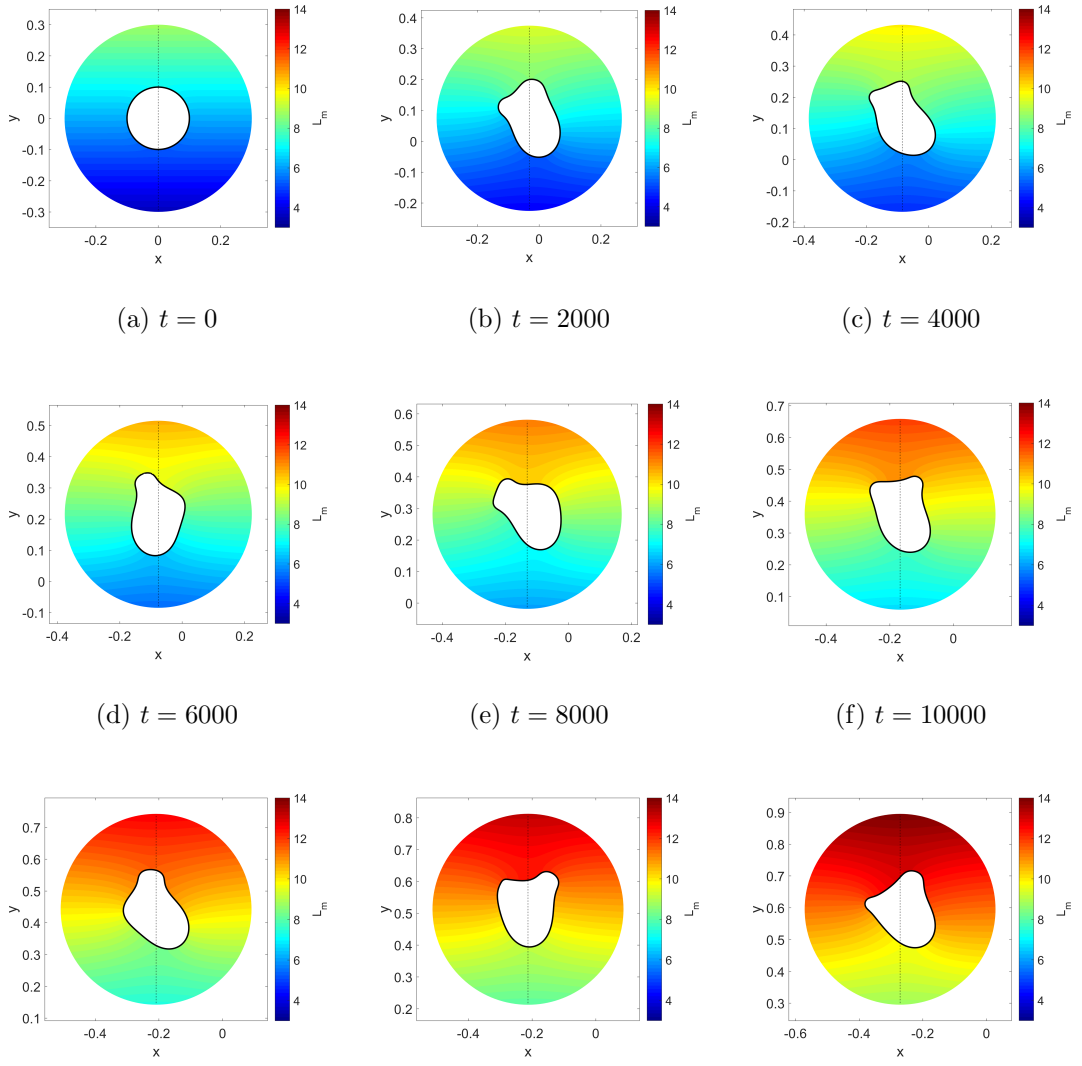


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 5.10: An example of a cell migrating in a linear background at various time-steps. The protrusions coincide with regions of high activator as shown in Figure 5.12.

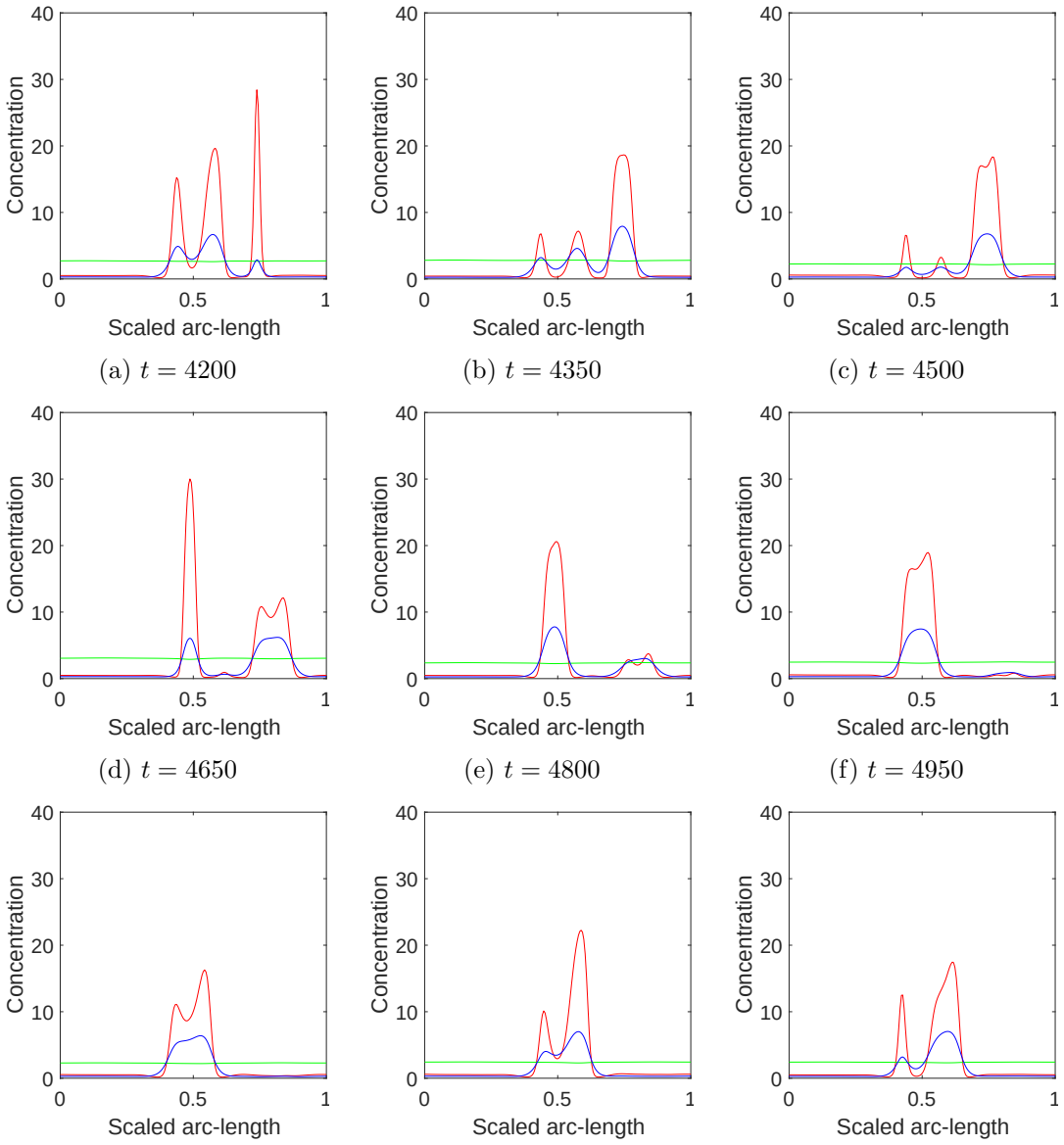


(a) $t = 0$ (b) $t = 2000$ (c) $t = 4000$

(d) $t = 6000$ (e) $t = 8000$ (f) $t = 10000$

(g) $t = 12000$ (h) $t = 14000$ (i) $t = 16000$

Figure 5.11: Example cell evolving with $\rho = 0.2$. The black line indicates the data used for the plot in Figure 5.14.

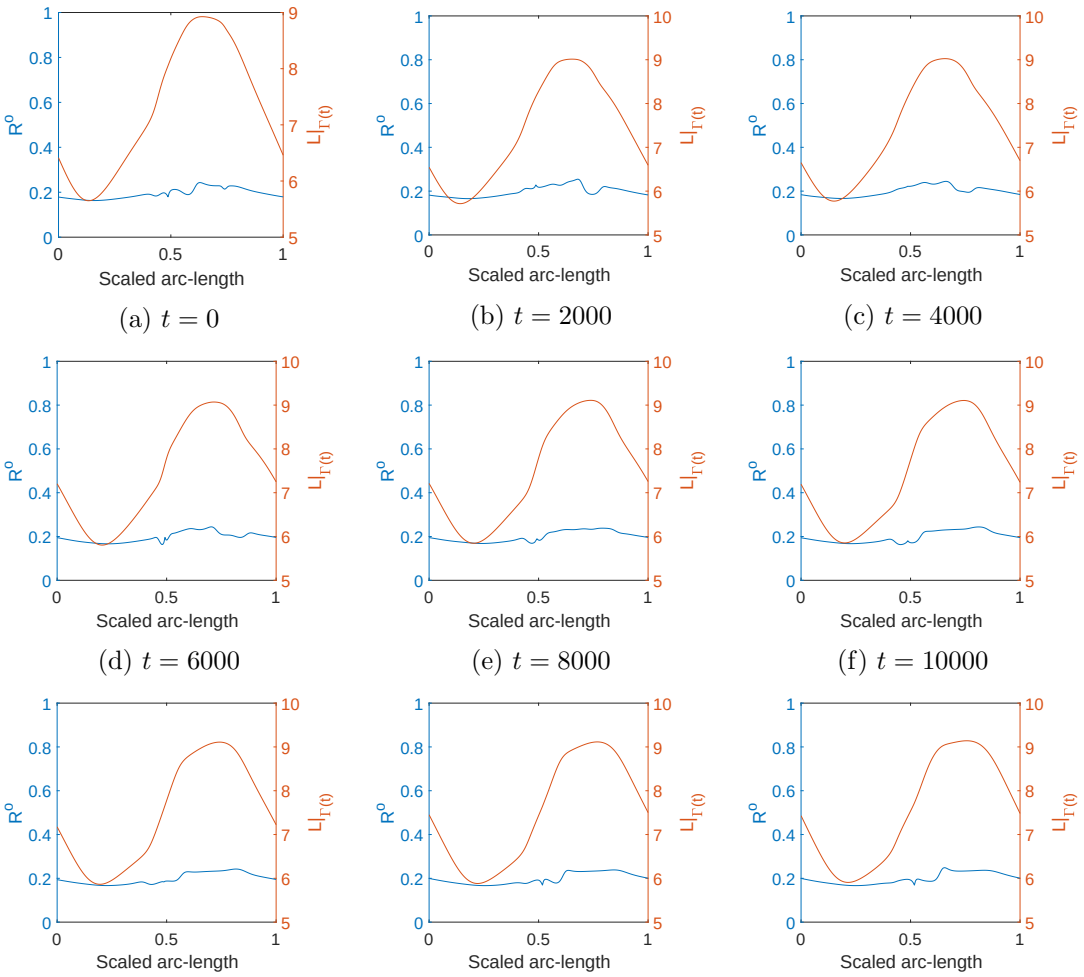


(a) $t = 4200$ (b) $t = 4350$ (c) $t = 4500$

(d) $t = 4650$ (e) $t = 4800$ (f) $t = 4950$

(g) $t = 5100$ (h) $t = 5250$ (i) $t = 5400$

Figure 5.12: The corresponding solutions to the Meinhardt system for the cell simulation shown in Figure 5.10. Here the activator a is in red, the global inhibitor b green and the local inhibitor c blue.



(a) $t = 0$ (b) $t = 2000$ (c) $t = 4000$

(d) $t = 6000$ (e) $t = 8000$ (f) $t = 10000$

(g) $t = 12000$ (h) $t = 14000$ (i) $t = 16000$

Figure 5.13: The corresponding fractional receptor occupancy and free-ligand concentration for the cell simulation shown in Figure 5.10. The free-ligand concentration in the bulk at the immediate boundary of the membrane is coloured in red, with the corresponding fractional receptor occupancy at the same point along the membrane given in blue.

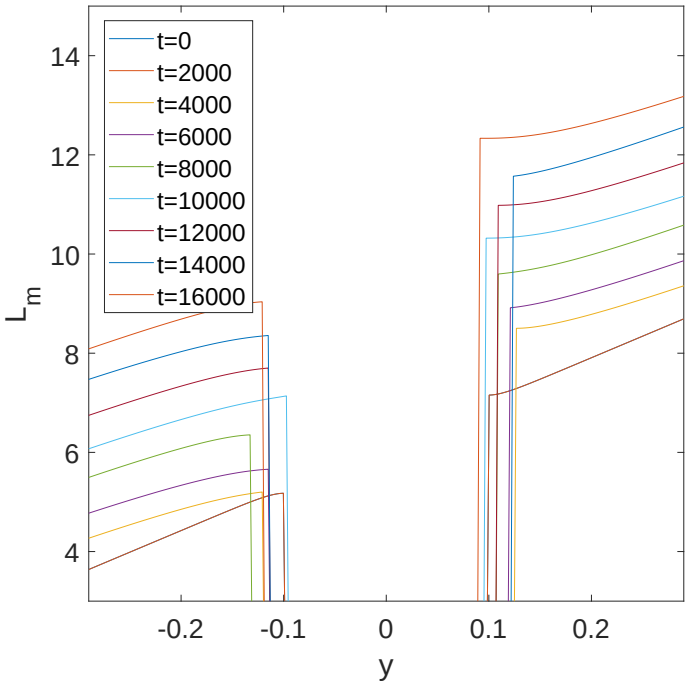
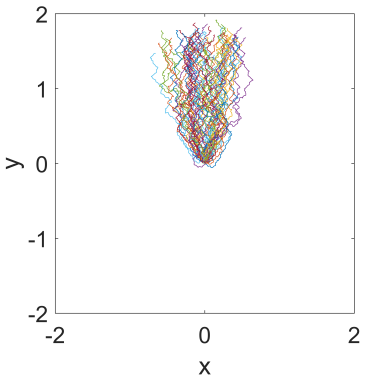
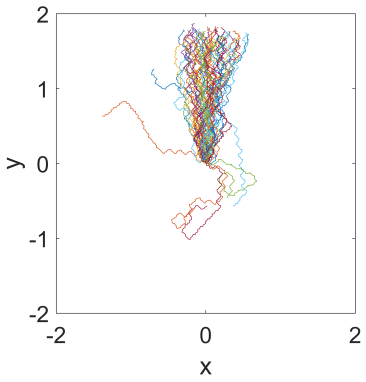


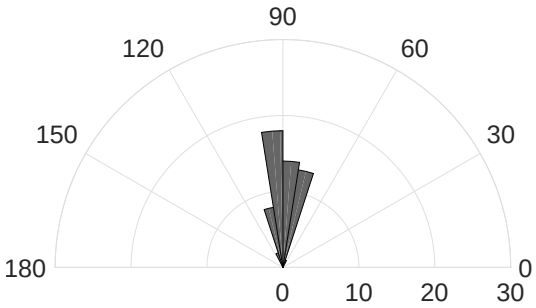
Figure 5.14: A slice through of L along the vertical line through the cell centre in Figure 5.11 which shows a more noticeable wind-shield effect induced by the propagation of the cell through the environment.



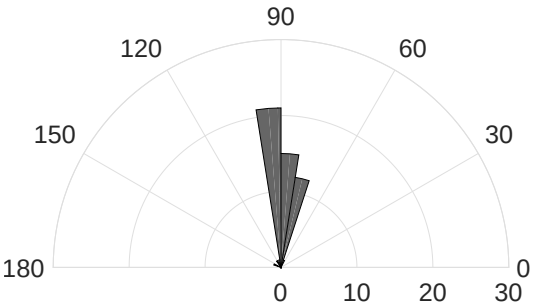
(a) Trajectory plot of the centroids of cells generated using the one-dimensional model with $\rho = 0.2$.



(b) Trajectory plot of the centroids of cells generated using the two-dimensional model with $\rho = 0.2$.

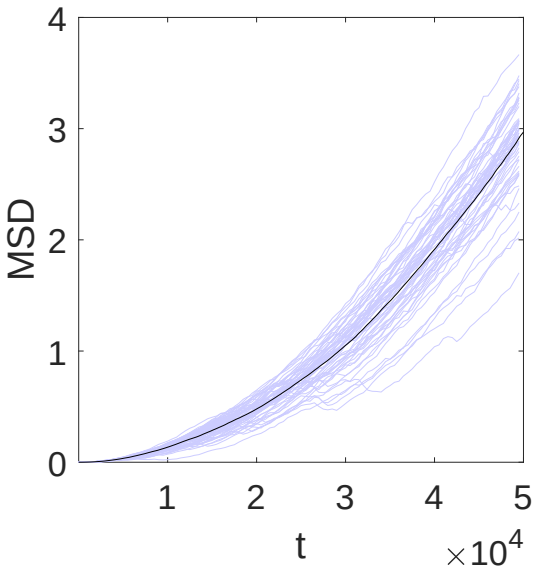


(c) Upper half of polar histograms for the trajectories of the one-dimensional cells with $\rho = 0.2$.

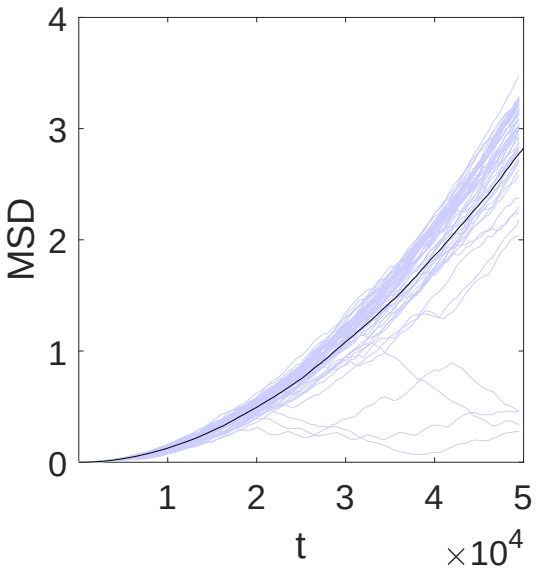


(d) Upper half of polar histograms for the trajectories of the two-dimensional cells with $\rho = 0.2$.

Figure 5.15: A comparison of trajectory plots for the $\rho = 0$ and $\rho = 0.2$ cases using the one-dimensional cell model of Chapter 3 in (a) and (c) and two-dimensional cell model in (b) and (d). The upper half of polar histograms for the $\rho = 0.2$ cases are shown in (e) and (f).



(a) Mean square displacement plot for one-dimensional cell model with $\rho = 0.2$.



(b) Mean square displacement plot for two-dimensional model with $\rho = 0.2$.

Figure 5.16: A comparison of the mean square displacement for the $\rho = 0$ and $\rho = 0.2$ cases using the one-dimensional cell model of Chapter 3 in (a) and (c) and two-dimensional cell model in (b) and (d).

5.4 Conclusions

In conclusion:

- This chapter introduces a new model for cell migration and chemotaxis build on an established model into two dimensions by utilising all of the methods developed and discussed in all other Chapters.

- This new two dimensional model resolves the free-ligand field on the exterior of the cell in order to more faithfully determine the solution to the receptor occupancy along the membrane. It does this by coupling the solutions of the two-dimensional convective-diffusion equation for the free-ligand chemical to the one-dimensional model of receptor occupancy by means of the flux coupling boundary condition in the ALEFEM2D method introduced in the chapter previous.

- The original one dimensional model is then calculated in the manner described in Algorithm 3.1 using ALEFEM1D but with the signal term of the Meinhardt system now being composed of the newly derived more accurately calculated receptor occupancy.

- The membrane derived from Algorithm 3.1 is then used to generate an exterior mesh using the MMPDE2D method for mesh generation described in the previous chapter.

- This model takes into account one of the chemical processes occurring in the cell's exterior which is the evolution of the free-ligand which is in a very fundamental way the cell's source of food and thus of crucial importance in modelling chemotaxis accurately. The perturbation which the cell does itself to its own environment in a homogeneous background gradient may also be important for cell migration in general though further work must be done to investigate this.

- Numerical results demonstrate a small loss of chemotaxic ability in the new model but this is not unexpected. Biologically there are many mechanisms taken place

involving the cells' receptors and these were not accurately investigated before because these methods were not in place but now they are being investigated by researchers at University of Strathclyde and the Beatson Institute and results are already showing that certain biological mechanisms may actually be key to helping the cell chemotax in weak environments such as enzyme degradation [86].

- The model presented in this chapter does give a more truthful representation of what a model of cell migration should be. It is consequently already being used in publications and collaborative work to develop new insights into cell migration and chemotaxis by researchers at University of Strathclyde and the Beatson Institute [86].

- The new model is more true to life as it takes account of the cell movement through the environment. The cells travel at slower speeds and are not as easily directed as the original model. It is thought that persistence and speed could be improved by considering some potential biological mechanisms which take place at the membrane to help cell environment sensing and work is already underway to look into some of these mechanisms [86].

Chapter 6

Conclusions and further work

6.1 Conclusions

The computational modelling of the underlying processes which drive cell motility is a rapidly growing area of research, gaining much mainstream popularity as computational power becomes less expensive and the numerical methods which can be used to investigate cell models become increasingly developed. The methods presented in this thesis are intended to be made use of by modellers and future researchers to assist in developing new understandings of the systems and processes underpinning cell migration and chemotaxis.

This thesis has attempted to introduce new methods and existing methods for the context of cell migration and chemotaxis modelling and has done this in the following ways:

- In Chapter 2, an ALEFEM1D method was discussed for the solution to reaction-diffusion equations on an evolving curve domains in such a way that the correct concentrations could be extracted without the need to re-mesh or use interpolation. The method was shown to be numerically second order accurate in both time and space for a sample of numerical test experiments. Further, the method was shown to be conservative for at least one test problem and while an analytical proof of the conservation property in general is not presented in this thesis, it has been since proven and is presented in up coming publications which relate to the

content of this thesis.

- In the same chapter, the MMPDE1D method was introduced for the first time. This method is a novel approach to mesh relocation for evolving domains which provides control over the mesh node velocities in such a way as to bring r-adaption into the evolving curve mesh generation process; thus potentially improving accuracy of the overlaying numerical methods without the need for a denser uniform mesh. Numerical experiments were also conducted to show that the method was second order accurate as well as various demonstrations of the adaptive property that were presented to compare the new method with some examples used by other methods.

- In Chapter 3, a reaction-diffusion model which builds on a biological model from Hans Meinhardt [94] was discussed. Using the numerical methods introduced in Chapter 2, this chapter built on the work carried out by Neilson et al. [105], who had adapted Meinhardt's model for the cell migration and chemotaxis problem. In this chapter simulations reiterate their results using the new MMPDE1D approach for the moving of the mesh points and show that the new moving mesh method was comparable with the one used by Neilson et al. The figures also showed that the model represents well certain characteristics of real cells.

- Further to this, in the same chapter there were also experiments using the newly created MMPDE1D method with a mesh adaption function. The improvements were slight but did show that the simulations could potentially be made more accurate or their computation potentially made more efficient by using the new method. The main purpose of developing MMPDE1D for use in the cell problem however was to introduce a degree of control over the mesh velocities which was not there previous so that modelling in higher dimensions could be investigated.

- In Chapter 4, we begun to examine methods needed for taking the model into higher dimensions. Here, the arbitrary Eulerian-Lagrangian approach for finding the solutions of reaction-diffusion equations on evolving curves was examined in two dimensions under the view of solving PDEs on planar surfaces. In addition

to this, a semi-implicit Galerkin finite element method for coupling systems on the external, boundary and interior through flux terms was also introduced which facilitates the coupling of models operating on different domains.

- Numerical experiments were conducted on moving domains using the ALEFEM method to compute solutions and compare with derived analytical solutions. These solutions demonstrate that the ALEFEM scheme was second order accurate in time and space for the given problems. This was a non-trivial realisation as the method is coupling the solutions different reactions together, not only that, the different reactions were taken place on different dimensions with the bulk species existing in two spatial dimensions and the surface-bound species existing in only one.

- The movement of the underlying two-dimensional mesh was also investigated in this chapter and a method developed by Cao, *et al* [29] was examined and found to have key elements thought to be highly desirable for complementing the existing methods developed for the cell problem. For instance, the variation form of the moving mesh PDE, on which their method was based, permits mesh adaption through a monitor function. This could be simultaneously used with MMPDE1D to give a smooth and natural progression from curve based mesh adaption to bulk based mesh adaption. This property may even prove vital in future as more complex biological systems are uncovered and demand grows for such processes to be modelled using the methods presented here. In such instances where a particular protein is prolific at the leading edge of the cell but virtually non existent at the trailing edge then adaption could be utilised to ensure that the desired accuracy is achieved without unwanted extra computation.

- In Chapter 5, the methods in all of the previous Chapters were utilised to develop an improved model of cell migration and chemotaxis in two-dimensional space. There, the ALEFEM2D method of Chapter 4 was used to create an enhanced model of receptor occupancy using the simulated value for the free-ligand field on the exterior of the cell. This field was contained within a small mesh which was

moved using the MMPDE2D method which was also introduced in Chapter 4. The membrane itself continued to move according to Algorithm 3.1, which was the one-dimensional model outlined in Chapter 3 but, in the two-dimensional model, the signal term ϱ was composed of the more accurately derived receptor occupancy term.

Numerical results were presented which showed the new two-dimensional model and compared it to the original model. The results showed that the chemotactic efficiency of the cell had decreased however it was noticed that true cells do not rely solely on this binding/unbinding model for the receptor occupancy and so, in a separate paper published by Mackenzie et al. [86], the two-dimensional model given in Chapter 5 is actually further developed and the cell process of enzyme degradation is investigated using the methods outlined in this thesis. The work done in [86] suggests that if the degradation rate is fast enough and the rate of extracellular diffusion is slow enough, then the moving cell displays significant directional persistence in a homogeneous ligand field and improved chemotaxis in shallow saturating fields. The predictions of this model are currently being investigated using experimental data.

6.2 Further work

Much work remains to be done on the modelling of cell migration and chemotaxis. The computational methods presented here open up new ways in which to test hypotheses. However, there is room for further research and development into these methods and the model presented here. For instance:

- The stability and convergence properties of the MMPDE1D method, which makes use of Picard iteration to solve the non-linear equations, have not been fully uncovered.

- Further to this, only isotropic curve shortening flow was investigated using the MMPDE1D method. There remains other problems to which this method could

be applied to, such as Willmore flow [123, 148] and anisotropic mean curvature flow [134].

- The MMPDE1D method could also potentially be extended to the evolution of surfaces in three dimensions based on a FEM discretisation of an approximate system of normal and tangential velocity equations. This could provide a more realistic representation of the cell.

- In Chapter 4, an iterative method for coupled bulk-surface problems is introduced but an analysis of the stability and convergence characteristics of this method remains to be carried out.

- The new computational framework for the new model could of course always be used to model intracellular signalling pathways and the possible interaction with the cell nucleus. The numerical techniques could also be used with more sophisticated models of cell mechanics and the mechanical interaction of the cell on substrates. To do this appropriate model equations will have to be solved to supply the intracellular material velocity; such models could be viscoelastic or poroelastic type [97].

- The numerical techniques could also be extended to model cell-cell and cell-obstacle interactions which would be difficult to do accurately without the two-dimensional framework. Finally, since there many different time-scales involved in cell migration, an adaptive time-stepping approach could be developed to potentially take advantage of the relatively slow kinetics of the Meinhardt model to gain more efficient simulations.

- Finally, cells in nature have three spatial dimensions and therefore a natural progression to the framework given here would be to develop the methods into for three-dimensional space. In order to do this, it is expected that a surface-based FEM implementation of MMPDE1D would be required first but the ALEFEM2D and MMPDE2D methods are in no way limited to two-dimensional space and could readily be adapted for modelling in higher dimensions.

Bibliography

[1] M. Abercrombie. The Croonian Lecture, 1978: The Crawling Movement of Metazoan Cells. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167):129–147, 1980.

[2] R. W. Anderson, V. A. Dobrev, T. V. Kolev, R. N. Rieben, and V. Z. Tomov. High-order multi-material ale hydrodynamics. *SIAM Journal on Scientific Computing*, 40(1):B32–B58, 2018.

[3] N. Andrew and R. H. Insall. Chemotaxis in shallow gradients is mediated independently of PtdIns 3-kinase by biased choices between random protrusions. *Nature Cell Biology*, 9:193–200, 2007.

[4] H. Antoniades, C. Scher, and C. Stiles. Purification of human platelet-derived growth factor. *Proceedings of the National Academy of Sciences of the United States of America*, 76(4):1809–1813, 1979.

[5] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151 – 167, 1997.

[6] M. J. Baines. Grid adaptation via node movement. *Applied Numerical Mathematics*, 26(1-2):77–96, 1998.

[7] G. A. Baker. Finite element methods for elliptic equations using nonconforming elements. *Mathematics of Computation*, 31(137):45–59, 1977.

[8] M. Balažovjech and K. Mikula. A Higher Order Scheme for a Tangentially Stabilized Plane Curve Shortening Flow with a Driving Force. *SIAM Journal on Scientific Computing*, 33(5):2277–2294, 2011.

[9] R. Barreira, C. M. Elliott, and A. Madzvamuse. The surface finite element method for pattern formation on evolving biological surfaces. *Journal of Mathematical Biology*, 63(6):1095–1119, 2011.

[10] J. W. Barrett, H. Garcke, and R. Nürnberg. On the variational approximation of combined second and fourth-order geometric evolution equations. *SIAM Journal on Scientific Computing*, 29(3):1006–1041, 2007.

[11] J. W. Barrett, H. Garcke, and R. Nürnberg. Numerical approximation of anisotropic geometric evolution equations in the plane. *IMA Journal of Numerical Analysis*, 28(2):292–330, 2008.

[12] J. W. Barrett, H. Garcke, and R. Nürnberg. Numerical approximation of gradient flows for closed curves in $\mathbb{R}^d$. *IMA Journal of Numerical Analysis*, 30(1):4–60, 2010.

[13] G. Beckett, J. A. Mackenzie, and M. L. Robertson. An r-adaptive finite element method for the solution of the two-dimensional phase-field equations. *Communications in Computational Physics*, 1(5):805–826, 2006.

[14] G. Beckett, J.A. Mackenzie, A. Ramage, and D.M. Sloan. On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution. *Journal of Computational Physics*, 167(2):372–392, 2001.

[15] G. Beckett, J.A. Mackenzie, A. Ramage, and D.M. Sloan. Computational solution of two-dimensional unsteady PDEs using moving mesh methods. *Journal of Computational Physics*, 182(2):478–495, 2002.

[16] A. Bellen. Numerical methods for delay differential equations: Accuracy and stability problems. *IFAC Proceedings Volumes*, 33(23)(23):127–128, 2000.

[17] P. Berens. CircStat: A MATLAB Toolbox for Circular Statistics. *Journal of Statistical Software, Articles*, 31(10):1–21, 2009.

[18] M. Bergert, S. D. Chandradoss, R. A. Desai, and E. Paluch. Cell mechanics control rapid transitions between blebs and lamellipodia during migration. *Proceedings of the National Academy of Sciences of the United States of America*, 109(36):14434–14439, 2012.

[19] T. Biben, K. Kassner, and C. Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72(4):041921, 2005.

[20] P. G. Blackwell. *Ornstein-Uhlenbeck Process: Overview*. John Wiley and Sons, Ltd, 2014.

[21] H. Blaser, M. Reichman-Fried, I. Castanon, K. Dumstrei, F. L. Marlow, K. Kawakami, L. Solnica-Krezel, C-P. Heisenberg, and E. Raz. Migration of Zebrafish Primordial Germ Cells: A Role for Myosin Contraction and Cytoplasmic Flow. *Developmental Cell*, 11(5):613–627, 2006.

[22] H. Borouchaki, P. George, F. Hecht, P. Laug, and E. Saltel. Delaunay mesh generation governed by metric specifications. part i. *Algorithms*, 25:61–83, 1997.

[23] H. Borouchaki, P. George, and B. Mohammadi. Delaunay mesh generation governed by metric specifications. part ii. *Algorithms*, 25:85–109, 1997.

[24] L. Bosgraaf and P. J. M. van Haastert. Navigation of chemotactic cells by parallel signalling to pseudopod persistence and orientation. *PLoS One*, 4:e6842, 2009.

[25] L. Bosgraaf and P. J. M. van Haastert. The ordered extension of pseudopodia by amoeboid cells in the absence of external cues. *PLoS One*, 4:e5253, 2009.

[26] J. U. Brackbill. An adaptive grid with directional control. *Journal of Computational Physics*, 108(1):38–50, 1993.

[27] A. N. Brooks and T. J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incom-

pressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.

[28] M. P. Calvo, J. de Frutos, and J. Novo. Linearly implicit rungekutta methods for advectionreactiondiffusion equations. *Applied Numerical Mathematics*, 37(4):535–549, 2001.

[29] W. Cao, W. Huang, and R. D. Russell. An r-adaptive finite element method based upon moving mesh PDEs. *Journal of Computational Physics*, 149:221–244, 1999.

[30] M. J. Castro-Daz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, 25:475–491, 1997.

[31] J. Dai, H. P. Ting-Beall, R. M. Hochmuth, M. P. Sheetz, and M. A. Titus. Myosin I contributes to the generation of resting cortical tension. *Biophysical Journal*, 77:1168– 1176, 1999.

[32] N. Dainiak, G. Davies, M. Kalmanti, J. Lawler, and V. Kulkarni. Platelet-derived growth factor promotes proliferation of erythropoietic progenitor cells in vitro. *Journal of Clinical Investigation*, 71(5):1206–1214, 1983.

[33] E. F. D'Azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM Journal of Scientific Computing*, 12:755–786, 1991.

[34] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numerical Mathematics*, 59:321–348, 1991.

[35] C. de Boor. Good Approximation by Splines with Variable Knots. *International Series of Numerical Mathematics*, 21:57–72, 1973.

[36] P. A. DiMilla, K. Barbee, and D. A. Lauffenburger. Mathematical model for the effects of adhesion and mechanics on cell migration speed. *Biophysical Journal*, 60(1):15–37, 1991.

[37] J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1):689–723, 1982.

[38] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. *Arbitrary LagrangianEulerian Methods*. John Wiley & Sons, Ltd, 2004.

[39] Q. Du, C. Liu, and X. Wang. Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions. *Journal of Computational Physics*, 212(2):757–777, 2006.

[40] M. Dworkin and K. H. Keller. Solubility and diffusion coeffcient of adenosine 3':5'-monophosphate. *Journal of Biological Chemistry*, 252:864–865, 1977.

[41] G. Dziuk and C. M. Elliott. Finite element methods for surface PDEs. *Acta Numerica*, 22:289–396, 2013.

[42] E. Evans E and A. Yeung. Apparent viscosity and cortical tension of blood granulocytes determined by micropipet aspiration. *Biophysical Journal*, 56(1):151–160, 1989.

[43] E. Evans E and A. Yeung. Cortical shell-liquid core model for passive flow of liquid-like spherical cells into micropipets. *Biophysical Journal*, 56(1):139–149, 1989.

[44] C. Eilks and C. M. Elliott. Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method. *Journal of Computational Physics*, 227(23):9727–9741, 2008.

[45] M. Eisenbach and L. C. Giojalas. Sperm guidance in mammals - an unpaved road to the egg. *Nature Reviews Molecular Cell Biology*, 7:276–285, 2006.

[46] C. M. Elliott and T. Ranner. Finite element analysis for a coupled bulk-surface partial differential equation. *IMA Journal of Numerical Analysis*, 33(2):377–402, 2013.

[47] C. M. Elliott, B. Stinner, V. Styles, and R. Welford. Numerical computation of advection and diffusion on evolving diffuse interfaces. *IMA Journal of Numerical Analysis*, 31(3):786–812, 2011.

[48] C. M. Elliott, B. Stinner, and C. Venkataraman. Modelling cell motility and chemotaxis with evolving surface finite elements. *Journal of The Royal Society Interface*, 9(76):3027–3044, 2012.

[49] C. M. Elliott and V. Styles. An ALE ESFEM for solving PDEs on evolving surfaces. *Milan Journal of Mathematics*, 80(2):469–501, 2012.

[50] R. G. Endres and N. S. Wingreen. Accuracy of direct gradient sensing by single cells. *Proceedings of the National Academy of Sciences of the United States of America*, 105(41):15749–15754, 2008.

[51] D. Engwirda. Locally-orthogonal staggered unstructured grid-generation for general circulation modelling on the sphere. *Atmospheric and Oceanic Physics*, arXiv:1611.08996, 2017.

[52] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[53] R. FitzHugh. Mathematical models of threshold phenomena in the nerve membrane. *The Bulletin of Mathematical Biophysics*, 17:257–278, 1955.

[54] J. Frank, W. Hundsdorfer, and J. G. Verwer. On the stability of implicit-explicit linear multistep methods. *Applied Numerical Mathematics*, 25(2):193–205, 1997. Special Issue on Time Integration.

[55] S. L. Gupton and C. M. Waterman-Storer. Spatiotemporal Feedback between Actomyosin and Focal-Adhesion Systems Optimizes Rapid Cell Migration. *Cell*, 125(7):1361–1374, 2006.

[56] M. H. Gutknecht. Variants of BICGSTAB for Matrices with Complex Spectrum. *SIAM Journal on Scientific Computing*, 14(5):1020–1033, 1993.

[57] T.L. Hill. Microfilament or microtubule assembly or disassembly against a force. *Proceedings of the National Academy of Sciences of the United States of America*, 78(9):5613–5617, 1981.

[58] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[59] D. Hoff. Stability and convergence of finite difference methods for systems of nonlinear reaction-diffusion equations. *SIAM Journal on Numerical Analysis*, 15(6):1161–1177, 1978.

[60] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *Journal of Computational Physics*, 171(2):753–775, 2001.

[61] W. Huang, L. Kamenski, and R. D. Russell. A comparative numerical study of meshing functionals for variational mesh adaptation. *Journal of Mathematical Study*, 48(2):168–186, 2015.

[62] W. Huang, Y. Ren, and R. D. Russell. Moving Mesh Partial Differential Equations (MMPDES) Based on the Equidistribution Principle. *SIAM Journal on Numerical Analysis*, 31(3):709–730, 1994.

[63] W. Huang and R. D. Russell. Analysis of moving mesh partial differential equations with spatial smoothing. *SIAM Journal on Numerical Analysis*, 34(3):1106–1126, 1997.

[64] W. Huang and R. D. Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM Journal on Scientific Computing*, 20(3):998–1015, 1998.

[65] W. Huang and R. D. Russell. *Adaptive moving mesh methods*. Springer, 2010.

[66] R. V. Iancu, S. W. Jones, and R. D. Harvey. Compartmentation of cAMP Signaling in Cardiac Myocytes: A Computational Study. *Biophysical Journal*, 92(9):3317–3331, 2007.

[67] P. A. Iglesias and A. Levchenko. Modeling the cell's guidance system. *Science Signaling*, 2002(148):re12, 2002.

[68] R. H. Insall. Understanding eukaryotic chemotaxis: a pseudopod-centred view. *Nature Reviews Molecular Cell Biology*, 11:453–458, 2010.

[69] R. H. Insall and L. M. Machesky. Actin dynamics at the leading edge: from simple machinery to complex networks. *Developmental Cell*, 17(3):310–322, 2009.

[70] E. Javierre, F. J. Vermolen, C. Vuik, and S. van der Zwaag. A mathematical analysis of physiological and morphological aspects of wound closure. *Journal of Mathematical Biology*, 59(5):605–630, 2009.

[71] R. L. Johnson, R. A. Vaughan, M. J. Caterina, P. J. Van Haastert, and P. N. Devreotes. Overexpression of the cAMP receptor 1 in growing Dictyostelium cells. *Biochemistry*, 30(28):6982–6986, 1991.

[72] T. Killich, P. J. Plath, X. Wei, H. Bultmann, L. Rensing, and M. G. Vicker. The locomotion, shape and pseudopodial dynamics of unstimulated Dictyostelium cells are not random. *Journal of Cell Science*, 106(4):1005–1013, 1993.

[73] D. C. Koppenol, F. J. Vermolen, G. V. Koppenol-Gonzalez, F. B. Niessen, P. P. M. van Zuijlen, and K. Vuik. A mathematical model for the simulation of the contraction of burns. *Journal of Mathematical Biology*, 75(1):1–31, 2016.

[74] B. Kutscher, R. Devreotes, and P. A. Iglesias. Local Excitation, Global Inhibition Mechanism for Gradient Sensing: An Interactive Applet. *Science Signaling*, 2004(219):PL3, 2004.

[75] K. Larripa and A. Mogilner. Transport of a 1D viscoelastic actin-myosin strip of gel as a model of a crawling cell. *Physica A*, 372(1):113–123, 2006.

[76] P. Lesaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. *Publications mathmatiques et informatique de Rennes*, S4:1–40, 1974.

[77] S. Leung and H. Zhao. A grid based particle method for moving interface problems. *Journal of Computational Physics*, 228(8):2993–3024, 2009.

[78] H. Levine, D. A. Kessler, and W-J. Rappel. Directional sensing in eukaryotic chemotaxis: A balanced inactivation model. *Proceedings of the National Academy of Sciences of the United States of America*, 103(26):9761–9766, 2006.

[79] J. Liang and H. Zhao. Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3):1461–1486, 2013.

[80] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[81] C. B. Macdonald, B. Merriman, and S. J. Ruuth. Simple computation of reaction-diffusion processes on point clouds. *Proceedings of the National Academy of Sciences of the United States of America*, 110(23):9209–9214, 2013.

[82] C. B. Macdonald and S. J. Ruuth. Level set equations on surfaces via the closest point method. *Journal of Scientific Computing*, 35(2):219–240, 2008.

[83] C. B. Macdonald and S. J. Ruuth. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing*, 31(6):4330–4350, 2010.

[84] G. MacDonald, J.A. Mackenzie, M. Nolan, and R.H. Insall. A computational method for the coupled solution of reaction-diffusion equations on evolving domains and manifolds: application to a model of cell migration and chemotaxis. *Journal of Computational Physics*, 309:207–226, 2016.

[85] J. Mackenzie and W. R. Mekwi. On the use of moving mesh methods to solve PDEs. *Adaptive communications: theory and algorithms*, 6:243–278, 2006.

[86] J. A. Mackenzie, M. Nolan, and R. H. Insall. Local modulation of chemoattractant concentrations by single cells: dissection using a bulk-surface computational model. *Interface Focus*, 6(5), 2016.

[87] A. Madzvamuse and R. Barreira. Exhibiting cross-diffusion-induced patterns for reaction-diffusion systems on evolving domains and surfaces. *Physical Review E*, 90(4):043307–043321, Oct 2014.

[88] A. Madzvamuse and A. H. W. Chung. Fully implicit time-stepping schemes and non-linear solvers for systems of reaction-diffusion equations. *Applied Mathematics and Computation*, 244:361–374, 2014.

[89] A. Madzvamuse and A. H. W. Chung. The bulk-surface finite element method for reaction-diffusion systems on stationary volumes. *Finite Elements in Analysis and Design*, 108:9–21, 2016.

[90] A. Madzvamuse, A. H. W. Chung, and C. Venkataraman. Stability analysis and simulations of coupled bulk-surface reaction–diffusion systems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2175), 2015.

[91] A. Madzvamuse, E. A. Gaffney, and P. K. Maini. Stability analysis of non-autonomous reaction-diffusion systems: the effects of growing domains. *Journal of Mathematical Biology*, 61(1):133–164, 2010.

[92] A. Madzvamuse, H. S. Ndakwo, and R. Barreira. Cross-diffusion-driven instability for reaction-diffusion systems: analysis and simulations. *Journal of Mathematical Biology*, 70(4):709–743, 2015.

[93] A. Madzvamuse, H. S. Ndakwo, and R. Barreira. Stability analysis of reaction-diffusion models on evolving domains: The effects of cross-diffusion. *Discrete and Continuous Dynamical Systems*, 36(4):2133–2170, 2016.

[94] H. Meinhardt. Orientation of chemotactic cells and growth cones: Models and mechanisms. *Journal of Cell Science*, 112:2867–2874, 1999.

[95] J. Meyers, J. Craig, and D. J. Odde. Potential for control of signaling pathways via cell size and shape. *Current Biology*, 16(17):1685–1693, 2006.

[96] K. Mikula and D. Ševčovič. Evolution of Plane Curves Driven by a Nonlinear Function of Curvature and Anisotropy. *SIAM Journal on Applied Mathematics*, 61(5):1473–1501, 2001.

[97] E. Moeendarbary, L. Valon, M. Fritzsche, A. R. Harris, D. A. Moulding, A. J. Thrasher, E. Stride, L. Mahadevan, and G. T. Charras. The cytoplasm of living cells behaves as a poroelastic material. *Nature Materials*, 12:253–261, 2013.

[98] N. Moelans, B. Blanpain, and P. Wollants. An introduction to phase-field modeling of microstructure evolution. *Calphad*, 32(2):268–294, 2008.

[99] A. Mogilner and A. Manhart. Agent-based modeling: Case study in cleavage furrow models. *Molecular Biology of the Cell*, 27(22):3379–3384, 2016.

[100] A. J. Muinonen-Martin, O. Susanto, Q. Zhang, E. Smethurst, W. J. Faller, D. M. Veltman, G. Kalna, C. Lindsay, D. C. Bennett, O. J. Sansom, R. Herd, R. Jones, L. M. Machesky, M. J. O. Wakelam, D. A. Knecht, and R. H. Insall. Melanoma cells break down LPA to establish local gradients that drive chemotactic dispersal. *PLOS Biology*, 12(10):e1001966, 2014.

[101] J. D. Murray. A pre-pattern formation mechanism for animal coat markings. *Journal of Theoretical Biology*, 88(1):161–199, 1981.

[102] J. D. Murray. Parameter space for turing instability in reaction diffusion mechanisms: A comparison of models. *Journal of Theoretical Biology*, 98(1):143 – 163, 1982.

[103] J. D. Murray. *Mathematical Biology An Introduction, third edition.* Springer, 2002.

[104] M. P. Neilson, J. A. Mackenzie, S. D. Webb, and R. H. Insall. Use of the parameterised finite element method to robustly and efficiently evolve the edge of a moving cell. *Integrative Biology*, 2(2):687–695, 2010.

[105] M. P. Neilson, J. A. Mackenzie, S. D. Webb, and R. H. Insall. Modelling cell movement and chemotaxis using pseudopod-based feedback. *SIAM Journal on Scientific Computing*, 33(3):1035–1057, 2011.

[106] M. P. Neilson, D. M. Veltman, P. J. van Haastert, S. D. Webb, J. A. Mackenzie, and R. H. Insall. Chemotaxis: a feedback-based computational model robustly predicts multiple aspects of real cell behaviour. *PLoS Biology*, 9:e1000618, 2011.

[107] S. R. Neves, P. Tsokas, A. Sarkar, E. A. Grace, P. Rangamani, S. M. Taubenfeld, C. M. Alberini, J. C. Schaff, R. D. Blitzer, I. I. Moraru, and R. Iyengar. Cell shape and negative links in regulatory motifs together control spatial information flow in signaling networks. *Cell*, 133(4):666–680, 2008.

[108] I. L. Novak, F. Gao, Y.-S. Choi, D. Resasco, J. C. Schaff, and B. M. Slepchenko. Diffusion on a curved surface coupled to diffusion in the volume: application to cell biology. *Journal of Computational Physics*, 226(2):1271–1290, 2007.

[109] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[110] G. Y. Ouaknin and P.Z. Bar-Yoseph. Stochastic collective movement of cells and fingering morphology: No maverick cells. *Biophysical Journal*, 97(7):1811–1821, 2009.

[111] Z. Pan and B. Wetton. Numerical methods for coupled surface and grain boundary motion. *European Journal of Applied Mathematics*, 19:311–327, 2008.

[112] C. Parent and P. N. Devreotes. A cell's sense of direction. *Science*, 284(5415):765–770, 1999.

[113] P. O. Persson. *Mesh Generation for Implicit Geometries.* Ph.D. thesis, Department of Mathematics, MIT, 2004.

[114] P. O. Persson and G. Strang. A Simple Mesh generator in MATLAB. *SIAM Review*, 46(2):329–345, 2004.

[115] C.S. Peskin, G.M. Odell, and G.F. Oster. Cellular motions and thermal fluctuations: the Brownian ratchet. *Biophysical Journal*, 65:16–24, 1993.

[116] R. G. Plaza, F. Sánchez-Garduño, P. Padilla, R. A. Barrio, and P. K. Maini. The Effect of Growth and Curvature on Pattern Formation. *Journal of Dynamics and Differential Equations*, 16(4):1093–1121, 2004.

[117] M. Prastawaa, E. Bullittb, and G. Geriga. Simulation of brain tumors in MR images for evaluation of segmentation efficacy. *Medical Image Analysis*, 13(2):297–311, 2009.

[118] J .D. Pryce. On the convergence of iterated remeshing. *IMA Journal of Numerical Analysis*, 9(3):315–335, 1989.

[119] P. Rajagopalan, W. A. Marganski, X. Q. Brown, and J. Y. Wong. Direct Comparison of the Spread Area, Contractility, and Migration of balb/c 3T3 Fibroblasts Adhered to Fibronectin- and RGD-Modified Substrata. *Biophysical Journal*, 87(4):2818–2827, 2004.

[120] M. Richards, O. Lomas, K. Jalink, K. L. Ford, R. D. Vaughan-Jones, K. Lefkimmiatis, and P. Swietach P. Intracellular tortuosity underlies slow camp diffusion in adult ventricular myocytes. *Cardiovascular Research*, 110(3):395–407, 2016.

[121] G. Rote. Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.

[122] E. T. Roussos, J. S. Condeelis, and A. Patsialou. Chemotaxis in cancer. *Nature Reviews Cancer*, 11(8):573–587, 2011.

[123] M. Droskeand M. Rumpf. A level set formulation for Willmore flow. *Interfaces and Free Boundaries*, 6(3):361–378, 2004.

[124] Y. Saad. *Iterative methods for sparse linear systems*. PWS, 1996.

[125] J. J. Saucerman, E. C. Greenwald, and R. Polanowska-Grabowska. Mechanisms of cyclic AMP compartmentation revealed by computational models. *Journal of General Physiology*, 143:39–48, 2014.

[126] J. J. Saucerman, J. Zhang, J. C. Martin, L. X. Peng, A. E. Stenbit, R. Y. Tsien, and A. D. McCulloch. Systems analysis of pka-mediated phosphorylation gradients in live cardiac myocytes. *Proceedings of the National Academy of Sciences of the United States of America*, 103(34):12923–12928, 2006.

[127] J. A. Sethian. *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[128] D. Ševčovič and S. Yazaki. Evolution of plane curves with a curvature adjusted tangential velocity. *Japan Journal of Industrial and Applied Mathematics*, 28(3):413–442, 2011.

[129] D. Shao, H. Levine, and W-J. Rappel. Coupling actin flow, adhesion, and morphology in a computational cell motility model. *Proceedings of the National Academy of Sciences of the United States of America*, 109(18):6851–6856, 2012.

[130] I. Singer-Loginova and H. M. Singer. The phase field technique for modeling multiphase materials. *Reports on Progress in Physics*, 71(10):106501, 2008.

[131] D. R. Soll, J. Yarger, and M. Mirick. Stationary phase and the cell cycle of Dictyostelium Discoidium in liquid nutrient medium. *Journal of Cell Science*, 20(3):513–523, 1976.

[132] S. J. Streichan, G. Valentin, D. Gilmour, and L. Hufnagel. Collective cell migration guided by dynamically maintained gradients. *Physical Biology*, 8(4):045004, 2011.

[133] K. F. Swaney, C.-H. Huang, and P. N. Devreotes. Eukaryotic chemotaxis: A network of signaling pathways controls motility, directional sensing, and polarity. *Annual Review of Biophysics*, 39:265–289, 2010.

[134] J. E. Taylor, J. W. Cahn, and C. A. Handwerker. Overview No. 98 I-Geometric models of crystal growth. *Acta Metallurgica et Materialia*, 40(7)(7):1443–1474, 1992.

[135] J. A. Theriot and T. J. Mitchison. Actin microfilament dynamics in locomoting cells. *Nature*, 352:126–131, 1991.

[136] J. E. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications*. Elsevier Science Ltd, 1985.

[137] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of Grid Generation*. CRC Press, 1998.

[138] L. Tweedy, D. A. Knecht, G. M. Mackay, and R. H. Insall. Self-generated chemoattractant gradients: Attractant depletion extends the range and robustness of chemotaxis. *PLOS Biology*, 14(3):1–22, 2016.

[139] M. Ueda, Y. Sako, T. Tanaka, P. Devreotes, and T. Yanagida. Single-molecule analysis of chemotactic signaling in Dictyostelium cells. *Science*, 294:864–867, 2001.

[140] H. A. van der Vorst. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[141] P. J. M. van Haastert. Chemotaxis: insights from the extending pseudopod. *Journal of Cell Science*, 123(18):3031–3037, 2010.

[142] P. J. M. van Haastert and R. J. De Wit. Demonstration of receptor heterogeneity and affinity modulation by nonequilibrium binding experiments. the cell surface cAMP receptor of Dictyostelium discoideum. *Journal of Biological Chemistry*, 259:13321–13328, 1991.

[143] Y. L. Wang. Exchange of actin subunits at the leading edge of living fibroblasts: possible role of treadmilling. *Journal of Cell Biology*, 101(2):597–602, 1985.

[144] Z. J. Wang. High-order methods for the Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43:1–41, 2007.

[145] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[146] E. S. Welf and J. M. Haugh. Signaling pathways that control cell migration: models and analysis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 3(2):231–240, 2011.

[147] S. Whitelam, T. Bretschneider, and N. J. Burroughs. Transformation from spots to waves in a model of actin pattern formation. *Physical Review Letters*, 102(19):198103, 2009.

[148] T. J. Willmore. *Riemannian Geometry*. Clarendon Press, 1996.

[149] A. Winslow. Numerical solution of quasi-linear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1:149–172, 1967.

[150] R.B. Wysolmerski and D. Lagunoff. Involvement of myosin light-chain kinase in endothelial cell retraction. *Proceedings of the National Academy of Sciences of the United States of America*, 87(1):16–20, 1990.

[151] Y. Xiong, C-H. Huang, P. A. Iglesias, and P. N. Devreotes. Cells navigate with a local-excitation, global-inhibition-biased excitable network. *Proceedings of the National Academy of Sciences of the United States of America*, 107(40):17079–17086, 2010.

[152] L. Yang, J. C. Effler, B. L. Kutscher, S. E. Sullivan, D. N. Robinson, and P. A. Iglesias. Modeling cellular deformations using the level set formalism. *BMC Systems Biology*, 2(1):68, 2008.