A Moving Mesh Finite Element Method And Its Application To Population Dynamics

Anna Watkins

Department of Mathematics

Reading University

This dissertation is submitted for the degree of

Doctor of Philosophy

June 2017

Declaration

I con rm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Anna Watkins
June 2017

Acknowledgements

There are many who deserve acknowledgement for their part in the completion of this thesis. The work here must be one of the longest running PhD theses of all time at 11 years, and has been completed in part-time bursts whenever life and time allowed. The beginning was as a new graduate in a rented at, coding on my rst laptop. During the course of the work, there have been two house moves, a wedding, a house renovation, a career as an athlete spanning two Olympic games, a start-up business, a new career in analytics, and two children. The work was done on planes, buses and commuter trains, at home with a baby in a sling, in any number of hotel rooms whilst on training camp and and even greater number of coffee shops, and even occasionally in the department at Reading. Therefore the support of those around me has been fundamental.

I'm very grateful to Paul Thompson, my rowing coach, who believed that space needs to be made for things like this in an athletic career. This attitude meant that I have been able to cope with retirement from sport much better than I would have otherwise. I'm also grateful to Reading University Boat Club, who gave me a sports scholarship, and to UK Sport, who gave me an education grant, between which I was able to make ends meet in the early days when money was tight. I'm grateful to my parents for the encouragement they gave. I'd also like to thank Paul Glaister and Peter Grindrod for their generous time and friendly support, in supervising me and sharing ideas and research.

There are certain people I'd like to thank simply for putting up with me. In this group are my rowing partners, particularly Annie Vernon, Elise Sherwell and Katherine Grainger, who put up with me ignoring them as we shared rooms whilst training, and even as I zoned out to think about maths in the middle of a training session in the boat. The other two important people who have put up with me are my sons William and Richard, aged 2 and 3 now. I'm sorry I was tired and busy and that you weren't allowed to mash the keys on the keyboard or watch train videos on the screen.

However I do have two very major acknowledgements to share. I'm incredibly grateful to my main supervisor Mike Baines. Mike and I have always spent our time together with equal time given to maths and life in general. Mike has heard and advised on the trials and

tribulations of my rowing career and motherhood as well as the ups and downs of research. His relentless support, friendly hello and interest in discussing any and every challenge in life means that I will count him as a lifelong friend. I always looked forward to our time, even when things weren't going well. I'm sorry Mike that I took so long and was often absent for long periods.

Finally I would like to thank my wonderful husband Oliver. His contributions to this work are too numerous to detail, but stretch comprehensively across all three of the nancial, practical and emotional ranges. Thanks for paying for our at when we rst started out.

Abstract

The moving mesh finite element method (MMFEM) is a highly useful tool for the numerical solution of partial differential equations. In particular, for reaction-diffusion equations and multi-phase equations, the method provides the ability to track features of interest such as blow-up, the ability to track a free boundary, and the ability to model a dynamic interface between phases. This is achieved through a geometric conservation approach, whereby the integral of a suitable quantity is constant within a given patch of elements, but the footprint and location of those elements are dynamic. We apply the MMFEM to a variety of systems, including for the first time to various forms of the Lotka-Volterra competition equations. We derive a Lotka-Volterra based reaction-diffusion-aggregation system with two phases, representing spatially segregated species separated by a competitive interface. We model

Table of contents

Chapter 1

Introduction

In a great many areas of study, partial differential equations (PDEs) are used to describe models, laws and systems. From the simplest of examples, the equations governing heat transfer, through to trading models for global nancial markets, the PDE gives us an approach that can tackle a vast and ever-growing range of real-world problems. We may understand and make predictions about the behaviour of complex mechanical systems, we may study the weather, or we may gain insights into biological systems. The scope of PDEs and their relevance to our lives is beyond doubt. In many of these systems we have very complex interactions for which analytical solutions are not practicable or even possible. Direct experimentation and measurement may likewise not be practical and is generally expensive. Numerical modelling is therefore the key tool to unlock our understanding of how these systems are working or how they might evolve in time. Techniques for doing so are well established and are subject to continual re nement and improvement. One particular modelling technique, the use of nite elements, has plenty to recommend it. It involves dividing the domain into small discrete elements, and calculating the effect of each part upon its neighbours. In doing so an approximation to the whole system is produced. The size and spacing of these elements can be chosen to particularly suit the shape or dynamics of the domain, and is speci ed by a grid or mesh. The mesh may be uniformly distributed or otherwise. In the particular case of time dependent PDEs, there may be advantages to having a mesh that moves with time, so that features of interest may be tracked with accuracy without the computational expense of increasing the resolution everywhere. For certain phenomena such as boundary layers, interior moving interfaces and blow-up problems this can be especially true. This is the eld of moving mesh nite element modelling, and this eld is the subject of this thesis.

1.1 Mesh adaptation

In the body of work concerning mesh adaptation, there are three basic approaches which are usually given the following names:

h-re nement is the insertion of extra mesh points around an area of interest;

p-re nement is the use of a higher-order polynomial in each interval between mesh points, so that values between mesh points are better approximated;

r-re nement is the dynamic movement of existing mesh points to track a feature of interest.

Most commonly, h-re nement and p-re nement techniques are used and are often combined together. Their strength is that the algorithms produced are versatile; they do not need to utilise any particular dynamic properties of the underlying solution. This is also a weakness, since the dynamic properties of the solution can be an excellent guide to the most ef cient mesh adaptations.

In r-re nement, the mesh nodes are assigned a velocity at each time step. This approach naturally lends itself to the solving of time dependent systems, as the time integration for the mesh movement and the solution evolution can be performed alongside one another, using any chosen integration scheme. Also, the node velocity can be chosen to work with useful properties of the system; for example one might wish to conserve mass within each element. Taking advantage of this sort of property means that, if our scheme is well chosen, the mesh evolves to re ect the solution in an ef cient and elegant way. The nodes move smoothly along with the solution. We do not need to add or remove nodes, and we do not need to interpolate the solution between nodes. The node positions and the solution are completely linked. An excellent summary of the theory and practice of r-re nement techniques can be found in Huang and Russell's book [49].

1.2 Scope of work

In this thesis, we consider in particular the application of one r-re nement technique. The technique of interest is termed the moving mesh nite element method (MMFEM). This method was developed in 2005 by Baines, Hubbard and Jimack [5], and uses a geometric conservation approach to generate mesh adaptation. A nite element construction provides the framework. We apply this method to a variety of reaction-diffusion PDE systems. We have a particular focus on multi-phase systems, where a dynamic interface exists between

phases. The MMFEM has previously been applied to the Stefan problem [8] where the dynamic interface represents the melting of ice into water. We extend this work with a simpli ed method. We then consider the application of the MMFEM to models of population dynamics. We take a version of the Lotka-Volterra competition model that, like Stefan, describes a two-phase reaction-diffusion system, and implement the MMFEM for this system. We then consider the application of the MMFEM to systems of intraspecies and interspecies interactions with aggregating dynamics. Finally, we present a new model for interspecies reactions that permits a dynamic interface combined with aggregating dynamics, as well as the more familiar reaction-diffusion dynamics. We implement the MMFEM for this model in chapter 7, and demonstrate its utility.

1.3 Novel material

This thesis contains the following novel material

- An application of the equidistribution method to a vertical water column under wind shear;
- A two dimensional MMFEM implementation for the Fisher's equation for the rst time;
- A two dimensional MMFEM implementation for the Keller-Segel model for the rst time;
- The rst numerical model of the two phase Lotka Volterra competition system derived by Hilhorst et al. [31]. We use the MMFEM to achieve this;
-

Chapter 2

Technical background

In this thesis we apply a moving mesh finite element method to a variety of systems, with a particular focus on population dynamics. Here we set out the historical evolution of moving mesh methods, and also a history of PDE systems for population modelling.

2.1 Moving mesh methods

In moving the mesh, we have two fundamentally different approaches. We may use a system that provides a mapping to move the nodes at each time step in a fixed, Eulerian frame, or we may construct the entire system in a Lagrangian, or moving, co-ordinate system. Following [18], we will call these location-based, and velocity-based methods, respectively. An overview of these methods is given here. For a more detailed summary, the 2009 paper by Budd, Huang and Russell [15] is recommended.

2.1.1 Location-based methods

The common feature of this class of methods is that the location of the mesh nodes at a particular time step is directly controlled by a mapping function. The principle most often used to achieve this is equidistribution. Equidistribution is a term used to describe the locating of points such that a particular monitor function, for example arc length, is the same for all intervals between nodes. This is achieved either directly, or by defining the mapping as the minimiser of a functional. In one dimension, consider the case of an adaptive mapping $x(\xi;t)$ from a computational domain Ω_c to a physical domain Ω. If we are using a uniform computational mesh then $\frac{\partial x}{\partial \xi}$ is the density of the mesh on Ω. We then

choose a monitor function $M(x) > 0$ and require the mesh density to be proportional to it,

$$\frac{\partial x}{\partial \xi} = c\, M(x). \tag{2.1}$$

The equivalence to a functional approach is apparent if we take the quadratic functional:

$$I[x] = \int_W [M(x)]^{-1} \left(\frac{\partial x}{\partial \xi}\right)^2 dx \tag{2.2}$$

for which the corresponding Euler-Lagrange equation is:

$$\frac{\partial}{\partial \xi}\left([M(x)]^{-1}\frac{\partial x}{\partial \xi}\right) = 0 \tag{2.3}$$

which is the same as dividing (2.1) by $M(x)$ and differentiating, and can be solved with a given M to give x in terms of ξ. The functional approach is useful as it is comparatively easily extended to higher dimensions.

An early example of the use of equidistribution is given by White [52]. He uses the integral version of the equidistribution principle (2.1) which is, in continuous form:

$$\int_0^{x(\xi,t)} M(x(\xi,t);t)\,dx = \xi \int_0^1 M(x(\xi,t);t)\,dx \qquad \forall t. \tag{2.4}$$

If this is differentiated with respect to ξ we obtain

$$M(x(\xi,t);t)\frac{\partial}{\partial \xi}x(\xi,t) = q(t) \tag{2.5}$$

where

$$q(t) = \int_0^1 M(x(\xi,t);t)\,dx$$

and differentiating with respect to ξ again gives

$$\frac{\partial}{\partial \xi}\left(M(x(\xi,t);t)\frac{\partial}{\partial \xi}x(\xi,t)\right) = 0. \tag{2.6}$$

This will generally be nonlinear and so has been solved using an iterative approach by Baines [3]. We use this approach in Chapter 4, where we use an arc length monitor function to update the node spacing for a water column model with coriolis forces.

2.1.2 Moving mesh partial differential equations (MMPDEs)

It is recommended by Huang, Ren and Russell in their 1994 paper [32] to choose a method that generates moving mesh equations in a continuous form. A simple algorithm is also very desirable. This is achieved in their work by constructing moving mesh partial differential equations (MMPDEs) directly from an equidistribution principle. This is a neat and elegant construction that avoids having to consider user-de ned input parameters in the mesh mapping. In taking this approach a more stable and more general algorithm can be produced. A simple example is given here. Huang*et al.* derive a MMPDE by differentiating (2.6) with respect to time to give

$$\frac{d}{dt}\left[\frac{\partial}{\partial x}\left[M(x(x,t),t)\frac{\partial}{\partial x}x(x,t)\right]\right] = 0 \tag{2.7}$$

which can be rearranged to give the MMPDE

$$\frac{\partial}{\partial x}\left[M\frac{\partial \dot{x}}{\partial x}\right] + \frac{\partial}{\partial x}\left[\frac{\partial M}{\partial x}\dot{x}\right] = -\frac{\partial}{\partial x}\left[\frac{\partial M}{\partial t}\frac{\partial x}{\partial x}\right] \tag{2.8}$$

where$\dot{x}(x,t)$ is the mesh velocity. A great variety of MMPDEs exist, which vary in their approach to temporal and spatial smoothing and regularisation. The power of selecting the right one was demonstrated by Budd*et al.* in 1996 [14]. They took an MMPDE from a 1986 paper [1] and applied it to a blow up problem. The MMPDE they used was derived from (2.7) using temporal smoothing and is

$$\frac{\partial^2 x}{\partial x^2} = -\frac{1}{r}\frac{\partial}{\partial x}\left[M\frac{\partial x}{\partial x}\right] \tag{2.9}$$

where$r$ is a small relaxation time after which the mesh is to reach equidistribution. This form has scale invariance properties. Here it is demonstrated that the use of monitor functions which incorporate such key properties of the original PDE can be particularly useful, as they allow features such as scaling invariance to be preserved. Natural spatial features of the PDE are inherited by the MMPDE. In this paper, self-similar or approximately self-similar solutions of blow-up equations are shown to be successful.

Another key concept was introduced by Budd and Williams in their 2006 paper [16]. They solve a relaxed form of the Monge–Ampere equation to compute a transformation from a regular (computational) to the desired spatially non-uniform mesh. The method involves the creation of a mesh potential which determines the location of the mesh points. Using the Legendre transformation, the equidistribution principle is transformed into the

Monge–Ampere equation giving the mesh potential.

2.1.3 Velocity-based methods

The following velocity based methods make use of the Arbitrary Langrangian Eulerian (ALE) form of the PDE; that is to say that a moving co-ordinate system is used to directly provide the mesh velocity. The form provides a mapping from the xed to the moving frame. Consider the time dependent PDE

$$\frac{\P u}{\P t} = L\ u; \tag{2.10}$$

where $u = u(x;t)$ is de ned in a xed (Eulerian) reference frame, and L is a differential operator involving only space derivatives. To rewrite this in a moving (Lagrangian) frame, we allow x to be a moving co-ordinate $x(t)$, which is related to a set of reference co-ordinates a by the invertible mapping

$$x = \hat{x}(a;t) \tag{2.11}$$

where the hat denotes a mapping from the Eulerian frame to the moving frame. We can then de ne the solution $u(x;t)$ in the moving frame:

$$u(x;t) = u(\hat{x}(a;t);t) = \hat{u}(a;t) \tag{2.12}$$

and then by the chain rule

$$\frac{\P \hat{u}}{\P t} = \frac{\P \hat{x}}{\P t} \cdot \tilde{N}u + \frac{\P u}{\P t} \tag{2.13}$$

where we clarify that

$$u = \frac{\P \hat{u}}{\P t}; x = \frac{\P \hat{x}}{\P t}: \tag{2.14}$$

The ALE form of the PDE is then

$$u - x \cdot \tilde{N}u = L\ u: \tag{2.15}$$

There are now two unknowns, u and x, so we must know the mesh velocities before we are able to nd the solution. The speci c method for constructing these velocities varies from using a real physical motion that provides a natural reference frame, through to de ning the motion with the sole aim of optimizing geometric properties of the mesh. The velocity can be de ned in any 39r323elo6cwvid Tf 171.393i61255(de lw)]TJs93i612struct9(sd(95

Moving finite elements

The MFE method of Miller and Miller, [38] and [39], involves taking the PDE (2.10) and determining the solution and the mesh simultaneously. This is achieved by minimising a discrete residual of the ALE form of the PDE (2.15) in a moving frame. Miller and Miller made the first attempts at a moving mesh of finite elements to deal with a model involving a sharp transition layer. These attempts made use of Burgers' equation as a test equation

innovation in the paper is the use of a mesh velocity potential in the calculation of grid velocities; this can make the nite element formulation better conditioned as certain asymmetric matrices can be substituted out. Furthermore, velocities in two or more space dimensions can be uniquely calculated from it. The mesh velocity potential idea is extensively used in this eld after this publication.

The conservation method

The 2005 paper by Baines, Hubbard and Jimack [5] takes the GCL concept and rmly establishes it from a nite element perspective. This method shares common roots with the GCL, but instead of using the variational principle to nd the mesh velocities they are directly calculated from the integral form of the PDE. This is achieved by taking a weak form of the PDE that includes a set of weight functions that move with the mesh. Then the Reynolds Transport Theorem is used to provide a link between the Eulerian and Lagrangian perspectives. A system is constructed where the mesh velocity is given in terms of a potential at a particular location (Eulerian view), but the elements themselves track the movement of mass (Lagrangian view). This gives rise to the Arbitrary Lagrangian Eulerian (ALE) equation, where a single equation ties together the relationship between the moving and static reference frames. The examples demonstrated each conserve a proportion of a quantity within each patch of elements. This may be mass itself for systems where mass is conserved overall, in which case the simplest form of the theory can be used. This is demonstrated for the porous medium equation and a fourth-order nonlinear diffusion equation. For non-conservative systems, the theory uses the concept of relative mass, the proportion of total mass associated with each element patch, and this is applied to a Stefan problem and a diffusion problem with a negative source term. The method is extended in their 2006 paper [7] to include the solution of scale invariant PDEs. This exploits the inherent independence of physical systems from any given unit system. Again using moving mesh nite element systems, the time stepping is coupled to the mesh resolution, resulting in a scheme that provides uniform local accuracy in time. This exploitation of scale invariance is not an option for xed mesh models since they are time-independent and therefore cannot exploit the coupling of dependent and independent variables in time.

2.1.4 Monitor functions

The choice of a suitable monitor function is of course key. The choice will be in uenced by the underlying physics of the system as well as the moving mesh method itself. There are

three classes of construction:

- An estimate of a quantity related to the solution such as arc length or mass, that can be made at the prior time step;
- An estimate of the error at each node or across each element, which can then be corrected by a suitable mesh adjustment. This is the approach used in moving nite element methods, where the mesh movement is determined by the velocity term in an ALE equation;
-

where k_1 and k_2 are the carrying capacities of species 1 and 2 respectively, K_1 is a measure of the effect that species 1 has on species 2, and K_2 is a measure of the effect species 2 has on species 1. The parameters r_1 and r_2 are a measure of the timescales upon which births and deaths operate.

These early sets of equations did not consider spatial effects, so an important development was made by Conway and Smoller in 1977 [22], where a diffusion term was included along with spatial dependence. This allowed the study of a vastly increased range of phenomena, such as the geographic spread of invasive species, or of disease, or the effect of non-homogenous resource distribution. When random motion of the individuals is considered in the form of a diffusion term, the Lotka-Volterra equations are of reaction-diffusion form. We have

$$\begin{aligned} \frac{\partial u_1}{\partial t} &= d_1 \tilde{N}^2 u_1 + f(u_1; u_2) u_1 \\ \frac{\partial u_2}{\partial t} &= d_1 \tilde{N}^2 u_2 + g(u_1; u_2) u_2 \end{aligned} \tag{2.18}$$

where d_1, d_2 are constant diffusion coefficients, and with $f(u_1; u_2)$ and $g(u_1; u_2)$ given by the logistic equations

$$\begin{aligned} f(u_1; u_2) &= r_1 \left(1 - \frac{u_1 - K_1 u_2}{k_1}\right) \\ g(u_1; u_2) &= r_2 \left(1 - \frac{u_2 - K_2 u_1}{k_2}\right) : \end{aligned} \tag{2.19}$$

It is this set of equations which is of interest to us here.

Chapter 3

The MMFEM and existing applications

3.1 The moving mesh nite element method

The conservation method of Baines, Hubbard and Jimack [5] can be implemented from either a nite difference or nite element perspective. Using the nite element method can be more computationally expensive than the nite difference method, but can be more easily extended to higher dimensions, and depending on the system, more stable. Furthermore, nite elements lend themselves well to being applied to complex geometries, although that is beyond the scope of this work. In this thesis we will take a nite element approach. This approach is termed the Moving Mesh Finite Element Method (MMFEM), and is the foundation of the methods in this thesis. The mass conservation concept involves assigning

equation (3.7) we can cancel out terms

$$\int_{W(t)} w_i \frac{\partial u}{\partial t} + w_i \nabla \cdot (u\mathbf{x}) \; dW = 0 \tag{3.12}$$

giving us the weak form of a PDE in the moving frame,

$$-\int_{W(t)} w_i \nabla \cdot (u\mathbf{x}) dW = \int_{W(t)} w_i L u \, dW \tag{3.13}$$

or, after integration by parts,

$$-\int_{S(t)} w_i u\mathbf{x} \cdot \hat{n} \, dS + \int_{W(t)} u\mathbf{x} \cdot \nabla w_i dW = \int_{W(t)} w_i L u \, dW \tag{3.14}$$

where $\hat{n}$

form of the definition (3.15) of f,

$$\int_{W(t)} w_i \chi \, dW = \int_{W(t)} w_i \tilde{N} f \, dW. \tag{3.17}$$

The Eulerian velocity χ is now known, and a moving reference frame can be generated. This can be considered as a deformation of $\hat{x}$ in time, derived from the ODE system

$$\frac{d\hat{x}}{dt} = \chi(\hat{x}, t) \tag{3.18}$$

with initial condition $\hat{x} = x$. Once $\hat{x}$ has been found we can recover the solution from the mass conservation principle (3.8) in the form

$$\int_{W(t)} w_i(\hat{x}(t), t) u(\hat{x}(t), t) dW = \int_{W(0)} w_i(\hat{x}(0), 0) u(\hat{x}(0), 0) dW \tag{3.19}$$

at any later time t.

Algorithm 1

The solution of the mass conserving equation (3.1) on the moving mesh therefore consists of the following steps.
Given functions u and χ initially, for each time t:

1.

and the test volume $W(t)$ is defined to be the total spatial domain of the model at time t, moving with velocity x. We introduce again our weight function w_i, also moving with velocity x, as in (3.7). Again we require that w_i is part of a set of functions that together form a partition of unity. We now define the moving co-ordinate system by requiring that the integral of u multiplied by that moving weight function is a constant proportion of the total mass in the system, i.e.

$$\int_{W(t)} w_i u \, dW = c_i q(t) \tag{3.21}$$

where the constant c_i is determined by the initial w_i and the initial data. Since $\sum_i w_i = 1$, it follows that $\sum_i c_i = 1$ also. Differentiating with respect to time gives

$$\frac{d}{dt} \int_{W(t)} w_i u \, dW = c_i \frac{dq}{dt} = c_i q(t). \tag{3.22}$$

As in the case of conserved total mass, we define a reference test domain $W(0)$ at $t = 0$ and a moving test volume $W(t)$ in the moving frame x. Applying the Reynolds Transport Theorem to $w_i u$, we obtain

$$\begin{aligned} \frac{d}{dt} \int_{W(t)} w_i u \, dW &= \int_{W(t)} \frac{\partial}{\partial t}(w_i u) dW + \int_{S(t)} w_i u x \cdot \hat{n} \, dS \\ &= \int_{W(t)} w_i \frac{\partial u}{\partial t} + u \frac{\partial w_i}{\partial t} + \tilde{\nabla} \cdot (w_i u x) \quad dW \end{aligned} \tag{3.23}$$

for the generalised weak form of the PDE. Using the advection equation (3.7) we can cancel out terms as before, giving us the weak form of the PDE in the moving frame,

$$\frac{d}{dt} \int_{W(t)} w_i u \, dW - \int_{W(t)} w_i \tilde{\nabla} \cdot (u x) dW = \int_{W(t)} w_i L u \, dW. \tag{3.24}$$

We now use the relative conservation principle (3.21) to make a substitution. We use the weak form (3.22) to give

$$c_i q(t) - \int_{W(t)} w_i \tilde{\nabla} \cdot (u x) dW = \int_{W(t)} w_i L u \, dW. \tag{3.25}$$

After integration by parts we obtain

$$c_i q(t) + \int_{W(t)} u x \cdot \tilde{\nabla} w_i dW = \int_{W(t)} w_i L u \, dW + \int_{S(t)} w_i u x \cdot \hat{n} \, dS. \tag{3.26}$$

The boundary $uxux \cdot \hat{n}$ is again assumed to be given by the boundary conditions. We now

have an expression for $\dot{x}$ in terms of u and $\dot{q}$. So long as we select weight functions w_i that form a partition of unity, $\sum_i w_i = 1$, we can calculate $\dot{q}$ by summing this expression over all weight functions in the model and using the boundary conditions. Recalling that $\sum_i c_i = 1$, we sum equation (3.26) over i, to give

$$\sum_i c_i \dot{q}(t) + \sum_i \int_{W(t)} u\dot{x} \cdot \nabla w_i \, dW = \sum_i \int_{W(t)} w_i Lu \, dW + \sum_i \int_{S(t)} w_i u\dot{x} \cdot \hat{n} \, dS \, . \tag{3.27}$$

Noting that $\nabla \sum w_i = 0$,

$$\dot{q}(t) = \int_{W(t)} Lu \, dW + \int_{S(t)} u\dot{x} \cdot \hat{n} \, dS \tag{3.28}$$

will determine q, providing that the boundary $u\dot{x}$ is indeed known.

A velocity potential is then introduced in the same way as for the conservative case. A velocity potential, f is defined,

$$\dot{x} = \nabla f \tag{3.29}$$

so that equation (3.26) can be rewritten as

$$c_i \dot{q}(t) + \int_{W(t)} u \nabla f \cdot \nabla w_i \, dW = \int_{W(t)} w_i Lu \, dW + \int_{S(t)} w_i u\dot{x} \cdot \hat{n} \, dS \tag{3.30}$$

and we are now able to uniquely determine $\dot{x}$ in terms of u and $\dot{q}$, as long as f is given at one point at least. As before, the recovery of $\dot{x}$ is made using the weak form of the definition (3.29) of f

$$\int_{W(t)} w_i \dot{x} \, dW = \int_{W(t)} w_i \nabla f \, dW \, . \tag{3.31}$$

Having updated $\dot{x}(t)$ from $\dot{x}$ and $\dot{q}(t)$ from $\dot{q}$ using a suitable integration procedure, we can now recover u from the relative conservation principle,

$$\frac{1}{q(t)} \int_{W(t)} w_i(\hat{x}(t);t) u(\hat{x}(t);t) dW = \frac{1}{q(0)} \int_{W(0} \int_{W} \int_{0}^{t}$$

1. Find $q(t)$ from (3.28)
2. Find the velocity potential by solving equation (3.30) for $\phi(x,t)$;
3. Find the deformation velocity by solving equation (3.31) for $\dot{x}(t)$;
4. Generate the moving co-ordinate $\hat{x}$ at the next time step $t + dt$ by integrating (3.18). Similarly, update $q(t + dt)$ from $q(t)$;
5. Find the solution $u(\hat{x}(t + dt), t + dt)$ by solving the relative conservation equation coi10 G [

form of (3.28),

$$q = \int_{a(t)}^{b(t)} Lu\,dx + [ux]_{a(t)}^{b(t)}. \tag{3.36}$$

The weak integral form defining the one dimensional velocity potential is

$$\int_{a(t)}^{b(t)} w_i x\,dx = \int_{a(t)}^{b(t)} w_i \P f$$

Fig. 3.1 Weight or basis functions for nodes x_0 and x_i

ically uses the weight functions $W_i(x)$ rather than the generic weight function w_i, so

$$c_i q(t) - \left[W_i u \frac{\partial f}{\partial t} \right]_{a(t)}^{b(t)} + \int_{a(t)}^{b(t)} u \frac{\partial f}{\partial x} \frac{\partial W_i}{\partial x} dx = \int_{a(t)}^{b(t)} W_i Lu \, dx \qquad (i = 0, \ldots, N+1). \tag{3.39}$$

Now we can substitute finite element approximations F, X and U for f, x and u respectively. These are also piecewise linear in form, and are linear combinations of basis functions $W_j(t)$. The basis functions $W_j(t)$ are often chosen to be the same set of functions as the weight functions $W_i(t)$, although this does not have to be the case. Here we will use the same definitions for $W_j(t)$ and $W_i(t)$ unless we specify otherwise. We will use the subscript i for weight functions, and the subscript j for basis functions. For example, the function $F(x,t)$ is defined as

$$F(x,t) = \sum_{j=0}^{N+1} F_j(t) W_j(x,t). \tag{3.40}$$

In this formulation each of the $N+2$ nodes will have a coefficient F_j associated with it and (3.40) will form a linear spline. Similarly we also define

$$X(x,t) = \sum_{j=0}^{N+1} X_j(t) W_j(x,t) \tag{3.41}$$

and

$$U(x,t) = \sum_{j=0}^{N+1} U_j(t) W_j(x,t). \tag{3.42}$$

We can also use the result that

$$\frac{\partial F}{\partial x} = \sum_{j=0}^{N+1} F_j \frac{\partial W_j}{\partial x}. \tag{3.43}$$

The process for assembly of $\underline{\underline{K}}(\underline{U})$ is outlined in section 3.1.2. The matrix is singular but a value of f is imposed at one point.

The right hand side $\underline{L}$ of equation (3.49) can take many forms depending on the nature of the operator L and the boundary conditions. It may be necessary to make substitutions and/or perform integration by parts to obtain a computable form: a weak form requiring functions once-differentiable between nodes only. Note that if we sum over all rows of (3.49) the rows of the stiffness term $\underline{\underline{K}}(\underline{U})$ of (3.49) will sum to zero, and the c_i values will sum to unity. Providing that $\int_{a(t)}^{b(t)} L\,dx$ is known, this makes it possible to recover v_p as the

RecoveringU

Equation (3.34) is

$$\int_{a(t)}^{b(t)} w_i u \, dx = c_i q(t). \tag{3.56}$$

Since the c_i are constant, we may write

$$\frac{1}{q(t)} \int_{a(t)}^{b(t)} w_i$$

3.1 The moving mesh nite element method

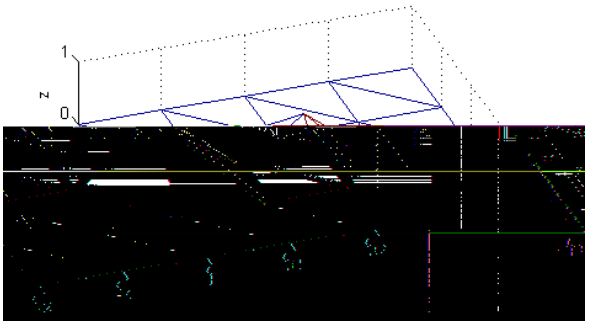


Fig. 3.2 Weight function or basis function centred at the node X_i

variety of suitable functions available but in this thesis we will use the simplest option, that of piecewise linear functions on a triangulated domain.

We triangulate the domain $\Omega(t)$ of the PDE we wish to solve. The nodes of the triangulation will be $\{X_i\}$, $(i = 1, \ldots, N)$.

We define the two-dimensional weight function $W_i(x)$ as the piecewise linear function

We use the result that

$$\tilde{\nabla} F = \sum_{j=1}^{N} F_j \tilde{\nabla} W_j \tag{3.65}$$

and make a substitution for $\tilde{\nabla} f$ and u into equation (3.62), giving

$$c_i \dot{q}(t) + \sum_{j=1}^{N} \int_{W(t)} U \tilde{\nabla} W_j \cdot \tilde{\nabla} W_i \, dW \; F_j = \int_{W(t)} W_i L U \, dW + \int_{S(t)} W_i U \mathbf{x} \cdot \hat{n} \, dS. \tag{3.66}$$

We also use the weight functions to evaluate the values c_i, recalling equation (3.21)

$$c_i = \frac{1}{q(t)} \int_{W(t)} W_i U \, dx. \tag{3.67}$$

We may construct (3.66) for every triangle and node combination, and thus obtain a linear system of equations. When doing so we must take special care to include the boundary term for domain edge boundary triangles. The boundary terms for internal triangle edges will cancel out, since each edge connects two triangles which will have opposite outwards pointing normals $\hat{n}$, and U is continuous. Each triangle with an edge lying along the boundary does make a contribution to the boundary term, so that in the sum of these contributions the whole boundary has been considered.
The assembled matrix equation has exactly the same form as the 1D case,

$$\underline{C} \dot{q}(t) + K(\underline{U}) \underline{F} = \underline{L}. \tag{3.68}$$

However, the weighted stiffness matrix in 2D is given by

$$K(\underline{U})_{ij} = \int_{W(t)} U(x) \tilde{\nabla} W_i(x) \cdot \tilde{\nabla} W_j(x) dW. \tag{3.69}$$

As in the 1-D case, we can use (3.68) to obtain $\dot{q}$ by summing over all rows. The stiffness term will sum to zero and the c_i values will sum to unity, leaving the boundary terms assumed known and $\dot{q}$ as the only unknown. We can then use (3.68) in full form to determine the vector $\underline{F}$, imposing a value of F_j at one point.

Recovering X

To find x, we work from the definition of f (3.16)

$$x = \tilde{\nabla} f \tag{3.70}$$

for which a weak form is

$$\int_{W(t)} w_i \dot{x}\, dW = \int_{W(t)} w_i \tilde{N} f\, dW. \tag{3.71}$$

Using again the linear weight functions $w_i = W_i(x,t)$, and (using basis functions W_j) the piecewise linear approximations $x = \sum_j X_j(t) W_j(x,t)$ and $\tilde{N}F = \sum_j F_j(t) \tilde{N} W_j(x,t)$

Algorithm 4

The finite element solution of the non mass conserving equation (3.1) on the moving mesh therefore consists of the following steps. Given the initial U and X, and having calculated $\underline{C}$ and the initial q from the definition (3.20), then for each time-step

1. Find $\dot{q}(t)$ by summing over all rows of the matrix equation (3.68);
2. Find the velocity potential by solving equation (3.68) for the $\Phi_j(t)$ values;
3. Find the node velocity by solving equation (3.73) for the $\dot{X}_j(t)$ values;
4. Generate the moving nodes $X_j(t + dt)$ at the next time-step by solving (3.18) using the forward Euler approximation. Update $q_j(t + dt)$ from $q(t)$ in the same way;
5. Find the solution $U(t + dt)$ by solving the relative conservation equation in the form (3.75).

Constructing the 2-D weighted stiffness matrix

The entries of the weighted stiffness matrix (3.69) are

$$K(\underline{U})_{ij} = \int_{W(t)} U(x,t)\tilde{\nabla}W_i(x,t) \cdot \tilde{\nabla}W_j(x,t)dW.$$

In order to determine the entries for each element matrix we examine a weight or basis function W_A on triangle w_e represented by the co-ordinates x_k labelled (A, B, C) (figure 3.3). The triangle has angles a, b, g as shown in figure 3.3. Such a triangle actually contains three local linear functions W_A, W_B, W_C, one associated with each node. The gradient of each weight or basis function can be calculated from the properties of the triangle. If $height_A$ is the height of triangle w_e in the direction of the normal to side BC,

$$|\tilde{\nabla}W_A| = \frac{1}{height_A} = \frac{1}{b\sin g}. \tag{3.76}$$

Likewise,

$$|\tilde{\nabla}W_B| = \frac{1}{height_B} = \frac{1}{c\sin a} \tag{3.77}$$

and

$$|\tilde{\nabla}W_C| = \frac{1}{height_C} = \frac{1}{a\sin b}. \tag{3.78}$$

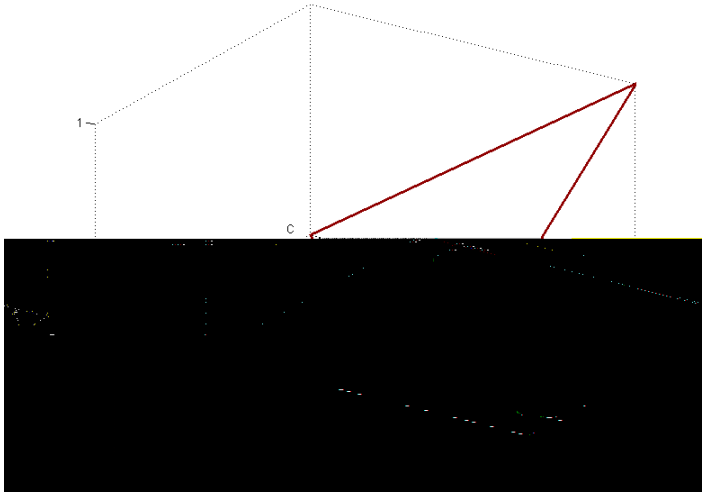


Fig. 3.3 Weight or basis function W_A centred at the node A

The area of the triangle can be calculated from any of these heights as

$$\begin{aligned} \text{area}_e &= \frac{1}{2}a(\text{height}_A) = \frac{1}{2}b(\text{height}_B) = \frac{1}{2}c(\text{height}_C) \\ &= \frac{1}{2}bc\sin\alpha = \frac{1}{2}ca\sin\beta = \frac{1}{2}ab\sin\gamma. \end{aligned} \tag{3.79}$$

The piecewise linear construction of function U means that U is linear in each triangle. We can therefore use the mean to give U in any triangle. Taking each combination of two nodes at a time, the entries for the element stiffness matrix can be determined. For example,

$$\begin{aligned} K_{BC} &= \int_{w_e} U\nabla W_B \cdot \nabla W_C dx \\ &= \frac{U_A + U_B + U_C}{3} \quad \frac{\text{area}_e}{(\text{height}_B)(\text{height}_C)}(-\cos\alpha) \\ &= \frac{U_A + U_B + U_C}{3} \quad \frac{\frac{1}{2}bc\sin\alpha}{bc\sin^2\alpha}(-\cos\alpha) \\ &= \frac{U_A + U_B + U_C}{3} \quad \frac{1}{2\sin\alpha}(-\cos\alpha) \\ &= \frac{(U_A + U_B + U_C)}{3} \frac{(-\cot)}{2}\alpha \end{aligned} \tag{3.80}$$

and since $\tilde{N}(W_A + W_B +$

after integration by parts, and imposing the boundary condition $u = 0$. In [5] F was computed using a finite element approximation in both one and two dimensions. The model was found to be second-order accurate for $n = 1$, although of lower accuracy for $n = 3$. An

3.2.3 A Stefan problem

A non mass conserving example is given in the original 2005 Baines paper [5]. The single phase Stefan problem describes heat diffusion in two dimensions given by the PDE

$$\frac{\partial u}{\partial t} = k\tilde{\nabla}^2 u \tag{3.89}$$

with different values for k and an interface boundary condition

$$\frac{\partial u}{\partial n}_{G_1} = C_L x \cdot n \qquad u_{G_1} = u_B \tag{3.90}$$

where C_L is the heat of phase change per unit volume, and the temperature at the interface is the constant u_B. Here G_1 represents a moving boundary. This is the basis for the later two-phase method in [8].

3.2.4 Finite difference implementations

As has been mentioned, the underlying conservation method behind the MMFEM can also be implemented numerically from a finite difference perspective. The conservation method with a finite difference approach has been applied to a wide variety of problems in one and two dimensions. herem7(tw)-267(the)-267(heat)-2ons.d

A different combined system of equations was studied by S.Cole in [21]. Models of chemotaxis using the Keller-Segel equations were solved in one and two (radial) dimensions using a nite difference conservation method. The system involves a substrate and a reactant, and the PDEs are

$$u_t = \tilde{N}\cdot(k_1(u,v)\tilde{N}u - k_2(u,v)u\tilde{N}v) + k_3(u,v) \tag{3.93}$$

and

$$v_t = D_v\tilde{N}^2v + k_4(u,v) - k_5(u,v)v \tag{3.94}$$

where u is cell density, v is concentration of substrate, k_1 is diffusivity, k_2 is chemotactic sensitivity, k_3 is cell growth and death, k_4 is production of substrate and k_5 is degradation of substrate.

Free boundary problems

In their 2015 paper [36], Lee, Baines and Langdon use the nite difference implementation of the method to examine free boundary problems in one dimension. These included the Porous Medium equation, Richards equation and the Crank Gupta problem. A moving boundary is introduced with a ux boundary condition. For a boundary $x = b(t)$, the boundary conditions are

$$u(b(t),t) = 0, \qquad u(b(t),t)\frac{db}{dt} = 0 \tag{3.95}$$

This is found to provide solutions accurate to second order.

In this thesis we will take a selection of these nite difference implementations and derive, implement and study the corresponding MMFEM.

3.3 Extensions to the MMFEM

Since the Baines, Hubbard and Jimack papers [5] and [7], a variety of interesting applications and extensions to the method have been investigated. Excellent overviews are given in the review papers [6] and [36]. It has been demonstrated that forms of the method can

High order nonlinear diffusion

In his PhD thesis [10], N.Bird considers nonlinear diffusion of second, fourth and sixth order. The MMFEM is applied, and interestingly an alternative type of higher order basis function is also tried. In one dimension Lagrange polynomials of linear, quadratic and fourth-order forms are used to provide a basis for the nite element approximation. The MMFEM is compared with a nite difference method. Some practical dif culties in applying the nite difference method are considered. These arise when the boundaries are permitted to move, resulting in certain functions becoming unbounded and singularities being introduced. It is found that the MMFEM alleviates this problem partially, although undesirable oscillations are still observed.

Two phase Stefan problem

In 2009, Baines, Hubbard and Jimack together with Mahmood [8] present a version of the algorithm from [5] in the form of an Arbitrary Lagrangian-Eulerian equation (ALE) that is suf ciently general to be able to model a two phase problem with a moving interface. Each phase is a diffusion system with driving PDEs

$$K_S \frac{\P u}{\P t}$$

93.6(h)]TJ/F8711.9552Tf1208.790-9.4265Td[(K)]TJ/F878.9664Tf7.675-17913Td[LS

¶t

solved as a single system to provide the mesh velocities. The masses are recovered for each phase separately since they are decoupled by the interface; these are again obtained from the conservation properties. This work is extended and developed for a new system in Chapter 5 of this thesis.

Ice sheets

In the 2013 PhD thesis [40] by Partridge and the subsequent paper in collaboration with Bonanet al. [11], a 1-D MMFEM is applied to dynamic ice ow equations to model the evolution of a glacier. The method is able to accurately capture and track the glacial front using a moving boundary framework, and the model is extended to two dimensions. In addition real world data is assimilated using the 3d-var scheme. This is found to work well in one dimension and to improve the accuracy of the pro le of the ice front. In two dimensions the moving mesh alone works well, but the data assimilation aspect of the problem remains open-ended.

Explicit and implicit time-stepping schemes

In the methods described above, the time-stepping schemes are usually simple choices such as the explicit Euler method. These can place considerable constraints on the size of the time-step that can be made, because mesh tangling can occur. This is caused by nodes overtaking one another, and imposes a limitation on the speed of computation such that it becomes impractical to run models for long time horizons. A particular semi implicit or implicit method is proposed by Baines and Lee in the 2014 paper [9] that can make it impossible for nodes to tangle in one dimension. This allows us to choose a larger time-step. The method involves manipulating the structure of the velocity equation so that it makes use of its similarity to a variable co-ef cient heat equation. A maximum/minimum principle can then be employed which makes it impossible for nodes to overtake. An alternative explicit method is given by Baines in his 2015 paper [4]. This method focuses on the node spacings or edge lengths (in 2D) and employs an ampli cation factor to calculate the distances. This factor is always positive and prevents overtaking. This is implemented in a nite difference framework in one dimension and the extension to two dimensions is outlined. It is noted that smoothness problems in ancillary variables may occur in certain circumstances.

Phase eld models

Another approach to the study of phase transitions and interfaces is to attempt the modelling of a small but nite transition layer between two uniform bulk phases. These are known as phase eld models, and a moving mesh nite element approach is discussed by Zhang and Du [53]. In these models the eld varies smoothly but with a steep gradient in the transition layer. The example used is the Allen-Cahn equation, and the challenge of suitably resolving the thin interface layer is discussed with reference to appropriate time-stepping schemes and numerical stability. The paper also examines cases where such layers move over time such that dynamically evolving fronts can be tracked with an appropriately adapting mesh.

Chapter 4

New applications for MMFEMs

We shall begin this chapter by illustrating methods that form a part of a development pathway for MMFEMs. This will allow us to become familiar with useful techniques as well as to assess the incremental bene ts offered by each evolutionary step in the development of the MMFEM.

4.1 An Illustration of the Equidistribution Method: a vertical velocity pro le

The Ekman spiral [25] is a structure of currents near the ocean surface in which the ow direction rotates as one moves away from the surface. It was rst noted by Swedish oceanographer Fridtjof Nansen, who observed an ice oe drifting at a tangent to the wind direction, and whose observations allowed Ekman to develop his model. The rotation is driven by the Coriolis effect. A feature of this structure of currents is the development of a shallow layer (Ekman layer) with behaviour that differs from the water below. The development of this layer in an initially stationary water column subject to wind stress is an interesting candidate for a moving mesh model, because we might wish to adapt the mesh to better resolve the emerging layer. Here we illustrate the equidistribution method and assess its utility for resolving the Ekman layer. A column of water is modelled under wind stress and with a Coriolis effect taken into account. A 1-D nite element method is used, and both the xed mesh and an adaptive scheme for the mesh are considered. In this example, the adaptive scheme will be the equidistribution method, using arc length as a monitor function. We also consider alternative monitor functions. Time integration is performed using an Adams Bashforth method of third order.

The PDE of interest is

$$\frac{\P u}{\P t} + F e_z \times u = \frac{\P}{\P z}\left(k_u \frac{\P u}{\P z}\right) \tag{4.1}$$

where $u(z;t)$ is the velocity, a function of the depth z. The velocity has two horizontal components u_x and u_y. The physical constants are $F = 10^{-4}s^{-1}$ (the Coriolis force) and $K_u = 10^{-2}m^{-2}s^{-1}$ (eddy viscosity, a function of density, here assumed to be constant). The boundary condition at the deepest extent of the water column $z = -h$ is the Dirichlet condition $u(z = -h;t) = 0$. On the surface $z = 0$, we have a wind shear providing a ux boundary condition $\frac{\P u}{\P z}\Big|_0 = \boldsymbol{b}$ with components $b_x = 10^{-2}s^{-1}$ and $b_y = 0$. The initial conditions are $u(z;t = 0) = 0$.

We separate (4.1) into x and y components. This generates two interdependent equations in 1-D. The PDEs for each component are

$$\frac{\P u_x}{\P t} = \frac{\P}{\P z}\left(k_u \frac{\P u_x}{\P z}\right) + F u_y \tag{4.2}$$

$$\frac{\P u_y}{\P t} = \frac{\P}{\P z}\left(k_u \frac{\P u_y}{\P z}\right) - F u_x: \tag{4.3}$$

4.1.1 Weak forms

To enable substitution of piecewise linear forms suitable for the nite element method, we must obtain the weak forms of the PDEs. The rst step is to multiply the PDEs by a weight function w_i, $\P t$

$$w_i \frac{\P u_x}{\P t} = w_i \frac{\P}{\P z}\left(k_u \frac{\P u_x}{\P z}\right) + w_i F u_y \tag{4.4}$$

$$w_i \frac{\P u_y}{\P t} = w_i \frac{\P}{\P z}\left(k_u \frac{\P u_y}{\P z}\right) - w_i F u_x \tag{4.5}$$

and integrate from $-h$ to 0. Then integration by parts gives the weak forms. These are, for each component respectively,

$$\int_{-h}^{0} w_i \frac{\P u_x}{\P t} dz = \int_{-h}^{0} w_i \frac{\P}{\P}$$

and

$$\sum_{j=0}^{N+1} \frac{\P U_{y_j}}{\P t} \int_{-h}^{0} W_i W_j \, dz = k_u W_i \frac{\P u_y}{\P z}\Big|_{-h}^{0} - \sum_{j=0}^{N+1} U_{y_j} \int_{-h}^{0} k_u \frac{\P W_i}{\P z}\frac{\P W_j}{\P z} dz - \sum_{j=0}^{N+1} U_{x_j} \int_{-h}^{0} F W_i W_j dz. \tag{4.15}$$

The Dirichlet boundary condition $u(z = -h; t) = 0$, is strongly imposed; therefore we will not need to calculate (4.14) and (4.15) for the zeroth node at $z = -h$. However, at $z = 0$ we will need to incorporate the boundary condition $\frac{\P u}{\P z} = b$. For the weight functions w_i, a convenient choice is the collection of piecewise linear hat functions from Chapter 3. These functions are also used as the basis functions for the piecewise linear approximations (4.8) and (4.9), so that in this case basis functions and weight functions are the same. The weak forms are now

$$\sum_{j=1}^{N+1} \frac{\P U_{x_j}}{\P t} \int_{-h}^{0} W_i W_j \, dz = k_u W_i b_x j_0 - \sum_{j=1}^{N+1} U_{x_j} \int_{-h}^{0} k_u \frac{\P W_i}{\P z}\frac{\P W_j}{\P z} dz + \sum_{j=1}^{N+1} U_{y_j} \int_{-h}^{0} F W_i W_j dz \tag{4.16}$$

and

$$\sum_{j=1}^{N+1} \frac{\P U_{y_j}}{\P t} \int_{-h}^{0} W_i W_j \, dz = -\sum_{j=1}^{N+1} U_{y_j} \int_{-h}^{0} k_u \frac{\P W_i}{\P z}\frac{\P W_j}{\P z} dz - \sum_{j=1}^{N+1} U_{x_j} \int_{-h}^{0} W_i F W_j dz. \tag{4.17}$$

This pair of equations can be written for any choice of i, so that $N + 1$ pairs of equations must be considered (recall that the zeroth node due to the Dirichlet condition need not be considered). The equations hold for all internal nodes, with the boundary term $k_u W_i b_x j_0$ being relevant to the $j = N + 1$ case only. This set of equations lends itself therefore to be written in matrix form. We de ne matrices $M = f M_{ij} g$, $K = f K_{ij} g$ and $F = f F_{ij} g$ with the following entries

$$M_{ij} = \int_{z_{i-1}}^{z_{i+1}} W_i(z) W_j(z) dz \tag{4.18}$$

$$K_{ij} = \int_{z_{i-1}}^{z_{i+1}} \frac{\P W_i}{\P z}\frac{\P W_j}{\P z} \, dz \tag{4.19}$$

$$F_{ij} = \int_{z_{i-1}}^{z_{i+1}} F W_i(z) W_j(z) dz. \tag{4.20}$$

We may now write (4.16) and (4.17) in matrix form. For $i = [1:::N]$, this will be:

$$M\underline{U}_x + K\underline{U}_x = F\underline{U}_y \tag{4.21}$$

$$M\underline{U}_y + K\underline{U}_y = -F\underline{U}_x \tag{4.22}$$

with $\underline{U}_x$ denoting a vector of entries $\{U_{x_i}\}$ and so on. The boundary term is evaluated separately and added on to the +f 148r0

4.1 An Illustration of the Equidistribution Method: a vertical velocity pro le

where

$$s(z) = \int \left(1 + g\left(\frac{\partial u}{\partial z}\right)^2\right)^{1/2} dz. \tag{4.25}$$

Here g is a scaling factor appropriate to the scaling of the system being studied. For this example, $g = 1000$ is used since the lateral velocity variations are three or four orders of magnitude smaller than the vertical scale. For our system this function can be expressed as

$$s(z) = \int_{-h}^{z} \left(1 + g\left(\frac{\partial u_x}{\partial z}\right)^2 + g\left(\frac{\partial u_y}{\partial z}\right)^2\right)^{\frac{1}{2}} dz \tag{4.26}$$

where u_x and u_y are the horizontal components of the velocity. New grid points z_i are selected such that

$$h(z) = \frac{\int_{-h}^{z} M(z)dz}{\int_{-h}^{0} M(z)dz} \tag{4.27}$$

for a set of regularly spaced grid points $0 \le h_i \le 1$. By differentiating (4.27) with respect to h twice, we arrive at the differential equation

$$\frac{\partial}{\partial h}\left(M(z)\frac{\partial z}{\partial h}\right) = 0. \tag{4.28}$$

This is a nonlinear PDE, so we will solve it iteratively and therefore write

$$\frac{\partial}{\partial h}\left(M(\underline{z}^p)\frac{\partial \underline{z}^{p+1}}{\partial h}\right) = 0 \qquad (p = 0, 1, \ldots) \tag{4.29}$$

with an initial guess for a vector of discrete points $\underline{z}^0$. As we have only discrete values of $\underline{z}$ and therefore $M(\underline{z})$ to work with, we aim for approximate equidistribution and approximate (4.29) as

$$M(z^p_{j+1/2})(z^{p+1}_{j+1} - z^{p+1}_j) - M(z^p_{j-1/2})(z^{p+1}_j - z^{p+1}_{j-1}) = 0 \tag{4.30}$$

with our discretised monitor function M as

$$M(z_{j+1/2}) = \left(1 + g\left(\frac{U_{x_{j+1}} - U_{x_j}}{z_{j+1} - z_j}\right)^2 + g\left(\frac{U_{y_{j+1}} - U_{y_j}}{z_{j+1} - z_j}\right)^2\right)^{1/2}. \tag{4.31}$$

We assemble (4.30) into a matrix system,

$$T(\underline{z}^p)\underline{z}^{p+1} = \underline{b} \tag{4.32}$$

3. Steps 1 and 2 are repeated for 10 iterations;

4. Using equation (4.32), calculate $\underline{z}^{p+1}$ from $\underline{z}^{p}$ and repeat until satisfactory convergence ($|\underline{z}^{p+1} - \underline{z}^{p}| < 10^{-5}$) is achieved between the two. This is then $\underline{z}(t + dt)$;

5. Interpolate the solutions $\underline{U}_x(t + dt)$ and $\underline{U}_y(t + dt)$ onto the new grid $\underline{z}(t)$.

Figure 4.2 shows a comparison of the xed grid and an adapted grid at t=100s. It is easy to see the clear improvements to the model that come with increased resolution. The xed grid with 160 elements is the highest resolution that the current MATLAB implementation can reasonably compute. If we take this as our reference solution, we see that the moving mesh equidistribution models are more accurate than the xed mesh models with the same number of elements, without a corresponding leap in the computational cost. The improvement is primarily in the gradient of the top portion of the line. Figure 4.3 shows where the grid adaptation has taken place. We see most adaptation around node 8, where the accelerations of the uid integrated over time have been the greatest. Node 8 is at a depth of around 7m. However, there is a physically important transition from a shearing of the uid to a stationary uid (Ekman layer) at around 10-20m depth at time=100 and this is poorly resolved. We investigate an alternative monitor function in search of a method that can better resolve this transition.

Equidistribution by curvature

As the region we wish to better resolve is a region of high curvature, we shall attempt

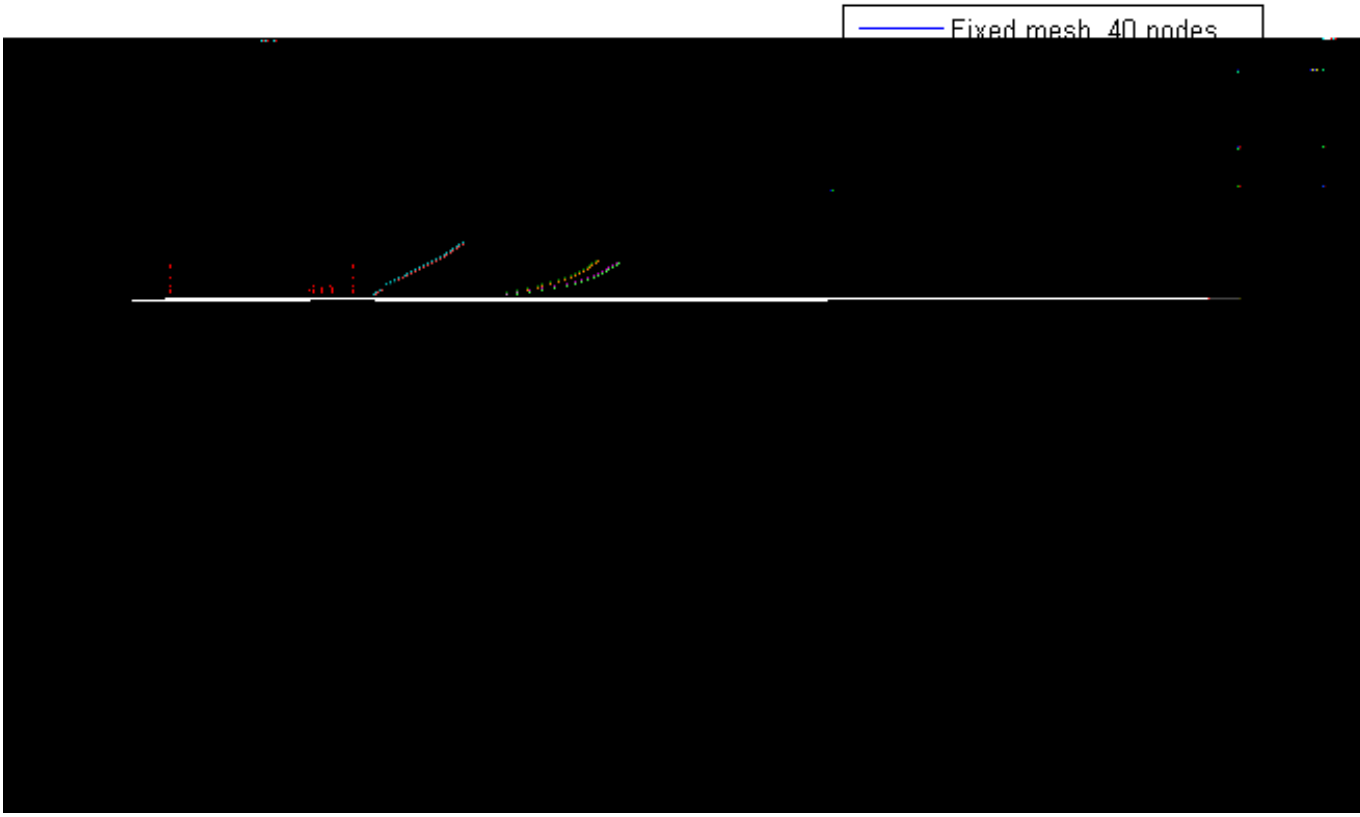


Fig. 4.2 A comparison of the solution given by xed and equidistributed grids for 40, 100 and 160 elements at t=100. See gure 4.1 for a detailed explanation of how this chart represents a velocity pro le. The reference solution (red) is computed on a xed grid with 160 elements. Assuming this higher resolution computation to be the most accurate, we compare the green and purple solutions, computed on 100 elements. We see that we can

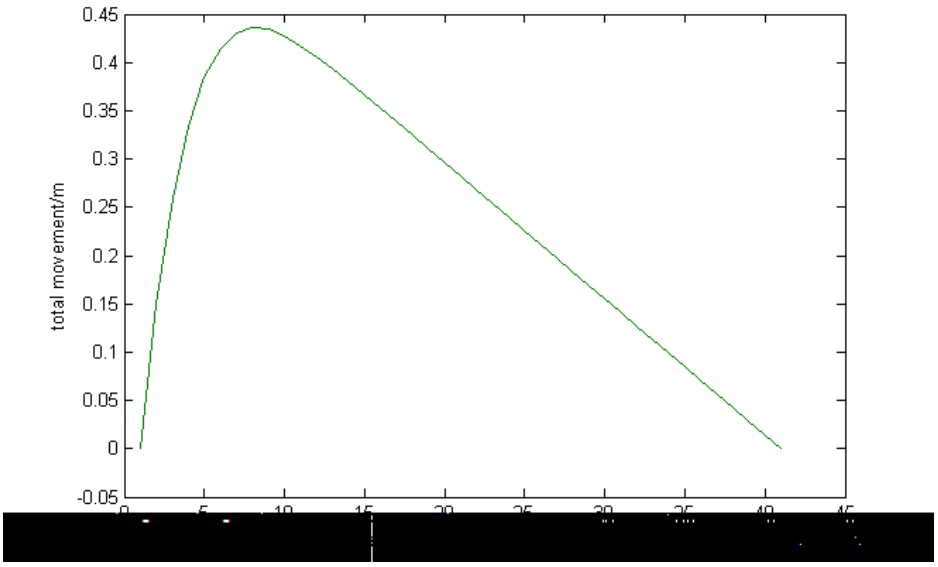


Fig. 4.3 Overall mesh movement for 40 nodes. Total movement for node i is given by $z_i(t) = z_i(0)$.

on the size of the interval in z for accuracy. A sensible solution to this would be to integrate C(z) along the curve;

- Using the integral of curvature as a monitor function we find, at the iterative stage (4.29), many cases that do not converge;
- We achieve sufficient stability to run the model in limited cases ($t < 30$, 1000). However, the node movement cannot take place in advance of the feature of interest forming, so the transition zone is not better resolved. Instead, the increased resolution is observed where the transition zone had previously been located.

From these examples we can see that the equidistribution method has improved the resolution of the Ekman layer for the arc length monitor function, but does not compare well with a grii9uy(v)15(er(ut)-289Teob)-3y(v)15(hig (has)-20(do)swe.7[ly)-1yccbybehir fun5(hiper7 To

based method, interesting features in the ow should be tracked as they develop, and therefore be tracked before they are observed in the features of the solution. The grid will not need remeshing periodically so we will not need to interpolate the solution at any point, instead the solution is tied to the grid at all times by the moving basis functions. Furthermore, the need for an iterative step is eliminated. We will now illustrate the conservation method using Fisher's equation.

4.2 An Illustration of the Conservation Method: Fisher's Equation

Fisher's equation is a reaction diffusion system that describes a balance between linear diffusion and nonlinear reaction. It arises in ecology where it is known as a population growth model, but it can also be used to describe biological invasion, or a simple combustion model for flame propagation, amongst others. In contrast to the alternative population models described later, it involves only a reactant, i.e. any substrate is not relevant. The Fisher's equation is known for exhibiting blow-up, which makes it a particularly interesting target for an adaptive mesh method. We consider here an illustration of Fisher's equation using the conservation method. The aim will be to derive a moving mesh that increases resolution around the blow-up.

4.2.1 Fisher's Equation in 1D

Fisher's equation has a variety of common forms but following the Beth al. paper [14], we look at the particular form describing the temperature u of a reacting or combusting medium. The Masters theses by Edgington, 2011 [24], and Cole, 2009 [21], both examine this same version of Fisher's equation on moving meshes from a finite difference perspective, but here we look at a finite element perspective. As discussed in Chapter 3, for ease of comparison between studies, we will refer to u as mass rather than temperature. Fisher's equation is

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^p$$

conditions are $u(a(t);t) = u(b(t);t) = 0$ where the points $a(t)$ and $b(t)$ may have a non zero velocity. We call this case 2. This allows all nodes including boundary nodes to respond to the mass dynamics. This is a useful alternative system to model because it will allow us to develop the approach that can later be used for a two phase system with a moving interface.

Conservation of relative mass

The approach to moving the nodes is now driven entirely by a conservation of mass in each patch of elements; we have no specialised monitor function with this approach. As the domain moves, the elements must shrink or grow to keep the proportions of mass constant in each. However, for this particular problem we do not have the advantage of a conservative total mass. Instead, as in the generic example in Chapter 3, we will introduce the concept of a relative total mass. This will be de ned as the proportion of mass in each patch of elements. These proportions will remain constant with respect to time. This principle is set out as follows for the Fisher's equation. De ne $q(t)$ to be the area (mass) under the entire solution curve at time t,

$$q(t) = \int_{a(t)}^{b(t)} u(x;t)\, dx. \tag{4.41}$$

We may use (4.41) to calculate

Using Leibnitz' integral rule we have

$$-\frac{q}{q}\int_{a(t)}^{b(t)} w_i u\,dx + \int_{a(t)}^{b(t)} \frac{\partial}{\partial t}(w_i u)\,dx + (w_i u)j_b \frac{db}{dt} - (w_i u)j_a \frac{da}{dt} = 0 \qquad (4.45)$$

or

$$-\frac{q}{q}\int_{a(t)}^{b(t)} w_i u\,dx + \int_{a(t)}^{b(t)} \frac{\partial}{\partial t}(w_i u)\,dx + \int_{a(t)}^{b(t)} \frac{\partial}{\partial x}(\chi w_i u)\,dx = 0 \qquad (4.46)$$

where χ is any velocity consistent with db/dt and da/dt, and then

$$-\frac{q}{q}\int_{a(t)}^{b(t)} w_i u\,dx + \int_{a(t)}^{b(t)} w_i \frac{\partial u}{\partial t} + u\frac{\partial w_i}{\partial t} + w_i \frac{\partial}{\partial x}(u\chi) + u\chi\frac{\partial w_i}{\partial x}\,dx = 0. \qquad (4.47)$$

After substitution of (4.43)

$$-c_i q + \int_{a(t)}^{b(t)} w_i \frac{\partial u}{\partial t} + u\frac{\partial w_i}{\partial t} + w_i \frac{\partial}{\partial x}(u\chi) + u\chi\frac{\partial w_i}{\partial x}\,dx = 0. \qquad (4.48)$$

We fix our weight functions w_i to the domain that moves with velocity χ. Therefore we can argue, by analogy to a convecting system, that

$$\frac{\partial w_i}{\partial t} + \chi\frac{\partial w_i}{\partial x} = 0. \qquad (4.49)$$

We can multiply (4.49) by u and take out this term from equation (4.48). Rearrangement yields

$$\int_{a(t)}^{b(t)} w_i \frac{\partial}{\partial x}(\chi u)\,dx = -\int_{a(t)}^{b(t)} w_i \frac{\partial u}{\partial t}\,dx + c_i q. \qquad (4.50)$$

Substituting from the weak form of Fisher's equation (4.40),

$$\int_{a(t)}^{b(t)} w_i \frac{\partial}{\partial x}(\chi u)\,dx = -\int_{a(t)}^{b(t)} w_i \left(\frac{\partial^2 u}{\partial x^2} + u^2\right) dx + c_i q. \qquad (4.51)$$

Integrating the first term on the left hand side by parts (assuming w_i is sufficiently smooth),

$$[w_i \chi u]_{a(t)}^{b(t)} - \int_{a(t)}^{b(t)} \frac{\partial w_i}{\partial x}\chi u\,dx = -\int_{a(t)}^{b(t)} w_i \left(\frac{\partial^2 u}{\partial x^2} + u^2\right) dx + c_i q. \qquad (4.52)$$

For both case 1 and case 2, we note that the boundary term on the left hand side of (4.52) is zero due to the Dirichlet conditions $u(a(t),t) = u(b(t),t) = 0$. We no

right hand side

$$\int_{a(t)}^{b(t)} \frac{\partial w_i}{\partial x} u\,dx = \left[w_i \frac{\partial u}{\partial x} \right]_{a(t)}^{b(t)} - \int_{a(t)}^{b(t)} \frac{\partial w_i}{\partial x} \frac{\partial u}{\partial x}\,dx + \int_{a(t)}^{b(t)} w_i u^2\,dx - c_i q. \tag{4.53}$$

We refer to the w_i as weight functions. Equation (4.53) is now in a suitable form for finite element functions to be substituted.

Finite elements

We choose the set of functions W_i for our weight functions. Consider the boundary term $\left[w_i \frac{\partial u}{\partial x} \right]_{a(t)}^{b(t)}$ in (4.53). In a finite element framework with Dirichlet conditions, the usual approach is to solve (4.53) for internal nodes only, and in those cases the boundary term would be equal to zero. Therefore the boundary term disappears. The given solution on the boundary can then be strongly imposed. However, in a conservation based system, ignoring boundary terms would destroy conservation in general. In this circumstance, following [33] we switch to a modified set of weight functions, which we will call $\tilde{W}_i$. These weight functions include a combined weight function for the boundary node and its nearest neighbour. This will allow us to strongly impose the Dirichlet conditions without destroying mass conservation. Our approach from here varies depending on the presence or otherwise of a free boundary.

Case 1: Fixed boundaries: Boundary conditions are $u = 0, x = 0$

For the static boundary, case 1, we work in modified weight functions throughout. The modified weight functions $\tilde{W}_i$ are constructed from the original weight functions W_i as follows,

$$\tilde{W}_1(t;x) = W_0(t;x) + W_1(t;x) \tag{4.54}$$

and

$$\tilde{W}_N(t;x) = W_N(t;x) + W_{N+1}(t;x) \tag{4.55}$$

with the remaining W_i unaltered. These modified weight functions are illustrated in figure 4.4. Note the dimension of the subspace in which these functions reside is reduced from $N + 2$ to N. The c_i values of equation (4.43) must be adjusted accordingly,

$$\tilde{c}_1 = c_0 + c_1 = \int_a^b \frac{1}{q}(W_0 + W_1)u\,dx \tag{4.56}$$

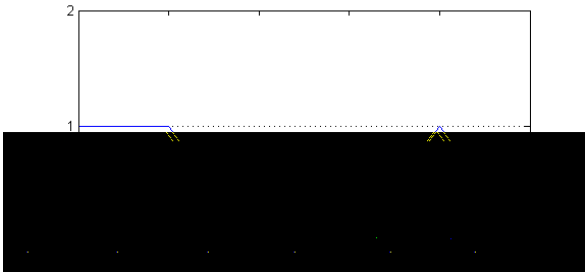


Fig. 4.4 Modi ed weight functions in 1-D for boundary node x_0 and internal node x_i. These W_i form a partition of unity and are compatible with strongly imposed Dirichlet conditions.

and

$$\tilde{c_N} = c_N + c_{N+1} = \int_a^b \frac{1}{q}(W_N + W_{N+1})u\, dx \tag{4.57}$$

We then have no c_0 or c_{N+1} values. The remaining c_i are unaltered. The use of $\tilde{W_i}$ and $\tilde{c_i}$ ensure that global conservation is not violated in (4.43

whereX

$\bar{B}(\underline{U})$ could be singular if U were constant and it had an odd number of rows and columns. However, the second caveat is that when we consider the system in two dimensions, the velocity is not unique because we could add an arbitrary curl vector to X (see equation (3.5)). By introducing a velocity potential F, we can avoid this problem since the velocity potential is unique. We then specify a curl of zero when we recover X from F. In order to keep the method consistent between one and two dimensions then, we will also work with a velocity potential in one dimension. We proceed therefore by introducing the velocity potential F, de ned by

$$X = \frac{\P F}{\P x} \tag{4.67}$$

where

$$F(x;t) = \sum_{j=0}^{N+1} W_j(x;t) F_j(t) \tag{4.68}$$

so that

$$\frac{\P F}{\P x} = \sum_{j=0}^{N+1} \frac{\P W_j}{\P x} F_j: \tag{4.69}$$

Substituting this into equation (4.61), equation (4.62) becomes

$$\tilde{K}(\underline{U})\underline{F} = \tilde{\underline{g}} \tag{4.70}$$

where $\underline{F}$ is the vector containing $fF_j g$(¶m298.762 410.028 .ieA87 401 18Td [(4.yion)npning)]TJ/ |

uation (4.61

Getting X

We then obtain X from a finite element approximation of (4.67) at each node. From the definition (4.67) we write the weak form,

$$\int_a^b w_i X \, dx = \int_a^b w_i \frac{\partial F}{\partial x} dx. \tag{4.74}$$

For case 1, the fixed boundaries are equivalent to imposing the boundary conditions $x_{j_a} = 0$ and $x_{j_b} = 0$. To impose these without violating relative mass conservation in (4.43), modified weight functions are again required. We select the modified weight functions $w_i = \tilde{W}_i$ of (4.54), (4.55) and use the piecewise linear approximations (4.58) and (4.68). We obtain

$$\sum_{j=0}^{N+1} \int_a^b \tilde{W}_i W_j \, dx \; X_j = \sum_{j=0}^{N+1} \int_a^b \tilde{W}_i \frac{\partial W_j}{\partial x} dx \; F_j. \tag{4.75}$$

In matrix form this is

$$\tilde{M}\underline{X} = \tilde{B}\underline{f}. \tag{4.76}$$

The matrix $\tilde{M}$ is a positive definite and well-conditioned mass matrix with entries

$$\tilde{M}_{ij} = \int_a^b \tilde{W}_i W_j dx \tag{4.77}$$

and $\tilde{B}$ is an asymmetric matrix similar to (4.65), with entries

$$\tilde{B}_{ij} = \int_a^b \tilde{W}_i \frac{\partial W_j}{\partial x} dx. \tag{4.78}$$

Note that we will only need to invert $\tilde{M}$ in order to recover $\underline{X}$.

Finding X

A time integration approximation such as forward Euler is used to generate the grid at the next time step from the values of X.

RecoveringU

To generate the new solution for at the new time step we return to our relative conservation principle (4.43), with our modified basis functions $\tilde{W}_i = \tilde{W}_i$,

$$\frac{1}{q(t)} \int_a^b \tilde{W}_i u \, dx = \tilde{c}_i .$$

In discretised form this becomes, with $u(x,t) = \sum_j W_j(x,t) U_j(t)$,

$$\sum_{j=0}^{N+1} \int_a^b \tilde{W}_i W_j \, dx \; U_j = \tilde{c}_i q(t) \tag{4.79}$$

where the $\tilde{c}_i$ are given by the modified values

$$\tilde{c}_i = \frac{1}{q(t_0)} \int_a^b \tilde{W}_i U_0 \, dx = \frac{1}{q(t_0)} \int_a^b \tilde{W}_i u_0 \, dx \tag{4.80}$$

for initial data u_0, if the U_0 is the L^2 best fit to u_0. Then (4.79) is equivalent to the mass matrix system

$$\tilde{M} \underline{U} = q(t) \underline{\tilde{c}} \tag{4.81}$$

with $\underline{\tilde{c}}$ as the vector containing entries $\tilde{c}_i$, and $\tilde{M}$ the mass matrix calculated for the new nodal positions. We may then solve for U with the boundary condition $\tilde{W}_i U \Big|_a^b = 0$ strongly imposed on U, without violating relative mass conservation.

Algorithm 7

For case 1 with fixed boundaries.
Having initial u_0 and x_0, and having calculated the piecewise linear function U_0 at the nodes X_0, as well as q from (4.41), the finite element solution of Fisher's equation (4.39) on the moving mesh in 1-D consists of the following steps at each time

1. Find

4. Generate the co-ordinate system $\underline{X}(t + dt)$ at the next time-step by evaluating (3.18) using the forward Euler approximation. Similarly, update q from $q(t)$;

5. Find the solution $U(t + dt)$ by solving the relative conservation equation (4.81) using the strong form of the boundary conditions.

Case 2: Moving boundaries: Boundary conditions are $u = 0, u_x = 0$

For the free boundary, case 2, whilst we have Dirichlet conditions for u we do not have them for x. In fact there are no boundary conditions to impose on x. We will not need modified weight functions to obtain x and indeed, using them would prevent us from obtaining a solution for x at the boundaries $x = a$ and $x = b$. For this reas.82 Tf2(we)-291(t)-3a45 0 Td [(0)]TJ/F

and B is the asymmetric matrix of (4.78):

$$B_{ij} = \int^{Z_{b(t)}}$$

using the modi ed conservation principle.

Recovering U

U can now be recovered in exactly the same way as in case 1, as described in section 4.2.1.

Algorithm 8

For case 2 with free boundaries.
Having initial u_0 and x_0, and having calculated the piecewise linear function U_0 at the nodes X_0, as well as q from (4.41), the nite element solution of Fisher's equation (4.39) on the moving mesh in 1-D consists of the following steps at each time:

1. Find $q(t)$ by evaluating (4.66);
2. Find the velocity potential by solving equation (4.88) for the $F_j(t)$ values, with F speci ed at the central node;
3. Find the node velocity by calculating (4.91) for the $\dot{X}$ values at all nodes including boundary nodes;
4. Generate the co-ordinates $X(t + dt)$ at the next time-step from (3.18) using the forward Euler approximation. Similarly, update q from $q(t)$;
5. Find the solution $U(t + dt)$ by solving the relative conservation equation (4.81) using the strong form of the boundary conditions.

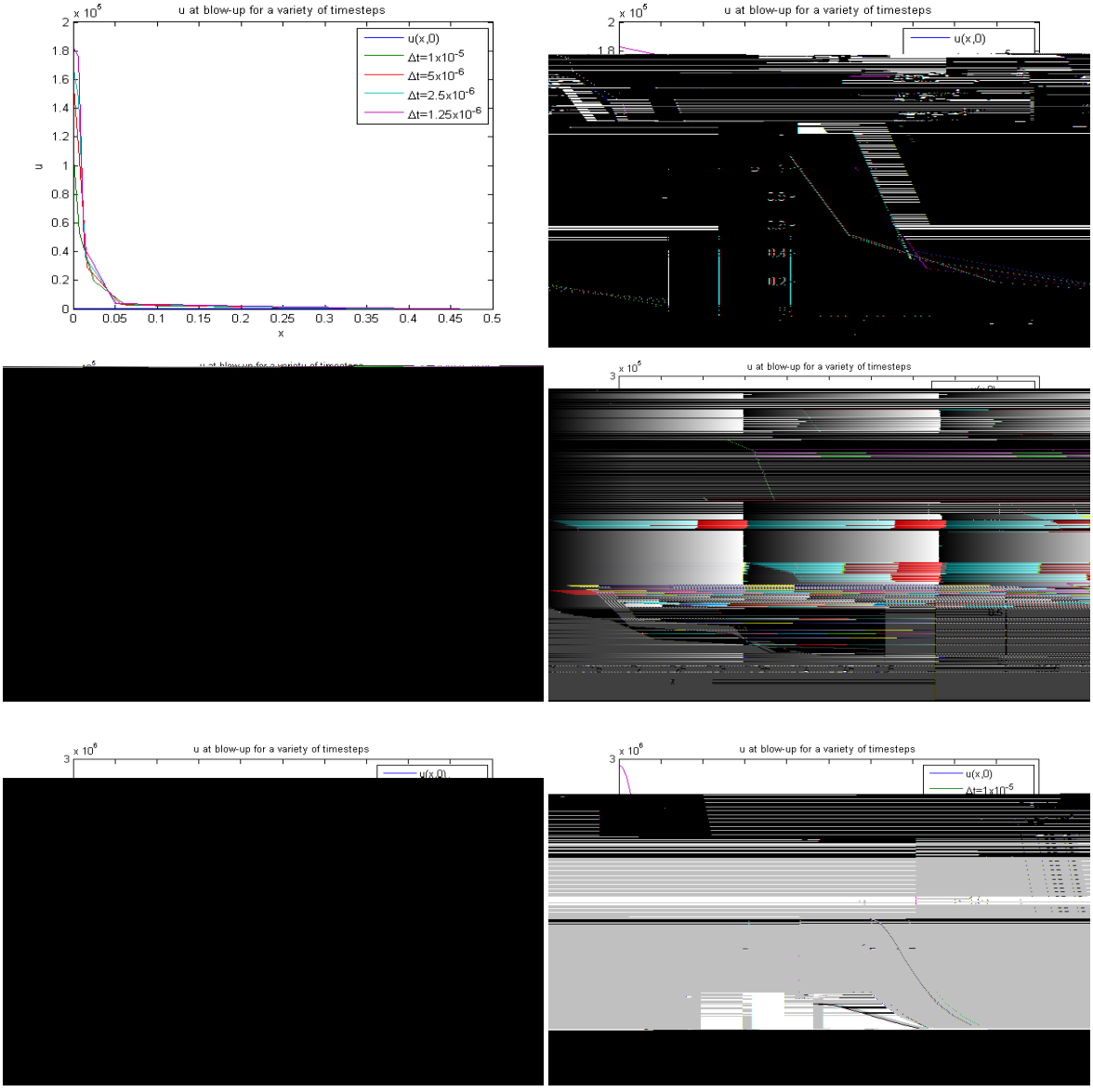


Fig. 4.6 Blow-up of solution $u(x,t)$ of Fisher's equation (4.39), with case 1 boundary conditions (xed boundaries). The model is run to t=0.0825, beyond which solutions begin to suffer from node crossing and other instabilities. The precise time that this occurs for each choice of Δt and Δx is given in table 4.1. The grid resolution is 6 nodes (top), 11 nodes (centre) and 21 nodes (bottom) in the half domain shown. Initial spacing Δx is regular. Various Δt choices are tested for each initial grid resolution. The gures on the right show the detail at x close to 0. As Δx is reduced a small improvement in the de nition of the peak is noted.

tion, that

$$\Delta t < \Delta x^2 = \left(\frac{0.5}{N}\right)^2 \tag{4.95}$$

which limits Δt to a maximum of 1.25×10^{-3}. In both [21], and [24], Cole and Edgington attempt a moving mesh solution of the same problem. They use an implicit finite difference method to compute a conservation-based approach to moving the mesh. The results from our finite element method are consistent with the approximate blow up time $T = 0.082372$ given in [13], as table 4.1 shows. We define the blow-up time of the model as the point of failure of the model to further resolve a solution, i.e. nodes are crossing or some other catastrophic instability. Looking in more detail, we are able to resolve a higher peak for at blow-up with values of the order of $N = 10^5$ (figure 4.6) rather than the $N = 10^4$ in [21]. We also observe a narrower, more defined peak at all values of Δx and Δt than in the Cole dissertation. Furthermore, we note from the results in [24] that the 11 node model performs better than the 6 node or 21 node models (defined as the number of nodes in the half-domain $0 \leq x \leq 0.5$). Presumably the 6 node model is limited by lack of resolution and the 21 node model is limited by node tangling. We do not see the same node tangling limit

Δx, when the variable time steps are used.

Results for the moving boundary problem: Case 2

Edgington extends the work in [21] by examining the effect of allowing the boundary nodes to move. He finds that in the finite difference model, the maximum u achieved is reduced when the boundaries are allowed to move, except in the coarsest 11 node model.

Fig. 4.7 A solution of the 1D Fisher's equation using a moving mesh with the free boundary of case 2. Here we use 20 elements and a time step of 0.00005.

We find that our results are somewhat mixed. The maximum u achieved is equalled or improved when the boundaries are allowed to move, when compared to the fixed boundary case. However it must be recognised that the problem is defined differently for each case. For the 21 node model, the maximum resolvable u_{max} is broadly unchanged at (near) blowup when the moving (case 2) and fixed boundary (case 1) versions are compared. For the 11 node model, allowing the moving boundary increases the resolvable u_{max} at blow-up by about a factor of 5. For the 6 node model, allowing the moving boundary increases the resolvable u_{max} at blow-up by about a factor of 10. However, the time taken to blow up is much less accurate with a moving boundary than with a static boundary. The model stops

running due to nodes crossing at around $T = 0.065$ in the moving boundary case, whereas we would aim for the model to run to $T = 0.0823$. It is probable that this is due to the greater nodal velocities observed when a free boundary is present.

Table 4.1 Blow-up times from MMFEM implementation of Fisher's equation, case 1 with xed time step

N	Dt	steps	$T_{blow-up}$
6	$1x10^{-5}$	8411	0.08411
6	$5x10^{-6}$	16815	0.08408
6	$2.5x10^{-6}$	33622	0.08406
6	$1.25x10^{-6}$		

Table 4.3 Blow-up times from MMFEM implementation of Fisher's equation, case 1 with smoothing, and fixed time steps. Blow-up happens later with smoothing.

N	Dt	steps	$T_{blow-up}$
6	$1x10^{-5}$	9821	0.0982
6	$5x10^{-6}$	19653	0.0983
6	$2.5x10^{-6}$	39317	0.0983
6	$1.25x10^{-6}$	78645	0.0983
11	$1x10^{-5}$	9892	0.0989
11	$5x10^{-6}$	19865	0.0993
11	$2.5x10^{-6}$	39817	0.0995
11	$1.25x10^{-6}$	79727	0.0997
21	$1x10^{-5}$	9691	0.0969
21	$5x10^{-6}$	19664	0.0983
21	$2.5x10^{-6}$	39663	0.0992
21	$1.25x10^{-6}$	79669	0.0996

Smoothing

We also note some saw-tooth instability in both the constant time step case and, to a lesser extent, the variable time step case finite element models. This is a common problem with finite element methods because of the central differences involved combined with explicit time stepping. We will attempt to smooth this out by introducing a viscosity term (Laplacian smoothing),

$$x_i^{new} = x_i + \frac{1}{4}d^2x_i; \qquad d^2 = x_{i+1} - 2x_i + x_{i-1}. \tag{4.97}$$

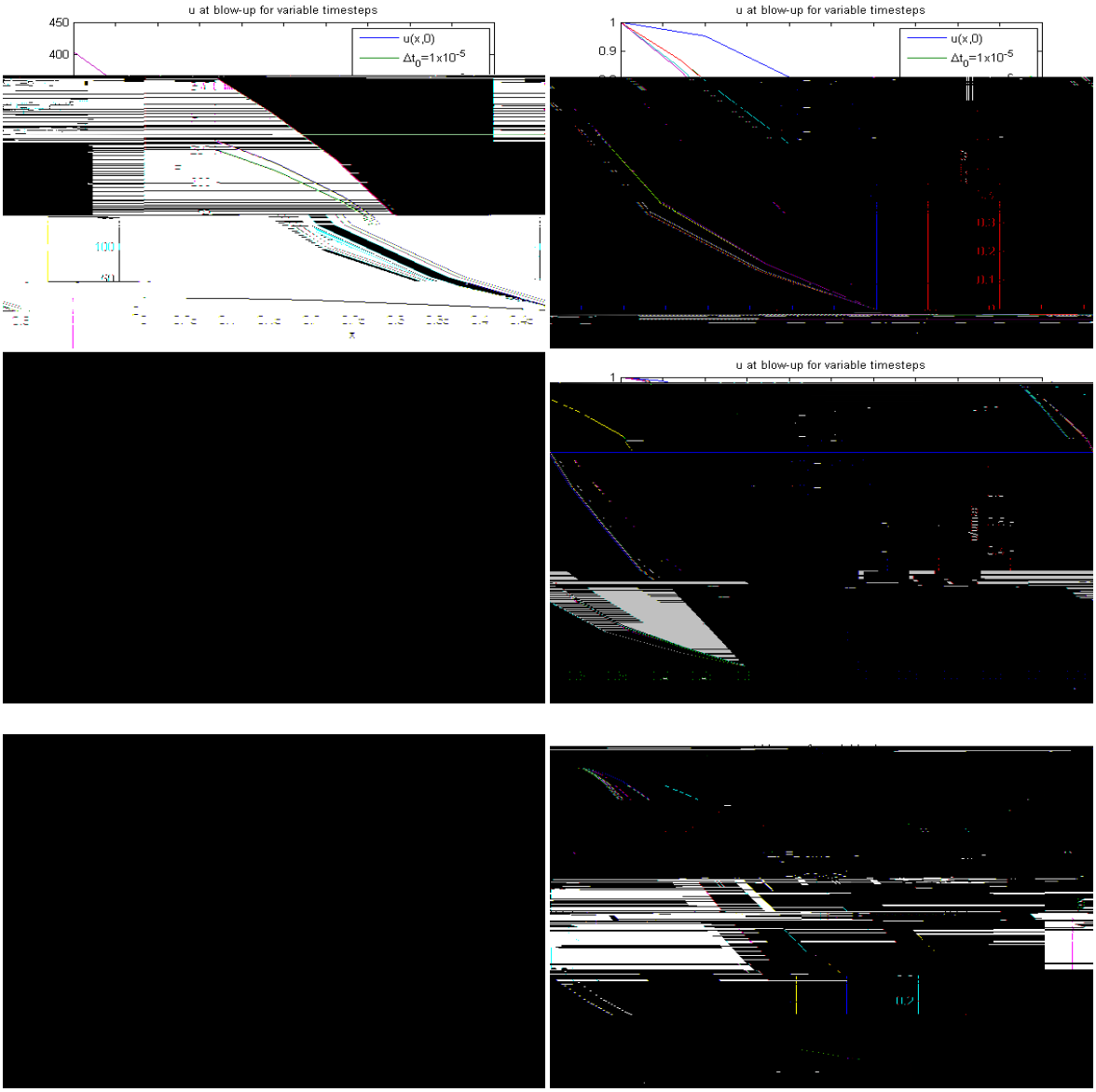


Fig. 4.8 Blow-up of the solution $u(x,t)$ of Fisher's equation (4.39) with fixed boundaries (case 1) and variable time steps. The models are run until t=0.0825. Grid resolutions are from top to bottom, 6 nodes, 11 nodes and 21 nodes. Variable time steps for comparison with [21] are used. The figures on the right show the results with a normalised

Order of convergence

Since we have a value for the blow up time from [13], we may examine the orders of convergence p or q with respect to time or space respectively. When Δt is varied with Δx held constant, we expect a fixed non-zero component of the spatial error, so we may estimate p and q by looking at the rate at which the differences between successive errors decrease. We assume

$$E_n = C$$

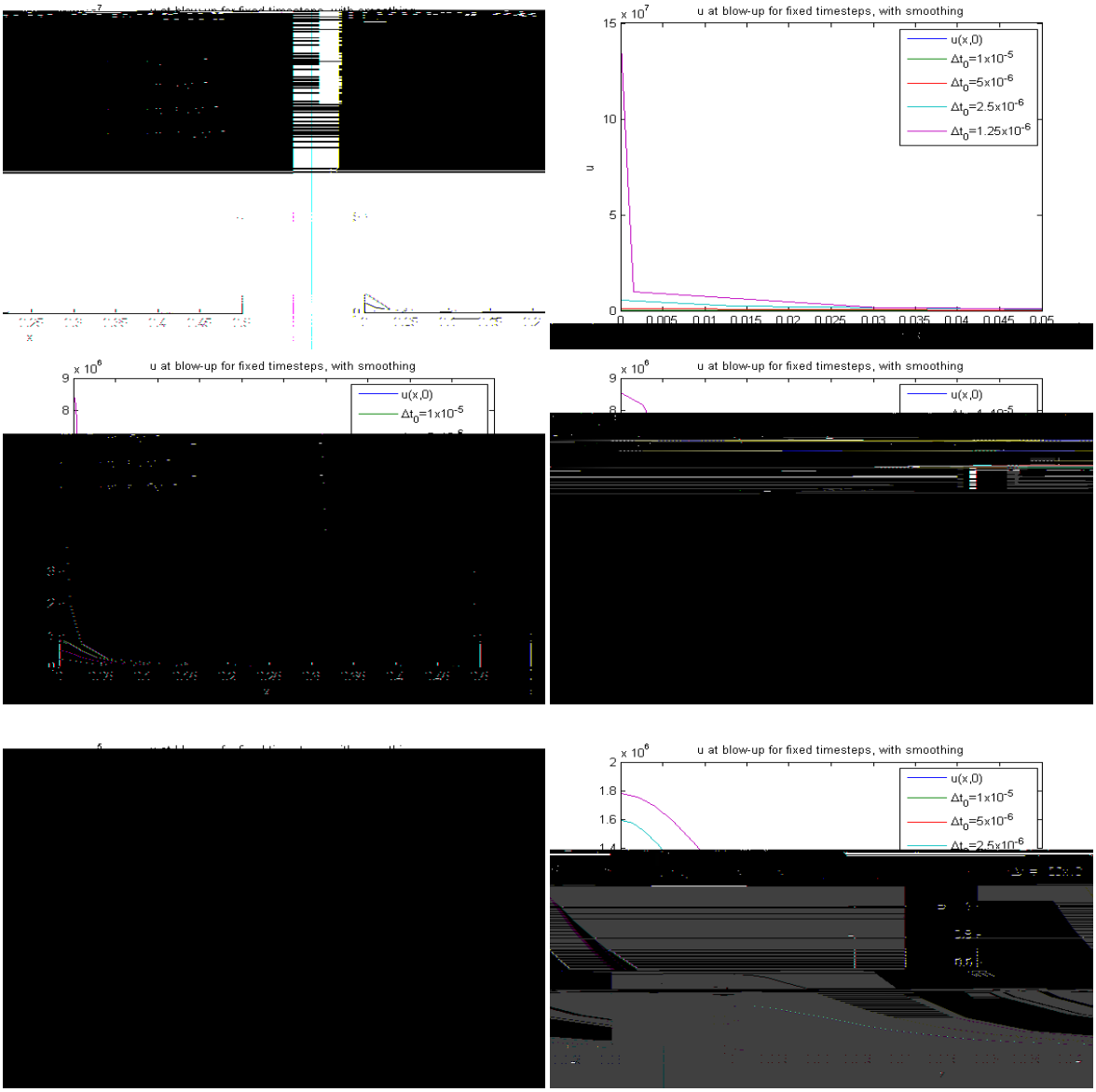


Fig. 4.9 The smoothed, xed time step results at t=0.0825 for blow-up of the solution $u(x,t)$ of Fisher's equation (4.39) with xed boundaries (case 1). Grid resolutions are from top to bottom, 6 nodes, 11 nodes and 21 nodes in the half domain. The gures on the right show the same results as the gures on the left but with a change of scale on the x axis. As Δx is reduced the peak actually gets wider as the smoothing becomes more effective.

Table 4.4 Errors in blow-up time from MMFEM implementation of Fisher's equation, case 1 with fixed time step, and their variation by time step and node spacing

N	Dx	Dt	E_n	$E_n - E_{n-1}$	$\frac{E_n - E_{n-1}}{E_{n+1} - E_n}$
21	0.025	1×10^{-5}	0.000248		
21	0.025	5×10^{-6}	0.000213	-0.000035	1.6
21	0.025	2.5×10^{-6}	0.000191	-0.000022	2.0
21	0.025	1.25×10^{-6}	0.000180	-0.00011	2.2
21	0.025	6.25×10^{-7}	0.000175	-0.00005	
6	0.1	1.25×10^{-6}	0.001674		
11	0.05	1.25×10^{-6}	0.000508	-0.001166	3.6
21	0.025	1.25×10^{-6}	0.000180	-0.000328	4.0
41	0.0125	1.25×10^{-6}	0.000099	-0.000081	4.8
81	0.00625	1.25×10^{-6}	0.000082	-0.000082	

We see in table 4.4 that successive differences between errors as you halve Dt go down by a factor of about 2, suggesting $p \approx 1$ or first order in time. When you halve Dx these differences go down by a factor of about 4, suggesting $q \approx 2$ or second-order accuracy in space. This is as expected from forward Euler in time and linear finite elements in space.

4.2.2 Fisher's Equation in 2D

The two dimensional solution of the Fisher's equation has not previously been attempted

We de ne a reference test domain $W(0)$ at $t = 0$ and a moving test domain $W(t)$. Applying the Reynolds Transport Theorem we obtain

$$\frac{d}{dt}\int_{W(t)} w_i u \, dW = \int_{W(t)} \frac{\partial}{\partial t}(w_i u) \, dW + \int_{W(t)} w_i u x \cdot \hat{n} \, dS$$
$$= \int_{W(t)} w_i \frac{\partial u}{\partial t} + u \frac{\partial w_i}{\partial t} + \nabla \cdot (w_i u x) \quad dW \qquad (4.111)$$

for the generalised weak form, where $\hat{n}$ is any normal velocity consistent with the normal boundary velocity. Using the advection equation (3.7) we can cancel out terms giving us the weak form of the Reynolds Transport Theorem in the moving frame,

$$\frac{d}{dt}\int_{W(t)} w_i u \, dW - \int_{W(t)} w_i \nabla \cdot (ux) \, dW = \int_{W(t)} w_i \frac{\partial u}{\partial t} \, dW. \qquad (4.112)$$

We now consider the speci c system described by Fisher's equation. We substitute the weak form of Fisher's equation (4.106), and obtain

$$\frac{d}{dt}\int_{W(t)} w_i u \, dW - \int_{W(t)} w_i \nabla \cdot (ux) \, dW = \int_{W(t)} (w_i \nabla^2 u + w_i u + \int_{W(t)} d 0 \; J \; 0.478 \; w \; 0 \; 0 \; m \; 13.366 \; 0 \; l \; S \; Q$$

where $\hat{n}$ is the outward pointing unit normal. The boundary $\mathbf{u}\mathbf{x} \cdot \hat{n}$ is zero due to the Dirichlet condition on u. We now have an equation for $\dot{q}$ in terms of u and q,

$$c_i q + \int_{W(t)} \mathbf{u}\mathbf{x} \cdot \tilde{N} w_i \, dW = \int_{S(t)} w_i \tilde{N} u \cdot \hat{n} \, dS - \int_{W(t)} \tilde{N} w_i \cdot \tilde{N} u \, dW + \int_{W(t)} w_i u^2 \, dW. \quad (4.117)$$

Providing that we select weight functions w_i that form a partition of unity $\sum w_i = 1$, we can calculate q by summing this expression over all weight functions in the model and using the boundary conditions. From (4.109), we define c_i as the proportion of mass associated with a particular weight function w_i. The sum is

$$\sum_i c_i q(t) = s(t)$$

giving

$$\tilde{\nabla}F(x;t) = \sum_{j=1}^{N} \tilde{\nabla}W_j(x;t)\,F_j(t). \tag{4.126}$$

We can now write (4.122) in the form

$$\sum_{j=1}^{N} \int_{W(t)} U\,\tilde{\nabla}W_i \cdot \tilde{\nabla}W_j \, dW \;\; F_j = \int_{S(t)} W_i \tilde{\nabla}U \cdot \hat{n} \, dS$$
$$- \sum_{j=1}^{N} \int_{W(t)} \tilde{\nabla}W_i \cdot \tilde{\nabla}W_j \, dW \;\; U_j + \int_{W(t)} W_i U^2 \, dW - c_i q(t) \tag{4.127}$$

or in matrix form

$$K(\underline{U})\underline{F} = \underline{f} \tag{4.128}$$

with the vector $\underline{F}$ containing the values F_i, and the vector $\underline{f}$ containing the values f_i given by

$$f_i = \int_{S(t)} W_i \tilde{\nabla}U \cdot \hat{n} \, dS - \sum_{j=1}^{N} \int_{W(t)} \tilde{\nabla}W_i \cdot \tilde{\nabla}W_j \, dW \;\; U_j + \int_{W(t)} W_i U^2 \, dW - c_i q(t). \tag{4.129}$$

The nonlinear term, $\int_{W(t)} W_i U^2 \, dW$, is evaluated using Gaussian quadrature (see Appendix B). Whilst not exact, the order of accuracy is high enough not to affect the order of accuracy of the complete algorithm.

$K(\underline{U})$ is the weighted stiffness matrix with elements

$$K(\underline{U})_{ij} = \int_{W(t)} U\,\tilde{\nabla}W_i \cdot \tilde{\nabla}W_j \, dW. \tag{4.130}$$

We obtain $\dot{X}$ from the finite element approximation of (4.120), for which the process is described in detail in Chapter 3, section 3.1.3. This gives the matrix form

$$M\dot{\underline{X}} = B\underline{F} \tag{4.131}$$

where $\underline{X} = \{x_i\}$, M is the standard mass matrix and B is an asymmetric matrix with elements

$$B_{ij} = \int_{W(t)} W_i \tilde{\nabla}W_j \, dW. \tag{4.132}$$

Modified weight functions in 2-D

Having obtained X we now rewrite the system in terms of modified weight functions, so that the Dirichlet condition on U can be imposed. Modified weight functions in this context are any suitable set of piecewise linear weight functions where the weighting normally associated with a boundary node has been transferred to an internal node, and where a partition of unity is preserved. We turn our attention firstly to describing our system in terms of modified weight functions, and afterwards will take a closer look at the form of these functions and how they may be used in calculating matrices. We use the tilde to denote the use of the modified weight functions, i.e. $w_i = \tilde{W}_i(x,y)$.

For the approximations to variables, we continue to make piecewise linear approximations in terms of standard (unmodified) basis functions,

$$U(x,t) = \sum_{j=1}^{N} W_j(x,t)U_j(t) \tag{4.133}$$

$$X(x,t) = \sum_{j=1}^{N} W_j(x,t)X_j(t). \tag{4.134}$$

The ALE equation (4.117) can now be written, with a little rearrangement, in terms of modified weight functions $\tilde{W}_i$ and unmodified basis functions W_j as

$$\tilde{c}_i = \frac{1}{q(t)} \left(- \sum_{j=1}^{N} \int_{W(t)} U\tilde{W}_i \cdot \tilde{\nabla}W_j \, dW \; X_j + \int_{S(t)} \tilde{W}_i \tilde{\nabla}U \cdot \hat{n} \, dS + \int_{S(t)} \tilde{W}_i U X \cdot \hat{n} \, dS \right.$$
$$\left. - \sum_{j=1}^{N} \int_{W(t)} \tilde{\nabla}\tilde{W}_i \cdot \tilde{\nabla}W_j \, dW \; U_j + \int_{W(t)} \tilde{W}_i U^2 \, dW \right). \tag{4.135}$$

We impose our X obtained from the unmodified system into this modified system, and thus obtain the correct values of $\tilde{c}_i$ for the modified system. The nonlinear term is calculated using Gaussian quadrature (see Appendix B). After time integration, we recover U using the finite element version of (4.109) with modified weight functions

$$\sum_{j=1;\, j \in S}^{N} \int_{W(t)} \tilde{W}_i W_j \, dW \; U_j = \tilde{c}_i q(t) \tag{4.136}$$

which references internal nodes only. In matrix form this is

$$\tilde{M}\underline{U} = \underline{\tilde{c}}q \tag{4.137}$$

for mass matrix $\tilde{M}$, and vectors $\underline{U}$ containing the U_j values and $\underline{c}$ containing the c_i values.

We now turn our attention to selecting a form for the modified weight functions and consider the implications for matrix construction. Following Hubbard, Baines and Jimack, 2009 [33] we are presented with the choice between two approaches for modifying the weight functions. These are termed the 'averaged modified approach' and the 'compact modified approach'. The two approaches are both derived and discussed only in the context of the mass matrix. The modified mass matrix alone is sufficient to solve the conservation equation (4.137), but here we require a more extensive implementation of the modified weight functions. In order to solve (4.135) we will require an evaluation of both a stiffness matrix and an asymmetric matrix. We must therefore extend one of the approaches from [33] in order to provide a way to construct any matrix from the modified weight functions. The averaged modified approach of [33] lends itself best to this, since it is defined in terms of the weight functions themselves. In [33] the modified weight functions are constructed in a similar way to the 1-D case, but with the added complication of increased connectivity. It is stated that the weight functions associated with boundary nodes are redistributed equally between their adjacent internal nodes. Therefore all basis functions defined on fully internal elements remain unaffected. With regard to the construction of the mass matrix, [33] sets out the following process. For triangles with two nodes on the boundary, all the weight associated with that triangle has only one internal node to go to, and the calculation is simple. For a given internal node j on a triangle with vertices $[j, J, K]$ where J and K are boundary nodes, the modified weight function $\tilde{W}_{j\ [j,J,K]}$ for triangle $[j, J, K]$ is given by

$$\tilde{W}_{j\ [j,J,K]} = W_j + W_J + W_K. \tag{4.138}$$

An example of such a triangle is number 3 of figure 4.10.

For triangles with one node on the boundary, the weight associated with that node is split equally between the two internal nodes. For a given internal node j on a triangle with vertices $[i, j, J]$ where only J is on the boundary, the modified weight function $\tilde{W}_{j\ [i,j,J]}$ for triangle $[i, j, J]$ is given by

$$\tilde{W}_{j\ [i,j,J]} = W_j + \frac{1}{2}W_J. \tag{4.139}$$

An example of such a triangle is number 2 of figure 4.10.

These sums are presented visually in figure 4.11. Recalling the standard 2-D basis functions W_i of figure 3.2, we obtain from equations (4.138) and (4.139) the coloured prisms $\tilde{W}_i$ of figure (4.11). The red volume represents $\tilde{W}_{j\ [j,J,K]}$, the contribution from triangle 3 to the modified basis function at internal node j. All the mass from triangle 3 has been as-

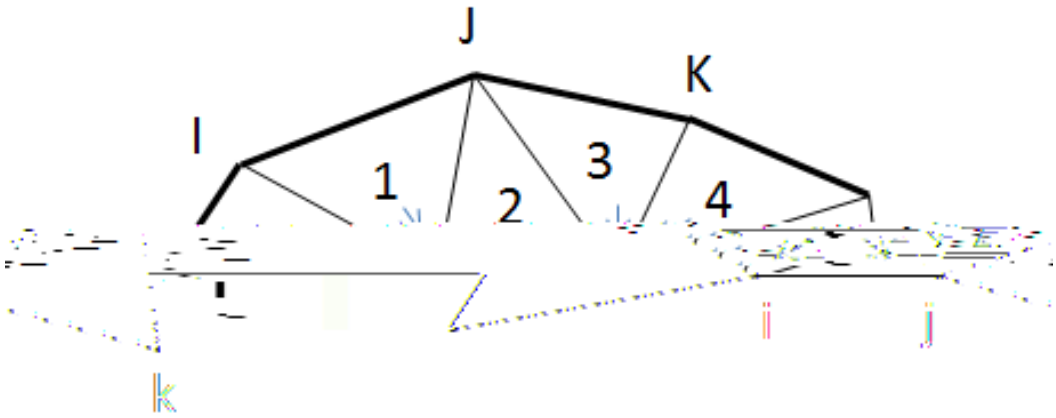


Fig. 4.10 Connectivity between boundary nodes (I,J and K) and internal nodes (i,j, and k). The arrows show where the weight function from each triangle will be transferred to under the modi ed system

signed to node j, so the red modi ed basis function is a triangular prism with height 1 at all three corners. The purple volume represents $\tilde{W}_{j[i;j;J]}$, the contribution from triangle 2 to the modi ed basis function at internal node j. Half of the mass normally assigned to boundary node J is transferred to internal node j, with the remaining half being transferred to internal node i. The purple modi ed basis function is therefore a modi ed prism with height 1 at j, height 0 at i and height 0.5 at J.

The practical implementation of this modi cation process takes place at the level of matrix assembly. The 2-D matrices are assembled as part of the algorithm by summing the element contributions from each triangle. When we require a matrix calculated from modi ed weight functions such as that of (4.137), the contributions from boundary triangles are adjusted before assembly according to (4.138) and (4.139). Contributions from triangles with no boundary nodes are unaffected.

The matrix assembly using these modi ed functions must consider the interactions between modi ed weight functions and unmodi ed basis functions. A generalised matrix A de ned in terms of functions F and G with standard weight functions W_i and basis functions W_j has entries

$$A_{ij} = \int_W F(W_i)G(W_j)\, dW \qquad (4.140)$$

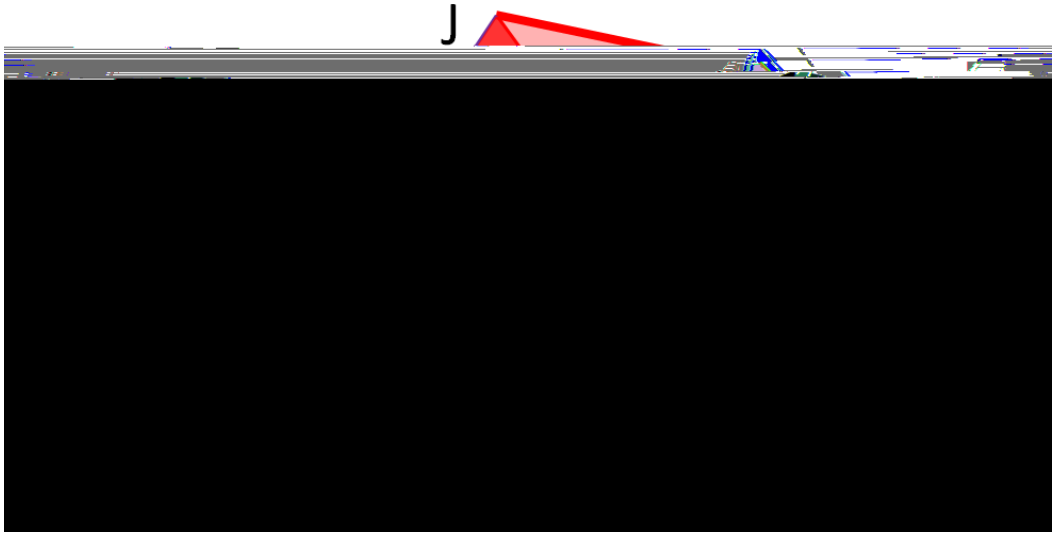


Fig. 4.11 Modi ed basis functions for internal nodes. The red modi ed weight function represents the mass contribution from triangle 3 to internal node j, and is a triangular prism

$$W_{j\ [i;j;J]} \tag{4.144}$$

and

$$W_J j_{[i;j;J]}: \tag{4.145}$$

The modified weight functions $\tilde{W}_i$ for triangle 2 are, in terms of those unmodified basis functions,

$$\tilde{W}_{i\ [i;j;J]} = W_i j_{[i;j;J]} + \frac{1}{2} W_J j_{[i;j;J]} \tag{4.146}$$

$$\tilde{W}_{j\ [i;j;J]} = W_{j\ [i;j;J]} + \frac{1}{2} W_J j_{[i;j;J]} \tag{4.147}$$

and

$$\tilde{W}_{J\ [i;j;J]} = 0: \tag{4.148}$$

The entries for the modified element matrix as defined by (4.142) can be calculated for triangle 2 from the local W_j and $\tilde{W}_i$ functions , (4.143) to (4.148). By reference to (4.140) and (4.141), the entries can be given in terms of the unmodified elements of (4.141) as

$$\tilde{A}_e = \begin{pmatrix} e_{ii} + \frac{1}{2}e_{iJ} & e_{ij} + \frac{1}{2}e_{jJ} & e_{iJ} + \frac{1}{2}e_{JJ} \\ e_{ji} + \frac{1}{2}e_{iJ} & e_{jj} + \frac{1}{2}e_{jJ} & e_{jJ} + \frac{1}{2}e_{JJ} \\ 0 & 0 & 0 \end{pmatrix}: \tag{4.149}$$

The matrix is partitioned into an upper left 2×2 matrix, a bottom row of all zeros, and a right hand column which refers to a known value obtained from the Dirichlet condition. For example to calculate $A\underline{U}$, we can see that we have

$$\left(\begin{array}{cc|c} e_{ii} + \frac{1}{2}e_{iJ} & e_{ij} + \frac{1}{2}e_{jJ} & e_{iJ} + \frac{1}{2}e_{JJ} \\ e_{ji} + \frac{1}{2}e_{iJ} & e_{jj} + \frac{1}{2}e_{jJ} & e_{jJ} + \frac{1}{2}e_{JJ} \\ \hline 0 & 0 & 0 \end{array}\right) \begin{pmatrix} U_i \\ U_j \\ U_J \end{pmatrix} = \begin{pmatrix} U_i \\ U_j \end{pmatrix}: \tag{4.150}$$

where U_i and U_j are free and U_J is fixed. The known terms generated by the right hand column of the matrix can be added directly into the rows of the calculation, allowing us to

use only the square matrix of the upper left in the matrix operation. This has the advantage of being invertible.

We can use this approach to generate the speci c matrices we will use. The unmodi ed mass matrix given by

$$M = \int_W W_i W_j \, dW \tag{4.151}$$

has the element mass matrix

$$M_e = \text{area}_4 \begin{pmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \end{pmatrix} \tag{4.152}$$

and the modi ed mass matrix given by

$$\tilde{M} = \int_W \tilde{W}_i W_j \, dW \tag{4.153}$$

has the element mass matrix (for a triangle such as w_{e_2} with two internal nodes and one boundary node) given by

$$M_{e_2} = \text{area}_4 \begin{pmatrix} \frac{5}{24} & \frac{3}{24} & \frac{1}{6} \\ \frac{3}{24} & \frac{5}{24} & \frac{1}{6} \\ 0 & 0 & 0 \end{pmatrix}. \tag{4.154}$$

We can calculate modi ed stiffness matrices in the same way. The standard element stiffness matrix is

$$K_e = \frac{1}{2} \begin{pmatrix} \cot\gamma + \cot\beta & -\cot\gamma & -\cot\beta \\ -\cot\gamma & \cot\alpha + \cot\gamma & -\cot\alpha \\ -\cot\beta & -\cot\alpha & \cot\beta + \cot\alpha \end{pmatrix}$$

by

$$u(0) = 75\sin(\pi(0.5 - r)). \qquad (4.160)$$

In this case, the reaction does build, and we observe blow-up in a similar manner to the 1-D case. We observe node movement towards the centre as u becomes large there. The solution tends towards a Dirac delta function before the model collapses due to node tangling. These results are presented in gures 4.12 to 4.16. We use 5 nodes on 20 concentric circles. The initial grid is presented in gure 4.17. Note that the outermost circle is different in having 10 nodes. This is to avoid the situation where if only 5 nodes were used, nodes from the next circle inward from the boundary would form part of the boundary, as a consequence of the alternating positioning of the nodes on adjacent circles. This would complicate the implementation of the boundary conditions, so additional nodes are added on the outer circle only. We use a time step of $dt = 10^{-5}$. Figure 4.13 shows the solution at $t = 0.01$, and gure 4.14 4.14 $t = 0.02$. We observe l]8(4.ap t)=-278(19-25gure]8(4.15)(4-13)]TJ bap

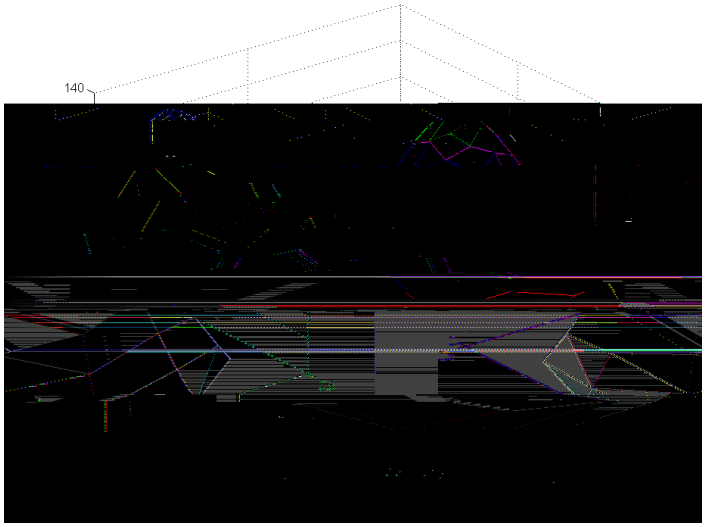


Fig. 4.13 Solution of the 2D Fisher's equation at $t = 0.01$. Note change of scale on the vertical axis.

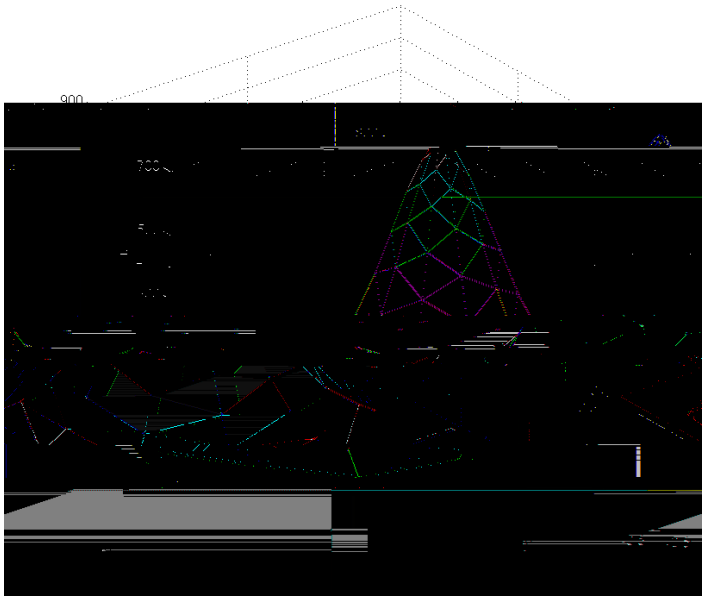


Fig. 4.14 Solution of the 2D Fisher's equation at $t = 0.02$. Note change of scale on the vertical axis.

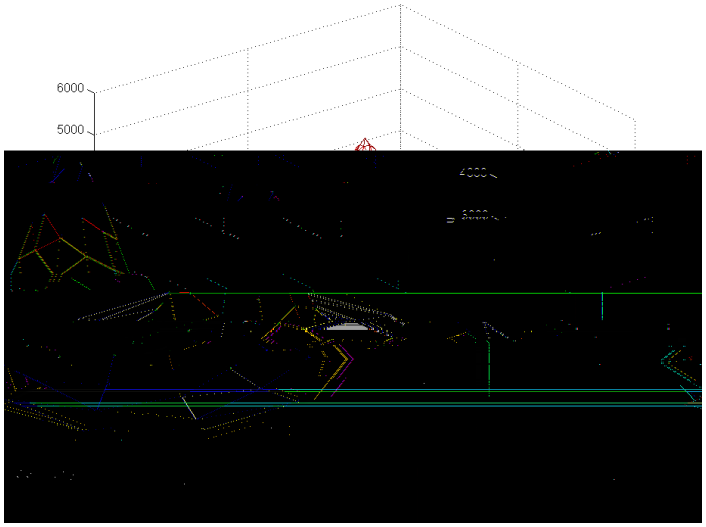


Fig. 4.15 Solution of the 2D Fisher's equation at $t = 0.0219$. Approaching blow-up. Note change of scale on the vertical axis.

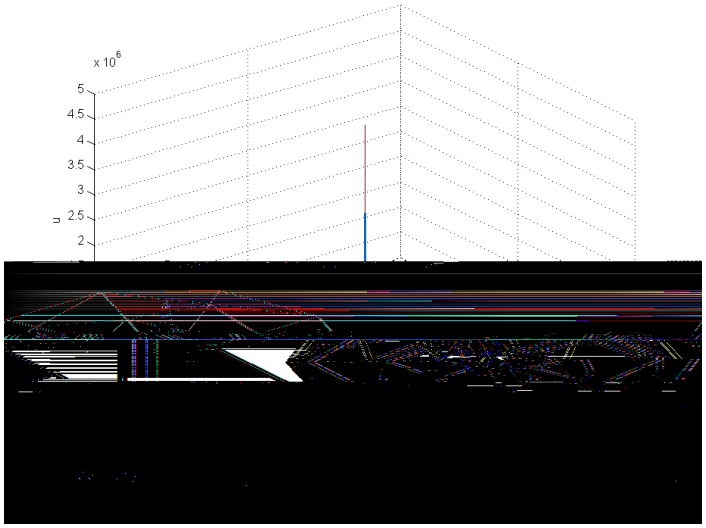


Fig. 4.16 Final solution of the 2D Fisher's equation. Here $t = 0.0225$. The solution approximates a Dirac delta function, and shortly after this time step the nodes become co-located and the model becomes unstable.

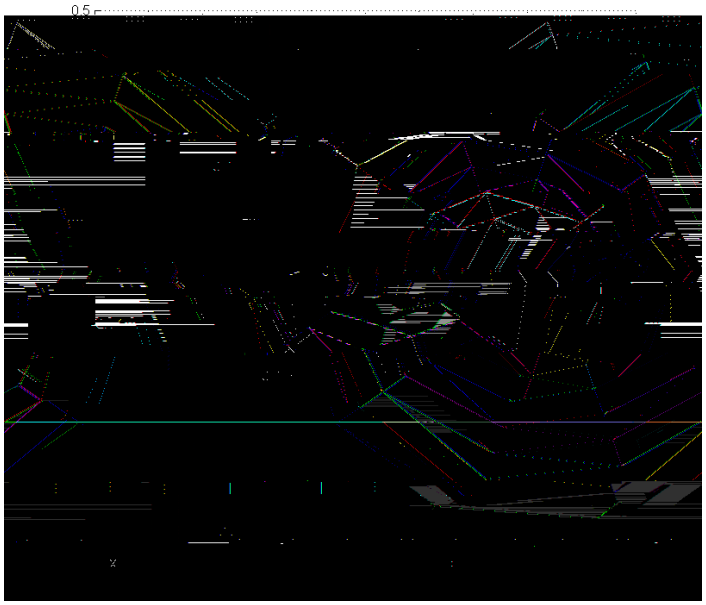


Fig. 4.17 Initial node positions for 2-D Fisher's equation at $t = 0$. We have 5 nodes on 20 equally spaced concentric circles.

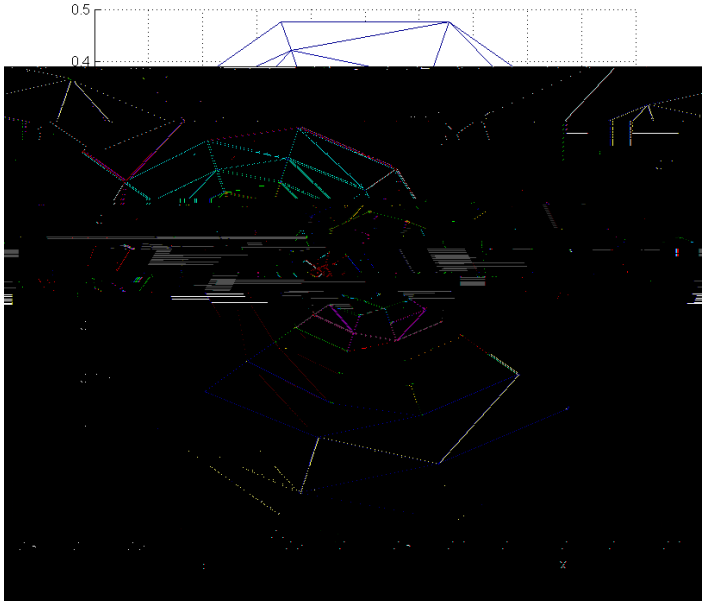


Fig. 4.18 Node positions for 2-D Fisher's equation at $t = 0.0219$ as we approach blow up. When compared to the initial grid, the movement towards the centre is clearly apparent.

4.3 Keller-Segel model in 2D

The Keller-Segel model [34] is a reaction-diffusion system related to the Fisher's equation. It differs from the Fisher's equation in that it involves both a substrate and a reactant, whereas the Fisher's equation is concerned with only the reactant. Both Cole [21] and Budd [13] consider the Keller-Segel system in two-dimensional, but radially symmetric, terms, on a moving mesh. Budd's paper [13] contains an equidistribution approach to moving the mesh, whereas Cole [21] demonstrates a conservation based method with a finite differences implementation. Here we move to a fully two dimensional approach, with a conservation based finite element method of solution (MMFEM).

This model, for chemotaxis of cells, takes the form of a pair of interdependent PDEs,

$$\frac{\partial u}{\partial t} = \nabla\cdot(k_1(u,v)\nabla u - k_2(u,v)u\nabla v) + k_3(u,v) \tag{4.161}$$

$$\frac{\partial v}{\partial t} = D_v\nabla^2 v + k_4(u,v) - k_5(u,v)v \tag{4.162}$$

where
u=cell density
v=concentration of substrate
k_1=diffusivity
k_2=chemotactic sensitivity
k_3=cell growth and death
k_4=production of substrate
k_5=degradation of substrate.

We model a system on a fixed domain W with boundary S. We take the Neumann boundary conditions used in [13], given by

$$\nabla u\cdot\hat{n}|_S = 0 \tag{4.163}$$

and

$$\nabla v\cdot\hat{n}|_S = 0. \tag{4.164}$$

We also take the initial values for u and v from [13], given by

$$u(r,0) = 1000e^{(-500r^2)} \tag{4.165}$$

$$v(r;0) = 10e^{(-500r^2)} \tag{4.166}$$

where $r \in W = \{r : krk \leq R\}$, and $R = 1$. A free boundary is unimportant here, since the initial conditions give a wide margin where

and

$$\frac{d}{dt}\int_W u \, dW = 0 = q; \tag{4.174}$$

i.e., mass is conserved. We define a distributed conservation principle using the weight functions w_i.

$$\int_W w_i u \, dW = c_i \tag{4.175}$$

or

$$\frac{d}{dt}\int_W w_i u \, dW = 0. \tag{4.176}$$

We differentiate using Leibnitz' rule and obtain

$$\int_W \frac{\partial}{\partial t}(w_i u) \, dW - \int_S w_i u \mathbf{x}\cdot\hat{n} \, dS = 0 \tag{4.177}$$

or

$$\int_W w_i \frac{\partial u}{\partial t} + u\frac{\partial w_i}{\partial t} + w_i \nabla\cdot(u\mathbf{x}) + u\mathbf{x}\cdot\nabla w_i \quad dW = 0. \tag{4.178}$$

If w_i

We arrive at

$$\int_W \tilde{N} w_i \cdot \chi_u \, dW = -\int_W \tilde{N} w_i \cdot \tilde{N} u \, dW + \int_W c \, \tilde{N} w_i \cdot u \tilde{N} v \, dW. \tag{4.184}$$

This is our weak form for χ in terms of u and v. We will move the nodes using a time integration scheme, and recover v using a conservation approach. We do however, require a weak form for v. We calculate v from the de nition of $\frac{\P v}{\P t}$, (4.168), the known nodal velocity χ and the material derivative

$$\frac{dv}{dt} = \frac{\P v}{\P t} + \tilde{N} v \cdot \chi. \tag{4.185}$$

The χ

We use the basis functions $\{W_i(x,y)\}$ and the piecewise linear approximations

$$\Phi(x,t) = \sum_{j=1}^{N} W_j(x,t)\Phi_j(t) \tag{4.191}$$

$$U(x,t) = \sum_{j=1}^{N} W_j(x,t)U_j(t) \tag{4.192}$$

$$V(x,t) = \sum_{j=1}^{N} W_j(x,t)V_j(t). \tag{4.193}$$

We can now write equation (4.184) in a finite element form.

$$\sum_{j=1}^{N} \int_\Omega U\nabla W_i.\nabla W_j \, d\Omega \;\; \Phi_j = -\sum_{j=1}^{N} \int_\Omega \nabla W_i.\nabla W_j \, d\Omega \;\; U_j + c\sum_{j=1}^{N} \int_\Omega (U\nabla W_i.\nabla W_j) d\Omega \;\; V_j. \tag{4.194}$$

In matrix form this is

$$K(\underline{U})\underline{\Phi} = \underline{f} \tag{4.195}$$

with $K(\underline{U})$ the weighted sto... i 3^3 f $\sum_{j=1}^{N} \int W_i \nabla W_j \, 3W$ $V_j + \sum_{j=1}^{N} \int_\Omega \nabla c \, W_i.\nabla W_j) d\Omega \;\; V_j($ ...chapterormhte3,orm73(se]TJ/Fg 0 G [())-rmhte3

together with the piecewise linear approximations of U (4.192) and V (4.193), we obtain from (4.188

Table 4.5 Blow up time for 2D Keller-Segel model with $m = 10$ showing variation by Δt, and n

m	n	Δt	$T_{blow-up}$
10	5	4×10^{-7}	2.00×10^{-5}
10	5	2×10^{-7}	1.90×10^{-5}
10	5	1×10^{-7}	1.90×10^{-5}
10	5	5×10^{-8}	2.00×10^{-5}
10	10	4×10^{-7}	1.48×10^{-5}
10	10	2×10^{-7}	1.24×10^{-5}
10	10	1×10^{-7}	1.10×10^{-5}
10	10	5×10^{-8}	1.00×10^{-5}

Table 4.7 Blow up time for 2D Keller-Segel model with $m = 40$ showing variation by Δt, and n

m	n	Δt	$T_{blow-up}$
40	5	$4x10^{-7}$	$5.52x10^{-5}$
40	5	$2x10^{-7}$	$5.54x10^{-5}$
40	5	$1x10^{-7}$	$5.56x10^{-5}$
40	5	$5x10^{-8}$	$5.56x10^{-5}$
40	10	$4x10^{-7}$	$1.52x10^{-5}$
40	10	$2x10^{-7}$	$1.52x10^{-5}$
40	10	$1x10^{-7}$	$1.52x10^{-5}$
40	10	$5x10^{-8}$	$1.52x10^{-5}$
40	20	$4x10^{-7}$	$8.80x10^{-6}$
40	20	$2x10^{-7}$	$8.80x10^{-6}$
40	20	$1x10^{-7}$	$8.80x10^{-6}$
40	20	$5x10^{-8}$	$8.85x10^{-6}$

Examples of the graphical results obtained are given in gures 4.19, 4.20 and 4.21. We observe the increasing height of the peak in u and the much lesser reduction in the height of v. Figure88 11.9552 .967 0 Td [(7)]TJ/Q 0 g 0 G BT /F86 7Td [(8)]TJ4.2-hwt91dtosi9a9552 T

method and it should be noted that for domains which involve large regions with zero mass or constant mass, node movement cannot occur in those regions.

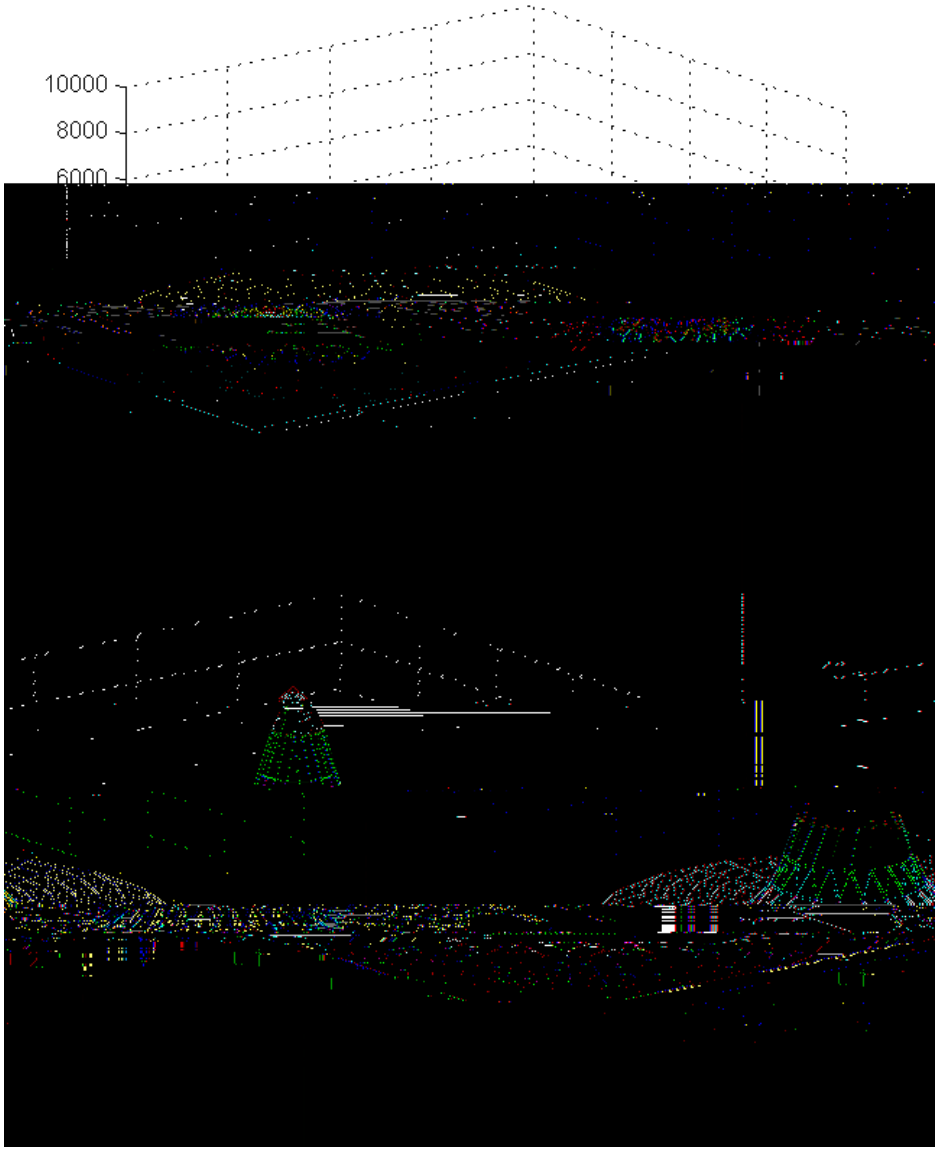


Fig. 4.19 Initial conditions for the Keller Segel model.

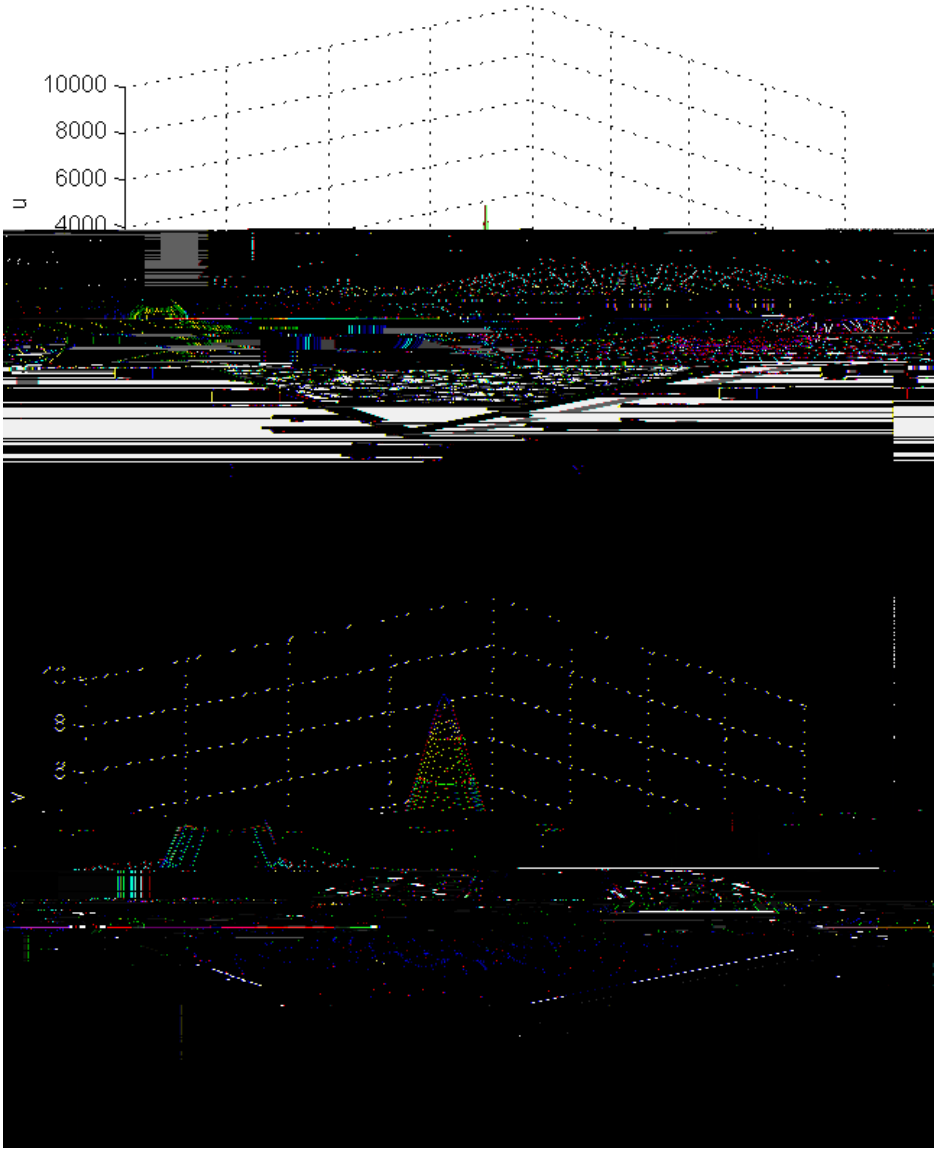


Fig. 4.20 Solution of the Keller Segel model on a grid with 20 nodes on 20 concentric circles at t = 5x10os/F86 11 gl6642 Tf 11.56 (4.54 Td [x)]TJ/F86 gl6642 Tf6gl663 0 Td [6x

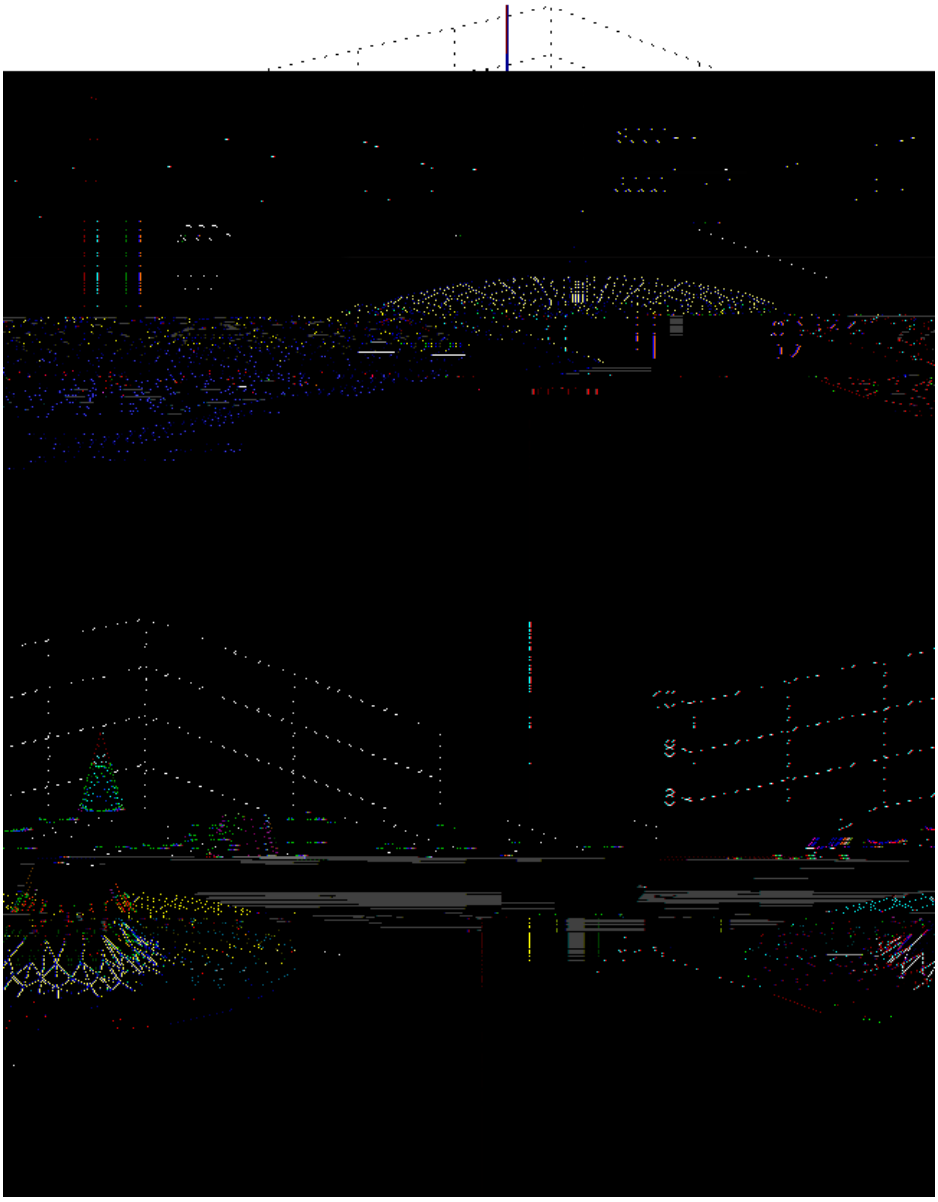


Fig. 4.21 Solution of the Keller Segel model on a grid with 20 nodes on 20 concentric circles as we approach blow-up.

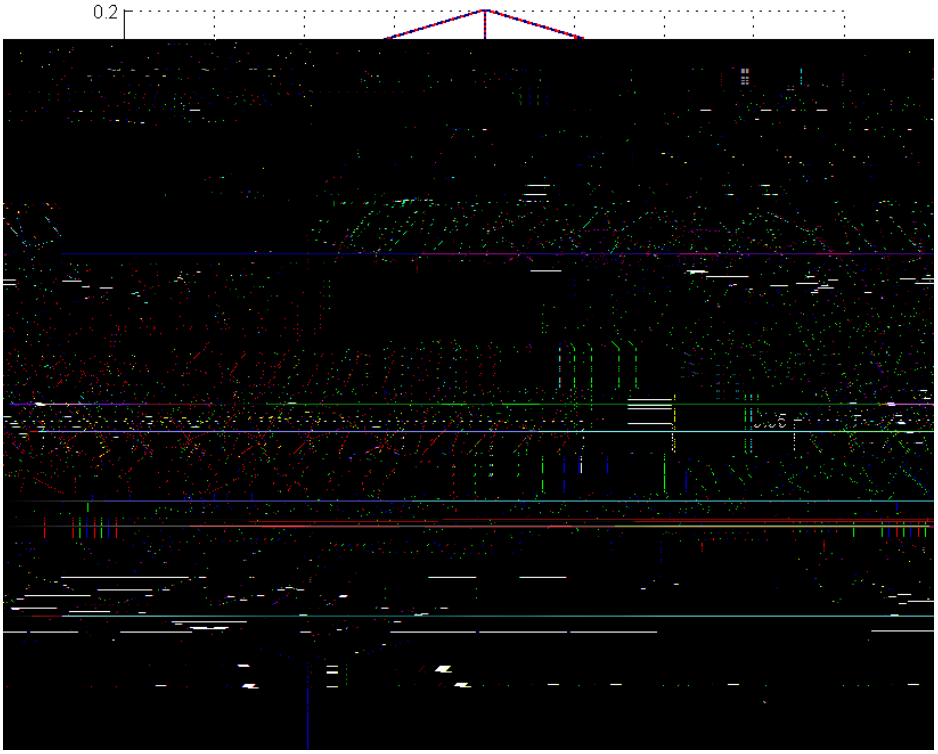


Fig. 4.22 Comparison of mesh movement between initial distribution (red dotted line) and approaching blow-up (blue solid line).

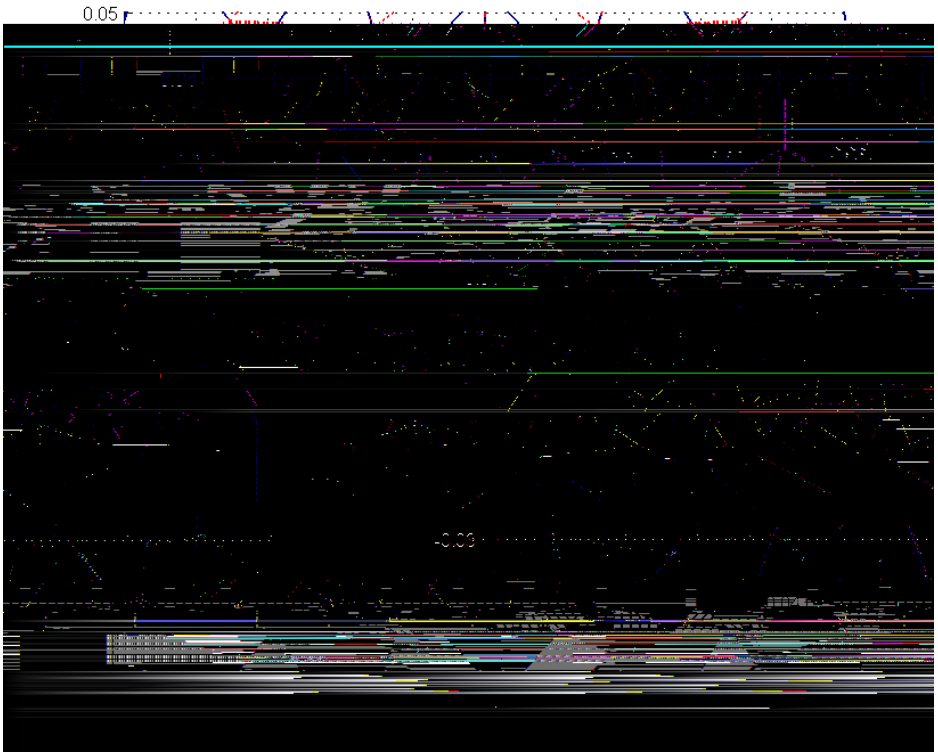


Fig. 4.23 Comparison of mesh movement between initial distribution (red dotted line) and approaching blow-up (blue solid line), a closer view.

Chapter 5

Moving interface models

5.1 The two phase Stefan problem in 1D

We now consider models with a moving interface between two phases. These models are a natural development from the free boundary variants of Chapter 4, for example the Fisher's model of section 4.2 with case (2) boundary conditions. We begin with a model of the two phase Stefan problem, constructed in a similar manner to that described in the Baines, Hubbard, Jimack and Mahmood (2009) paper [8]. The model describes the melting of ice into water. This model differs from those seen in this thesis so far in that the nodes at the phase boundary are themselves moving, as well as node movement within each phase. The model explicitly calculates the velocity of the interface between phases as the ice melts. This velocity comes from an interface condition, and this information is then incorporated into the model as a Dirichlet condition at the moving boundary. The model is constructed as a moving mesh nite element model. We present a modi cation to the paper [8]. In this problem we have Dirichlet boundary conditions on the boundary velocities as well as on the temperature of the ice or water. This makes it possible to construct the entire nite element model from start to nish in terms of the modi ed basis functions described in Chapter 4, section 4.2.1. We therefore do not need to switch basis systems via the ALE equation, as we did for the free boundary Fisher's problem (4.2) and as is derived in the paper [8]. We derive this alternative process and demonstrate that results equivalent to [8] can be obtained by it. The system is driven by the diffusion of heat. We consider the 1-D diffusion PDEs

$$K_S \frac{\P u}{\P t} = \frac{\P}{\P x} \quad k_S \frac{\P u}{\P x}$$

$$K_L \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(k_L \frac{\partial u}{\partial x}\right). \tag{5.1}$$

The parameters used are K_S and K_L, the volumetric heat capacities of the solid and liquid phases; k_S and k_L, the thermal conductivities; and u, the temperature.

At the interface, $u = u_m$, the temperature at which melting takes place. There is an energy balance across the phase-change boundary $\Gamma_m(t)$. This is described by the Stefan equation

$$k_S \frac{\partial u_S}{\partial x} - k_L \frac{\partial u_L}{\partial x} = \lambda \dot{x}_m \tag{5.2}$$

with λ, the heat of phase change per unit volume; and $\dot{x}_m$, the velocity of the interface. We assume that all parameters are constant within their respective phases. In this system the derivative $\frac{\partial u}{\partial x}$ is not continuous across the moving interface so we will need to be explicit about in which phase we are evaluating that gradient.

The particular case we will consider uses fixed outer boundaries $x \in [0, 1]$ with zero Dirichlet conditions on the velocity for external boundary nodes, and initial conditions taken from a system with an exact solution,

$$u_S = u \quad 1 -$$ erf¶ Tu Th1.926Tf()]TJ/F8611.9556BT/F9811.9553()-205.[

The following values for the parameters are used:

$u = 20$

$u_0 = 10$

$k_S = 2{:}22$

$k_L = 0{:}556$

$K_S = 1{:}762$

$K_L = 4{:}226$

$l = 338$

$y = 0{:}2054$

$t_{initial} = 0{:}0012$ (in order to avoid a singularity in (5.3))

$k = k{=}K{:}$

We consider a domain $R(t$

it may be helpful to think of this as 'mass' and we will use that shorthand here.

$$q(t) = \int_{R(t)} u \, dx \tag{5.11}$$

In terms of q and the constants this is

$$c_i q - \int_{R(t)} w_i \frac{\partial}{\partial x}(ux)\, dx = \int_{R(t)} w_i \frac{\partial u}{\partial t}\, dx. \tag{5.19}$$

For consistency of method with the 2-D version, we introduce the velocity potential defined by

$$x = \frac{\partial f}{\partial x} \tag{5.20}$$

so that

$$c_i q - \int_{R(t)} w_i \frac{\partial}{\partial x}\left(u \frac{\partial f}{\partial x}\right) dx = \int_{R(t)} w_i \frac{\partial u}{\partial t}\, dx \tag{5.21}$$

or, after integration by parts

$$c_i q + \int_{R(t)} u \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx -$$

with distributed (weak) forms

$$c_{S_i} q_S(t) = \int_{R_S(t)} w_i u \, dx \tag{5.27}$$

$$c_{L_i} q_L(t) = \int_{R_L(t)} w_i u \, dx \tag{5.28}$$

We rewrite (5.24) for each phase separately. Since $x = 0$ at the external boundaries and $u = u_m$

At the moving interface, the Stefan condition can replace the terms in equations involving f. We may run into difficulties with this if $u_m = 0$ or changes sign at any point, so as in [8] we will add a constant to the whole domain when constructing the algorithm. Equations (5.30) and (5.32) become

$$c_{S_i} q_S + \int_{R_S(t)} u \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx = -\int_{R_S(t)} k_S \frac{\partial w_i}{\partial x} \frac{\partial u}{\partial x}\, dx + w_i k_S \frac{\partial u}{\partial x}\Big|_{R_m(t)} - w_i k_S \frac{\partial u}{\partial x}\Big|_{R_f} + \frac{u_m}{l}\left(k_S w_i \frac{\partial u_S}{\partial x}\Big|_{R_m(t)} - k_L w_i \frac{\partial u_L}{\partial x}\Big|_{R_m(t)} \right) \tag{5.34}$$

$$c_{L_i} q_L + \int_{R_m(t)} u \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx = -\int_{R_L(t)} k_L \frac{\partial w_i}{\partial x} \frac{\partial u}{\partial x}\, dx + w_i k_L \frac{\partial u}{\partial x}\Big|_{R_f} - w_i k_L \frac{\partial u}{\partial x}\Big|_{R_m(t)} - \frac{u_m}{l}\left(k_S w_i \frac{\partial u_S}{\partial x}\Big|_{R_m(t)} - k_L w_i \frac{\partial u_L}{\partial x}\Big|_{R_m(t)} \right). \tag{5.35}$$

We can then sum equations (5.34) and (5.35) over $R_S(t)$ and $R_L(t)$ respectively, to give us the rate of change of total 'mass' q_S and q_L, in each phase. Providing that we have chosen a set of basis functions w_i that form a partition of unity, the full integral terms will sum to zero and the values of c_i will sum to 1. We obtain

$$q_S = k_S \frac{\partial u}{\partial x}\Big|_{R_m(t)} - k_S \frac{\partial u}{\partial x}\Big|_{R_f} + \frac{u_m}{l}\left(k_S \frac{\partial u_S}{\partial x}\Big|_{R_m(t)} - k_L \frac{\partial u_L}{\partial x}\Big|_{R_m(t)} \right) \tag{5.36}$$

$$q_L = k_L \frac{\partial u}{}$$

directly, i.e., for the interface, the velocity χ_m is given by

$$l\,\omega_i \chi_m|_{R_m(t)} = k_S \omega_i \frac{\partial u_S}{\partial x}\bigg|_{R_m(t)} - k_L \omega_i \frac{\partial u_L}{\partial x}\bigg|_{R_m(t)}. \tag{5.39}$$

The boundary conditions (5.7) and (5.8) that give

5.1.1 Construction of the finite element form

We solve the derived system using a finite element method. Since we have Dirichlet boundary conditions on equation (5.38) for the velocity, and also on (5.43) and (5.44) for the temperature, we use the modified piecewise linear weight functions $w_i = \tilde{W}_i$ of 4.2.1. We define an approximation to each of our variables in terms of a weighted linear combination of the W_j. These are given in Appendix A.

We also define the weightings c_i of (5.14) in terms of the same $\tilde{W}_i$. These are

$$\sum_{j \in Z_S} \int_{R(t)} \tilde{W}_i W_j \, dx \; U_{S_j} = \tilde{c}_{i_S} q_S(t) \tag{5.45}$$

$$\sum_{j \in Z_L} \int_{R(t)} \tilde{W}_i W_j \, dx \; U_{L_j} = \tilde{c}_{i_L} q_L(t). \tag{5.46}$$

Here Z_S and Z_L are the sets of nodes in the solid and liquid phases respectively. We can now express the system in finite element form. We make substitutions as necessary from equations (A.5) to (A.14) into equations (5.34) and (5.35) so that all our variables are expressed in terms of their piecewise linear approximations. We obtain

$$\begin{aligned} \tilde{c}_{i_S} q_S + \sum_{j \in Z_S} \int_{R_S(t)} U_S \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; F_j = &- \sum_{j \in Z_S} \left(\int_{R_S(t)} k_S \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; U_{S_j} + \tilde{W}_i k_S \frac{\partial U_S}{\partial x} \right)_{R_m(t)} \\ &- \tilde{W}_i k_S \frac{\partial U_S}{\partial x}_{R_f} + \frac{U_m}{l} \left(k_S \tilde{W}_i \frac{\partial U_S}{\partial x}_{R_m(t)} - k_L \tilde{W}_i \frac{\partial U_L}{\partial x}_{R_m(t)} \right) \end{aligned} \tag{5.47}$$

$$\begin{aligned} \tilde{c}_{i_L} q_L + \sum_{j \in Z_L} \int_{R_m(t)} U_L \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; F_j = &- \sum_{j \in Z_L} \left(\int_{R_L(t)} k_L \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; U_{L_j} + \tilde{W}_i k_L \frac{\partial U_L}{\partial x} \right)_{R_f} \\ &- \tilde{W}_i k_L \frac{\partial U_L}{\partial x}_{R_m(t)} - \frac{U_m}{l} \left(k_S \tilde{W}_i \frac{\partial U_S}{\partial x}_{R_m(t)} - k_L \tilde{W}_i \frac{\partial U_L}{\partial x}_{R_m(t)} \right) : \end{aligned} \tag{5.48}$$

In matrix form (5.47) is

$$\tilde{K}(\underline{U}_S) \; \underline{F}_S = \underline{\tilde{f}}_S \tag{5.49}$$

where $\tilde{K}(\underline{U}_S)$ is the weighted stiffness matrix of 3.1.2 constructed with modified basis functions, and $\underline{F}_S$ is the vector containing the values of F_{S_j}, and $\underline{\tilde{f}}_S$ is a vector with entries $\tilde{f}_{S_i}$

given by

$$\tilde{f}_{S_i} = -\tilde{c}_{i_S} q_S - \sum_{j \in Z_S} \int_{R_S(t)} k_S \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x}\, dx\; U_{S_j} + \tilde{W}_i k_S \frac{\partial U_S}{\partial x}\Big|_{R_m(t)} - \tilde{W}_i k_S \frac{\partial U_S}{\partial x}\Big|_{R_f} + \frac{U_m}{l}\Big(k_S \tilde{W}_i \frac{\partial U_S}{\partial x}\Big|_{R_m(t)} - k_L \tilde{W}_i \frac{\partial U_L}{\partial x}\Big|_{R_m(t)} \Big). \tag{5.50}$$

Similarly, (5.48) can be expressed as

$$\tilde{K}(\underline{U})_L\; \underline{F}_L = \underline{\tilde{f}}_L \tag{5.51}$$

with the vector $\underline{\tilde{f}}_L$ containing entries $\tilde{f}_{L_i}$ given by

$$\tilde{f}_{L_i} = -\tilde{c}_{i_L} q_L - \sum_{j \in Z_L} \int_{R_L(t)} k_L \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x}\, dx\; U_{L_j} + \tilde{W}_i k_L \frac{\partial U_L}{\partial x}\Big|_{R_f} - \tilde{W}_i k_L \frac{\partial U_L}{\partial x}\Big|_{R_m(t)} - \frac{U_m}{l}\Big(k_S \tilde{W}_i \frac{\partial U_S}{\partial x}\Big|_{R_m(t)} - k_L \tilde{W}_i \frac{\partial U_L}{\partial x}\Big|_{R_m(t)} \Big). \tag{5.52}$$

The matrix systems can be solved to obtain F_L and F_R. Since the weighted stiffness matrices $\tilde{K}(\underline{U})_S$ and $\tilde{K}(\underline{U})_L$ are singular, we have an in nity of solutions available and we set $F = 0$ at the interface node to reduce the system in order to give a unique solution. Note that the expressions for q_S (5.36) and q_L (5.37) can be obtained and solved in a straightforward manner by simply summing over the rows of (5.49) and (5.51).
To recover x, we use the approximation

$$X = \sum_j X_j W_j. \tag{5.53}$$

We substitute this into equation (5.38) to obtain the nite element form

$$\sum_{j \in Z_S \cup Z_L} \int_{R(t)} \tilde{W}_i W_j\, dx\; X_j = \sum_{j \in Z_S \cup Z_L} \int_{R(t)} \tilde{W}_i \frac{\partial W_j}{\partial x}\, dx\; F_j \tag{5.54}$$

or in matrix form,

$$M\underline{X} = B\underline{F}. \tag{5.55}$$

We impose the velocity on the interface obtained from (5.39), and we impose $X_j = 0$ on the

As in [33], we now move the nodes using Heun's scheme [30]. Using the same scheme, we update the values of q_S and q_L from the values of $\dot{q}_S$ (5.36) and $\dot{q}_L$ (5.37). The final step is the recovery of U. We can obtain U on the updated grid from the relative conservation of mass equations (5.45) and (5.46). These can be expressed in matrix form as

$$\tilde{M}\underline{U}_S = \tilde{\underline{c}}_{S_i} q_S(t) \tag{5.56}$$

and

$$\tilde{M}\underline{U}_L = \tilde{\underline{c}}_{L_i} q_L(t). \tag{5.57}$$

In the initial set up, we set $t = 0$ and use (5.56) and (5.57) to find the constant vectors $\tilde{\underline{c}}_{S_i}$ and $\tilde{\underline{c}}_{L_i}$. Then for each subsequent time step we proceed as follows. We take q_S and q_L at the new time step. We calculate the mass matrix $\tilde{M}$ for the updated grid. We can then obtain the updated $\underline{U}_S$ and $\underline{U}_L$ from inversions of (5.56) and (5.57) respectively.

Algorithm 11

The finite element solution of the Stefan problem given by equations (5.1) and with an interface condition given by (5.2) on the moving mesh in 1-D therefore consists of the following steps. We first add a constant to the domain so that we avoid any zero or negative values for U. Having obtained the values of $\tilde{c}_{S_i}$ and $\tilde{c}_{L_i}$ from (5.56) and (5.57):

1. Find the velocity potential by solving equation (5.49) and (5.51) for $F_j(t)$ values;
2. Find the node velocity by solving equation (5.55) for $\dot{x}_j(t)$ values;
3. Generate the co-ordinate system at the next time-step by solving (3.18) using Heun's approximation;

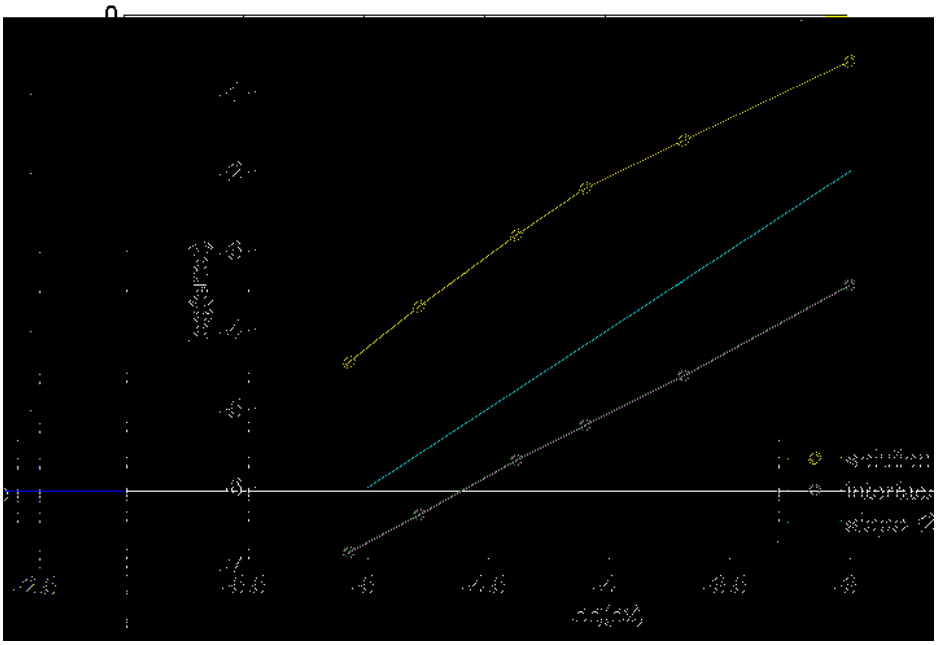


Fig. 5.1 Comparison of L^2 errors in the solution and the magnitudes of the errors in the interface node position for the two-phase Stefan problem in one space dimension, at $T = 0.5$. We observe an order of convergence of $p = 2$.

the convergence of the L^2 errors at $T = 0.5$ as the mesh resolution is increased. Both the normalised solution error and the interface position error have an order of convergence of approximately $p = 2$. This is consistent with the order of convergence given in [8] and demonstrates that this method is an acceptable alternative.

5.2 The two phase model of competition-diffusion

We now turn our attention to competition-diffusion models, in particular the Lotka-Volterra systems of theoretical ecology. As described in Chapter 2, there are many variations. After

where d_1, d_2 are constant diffusion coefficients, and with, in general

$$f(u_1, u_2) = r_1 \left(1 - \frac{u_1 + K_1 u_2}{k_1}\right) \tag{5.60}$$

$$g(u_1, u_2) = r_2 \left(1 - \frac{u_2 + K_2 u_1}{k_2}\right) : \tag{5.61}$$

Here u_1 and u_2 are the population densities of two competing species, k_1 and k_2 are the respective carrying capacities of the species, the K's are the species specific competition rates, and r_1 and r_2 are a reproductive rate parameter. The Hilhorst paper [31] demonstrates that this system can be reduced, if we have two species completely segregated, to

$$f(u_1, u_2) = r_1(1 - u_1/k_1) \tag{5.62}$$

$$g(u_1, u_2) = r_2(1 - u_2/k_2): \tag{5.63}$$

The resulting system represents the limit where the K values are very large; the competition rate is high enough that the two species cannot coexist in space. In the area populated by species 1, $u_2 = 0$, and in the area populated by species 2, $u_1 = 0$. At the interface, we have a condition that gives the relationship between the fluxes of the two species. In essence, the species both flow into the interface and annihilate each other in a ratio according to the competition coefficient, m. This condition is given by [31] as

$$m d_1 \frac{\partial u_1}{\partial x} = -d_2 \frac{\partial u_2}{\partial x} \tag{5.64}$$

where $m = K_2/K_1$. We will call m the interspecies competition rate. We work with Neumann boundary conditions on the external boundaries, which will be fixed,

$$\frac{\partial u_1}{\partial x}\bigg|_{r}$$

principle in terms of q, as

$$\frac{1}{q(t)} \int_{R(t)} u \, dx = 1. \tag{5.71}$$

We write this in a weighted form, introducing the weight function w_i,

$$\frac{1}{q(t)} \int_{R(t)} w_i u \, dx = c_i \tag{5.72}$$

or

$$\int_{R(t)} w_i u \, dx = c_i q(t) = c_i \int_{R(t)} u \, dx \tag{5.73}$$

where c_i is independent of time. The constant c_i is determined by the choice of weighting w_i. All of the weightings together should be chosen to provide a partition of unity. We differentiate (5.73) with respect to time using the Leibnitz integral rule on our moving frame $R(t)$,

$$\frac{d}{dt} \int_{R(t)} w_i u \, dx = \int_{R(t)} \frac{\partial (w_i u)}{\partial t} + \frac{\partial}{\partial x}(w_i u x) \quad dx. \qquad i,$$

or, after integration by parts,

$$c_i q + \int_{R(t)} u \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx - \left. u w_i \frac{\partial f}{\partial x} \right|_{\partial R(t)} = \int_{R(t)} w_i \frac{\partial u}{\partial t}\, dx. \tag{5.81}$$

We substitute in a weak form of the driving PDE, either (5.68) or (5.69), depending on the phase under consideration. For either phase $p \in [1, 2]$

$$c_p q_p + \int_{R_p(t)} u_p \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx - \left. u_p w_i \frac{\partial f}{\partial x} \right|_{\partial R_p(t)} = \int_{R_p(t)} w_i d_p \frac{\partial^2 u_p}{\partial x^2}\, dx + \int_{R_p(t)} w_i u_p r_p \left(1 - \frac{u_p}{k_p}\right) dx. \tag{5.82}$$

Again integrating by parts, this time on the right hand side

$$c_p \frac{\partial u_p}{\partial t}$$

429552 T6345 Td [(p)]TJ/F101 ;54.981 -1.793 .9664 6 Td [(R)]TJ/F87 6.9738 Tf 6.002 -1.345 Td [(p)]TJF87 8.9664 Tf 5.966

Equation (5.83) can be written now for each phase separately. For species 1 it becomes

$$c_{1_i} q_1 + \int_{R_1(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x}\, dx - u_1 w_i \frac{\partial f}{\partial x}\Big|_{\partial R_1(t)} = -\int_{R_1(t)} d_1 \frac{\partial w_i}{\partial x} \frac{\partial u_1}{\partial x}\, dx + w_i d_1 \frac{\partial u_1}{\partial x}\Big|_{\partial R_1(t)} + \int_{R_1(t)} w_i u_1 r_1 \left(1 - \frac{u_1}{k_1}\right) dx. \quad (5.88)$$

At the external boundaries $\frac{\partial u}{\partial x} = 0$ (5.65), and also $\frac{\partial f}{\partial x} = 0$ because the boundaries are xed. Together with the condition that $\frac{dR}{dt} = 0$ (5.67

(5.64) for the competition system is

$$m d_1 \frac{\partial u_1}{\partial x} = -d_2 \frac{\partial u_2}{\partial x} \tag{5.93}$$

which is equivalent to the Stefan condition with $h = 0$ and does not contain the velocity x. Note also that the Stefan condition is relevant to a situation where the gradients of either side of the interface are of the same sign in general. In contrast, equation (5.64) is relevant to an interface where the gradients either side are of opposite polarity. Since $u_1 = 0$ on the interface and we can't have a negative mass, we are in effect considering 'v' shaped interfaces. We note that whilst the interface velocity is not given by (5.64), the expression does implicitly contain information about the location of the interface. In particular, if we know the position of the mesh points adjacent to the interface and also the values of those points, we may use the fact that $u_1 = 0$ at the interface to infer an interface position that satisfies (5.64). We select an interface position such that the values of $\frac{\partial u_1}{\partial x}$ either side of the interface are in the ratio $-m$. We proceed as follows. At a given time step t_{N+1} we write the interface condition (5.64) in a finite difference form

$$m d_1 \left(u_{1_m}^N - \right.$$

the interface condition. We must allow the solution around the interface to evolve first, and then adapt the interface position in response to that. We cannot generate the position of the interface that satisfies (5.64) at the same time as we find the node velocities elsewhere, because we must solve the system for u on the updated grid before we can see where the interface ought to be positioned. After we have solved for u, we can obtain the interface position resulting from those u values, but we cannot impose it on the system straightaway. We would violate conservation of mass by doing so. Instead, we determine the new position at the next time step. A concern this raises is whether the interface position is effectively imposed one time step behind where it should be. The condition (5.64) is always slightly violated, since it is this violation that drives the interface movement. Philosophically, we can reconcile this difficulty by considering that there ought to be a force driving a movement of the interface *before* the interface starts to respond. In the real world, would our species retreat in anticipation of competition, or else compete and then accept the resulting boundary change? The subtleties of this interaction, and its timing or lag, are not considered in the Lotka-Volterra equations. We can therefore be confident that the explicit nature of our system does not violate any conditions of the system, and indeed it may better reflect reality than a predictive approach. Should we determine that a problem does exist in this regard, a suitable solution would be to use an implicit time integration method, which would accord the ability to reassign the interface movement to the prior time step if so desired.

To recover the nodal velocities we first solve (5.97) and (5.98) for q_i in each phase. We then solve (5.89) and (5.91) to give v in each phase, but not on the interface node due to the modified basis functions we will use. We then return to our definition of q_i (5.79), now written in distributed form,

$$\int_{R(t)} w_i x \, dx = \int_{R(t)} w_i \frac{\partial f}{\partial x} \, dx \tag{5.99}$$

which can be solved for x. Having obtained x, we move the domain using the explicit Euler integration scheme. We also update q_1 and q_2 from q_1 (5.97) and q_2 (5.98) using the same time integration procedure. For the interface itself, we calculate the new position by correcting the interface condition at the prior time step. We obtain the resultant interface velocity by solving equation (5.96) with $u_n = 0$ in the interface node.

We may now recover u. We determine the constant partial masses c_{1_i} and c_{2_i} from (5.86) and (5.87) and the initial conditions. We obtain, for $t = 0$

$$c_{1_i} = \frac{1}{q_1(0)} \int_{R_1(0)} w_i(x,0)u(x,0) \, dx \tag{5.100}$$

$$c_{2_i} = \frac{1}{q_2(0)} \int_{R_2(0)} w_i(x,0)u(x,0) \, dx. \tag{5.101}$$

We then use (5.86) and (5.87) again, to recover u_1 and u_2. We require q_1 and q_2 at the new time step. We move the weight functions with the domain. For species 1, u_1 can be recovered from

$$\int_{R_1(t)} w_i(x,t)u_1(x,t) \, dx = c_{1_i}(x)q_1(t) \tag{5.102}$$

and for species 2, u_2 can be recovered from

$$\int_{R_2(t)} w_i(x,t)u_2(x,t) \, dx = c_{2_i}(x)q_2(t). \tag{5.103}$$

In each case the Dirichlet condition that $u_n = 0$ at the interface is strongly imposed, and the Neumann condition at the external boundaries is also strongly imposed.

We solve the derived system using a finite element method. We have Dirichlet boundary conditions on equation (5.99) for the velocity, at both the interface and external boundaries. For the values of u_1 and u_2, given by equations (5.102) and (5.103), we have a Dirichlet condition at the interface only. At the external boundaries we have Neumann boundary conditions instead. However, all these conditions are compatible with using the modified

piecewise linear weight functions $w_i = \tilde{W}_i$ of 4.2.1, with a modified weight function at each external boundary, and also at each side of the interface. We may then strongly impose the values of the velocity and d_1 and u_2 at the interfaces and external boundaries. The values of u_1 and u_2 at the external boundaries can be transferred from their adjacent nodes because we have the Neumann conditions.

5.2.1 Construction of the finite element form

We begin the finite element method implementation by defining an approximation to each of our variables in terms of a weighted linear combination of the W_j. These are given in Appendix A. The weightings s_{p_i} of (5.102) and (5.103) are likewise defined in terms of $\tilde{W}_i$. We obtain

$$\sum_{j \in Z_1} \int_{R(t)} \tilde{W}_i W_j \, dx \; U_{1_j} = \tilde{c}_{1_i} q_1(t) \tag{5.104}$$

$$\sum_{j \in Z_2} \int_{R(t)} \tilde{W}_i W_j \, dx \; U_{2_j} = \tilde{c}_{2_i} q_2(t) \tag{5.105}$$

where Z_i is the set of nodes in phase i. We may rewrite the system in finite element form. We take the approximations (A.1) to (A.9) as necessary, and also (5.104) to (5.105), and make substitutions as necessary into equations (5.89) and (5.91). We obtain the following, with all variables now expressed in terms of their piecewise linear approximations. Equation (5.89) becomes

$$\begin{aligned} \tilde{c}_{1_i} q_1 + \sum_{j \in Z_1} \int_{R_1(t)} U_1 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; F_j &= -\sum_{j \in Z_1} \int_{R_1(t)} d_1 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; U_{1_j} \\ + \tilde{W}_i d_1 \frac{\partial U_1}{\partial x}\Big|_{R_m(t)} + \sum_{j \in Z_1} \int_{R_1(t)} \tilde{W}_i W_j r_1 \, dx \; U_{1_j} &- \int_{R_1(t)} \frac{r_1}{k_1} \tilde{W}_i U_1^2 \, dx \end{aligned} \tag{5.106}$$

and equation (5.91) becomes

$$\begin{aligned} \tilde{c}_{2_i} q_2 + \sum_{j \in Z_2} \int_{R_2(t)} U_2 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; F_j &= -\sum_{j \in Z_2} \int_{R_2(t)} d_2 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} \, dx \; U_{2_j} \\ - \tilde{W}_i d_2 \frac{\partial U_2}{\partial x}\Big|_{R_m(t)} + \sum_{j \in Z_2} \int_{R_2(t)} \tilde{W}_i W_j r_2 \, dx \; U_{2_j} &- \int_{R_2(t)} \frac{r_2}{k_2} \tilde{W}_i U_2^2 \, dx. \end{aligned} \tag{5.107}$$

In matrix form (5.106) is expressed as

$$\tilde{K}(\underline{U}_1)\ \underline{F}_1 = \underline{\tilde{f}}_1 \tag{5.108}$$

where $\tilde{K}(\underline{U}_1)$ is the weighted stiffness matrix of Chapter 3, section 3.1.2, constructed with the modified basis functions $\tilde{W}_i$, and $\underline{F}_1$ is the vector containing the values of F_{1_j}, and $\underline{\tilde{f}}_1$ is a vector with entries $\tilde{f}_{1_i}$ given by

$$\begin{aligned}\tilde{f}_{1_i} = &-\tilde{c}_{1_i} q_1 - \sum_{j \in Z_1} \int_{R_1(t)} d_1 \frac{\partial \tilde{W}_i}{\partial x}\frac{\partial W_j}{\partial x}\,dx\ U_{1_j} \\ &+ \tilde{W}_i d_1 \frac{\partial U_1}{\partial x}\Big|_{R_m(t)} + \sum_{j \in Z_1} \int_{R_1(t)} \tilde{W}_i W_j r_1\,dx\ U_{1_j} - \int_{R_1(t)} \frac{r_1}{k_1}\tilde{W}_i U_1^2\,dx.\end{aligned} \tag{5.109}$$

Similarly, (5.107) can be expressed as

$$\tilde{K}(\underline{U}_2)\underline{F}_2 = \underline{\tilde{f}}_2 \tag{5.110}$$

with the vector $\underline{\tilde{f}}_2$ containing entries $\tilde{}$

for which the nonlinear terms may be computed exactly using Simpson's rule (4.64). To recover X, we use the approximation

$$X = \sum_{j \in Z_1 \cup Z_2} X_j W_j : \tag{5.114}$$

We substitute this into equation (5.99) to obtain the finite element form

$$\sum_{j \in Z_1 \cup Z_2} \int_{R(t)} \tilde{W}_i W_j \, dx \; X_j = \sum_{j \in Z_1 \cup Z_2} \int_{R(t)} \tilde{W}_i \frac{\partial W_j}{\partial x} \, dx \; F_j \tag{5.115}$$

or in matrix form

$$\tilde{M} \underline{X} = \tilde{B}(u) \underline{F} : \tag{5.116}$$

We impose $v = 0$ on the external boundaries. We impose the interface velocity obtained from (5.96). Since we are using modified weight functions we will not interfere with the compatibility condition (5.40) by doing so. We solve (5.116) for the remaining velocities. We move the nodes using Euler's scheme. Using the same scheme, we update the values of q_1 and q_2 from the values of $\dot{q}_1$ (5.112) and $\dot{q}_2$ (5.113). We may now recover the values of U_1 and U_2. We can obtain U 5.99 4 to obtain Tf 1G [(5.99)]TJ 0 g 0 G [(5.)-365(W)80(e)-268(m

ues;

2. Find the internal node velocity by solving equation (5.116) for $X_j(t)$ values;
3. Find the interface node velocity by solving equation (5.96) for $X_{int}(t)$ value;
4. Generate the co-ordinate system at the next time-step $t + dt$ by solving (3.18) using Euler's approximation;
5. Update the values of q_1 and q_2 from the values of q_1 (5.112) and q_2 (5.113);
6. Find the solutions $U_1(t + dt)$ and $U_2(t + dt)$ by solving the conservation equations (5.117) and (5.118).

5.2.2 Results

We find that the model is stable and robust. Even using the simplest Euler integration scheme, we observe minimal oscillations affecting the smoothness of results. Figure 5.3 shows convergence in the solution of second or third order as $\Delta x \to 0$. This estimate is obtained by comparison of the result generated by each grid spacing with a high-resolution (641 node) result, since no absolute result is available. This order of convergence is at least as high as that reported for the very similar method in [8].

In the body of work concerning Lotka-Volterra equations, there are a vast range of parameter values in use, because there are so many varied but suitable examples of the type of competition that is described. We therefore select a conservatively representative set of parameters, chosen to demonstrate some of the interesting behaviours that this model is able

gains due to this alone (gure 5.9). However, as time goes on, the growth and competition characteristics become increasingly important. We see species 1 becoming more dominant over time, so that the interface velocity actually reverses direction. This is fascinating interface behaviour! Figure 5.10 shows the evolution of the system at $t = 12.3$, and gure 5.11 shows the movement of the interface with the direction reversal. These results give con dence that this model is likely to be able to satisfy the requirements of modelling a wide variety of competition systems. It is stable to a large choice of set-up parameters and

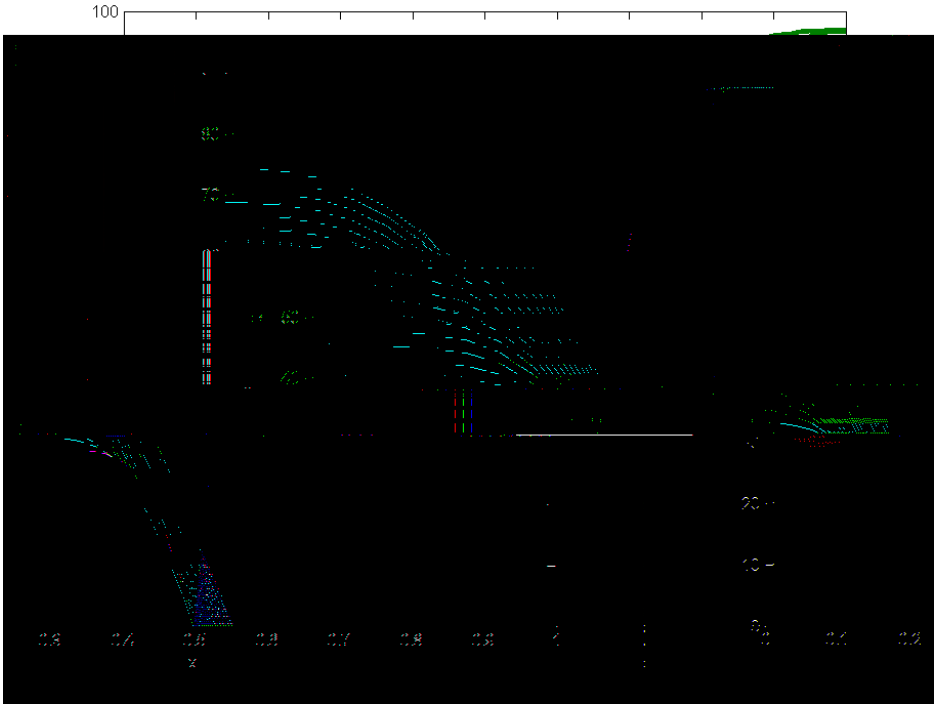


Fig. 5.4 Result of competition model at $t = 1.7$. Here we use $d_1 = d_2 = 0.01, k_1 = k_2 = 100$, $r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step $dt = 0.0001$ for 17000 iterations and plot the results every $dty = 0.1$. We see the internal dynamics of the species driving population density and interface fluxes, and the position of the interface responding to those fluxes. The initial conditions are shown in red, with species 1 in blue and species 2 in green.

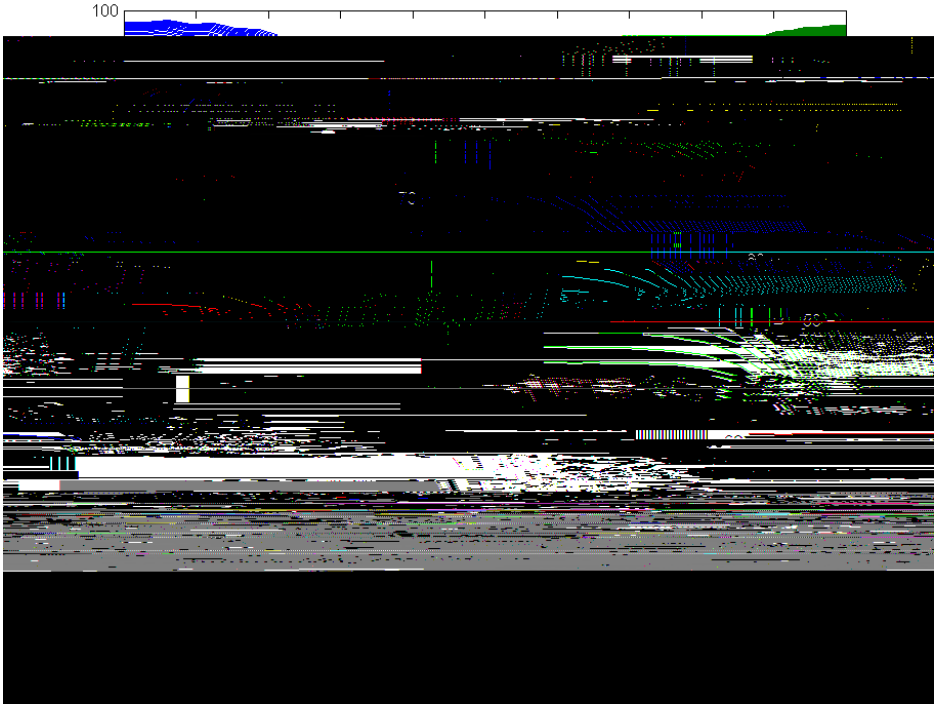


Fig. 5.5 Result of competition model at $t = 6.0$. Here we used $d_1 = d_2 = 0.01, k_1 = k_2 = 100$, $r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$ for 60000 iterations and plot the results every $Dt = 0.1$. The interface continues to evolve and the masses of the species are now limited by the respective carrying capacities. The initial conditions are shown in red, with species 1 in blue and species 2 in green

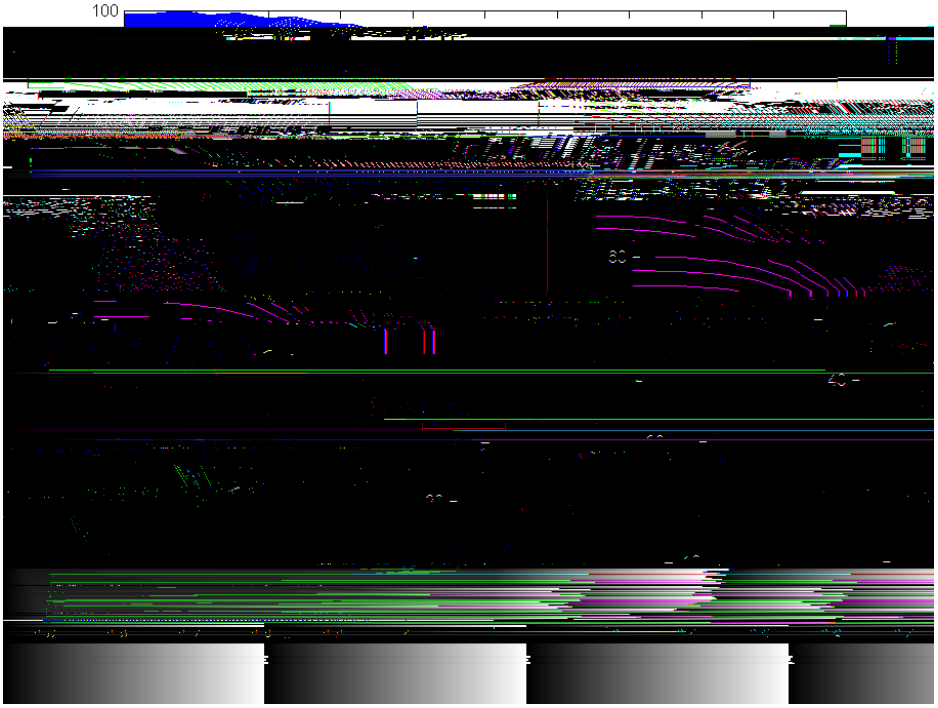


Fig. 5.6 Result of competition model at $t = 8.8$. Here we used $d_1 = d_2 = 0.01, k_1 = k_2 = 100$, $r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$ for 122000 iterations and plot the results every $dy = 0.1$. Final step before node crossing occurs. We observe that whilst species 2 initially grew in mass, it will now be wiped out by competition with species 1.

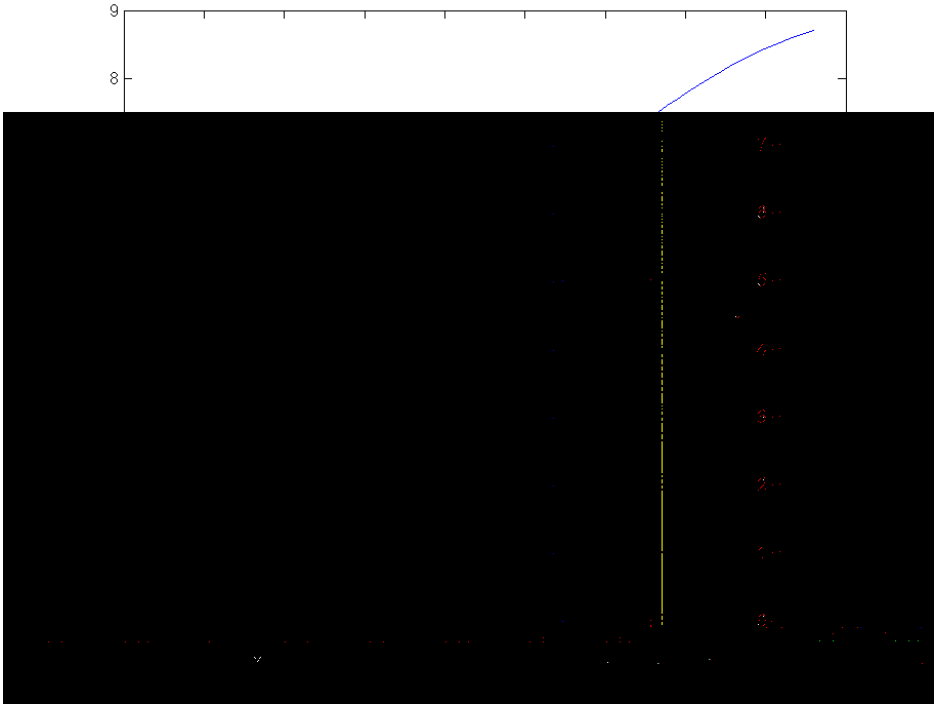


Fig. 5.7 Movement of interface position x_n for competition model with parameters $d_1 = d_2 = 0.01, k_1 = k_2 = 100, r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$. We see the interface increase in velocity after a slower initial phase where both species are experiencing population growth. We see the interface velocity accelerate as we approach an annihilation event.

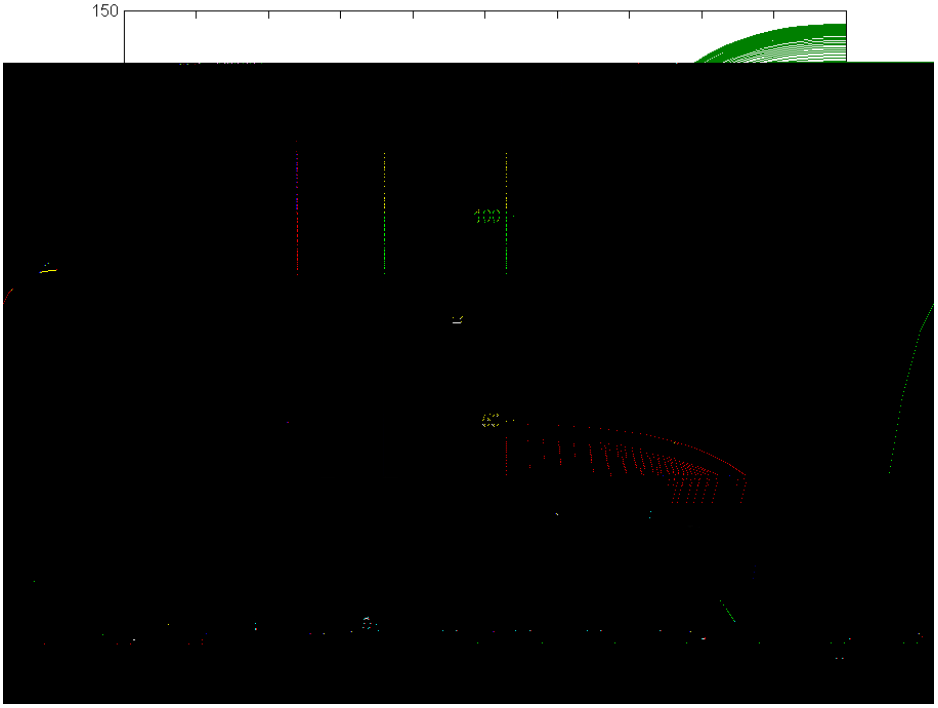


Fig. 5.8 Result of competition model at $t = 8$, considering the effect of altered carrying capacities. Here we used $d_1 = d_2 = 0.01$, $k_1 = 50, k_2 = 150, r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$ for 80000 iterations and plot the results every $dt = 0.1$. We see that with differently chosen carrying capacities we find the interface position is approximately steady and these two species are in balance.

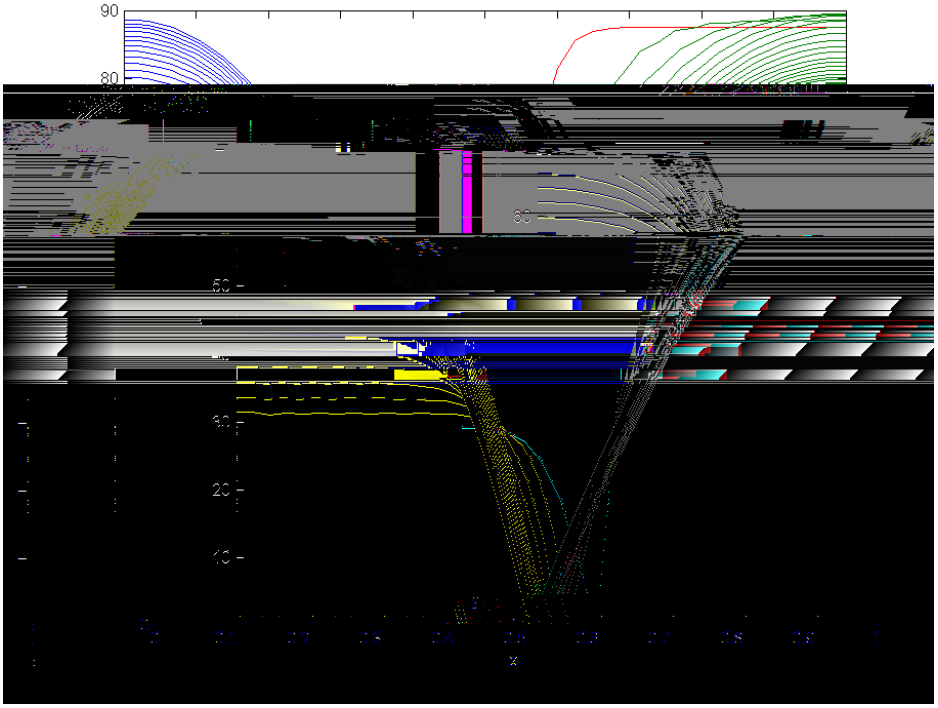


Fig. 5.9 Result of competition model at $t = 3.5$, considering the effect of an increased diffusion rate for species 2. Here we use $d_1 = 0.01, d_2 = 0.05, k_1 = k_2 = 100, r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$ for 35000 iterations, and plot the results every $dt = 0.1$. We observe that species 2 is able to make initial territory gains due to its high diffusion rate, even though the competition rate is unaltered.

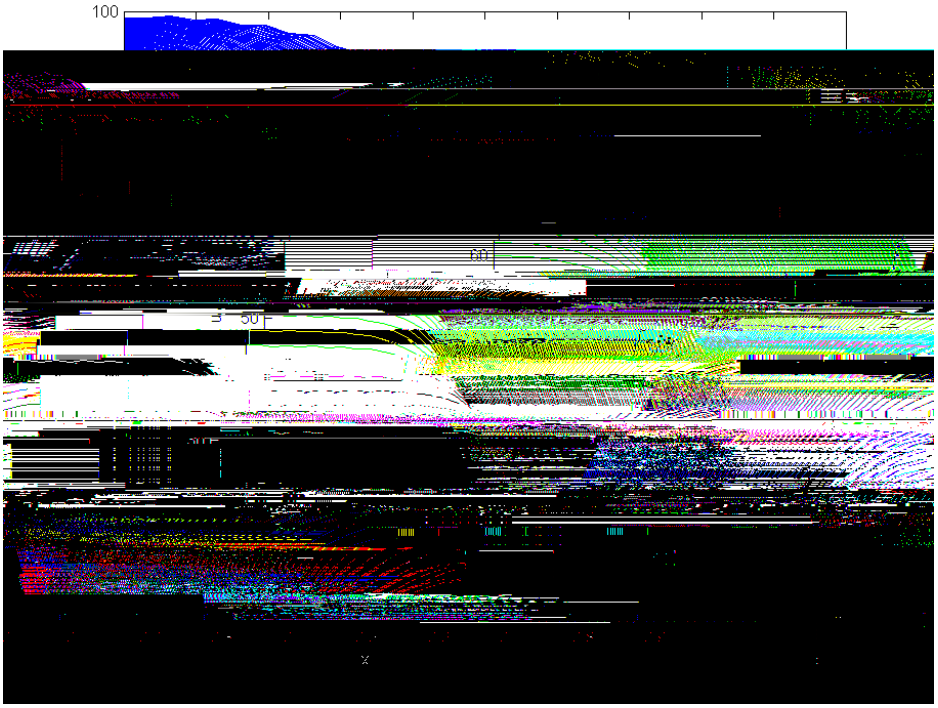


Fig. 5.10 Result of competition model at $t = 12.3$, considering the effect of an increased diffusion rate for species 2. Here we use $\delta t = 0.01$, $d_2 = 0.05$, $k_1 = k_2 = 100$, $r_1 = r_2 = 1$ and $I = 3$. We run the model with a time step of $dt = 0.0001$ for 123000 iterations, and plot the results every $dt = 0.1$. We see that the initial diffusion-driven gains by species 2 are reversed, and that the overall growth characteristics are dominating so that species 1 is gaining territory.

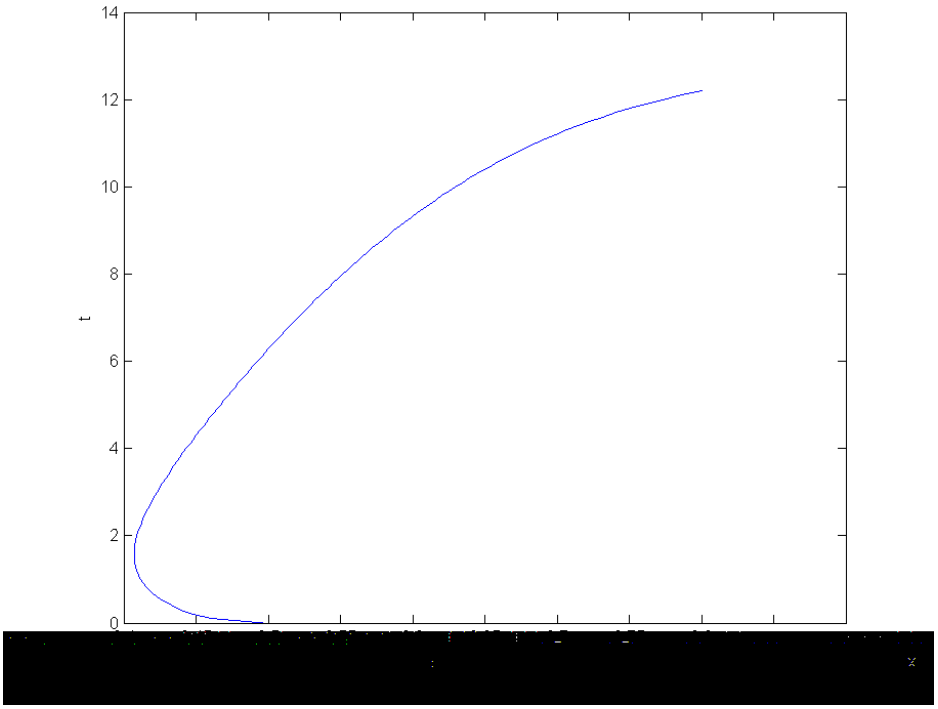


Fig. 5.11 Position of interface x_m, showing interface movement for the competition model at up to $t = 12.3$, considering the effect of an increased diffusion rate for species 2 (cf. figure 5.7). Here we use $d_1 = 0.01, d_2 = 0.05, k_1 = k_2 = 100, r_1 = r_2 = 1$ and $l = 3$. We run the model with a time step of $dt = 0.0001$ for 123000 iterations, and plot the results every $dt = 0.1$. Due to the growth characteristics we can see interesting temporal effects. Here the interface velocity has actually reversed directions as the system changes from diffusion dominated to growth dominated.

Chapter 6

Aggregation models

In [28] and [29], Grindrod presents a new consideration for population modelling. He points out that the derivation of the Lotka-Volterra competition models and similar single-species dispersion models rests on the assumption that the dispersal of individuals is due to random diffusive motion. This assumption is dif cult to justify, since it is readily apparent that in the real world, individuals group together to improve their chances of survival, do not voluntarily overcrowd themselves to death, and deliberately avoid predators. Grindrod therefore introduces an element of deterministic behaviour to his model. In the Grindrod models, we assume that the random motion of individuals is biased by an optimal velocity n. This velocity is selected so as to increase an individual's expected rate of reproduction. On average, the population is dispersing in the ideal direction. Grindrod produces results obtained from this model as derived for a single species, and demonstrates that from an initially random seeding of individuals, clusters are formed. This work is of interest to us for three reasons. Firstly, the aggregation model has not previously been constructed in nite element form, on either a static or moving grid. Secondly, the model has not previously been implemented for a two species competitive environment. Thirdly, the assumptions made by Hilhorst in [31] require a zero population condition on the interface that is entirely driven by high competition rates. Whilst we would need a high competition interface in any multi-phase scenario, having intelligent aggregation as a component of the model would seem to add somewhat more justi cation to the imposition of a zero population interface condition.

We may now derive the PDE giving population density time dependence. We have

$$\begin{aligned}\frac{\partial u}{\partial t} &= -\nabla\cdot(uv)\\ &= -\nabla\cdot(-d\nabla u + un)\\ &= d\nabla^2 u - \nabla\cdot(un)\\ &= d\nabla^2 u - \nabla\cdot(u\nabla q)\end{aligned} \tag{6.4}$$

with boundary conditions

$$\hat{n}\cdot\nabla u = 0 \qquad x \in \partial W,\quad t \geq 0, \tag{6.5}$$

$$\hat{n}\cdot n = 0 \qquad x \in \partial W,\quad t \geq 0,\quad n = \nabla q. \tag{6.6}$$

6.1.1 1D population clustering model for a single species

We examine the 1D analogues of the equations described in section 6.1.

$$\frac{\partial u}{\partial t} = d\frac{\partial^2 u}{\partial x^2} - \frac{\partial}{\partial x}\left(u\frac{\partial q}{\partial x}\right) \tag{6.7}$$

$$E(u) = -e\frac{\partial^2 q}{\partial x^2} + q \tag{6.8}$$

$$E(u) = (u-a)(1-u). \tag{6.9}$$

We have the boundary conditions

$$\frac{\partial u}{\partial x} = 0 \qquad x = A, B, \qquad t \geq 0, \tag{6.10}$$

$$\frac{\partial q}{\partial x} = 0 \qquad x = A, B, \qquad t \geq 0 \tag{6.11}$$

where A and B are fixed. We derive the moving-mesh, finite element model for this system. We consider the system with no births or deaths, so we have a true conservation of mass. Over the domain $x \in [A, B]$,

$$\int_A^B u\,dx = \text{constant}. \tag{6.12}$$

We de ne the distributed conservation principle, for a weight function w_i,

$$\int_A^B w_i u dx = \text{constant} \tag{6.13}$$

hence

$$\frac{d}{dt}\int_A^B w_i u \, dx = 0. \tag{6.14}$$

By the Reynolds Transport Theorem, we can say that

$$\int_A^B \frac{\partial}{\partial t}(w_i u)\, dx + \int_A^B \frac{\partial}{\partial x}(x w_i u)\, dx = 0 \tag{6.15}$$

$$\int_A^B w_i \frac{\partial u}{\partial t} + u\frac{\partial w_i}{\partial t} + w_i \frac{\partial}{\partial x}(ux) + ux\frac{\partial w_i}{\partial x} \; dx = 0. \tag{6.16}$$

Assuming the weight functions w_i move with the domain

$$\frac{\partial w_i}{}$$

We make the velocity potential substitution $u = \frac{\partial f}{\partial x}$. We obtain

$$-\int_A^B u\frac{\partial w_i}{\partial x}\frac{\partial f}{\partial x}dx = -\left[dw_i\frac{\partial u}{\partial x}\right]_A^B + \int_A^B d\frac{\partial w_i}{\partial x}\frac{\partial u}{\partial x}\,dx + \left[w_i u\frac{\partial q}{\partial x}\right]_A^B - \int_A^B u\frac{\partial q}{\partial x}\frac{\partial w_i}{\partial x}dx. \tag{6.22}$$

We note the presence of the zero flux boundary conditions (6.10), hence the two non-integral terms on the right hand side will be equal to zero. The equation to be solved for f is then

$$-\int_A^B u\frac{\partial w_i}{\partial x}\frac{\partial f}{\partial x}dx = \int_A^B d\frac{\partial w_i}{\partial x}\frac{\partial u}{\partial x}\,dx - \int_A^B u\frac{\partial q}{\partial x}\frac{\partial w_i}{\partial x}dx. \tag{6.23}$$

This expression requires known q. We return to the definition (6.8). We write this in weak form,

$$\int_A^B w_i E(u)\,dx = -e\int_A^B w_i\frac{\partial^2 q}{\partial x^2}dx + \int_A^B w_i q\,dx. \tag{6.24}$$

Integrating by parts on the right hand side, we obtain

$$\int_A^B w_i E(u)\,dx = -e\left[w_i\frac{\partial q}{\partial x}\right]_A^B + e\int_A^B \frac{\partial w_i}{\partial x}\frac{\partial q}{\partial x}dx + \int_A^B w_i q\,dx. \tag{6.25}$$

Noting again the zero flux boundary condition on q, we may simplify this to

$$\int_A^B w_i E(u)\,dx = e\int_A^B \frac{\partial w_i}{\partial x}\frac{\partial q}{\partial x}dx + \int_A^B w_i q\,dx \tag{6.26}$$

which may be solved for q.

6.1.2 Construction of the finite element form

We use the unmodified piecewise linear basis functions $w_i = W_i$, since we have only Neumann conditions to consider. We define our finite element variables U, Q and E in terms of the piecewise linear approximations $u = \sum_j W_j U_j$, $Q = \sum_j W_j Q_j$, $E = \sum_j W_j E_j$. Note that although E is itself a nonlinear function of U_j, we simply calculat(we)-31 Tf isd [(E)]TJ/F101 11

In terms of the standard mass and stiffness matrices this is

$$\epsilon K\underline{Q} + M\dot{\underline{Q}} = M\underline{E} \tag{6.28}$$

for vectors $\underline{Q}$ containing the values of Q_j, and $\underline{E}$ containing the values of E_j. To solve this we first obtain the values of $E(u)$ from equation (6.9). We can now obtain $\underline{Q}$ from the steady state system

$$\underline{Q} = (\epsilon K + M)^{-1} M\underline{E}. \tag{6.29}$$

We now require f, which can be obtain from equation (6.23). We make the same piecewise linear approximations and, after substitution, obtain

$$\sum_{j=0}^{N+1} \int_A^B U \frac{\partial W_i}{\partial x}\frac{\partial W_j}{\partial x}dx\ F_j = \sum_{j=0}^{N+1} \int_A^B d\frac{\partial W_i}{\partial x}\frac{\partial W_j}{\partial x}dx\ U_j - \sum_{j=0}^{N+1} \int_A^B U \frac{\partial W_i}{\partial x}\frac{\partial W_j}{\partial x}dx\ Q_j. \tag{6.30}$$

We solve for the vector $F = \{F_j\}$ using the matrix form

$$K(\underline{U})\underline{F} = dK\underline{U} - K(\underline{U})\underline{Q} \tag{6.31}$$

with $K(\underline{U})$ analogous to the stiffness matrix, and given by

$$K(\underline{U})_{ij} = \int_{x_{i-1}}^{x_{i+1}} U \frac{\partial W_i}{\partial x}\frac{\partial W_j}{\partial x}\,dx. \tag{6.32}$$

Once $\underline{F}$ is recovered, we obtain $\dot{X}$ and $\dot{U}$ in the manner now standard in this thesis. Briefly, we use the weak form of the definition for the velocity potential

where $\underline{X}$ and $\underline{F}_j$ are the vectors containing the unknown velocities X_{jk} and the known F_j, and M and B are the symmetric mass matrix and an asymmetric matrix respectively, as de ned in section 3.1.2.

In this way, (6.35) can be solved to obtain the X_j values. We then perform the time integration step using any chosen scheme. Once the grid position has been recalculated, the basis functions are likewise moved and the matrices de ned by them are recalculated.

We recover $\underline{U}$ from the conservation principle (6.13)

$$\int_A^B w_i$$

6.1.3 2D population clustering model for a single species

For the 2D model, we remind ourselves of the driving PDE system. This is

$$\frac{\partial u}{\partial t} = d\nabla^2 u - \nabla\cdot(u\nabla q) \tag{6.40}$$

$$E(u) = -e\nabla^2 q + q \tag{6.41}$$

$$E(u) = (u-a)(1-u) \tag{6.42}$$

with boundary conditions on the fixed boundary S

$$\hat{n}\cdot\nabla u = 0 \qquad x \in S, t \geq 0, \tag{6.43}$$

$$\hat{n}\cdot n = 0 \qquad x \in S, t \geq 0. \tag{6.44}$$

In this single species system we consider the case where we have no births or deaths, so that clustering effects are most apparent even if transient. We therefore have a true conservation of mass. Over the domain $x \in W$ the conservation principle is

$$\int_W u\,dW = \text{constant} \tag{6.45}$$

We define the distributed form, for a weight function w_i,

$$\int_W w_i u\,dW = c_i \tag{6.46}$$

where the constant c_i is determined by the choice of w_i. Hence

$$\frac{d}{dt}\int_W w_i u\,dW = 0. \tag{6.47}$$

Using the Reynolds Transport Theorem, we can write

$$\int_W \frac{\partial}{\partial t}(w_i u)\,dW + \int_W \nabla x\cdot(x w_i u)\,dW = 0 \tag{6.48}$$

leading to

$$\int_W w_i\frac{\partial u}{\partial t} + u\frac{\partial w_i}{\partial t} + w_i\nabla\cdot(ux) + ux\cdot\nabla w_i\,dW = 0. \tag{6.49}$$

We assume that the weight functions w_i move with the domain, which gives

$$\frac{\partial w_i}{\partial t} + \mathbf{x} \cdot \nabla w_i = 0 \tag{6.50}$$

hence (6.49) is

$$\int_W w_i \nabla \cdot (u\mathbf{x}) \, dW = -\int_W w_i \frac{\partial u}{\partial t} \, dW. \tag{6.51}$$

We make a substitution from the driving PDE (6.40) to obtain

$$\int_W w_i \nabla \cdot (u\mathbf{x}) \, dW = -\int_W w_i (d\nabla^2 u - \nabla \cdot (u\nabla q)) \, dW \tag{6.52}$$

and after integration by parts we obtain

$$\begin{aligned}\int_S w_i u\mathbf{x} \cdot \hat{\mathbf{n}} \, dS - \int_W \nabla w_i \cdot (u\mathbf{x}) \, dW = \int_W d\nabla w_i \cdot \nabla u \, dW - \int_W \nabla w_i \cdot u\nabla q \, dW \\ + \int_S w_i u\nabla q \cdot \hat{\mathbf{n}} \, dS - \int_S d w_i \nabla u \cdot \hat{\mathbf{n}} \, dS\end{aligned} \tag{6.53}$$

or, because we have fixed boundaries,

$$\begin{aligned}-\int_W \nabla w_i \cdot (u\mathbf{x}) \, dW = \int_W d\nabla w_i \cdot \nabla u \, dW - \int_W \nabla w_i \cdot u\nabla q \, dW \\ + \int_S w_i u\nabla q \cdot \hat{\mathbf{n}} \, dS - \int_S d w_i \nabla u \cdot \hat{\mathbf{n}} \, dS.\end{aligned} \tag{6.54}$$

We make the velocity potential substitution $\mathbf{x} = \nabla f$. We obtain

$$\begin{aligned}-\int_W u\nabla w_i \cdot \nabla f \, dW = -\int_S d w_i \nabla u \cdot \hat{\mathbf{n}} \, dS + \int_W d\nabla w_i \cdot \nabla u \, dW \\ + \int_S w_i u\nabla q \cdot \hat{\mathbf{n}} \, dS - \int_W u\nabla q \cdot \nabla w_i \, dW\end{aligned} \tag{6.55}$$

subject to zero flux conditions on the boundary (6.43), so the two boundary terms on the right hand side will be equal to zero. We obtain the equation to be solved for f,

$$-\int_W u\nabla w_i \cdot \nabla f \, dW = \int_W d\nabla w_i \cdot \nabla u \, dW - \int_W u\nabla q \cdot \nabla w_i \, dW. \tag{6.56}$$

Before this can be solved we need a value of f and this can be obtained, in a similar way to the 1D case (6.13), the equation (6.13),

Writing this in weak form, we have

$$\int_W w_i E(u)\, dW = -e \int_W w_i \tilde{N}^2 q\, dW + \int_W w_i q\, dW. \tag{6.58}$$

We integrate the right hand side by parts to obtain

$$\int_W w_i E(u)\, dW = -e \int_S w_i \tilde{N} q \cdot \hat{n}\, dS + e \int_W \tilde{N} w_i \cdot \tilde{N} q\, dW + \int_W w_i q\, dW \tag{6.59}$$

since $\tilde{N} q \cdot \hat{n} = 0$ on S, the boundary term is equal to zero. We therefore have

$$\int_W w_i E(u)\, dW = e \int_W \tilde{N} w_i \cdot \tilde{N} q\, dW + \int_W w_i q\, dW. \tag{6.60}$$

This allows us to obtain q once $E(u)$ is known. We may obtain the values of $E(u)$ from equation (6.42).

6.1.4 Construction of the finite element form

In order to solve equations (6.56) and (6.60), we use the finite element method. We use the unmodified two dimensional triangular weight functions $w_i = W_i$ described in Chapter 3 (3.1.3), since we have no Dirichlet conditions to impose. We define our finite element variables Q and E in terms of the same basis functions, using the approximations $Q(x,t) = \sum_j W_j(x) Q_j(t)$ and $E(x,t) = \sum_j W_j(x) E_j(t)$. After making the substitutions for these approximations, (6.60) becomes

$$\sum_{j=1}^{N} \int_W W_i W_j\, dW\ E_j = \sum_{j=1}^{N} e \int_W \tilde{N} W_i \cdot \tilde{N} W_j\, dW\ Q_j + \sum_{j=1}^{N} \int_{W_i} W_i W_j\, dW\ Q_j. \tag{6.61}$$

In terms of mass and stiffness matrices M and K this is written as

$$eK\underline{Q} + M\underline{\dot{Q}} = M\underline{E} \tag{6.62}$$

for vectors $\underline{Q}$ containing Q_i and $\underline{E}$ containing E_i. Rearranging, we obtain $\underline{\dot{Q}}$ from the steady state system

$$\underline{\dot{Q}} = (eK + M)^{-1} M\underline{E} \tag{6.63}$$

where $\underline{E}$ is given by the definition (6.42). Similarly, we make substitutions into equation (6.56), defining our variables in terms of piecewise linear approximations based on the W_i. In addition to the approximation $Q(x,t) = \sum_j W_j(x) Q_j(t)$ already given, we require

$U(x;t) = \sum_j W_j(x)U_j(t)$ as the approximation for u, and $F(x;t) = \sum_j W_j(x)F_j(t)$ as the approximation for f. Equation (6.56) becomes

$$-\sum_{j=1}^{N}\int_W \nabla W_i \cdot \nabla W_j U \, dW \; F_j = \sum_{j=1}^{N} d \int_W \nabla W_i \cdot \nabla W_j \, dW \; U_j + \sum_{j=1}^{N}\int_W \nabla W_i \cdot (U\nabla W_j) dW \; Q_j \tag{6.64}$$

or in matrix form

$$K(\underline{U})\underline{F} = dK\underline{U} + K(\underline{U})\underline{Q} \tag{6.65}$$

where $K(\underline{U})$ is the weighted stiffness matrix given by (3.69). Equation (6.65) is now sufficient to recover F. We then calculate X, perform the time integration and lastly recover $\underline{U}$ from the conservation of mass equation. This process is described fully in Chapter 3 but briefly, the definition of f is

$$x = \nabla f \tag{6.66}$$

for which a weak form is

$$\int_W w_i x \, dW = \int_W w_i \nabla f \, dW. \tag{6.67}$$

Using again the piecewise linear approximations $w_i = W_i(x)$, $X(x;t) = \sum_{j=1}^{N} X_j(t)W_j(x)$ and $\nabla F(x;t) = \sum_{j=1}^{N} F_j(t)\nabla W_j(x)$ we obtain

$$\sum_{j=1}^{N}\int_W W_i W_j \, dW \; X_j = \sum_{j=1}^{N}\int_W W_i \nabla W_j \, dW \; F_j. \tag{6.68}$$

Hence in matrix form, (6.68) can be solved for X using

$$M\underline{X} = B\underline{F} \tag{6.69}$$

where $\underline{X} = \{X_i\}$, M is the symmetric mass matrix, and B is an asymmetric matrix with elements $B_{ij} = \int_W W_i \nabla W_j \, dW$. Having found X, the nodes are repositioned using the forward Euler scheme. We recover $\underline{U}$ using our distributed mass conservation principle (6.13). Using the piecewise linear W_i that together form a partition of unity, equation (6.13) is, for each node i,

$$c_i = \int_W W_i U \, dW.$$

Using the piecewise linear approximation $U(x,t) = \sum_j U_j(t)W_j(x)$ we obtain

$$\sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_j = c_i \tag{6.70}$$

which is equivalent to the mass matrix system

$$M\underline{U} = \underline{c}_i. \tag{6.71}$$

This equation is used to calculate the initial (and constant) values of c_i, using the initial values of U_j and X_j. After repositioning the nodes we may recover $U_j(t)$ from the mass matrix system (6.71).

Algorithm 13

The finite element solution of the single species aggregation model defined by equations (6.40), (6.41) and (6.42) on the moving mesh in 2-D therefore consists of the following steps. We obtain the constant values of c_i from (6.71) calculated at $t = 0$, and then for each time step:

1. Calculate the reproductive potential by solving equation (6.42) for $E(u)$;
2. Find the values of Q by solving equation (6.63);
3. Find the velocity potential by solving equation (6.65) for the $F_j(t)$ values;
4. Find the node velocity by solving equation (6.69) for $X(t)$ values;
5. Generate the co-ordinate system at the next time-step $t + dt$ by solving (3.18) using Euler's approximation;
6. Find the solution $U(t + dt)$ by solving the conservation equation (6.71).

6.1.5 Results

In common with [29], we use a random seeding to provide the initial conditions for the model. The random seeding is selected from a normal distribution with a mean of 0.3 and a standard deviation of 0.01. The model is stable and robust. We are able to run the model sometimes to a blow up and sometimes to a solution where population growth and decline become approximately balanced. The outcome depends on the initial values used, and also

on the parameters and d. We are familiar from the diffusion models with the parameter d and its effects. As the parameter controlling the rate of diffusion, it has a smoothing effect when large. The parameter ϵ is less familiar. From the definition contained within (

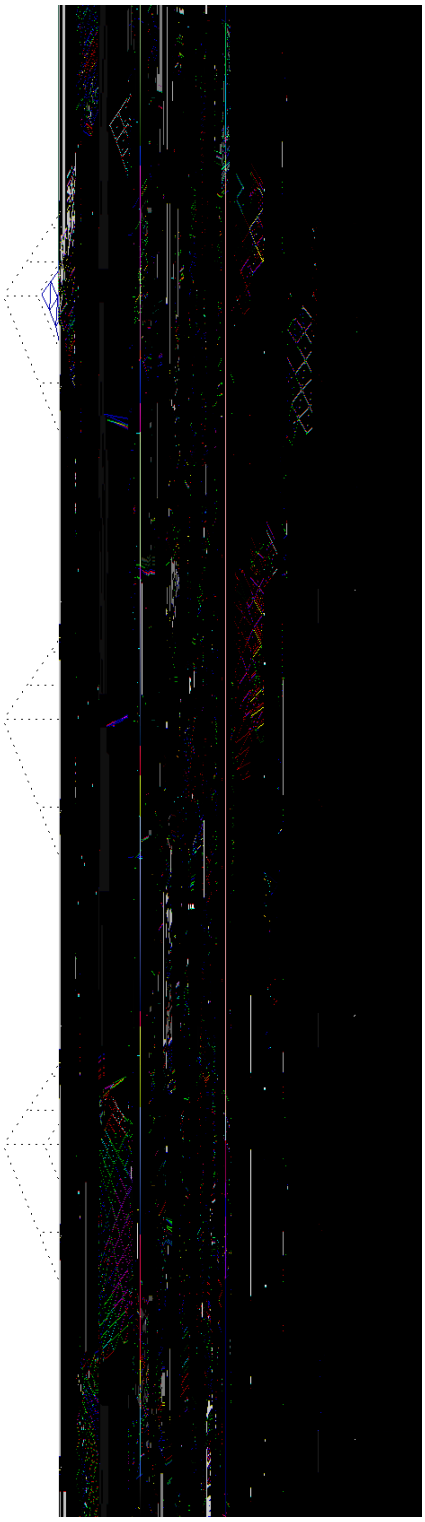


Fig. 6.1 A solution after 350 iterations at $t = 0.35$ of the 2D population equations, with $e = 0.005$ and $d = 0.01$. This solution has not yet reached a balance, but is approximating the 4th eigenmode of the Laplacian.

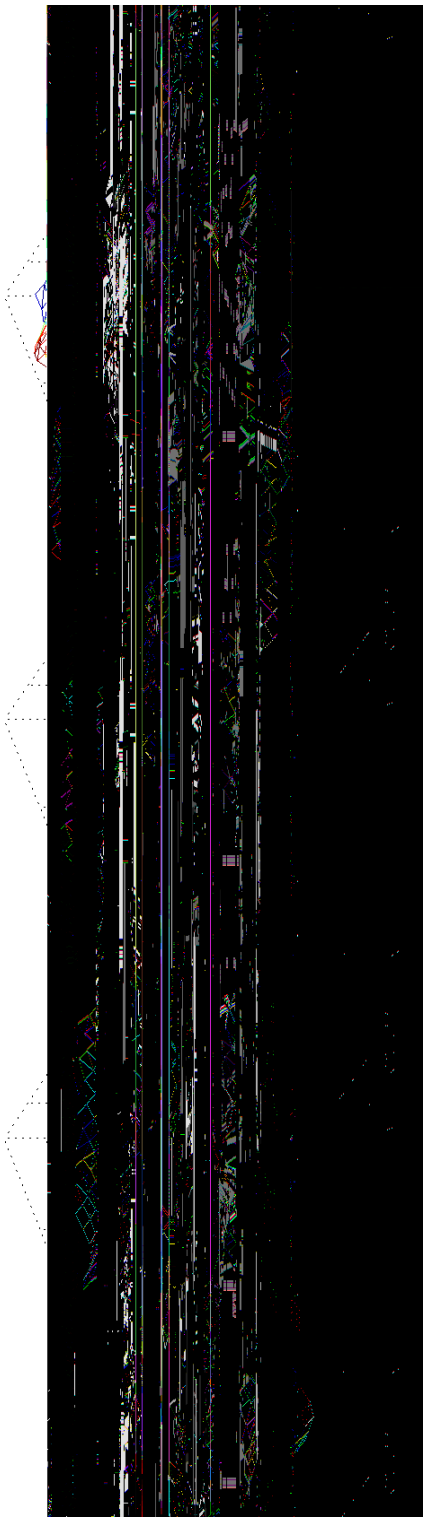


Fig. 6.2 A solution after 10 iterations at $t = 0.01$ of the 2D population equations, with $e = 0.001$ and $d = 0.01$. This solution has not yet reached a balance, but is approximating the 20th eigenmode of the Laplacian.

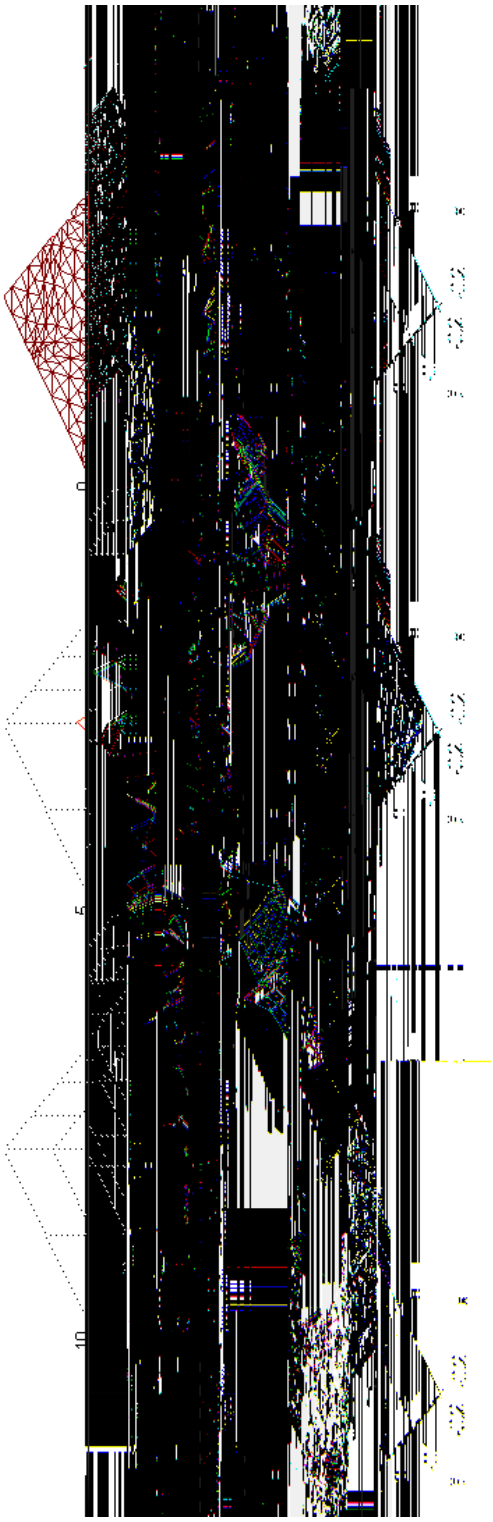


Fig. 6.3 An approximately balanced solution of the 2D population equations, with $\varepsilon = 0.001$ and $d = 0.01$, plotting (from left to right) u, q and $E(u)$. Whilst there is overcrowding in the centres of the clusters, giving a dramatically negative $E(u)$, the rate of population decline resulting from that is balanced by the attraction of the cluster to individuals nearby. These two effects mean that the shape of the solution does not evolve further, with only minor local effects observed.

6.2 Population clustering models for two competitive species

We now consider reaction-aggregation-diffusion models with two species. We consider the case where the species share a domain, so that we may examine clustering into species specific groups and the resulting claiming of territory. This may be of use in informing suitable starting conditions for a two phase model of competition.

A finite element formulation for the fixed mesh case

We begin with the Lotka-Volterra competition equations of Chapter 2, section 2.2, for two competing species with population densities u_1 and u_2. Following [29

(6.78) and (6.76) we obtain

$$A \quad au_1 \quad bu_2 = e_1\tilde{N}^2 q_1 + q_1: \tag{6.86}$$

We write this in weak form, using a weight function w_i to give

$$\int_w w_i A \, dW$$

and after integration by parts we have

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = d_1 \int_S w_i \tilde{\nabla} u_1 \cdot \hat{n}\, dS - d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW - \int_W w_i u_1 \tilde{\nabla}^2 q_1\, dW - \int_W w_i \tilde{\nabla} q_1 \cdot \tilde{\nabla} u_1\, dW \tag{6.93}$$

the first term on the right hand side of which is equal to zero at the domain boundaries due to the zero flux boundary conditions (6.82), leaving

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW - \int_W w_i u_1 \tilde{\nabla}^2 q_1\, dW - \int_W w_i \tilde{\nabla} q_1 \cdot \tilde{\nabla} u_1\, dW. \tag{6.94}$$

Turning our attention to the second term on the right hand side, again we integrate by parts and obtain

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW - \int_S w_i u_1 \tilde{\nabla} q_1 \cdot \hat{n}\, dS + \int_W \tilde{\nabla}(w_i u_1) \cdot \tilde{\nabla} q_1\, dW - \int_W w_i \tilde{\nabla} q_1 \cdot \tilde{\nabla} u_1\, dW. \tag{6.95}$$

Again the boundary integral is zero from the boundary condition (6.82). Therefore we can reduce this to

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW + \int_W \tilde{\nabla}(w_i u_1) \cdot \tilde{\nabla} q_1\, dW - \int_W w_i \tilde{\nabla} q_1 \cdot \tilde{\nabla} u_1\, dW. \tag{6.96}$$

By the product rule, this becomes

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW + \int_W u_1 \tilde{\nabla} w_i \cdot \tilde{\nabla} q\, dW + \int_W w_i \tilde{\nabla} u_1 \cdot \tilde{\nabla} q_1\, dW - \int_W w_i \tilde{\nabla} q_1 \cdot \tilde{\nabla} u_1\, dW \tag{6.97}$$

which simplifies to

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1\, dW + \int_W u_1 \tilde{\nabla} w_i \cdot \tilde{\nabla} q_1\, dW. \tag{6.98}$$

Equation (6.98) determines $\frac{\partial u_1}{\partial t}$ in terms of q_1 and u_1, and is ready for finite element substitutions to be made. We follow the same process for $\partial u_2/\partial t$ as we did for $\partial u_1/\partial t$. Equation

(6.85), the driving PDE for $\partial u_2/\partial t$, is

$$\frac{\partial u_2}{\partial t} = d_2 \nabla^2 u_2 - m \nabla \cdot (u_2 \nabla q_2) \tag{6.99}$$

or

$$\frac{\partial u_2}{\partial t} = d_2 \nabla^2 u_2 - m u_2 \nabla^2 q_2 - m \nabla q_2 \cdot \nabla u_2. \tag{6.100}$$

In weak form we rewrite this as

$$\int_W w_i \frac{\partial u_2}{\partial t}\, dW = d_2 \int_W w_i \nabla^2 u_2\, dW - m \int_W w_i u_2 \nabla^2 q_2\, dW - m \int_W w_i \nabla q_2 \cdot \nabla u_2\, dW. \tag{6.101}$$

The process that follows is identical to the

We substitute these approximations into (6.89) and obtain

$$\int_W W_i A \, dW - a \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{1_j} - b \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{2_j}$$
$$= -e_1 \sum_{j=1}^{N} \int_W \tilde{N} W_i \cdot \tilde{N} W_j \, dW \; Q_{1_j} + \sum_{j=1}^{N} \int_W W_i W_j \, dW \; Q_{1_j}. \qquad (6.107)$$

We may write (6.107) in terms of our mass and stiffness matrices M and K to obtain

$$M\underline{A} - aM\underline{U}_1 - bM\underline{U}_2 = -e_1 K\underline{Q}_1 + M\underline{Q}_1. \qquad (6.108)$$

Here $\underline{A}$ is a vector with all entries equal to A. We may rewrite this in terms of $\underline{Q}_1$

$$\underline{Q}_1 = (-e_1 K + M)^{-1} M(\underline{A} - a\underline{U}_1 - b\underline{U}_2). \qquad (6.109)$$

In exactly the same manner, we substitute the approximations (6.103) to (6.106) into (6.90) and obtain

$$\int_W W_i B \, dW - a \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{1_j} - b \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{2_j}$$
$$= -e_2 \sum_{j=1}^{N} \int_W \tilde{N} W_i \cdot \tilde{N} W_j \, dW \; Q_{2_j} + \sum_{j=1}^{N} \int_W W_i W_j \, dW \; Q_{2_j} \qquad (6.110)$$

which is, in matrix form

$$\underline{Q}_2 = (-e_2 K + M)^{-1} M(\underline{B} - a\,\underline{U}_1 - b\,\underline{U}_2). \qquad (6.111)$$

Here $\underline{B}$ is a vector with all entries equal to B. Equations (6.109) and (6.111) can be solved to obtain $\underline{Q}_1$ and $\underline{Q}_2$.

We now tackle the solution of $\frac{\partial U_1}{\partial t}$ by constructing equation (6.98) in finite element form. We again choose $w_i = W_i$ and make substitutions for the approximations (6.103) and (6.104), together with the derivatives

$$\frac{\partial U_1}{\partial t} = \dot{U}_1 = \sum_{j=1}^{N} W_j \dot{U}_{1_j} \qquad (6.112)$$

$$\frac{\partial U_2}{\partial t} = \dot{U}_2 = \sum_{j=1}^{N} W_j \dot{U}_{2_j}. \qquad (6.113)$$

Thus (6.98) can be rewritten in the form

$$\sum_{j=1}^{N} \int_W \Psi_i \Psi_j \, dW \; U_{1_j} = -d_1 \sum_{j=1}^{N} \int_W \tilde{\nabla}\Psi_i \cdot \tilde{\nabla}\Psi_j \, dW \; U_{1_j} + \sum_{j=1}^{N} \int_W U_1 \tilde{\nabla}\Psi_i \cdot \tilde{\nabla}\Psi_j \, dW \; Q_{1_j}. \tag{6.114}$$

In matrix form this is

$$M\underline{U}_1 = -d_1 K\underline{U}_1 + K(\underline{U}_1)\underline{Q}_1 \tag{6.115}$$

where the entries of matrix $K(\underline{U}_1)$

6.2.3 Results

The model has been encoded on the square domain de ned by $0:2 \leq x \leq 0:2$, $0:2 \leq y \leq 0:2$. We set 21 by 21 regularly spaced nodes, creating a mesh containing 512 triangular elements. The default variables used for the simulations are (from [29]):

$$A = 1$$
$$B = 1:5$$
$$a = 1$$
$$b = 2$$
$$a = 3$$
$$b = 1$$
$$e_1 = 0:025$$
$$e_2 = 0:025$$
$$d_1 = 0:1$$
$$d_2 = 0:1$$
$$m = 1:$$

The initial population is generated randomly. Boundary nodes are set at $u_1 = 0:4$, $u_2 = 0:3$ for $t = 0; x \in \P W$. Internal nodes are assigned a random value for u_1 from a set with a mean of $0:4$ and a standard deviation of $0:1$. For u_2 the random values are assigned from a set with a mean of $0:3$ and a standard deviation of $0:1$. These values are chosen so that E_1 and E_2 are zero (neutral survivability) at the boundaries, and have small perturbations from neutral elsewhere. These random perturbations seed the evolution of the population densities towards preferred locations. One such set of preferred locations is shown in gure 6.4. At $t = 0:7$, the two species have separated almost completely in space and four clusters are formed, each species inhabiting two corners of the domain with one favoured corner each. In gure 6.5, the same simulation is run to $t = 0:7$ with a different random initial seeding and this time only two larger clusters are formed. For the parameters used in this initial simulation, all the outputs fall broadly into one of these two categories. By experimenting with parameters, we are able to affect the number and size of clusters that are formed. Figure 6.6 shows a simulation at $t = 0:7$ with $e_1 = e_2 = 0:01$. The clusters produced are more compact and the four corners of the domain are more evenly populated. Running further simulations

with these parameters always produces this four-corner pattern. However, the question of which species inhabit which diagonal pair of corners is determined by the random seeding.

The simulations run smoothly with an initial diffusion dominated phase lasting to approximately $t = 0.05$, whilst groupings are established and peak population densities are reduced, then a much longer group growth stage where populations tend towards their groups containing maximum sustainable density, $u_1 = 1$ and $u_2 = 1.5$. Between groups populations are approximately zero, and the habitat appropriated by each species is clearly defined. Steady state is reached at around $t = 0.7$. These simulations are robust to significant experimentation with parameters and so provide a useful tool for understanding the behaviour described by the model.

6.2.4 The non-conservative population case

We now consider the evolution of a system that allows births and deaths to take place. This is non mass conserving so the treatment is slightly different. The equations for q_1, q_2, E_1 and E_2, and the boundary conditions are unchanged from the conservative case. These are given as

$$\epsilon_1 \tilde{N}^2 q_1 + q_1 = E_1 \tag{6.118}$$

$$\epsilon_2 \tilde{N}^2 q_2 + q_2 = E_2 \tag{6.119}$$

$$E_1 = A - a u_1 - b u_2 \tag{6.120}$$

$$E_2 = B - a\, u_1 - b\, u_2 \tag{6.121}$$

with boundary conditions

$$\tilde{N}u \cdot \hat{n} = 0 \qquad x \in \partial W, t \geq 0 \tag{6.122}$$

$$\tilde{N}q \cdot \hat{n} = 0 \qquad x \in \partial W, t \geq 0. \tag{6.123}$$

However, for the time dependent PDEs we have a different system. We set the reproduction parameter $r = 1$, so that (6.80) and (6.81) become

$$\frac{\partial u_1}{\partial t} = d_1 \tilde{N}^2 u_1 - \tilde{N} \cdot (u_1 \tilde{N} q_1) + u_1 E_1 \tag{6.124}$$

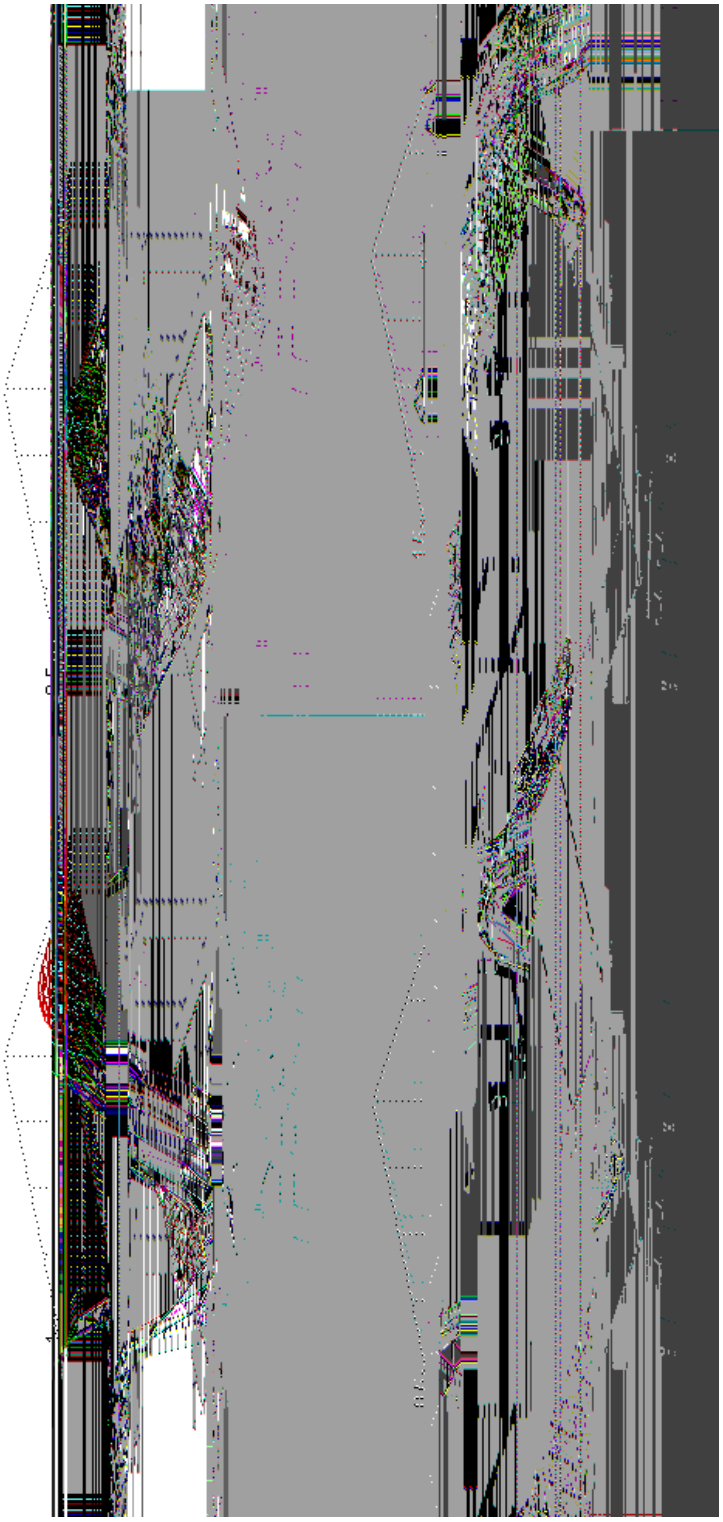


Fig. 6.4 A conservative, static mesh, two species simulation at $t = 0.7$ with $e_1 = e_2 = 0.025$ and $d_1 = d_2 = 0.1$. Initial seeding is random, so no two results are identical.

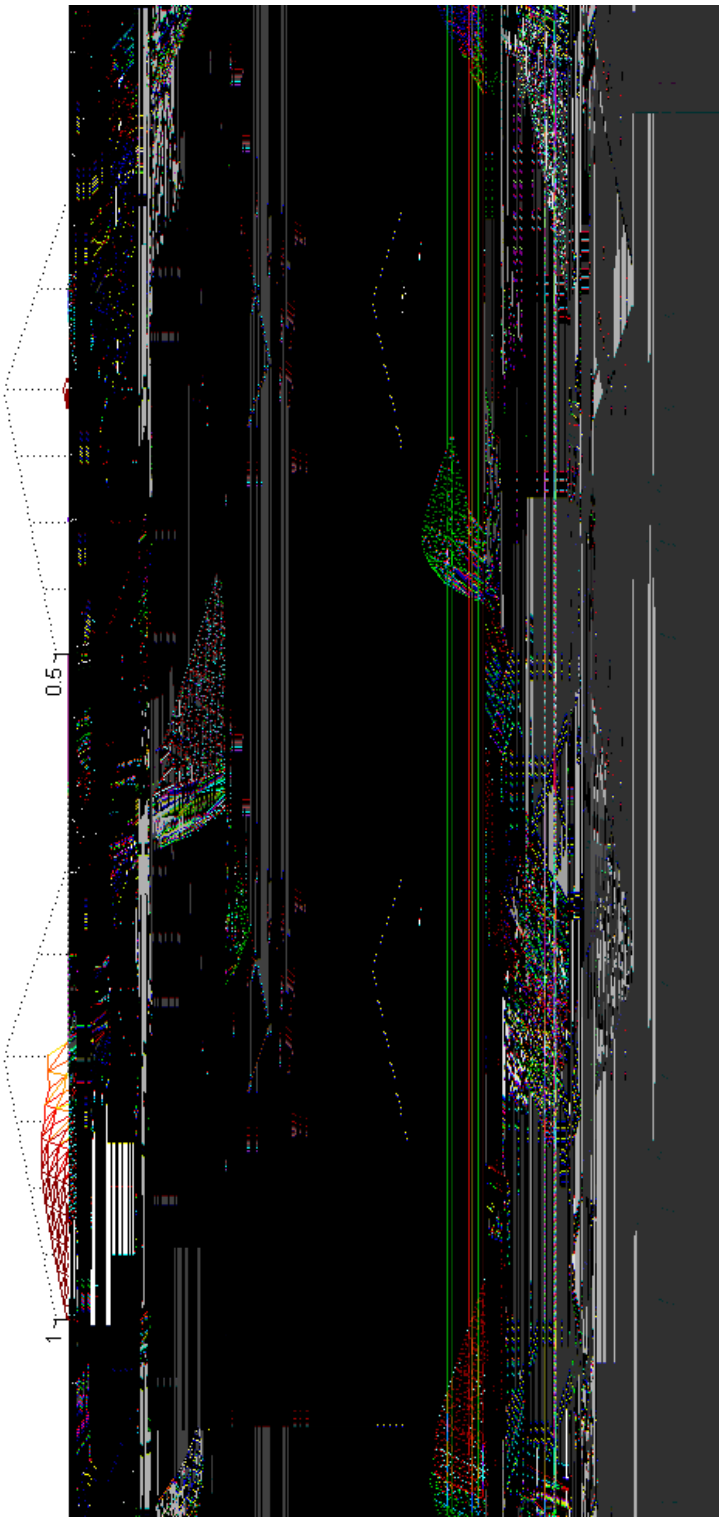


Fig. 6.5 An alternative result from the conservative, static mesh, two species simulation t = 0:7, the only difference being in the initial random population seeding. The parameters are identical to those for gure 6.4: $e_1 = e_2 = 0.025$ and $d_1 = d_2 = 0.1$.

$$\frac{\partial u_2}{\partial t} = d_2 \nabla^2 u_2 - m \nabla \cdot (u_2 \nabla q_2) + u_2 E_2. \tag{6.125}$$

This means that we have an extra term to consider in constructing the appropriate weak form for nite element substitutions. The weak form of (6.124) is

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = d_1 \int_W w_i \nabla^2 u_1\, dW - \int_W w_i u_1 \nabla^2 q_1\, dW - \int_W w_i \nabla q_1 \cdot \nabla u_1\, dW + \int_W w_i u_1 E_1\, dW \tag{6.126}$$

where w_i is part of a set of weight functions that together form a partition of unity. We treat the rst three terms on the right hand side in the same manner as in the conservative case, given by equations (6.92) to (6.98). We obtain the simpler weak form

$$\int_W w_i \frac{\partial u_1}{\partial t}\, dW = -d_1 \int_W \nabla w_i \cdot \nabla u_1\, dW + \int_W u_1 \nabla w_i \cdot \nabla q_1\, dW + \int_W w_i u_1 E_1\, dW \tag{6.127}$$

and similarly, from (6.125) we obtain the weak form

$$\int_W w_i \frac{\partial u_2}{\partial t}\, dW = -d_2 \int_W \nabla w_i \cdot \nabla u_2\, dW + \int_W m u_2 \nabla w_i \cdot \nabla q_2\, dW + \int_W w_i u_2 E_2\, dW. \tag{6.128}$$

6.2.5 Construction of the nite element form

We choose the piecewise linear weight functions $w_i = W_i$. We use the piecewise linear

The matrix form is then

$$M\underline{U}_1 = -d_1 K\underline{U}_1 + K(U_1)\underline{Q}_1 + \underline{N}_1 \tag{6.132}$$

where $\underline{N}_1 = {}^{R}$

$\underline{U}_1(t)$ and $\underline{U}_2(t)$ in forward Euler time integration.

6.2.6 Results

We use the same grid and default variables as in the conservative case, given in section 6.2.3. The initialisation is also unchanged from the conservative case. Boundary nodes are set at $u_1 = 0.4$, $u_2 = 0.3$ for $t = 0; x \in \Gamma$. Internal nodes are assigned a random value for u_1 at $t = 0$ from a set with a mean of 0.4 and a standard deviation of 0.01. For u_2 at $t = 0$ the random values are assigned from a set with a mean of 0.3 and a standard deviation of 0.01. Again the simulations run smoothly with the short initial diffusion dominated phase then the much longer group growth stage. Steady state, or at least a phase of very slow change, is reached at approximately $t = 0.7$ with no significant change thereafter to at least $t = 25$. The results shown here show a single simulation at different stages. We show progress of clusters forming at $t = 0.5$ (figure 6.7), smaller clusters becoming extinct at $t = 1.0$ (figure 6.8) and a straighter interface forming at $t = 1.5$ (figure 6.9). Compared to the conservative case, we see that only the larger groupings survive, which is to be expected if threatened populations are now allowed to suffer deaths. We also see the formation of a clear and increasingly straight interface between the two populations. As regards our aim of generating a system that truly tends towards a zero population species interface suitable for a spatially segregated multi-phase model, this is a success.

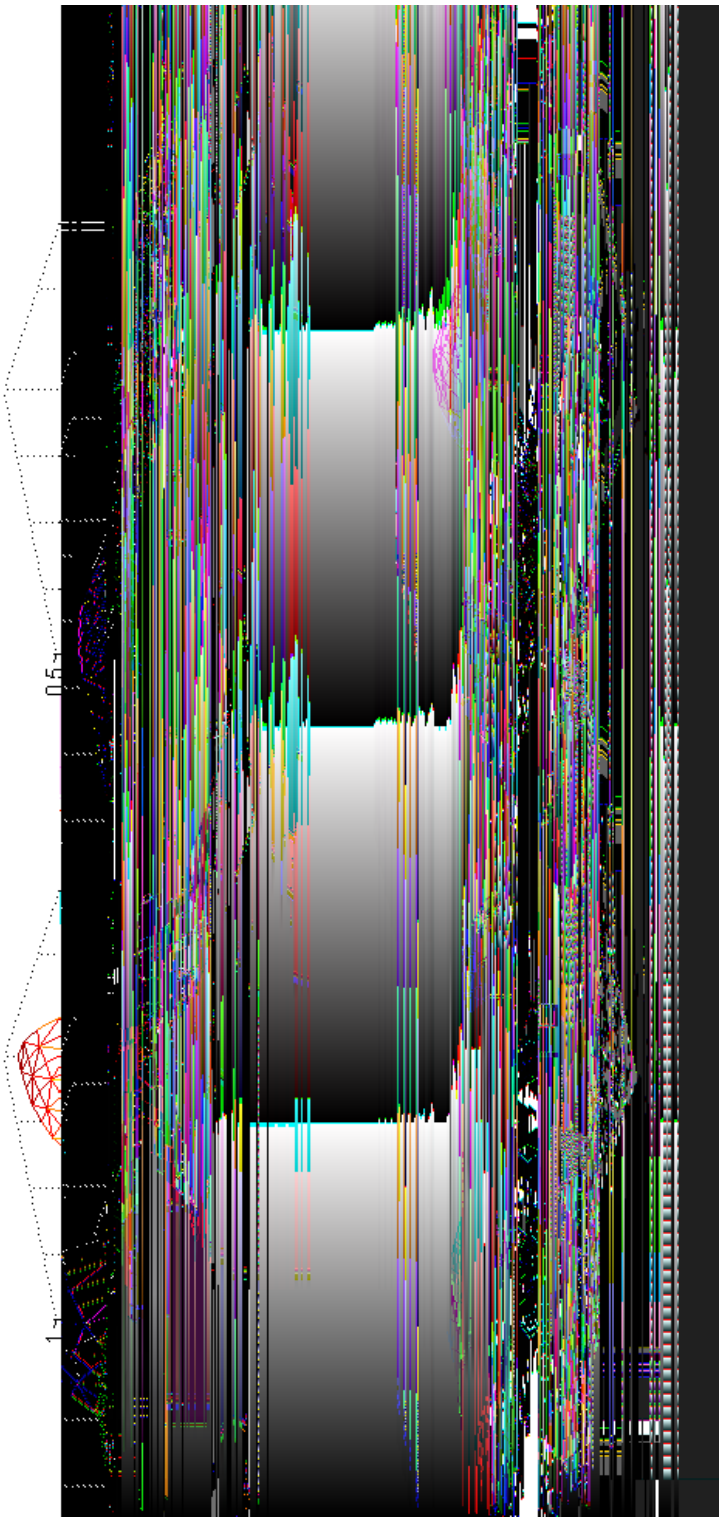


Fig. 6.7 An example result from the non-conservative static mesh at $t = 0.5$. Random seeding is used to produce the initial conditions. In this case two clusters of each species are formed.

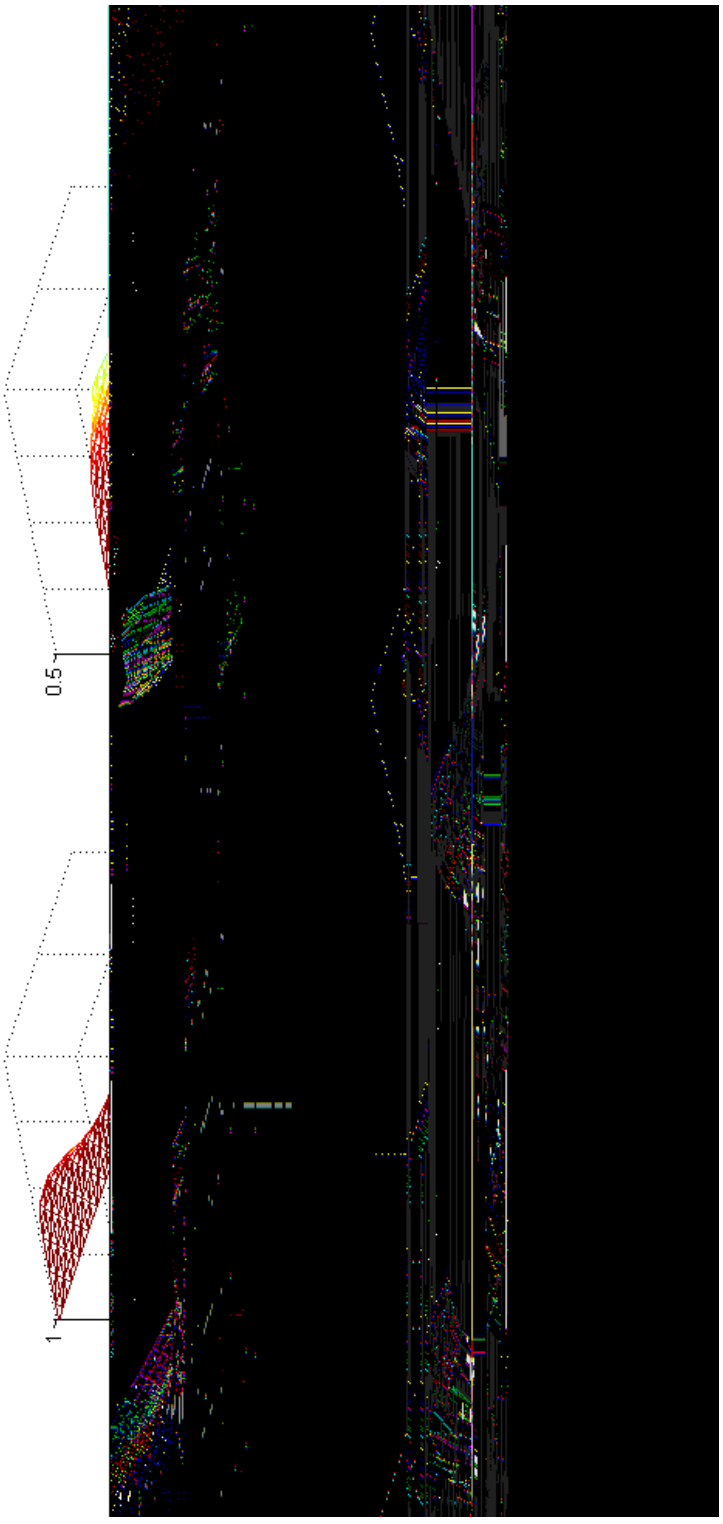


Fig. 6.8 An example result from the non-conservative static mesh at $t = 1.0$. Random seeding is used to produce the initial conditions. As the reproductive terms make impact, the number of clusters is reduced to one per species.

6.2.7 A change in the resource space

An interesting consideration is how changing the resource space affects the dynamics of the group. This is particularly relevant when we move on to restricting each species to its own domain. We can see how the shape of the interface will come into play, as well as our later look at the effect of the interface as it is moving. The variables A_{max} and B_{max} are the carrying capacities for species 1 and species 2 respectively, and can be considered to represent the maximum resource a species can access. In this simulation, we look at the effect of removing the resource from a part of the domain after a period of time during which groupings have become established. We allow the simulation to run as normal to $t = 10$ (gures 6.10 and 6.11), with the usual random population seeding, and then we reduce A_{max} and B_{max} to zero in one quadrant of the domain (gures 6.12 and 6.13). After the removal of

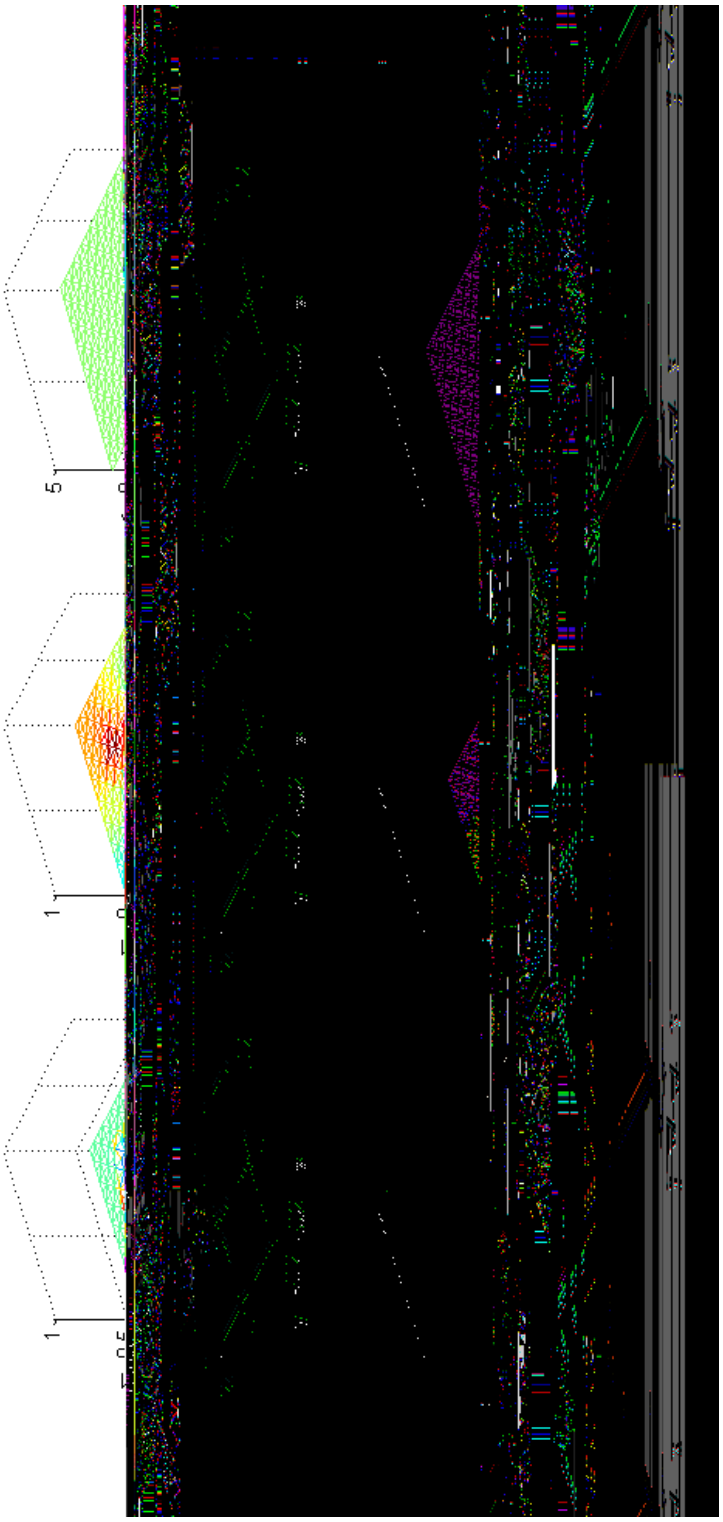


Fig. 6.10 An example with changing resource space, showing random population seeding at t = 0:0. At this stage, resource distribution is homogenous.

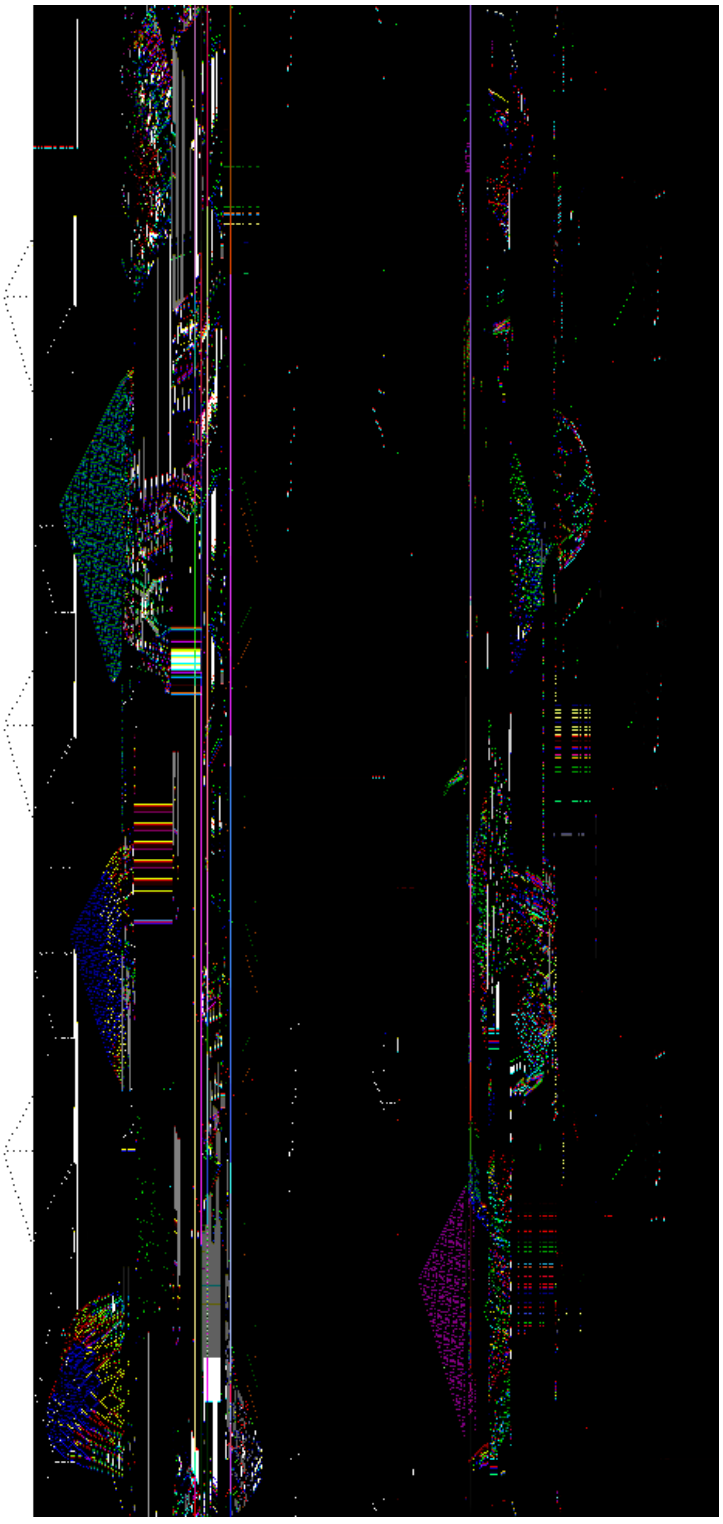


Fig. 6.11 An example with changing resource space, showing random population seeding at t = 1:0. At this stage, resource distribution is still homogenous.

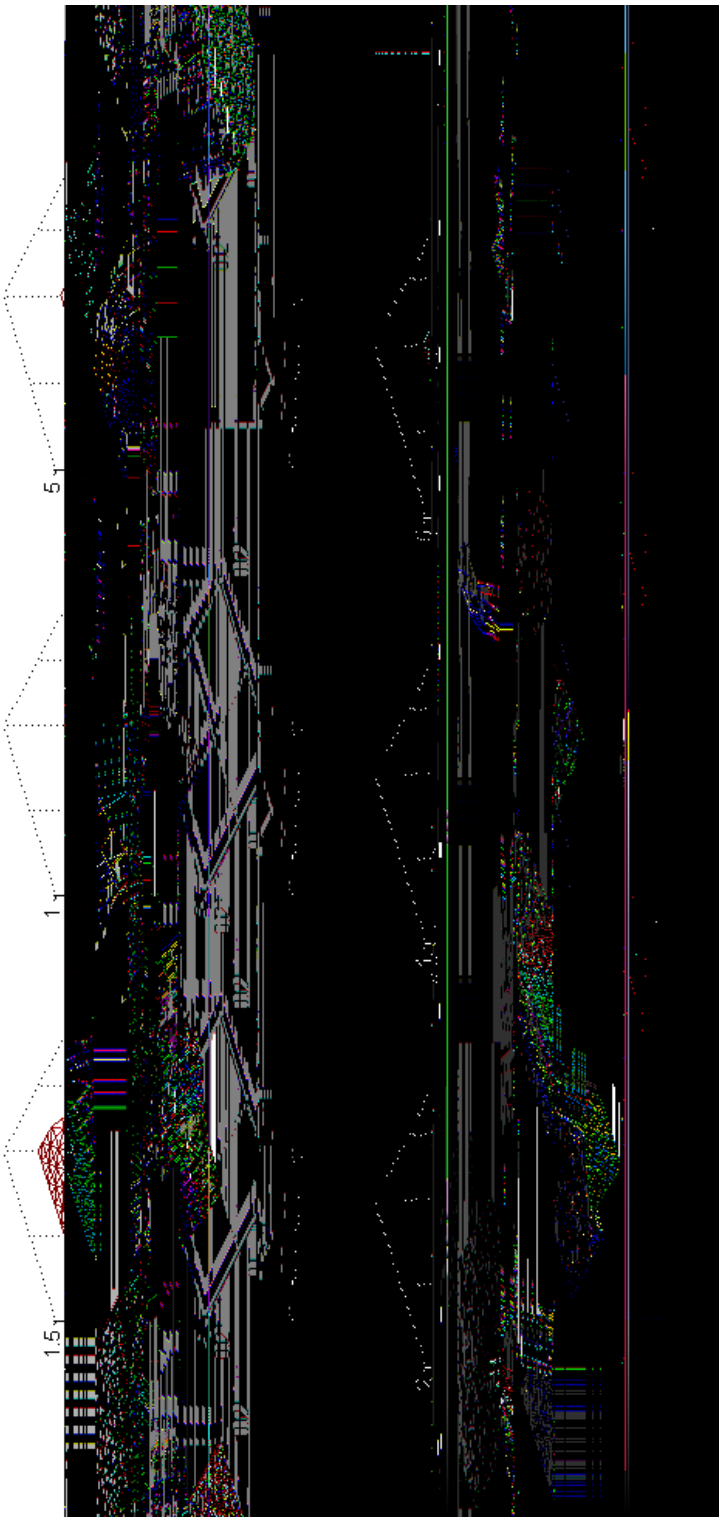


Fig. 6.12 An example with changing resource space, showing random population seeding at t = 1:5. At this stage, resource distribution is non-homogenous and species 2 is subject to a falling population.

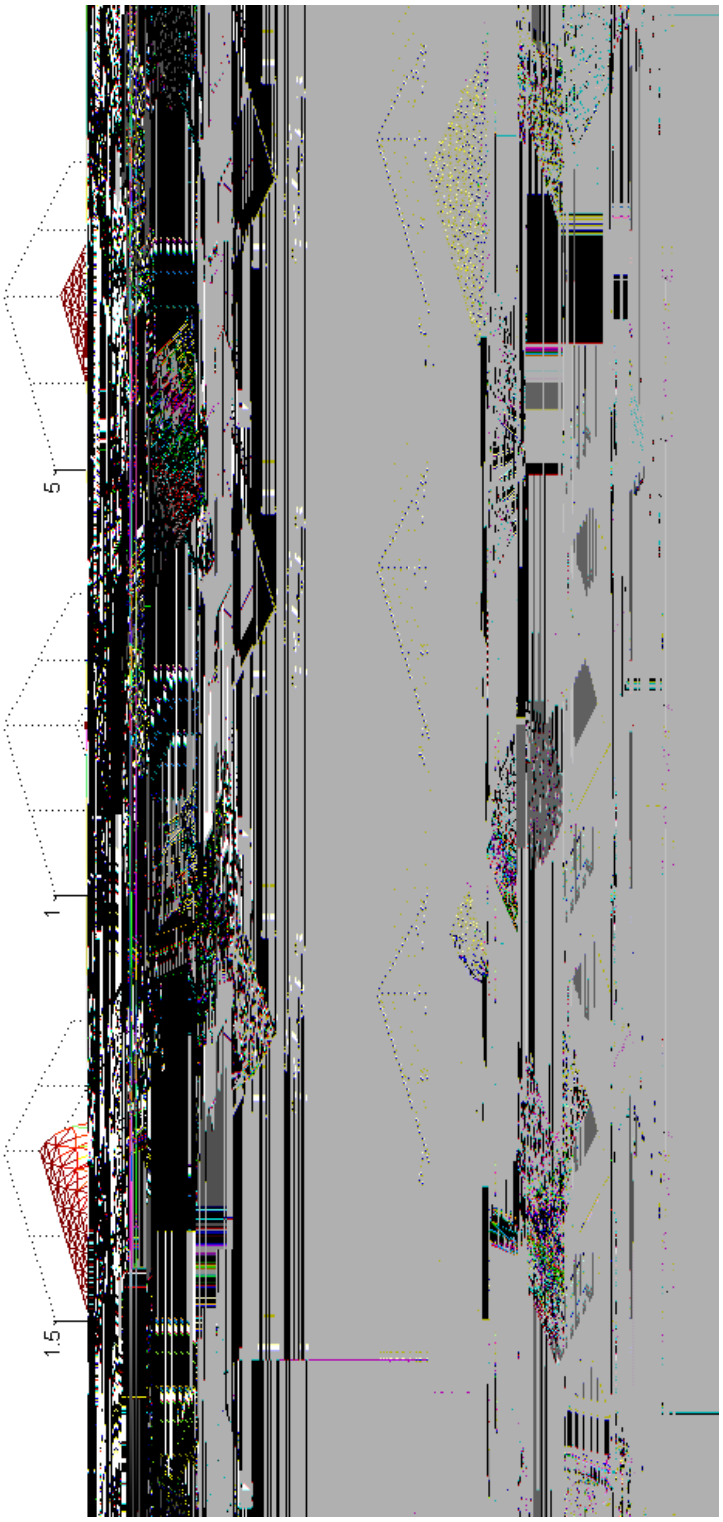


Fig. 6.13 An example with changing resource space, showing random population seeding at t = 2:0. At this stage, resource distribution is non-homogenous, but species 2 has adapted to a new domain and is forming a smaller cluster.

Chapter 7

A combined model with a moving interface

7.1 The two phase model of competition-diffusion-aggregation

We propose a model for a two component reaction-diffusion-aggregation system based on the Lotka-Volterra competition system, which will additionally incorporate the aggregation characteristics proposed by Grindrod [29] and the interface condition proposed by Hilhorst [31]. We construct the model in such a way that we will be able to utilise the two phase MMFEM of Baines, Hubbard et al. [8], with an adapting mesh based on a relative conservation principle. The PDE system that defines the basis of the model is given by the reaction-diffusion-aggregation PDEs from [29]. See Chapter 6, section 6.2 for a more detailed background. In Chapter 6 we derived a model based upon the same PDEs for two species sharing a domain, but here we are concerned with a truly two phase model. The driving PDEs are

$$\frac{\partial u_1}{\partial t} = d_1 \nabla^2 u_1 - \nabla (u_1 \nabla q_1) + r u_1 E_1 \qquad t > 0; x \in \Omega_1(t) \qquad (7.1)$$

and

$$\frac{\partial u_2}{\partial t} = d_2 \nabla^2 u_2 - r \nabla (u_2 \nabla q_2) + r u_2 E_2 \qquad t > 0; x \in \Omega_2(t). \qquad (7.2)$$

We use a fixed domain Ω bounded externally by Γ_e, but Ω is divided into two subdomain classes Ω_1 and Ω_2 which are separated by the moving interface $\Gamma_{int}(t)$. The 1-D analogies

are given by

$$\frac{\partial u_1}{\partial t} = d_1 \frac{\partial^2 u_1}{\partial x^2} - \frac{\partial}{\partial x}\left(u_1 \frac{\partial q_1}{\partial x}\right) + r u_1 E_1 \qquad t > 0; x \in (a; m(t)) \qquad (7.3)$$

and

$$\frac{\partial u_2}{\partial t} = d_2 \frac{\partial^2 u_2}{\partial x^2} - r \frac{\partial}{\partial x}\left(u_2 \frac{\partial q_2}{\partial x}\right) + r u_2 E_2 \qquad t > 0; x \in (m(t); b) \qquad (7.4)$$

for a domain with fixed boundaries a and b but with a moving interface between species $m(t)$. The parameters E_1 and E_2 are the net reproduction rates for each species, given by the logistic equations

$$E_1 = A - a u_1 - b u_2 \qquad (7.5)$$

$$E_2 = B - a\, u_1 - b\, u_2. \qquad (7.6)$$

We can see that this system also has parallels with the competition-diffusion model of Chapter 5. This system differs from that in Chapter 5 in the additional consideration of an aggregation component (the term containing r). We note that the parameters used in the expressions (7.5) and (7.6) for the reproduction rates are named differently to the competition diffusion model, but we can see that no material difference exists. For simplicity we adopt the naming conventions used by Hilhorst where we extend her work, and have followed the naming conventions used by Grindrod where we extend his work. This model

$$e_2 \frac{\P^2 q_2}{\P x^2} + q_2 = E_2: \tag{7.10}$$

7.2 1-D competition-aggregation-diffusion in a two phase model

We have boundary conditions given by

$$\begin{aligned} \frac{\P u}{\P x} &= 0 && x = a; b \\ \frac{\P q}{\P x} &= 0 && \\ u &= 0 && x = m(t) \end{aligned} \tag{7.11}$$

and we work in the high competition limit de ned by Hilhorst [31], so that the species cannot exist in the opposite species' domain. Formally,

$$\begin{aligned} u_1 &= 0 && x \, 2 \, [m(t); b] \\ u_2 &= 0 && x \, 2 \, [a; m(t)]: \end{aligned} \tag{7.12}$$

The interface condition is taken from [31], and is

$$m d_1 \frac{\P u_1}{\P x}_{m(t)} = -d_2 \frac{\P u_2}{\P x}_{m(t)} \tag{7.13}$$

where, once parameter naming conventions are compared between [31] and [29], $d_1 = d$, $a = b$. We will call m the interspecies competition rate. We work with Neumann boundary conditions on the external boundaries, which will be xed. We use parameter choices from [29] which are given in Chapter 6, section 6.2.3. In order to set suitable initial conditions, we consider the results of the shared-domain clustering models of Chapter 6. We note the steady state solutions that arise from the Chapter 6 models, and construct initial conditions that approximate those steady state results. These are given by gure 7.1. We begin by rede ning the driving Lotka-Volterra based equations (7.3) and (7.4) in weak form, incorporating the weight function w_i,

$$\int_a^b w_i \frac{\P u_1}{\P t} dx = \int_a^b d_1 w_i \frac{\P^2 u_1}{\P x^2} dx - \int_a^b w_i \frac{\P}{\P x} \left(u_1 \frac{\P q_1}{\P x} \right) dx + \int_a^b w_i r_1 u_1 E_1 \, dx \tag{7.14}$$

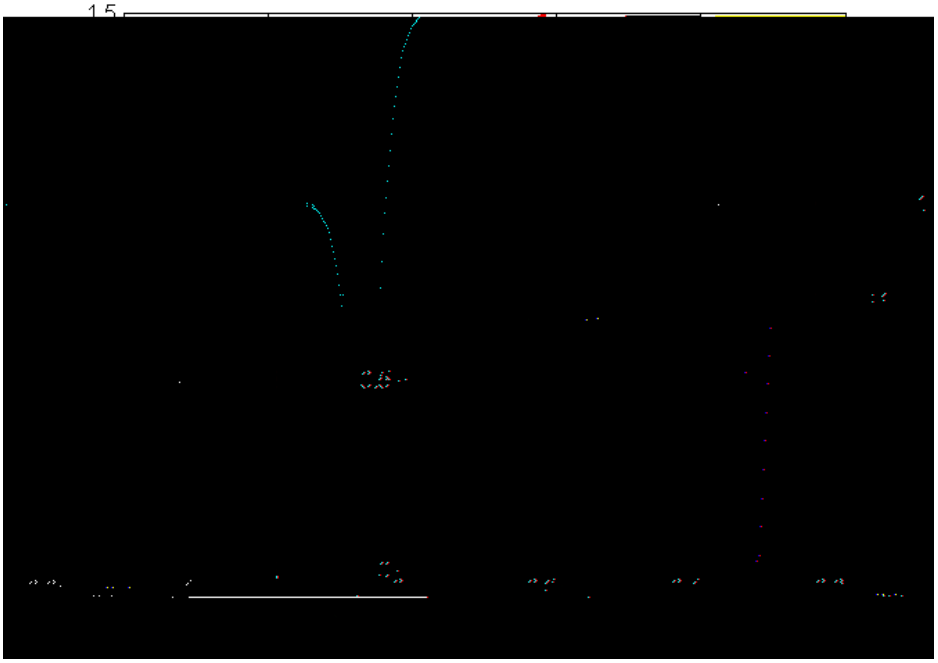


Fig. 7.1 Initial conditions for the two-phase reaction-diffusion-aggregation model. The amplitudes are taken from the steady state results arising from the shared domain model of Chapter 6. Species 1 in on the left and species 2 is on the right.

$$\int_a^b w_i \frac{\partial u_2}{\partial t}\,dx = \int_a^b d_1 w_i \frac{\partial^2 u_2}{\partial x^2}\,dx - \int_a^b w_i r \frac{\partial}{\partial x}\left(u_2 \frac{\partial q_2}{\partial x}\right) dx + \int_a^b w_i r_2 u_2 E_2\,dx. \quad (7.15)$$

We substitute the definitions for E_1 (7.5) and E_2 (7.6), noting that because we have the high competition limit, the terms containing b and a are equal to zero. We obtain

$$\int_a^{m(t)} w_i \frac{\partial u_1}{\partial t}\,dx = \int_a^{m(t)} d_1 w_i \frac{\partial^2 u_1}{\partial x^2}\,dx - \int_a^{m(t)} w_i \frac{\partial}{\partial x}\left(u_1 \frac{\partial q_1}{\partial x}\right) dx + \int_a^{m(t)} w_i r_1 u_1 (A - a u_1)\,dx \quad (7.16)$$

and

We impose the condition that the basis functions w_i move with the domain. Hence the basis functions also have velocity x. By analogy with advection, we write,

$$\frac{\partial w_i}{\partial t} + x\frac{\partial w_i}{\partial x} = 0 \tag{7.22}$$

hence

$$\frac{d}{dt}\int_{R(t)} w_i u\,dx = \int_{R(t)} w_i$$

$$c_{2_i} q_2(t) = \int_{m(t)}^{b} w_i u \, dx \tag{7.32}$$

Then equation (7.28) becomes, for species 1,

$$c_{1_i} q_1 + \int_a^{m(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx - \left[u_1 w_i \frac{\partial f}{\partial x} \right]_a^{m(t)} = \int_a^{m(t)} w_i \frac{\partial u_1}{\partial t} dx. \tag{7.33}$$

We now substitute (7.16). We obtain

$$\begin{aligned} c_{1_i} q_1 + \int_a^{m(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx - \left[u_1 w_i \frac{\partial f}{\partial x} \right]_a^{m(t)} &= \int_a^{m(t)} d_1 w_i \frac{\partial^2 u_1}{\partial x^2} dx \\ - \int_a^{m(t)} w_i \frac{\partial}{\partial x} \left(u_1 \frac{\partial q_1}{\partial x} \right) dx + \int_a^{m(t)} w_i r_1 u_1 (A - a u_1) \, dx. \end{aligned} \tag{7.34}$$

Integration by parts on the right leads to

$$\begin{aligned} c_{1_i} q_1 + \int_a^{m(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx - \left[u_1 w_i \frac{\partial f}{\partial x} \right]_a^{m(t)} &= - \int_a^{m(t)} d_1 \frac{\partial w_i}{\partial x} \frac{\partial u_1}{\partial x} dx + \left[w_i d_1 \frac{\partial u_1}{\partial x} \right]_a^{m(t)} \\ + \int_a^{m(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial q_1}{\partial x} dx - \left[w_i u_1 \frac{\partial q_1}{\partial x} \right]_a^{m(t)} + \int_a^{m(t)} w_i r_1 u_1 (A - a u_1) \, dx. \end{aligned} \tag{7.35}$$

We note the zero Neumann boundary conditions (7.11) on q_1 and u_1 at the external boundary, and the Dirichlet boundary condition (7.12) on u_1 at the interface. We also note the fixed external boundaries which mean that $\frac{\partial f}{\partial x} = 0$ on a. These conditions mean that most of the boundary terms in (7.35) are equal to zero. The remaining expression is

$$c_{1_i} q_1 + \int_a^{m(t)} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx = - \int_a^{m(t)} d_1 \frac{\partial w_i}{\partial x} \frac{\partial u_1}{\partial x} dx + \left[w_i d_1 \frac{\partial u_1}{\partial x} \right]_{m(t)}$$

After substitution of (7.17), equation (7.37) becomes

$$c_{2_i} q_2 + \int_{m(t)}^{b} u_2 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx - \left[u_2 w_i \frac{\partial f}{\partial x} \right]_{m(t)}^{b} = \int_{m(t)}^{b} d_1 w_i \frac{\partial^2 u_2}{\partial x^2} dx$$
$$- \int_{m(t)}^{b} w_i r \frac{\partial}{\partial x} \left[u_2 \frac{\partial q_2}{\partial x} \right] dx + \int_{m(t)}^{b} w_i r_2 u_2 (B - b\, u_1)\, dx. \qquad (7.38)$$

Integration by parts on the right leads to

$$c_{2_i} q_2 + \int_{m(t)}^{b} u_1 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx - \left[u_2 w_i \frac{\partial f}{\partial x} \right]_{m(t)}^{b} = - \int_{m(t)}^{b} d_2 \frac{\partial w_i}{\partial x} \frac{\partial u_2}{\partial x} dx + \left[w_i d_2 \frac{\partial u_2}{\partial x} \right]_{m(t)}^{b}$$
$$+ \int_{m(t)}^{b} r u_2 \frac{\partial w_i}{\partial x} \frac{\partial q_2}{\partial x} dx - \left[w_i r u_2 \frac{\partial q_2}{\partial x} \right]_{m(t)}^{b} + \int_{m(t)}^{b} w_i r_2 u_2 (B - b\, u_2)\, dx. \qquad (7.39)$$

After considering the boundary conditions (7.11) and (7.12) the remaining expression is

$$c_{2_i} q_2 + \int_{m(t)}^{b} u_2 \frac{\partial w_i}{\partial x} \frac{\partial f}{\partial x} dx = - \int_{m(t)}^{b} d_1 \frac{\partial w_i}{\partial x} \frac{\partial u_2}{\partial x} dx - \left[w_i d_2 \frac{\partial u_2}{\partial x} \right]_{m(t)}$$
$$+ \int_{m(t)}^{b} r u_2 \frac{\partial w_i}{\partial x} \frac{\partial q_2}{\partial x} dx + \int_{m(t)}^{b} w_i r_2 u_2 (B - b\, u_2)\, dx. \qquad (7.40)$$

We may solve (7.36-1.886 Td [(2)]TJ/F101 Td [(TJ/f.9552 Tf 5.305 9.408 Td [(w)]TJ/F87 254pC9

We have zero ux external boundary conditions (7.11), so the rst term on the right is equal to zero, leaving

$$\int_b^a w_i A\,dx - \int_b^a w_i a u_1\,dx - \int_b^a w_i b u_2\,dx = -e_1 \int_b^a \frac{\P w_i}{\P x}\frac{\P q_1}{\P x}dx + \int_b^a w_i q_1\,dx. \quad (7.44)$$

Equation (7.44) will give us q_1 in terms of u_1 and u_2. In exactly the same way, from (7.10) and (7.6) we obtain

$$\int_b^a w_i B\,dx - \int_b^a w_i a\,u_1\,dx - \int_b^a w_i b\,u_2\,dx = -e_2 \int_b^a \frac{\P w_i}{}$$

$$\int_{ja}^{} dx +$$

the interface node $x_{m(t)}$,

$$x^{N+1}_{m(t)} = \frac{(md_1u^N_{1_{m-1}}x^N_{m+1} + d_2u^N_{2_{m+1}}x^N_{m-1})}{(md_1u^N_{1_{m-1}} + d_2u^N_{2_{m+1}})}: \tag{7.50}$$

We use the finite differences approximation to calculate the interface velocity

$$x^{N+1}_{m(t)} = \frac{\frac{(md_1u^N_{1_{m-1}}x^N_{m+1}+d_2u^N_{2_{m+1}}x^N_{m-1})}{(md_1u^N_{1_{m-1}}+d_2u^N_{2_{m+1}})} - x^N_{m(t)}}{dt}: \tag{7.51}$$

This velocity can then be imposed on the interface when the velocity is recovered from We return to our definition of (7.26), now written in distributed form,

$$\int_{R(t)} w_i x dx = \int_{R(t)} w_i \frac{\P f}{\P x} dx: \tag{7.52}$$

This system of equations can be solved for For the interface itself, we calculate the new position by correcting the interface condition at the prior time step. We obtain the resultant interface velocity by solving equation (7.51) with $= 0$ imposed at the interface node. Having obtained, we move the domain using Euler integration. We also update and q_2 from q_1 (7.47) and q_2 (7.48) using the same time integration procedure. We may now recover. We determine the constant partial masses and c_{2_i} from (7.31) and (7.32) and the initial conditions. We obtain, for $= 0$

$$c_{1_i} = \frac{1}{q_1(0)} \int_a^{m(t)} w_i (266 f$$

andu_2 can be recovered from

$$\int_{m(t)}^{b} w_i(x;t)u_2(x;t)\,dx = c_{2_i}q_2(t). \tag{7.56}$$

In each case the Dirichlet condition that$u_2 = 0$ at the interface is strongly imposed, and the

$2 = \quad 2$

(A.1) to (A.4). We substitute those approximations into equation (7.44) and obtain

$$\int_a^b W_i A\,dx - a\sum_{j=0}^{N+1}\int_a^b W_i W_j\,dx\;U_{1_j} - b\sum_{j=0}^{N+1}\int_a^b W_i W_j\,dx\;U_{2_j}$$
$$= -e_1\sum_{j=0}^{N+1}\int_a^b \frac{\P W_i}{\P x}\frac{\P W_j}{\P x}dx\;Q_{1_j} + \sum_{j=0}^{N+1}\int_a^b W_i W_j dx\;Q_{1_j}: \qquad (7.57)$$

In terms of our mass and stiffness matrices M and K, equation (7.57) can be rewritten as

$$M\underline{A} - aM\underline{U}_1 - bM\underline{U}_2 = -e_1 K\underline{Q}_1 + M\underline{Q}_1: \qquad (7.58)$$

Here $\underline{A}$ is a vector with all entries equal to the resource parameter A

the following form for equation (7.36),

$$\tilde{c}_{1_i} q_1 + \sum_{j \in Z_1} \int_a^{m(t)} U_1 \frac{\partial \tilde{W}_i}{}$$

with the vector $\underline{\tilde{f}}_2$ containing entries $\tilde{f}_{2_i}$ given by

$$\tilde{f}_{2_i} = -\tilde{c}_{2_i} q_2 - \sum_{j \in Z_2} \int_{m(t)}^{b} d_2 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} dx \; U_{2_j} - \tilde{W}_i d_2 \frac{\partial U_2}{\partial x}\Big|_{m(t)}$$

$$+ \sum_{j \in Z_2} \int_{m(t)}^{b} r U_2 \frac{\partial \tilde{W}_i}{\partial x} \frac{\partial W_j}{\partial x} dx \; Q_{2_j} + \sum_{j \in Z_2} \int_{m(t)}^{b} \tilde{W}_i W_j r_2 B dx \; U_{2_j} \tag{7.68}$$

$$- \int_{m(t)}^{b} r_2 b \; \tilde{W}_i U_2^2 dx. \tag{7.69}$$

The nonlinear terms $\int_a^{m(t)} r_1 a \tilde{W}_i U_1^2 \, dx$ and $\int_{m(t)}^{b} r_2 b$

Obtaining the solution U_1 and U

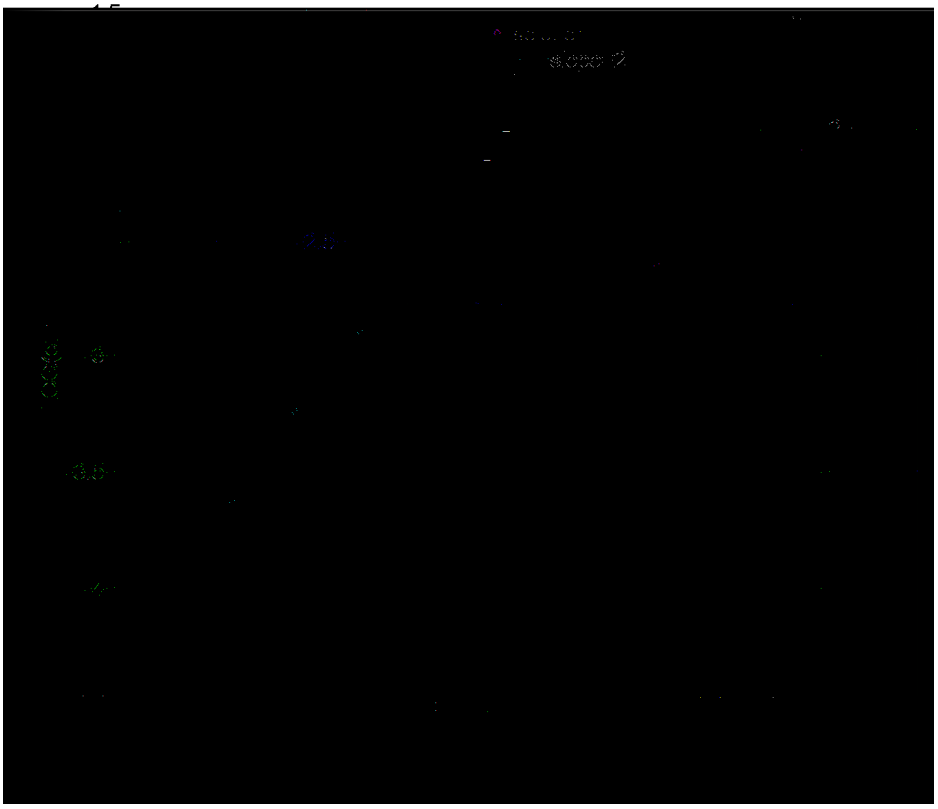


Fig. 7.2 Comparison of L^2 errors in the solution of algorithm 15. We observe an order of convergence of $p \approx 2$ in space, with time steps held constant at $\Delta t = 10^{-7}$.

7.2.2 Results

We find that the model is robust and the oscillations commonly found in finite element implementations, which are caused by the central differences approach, are minimal. Figure 7.2 shows convergence in the solution of approximately second order in space, Δx, and with time steps held constant at $\Delta t = 10^{-7}$. This estimate is obtained by comparison of the result generated by each grid spacing with a high-resolution (641 node) result, since no absolute result is available. This order of convergence is as reported for the similar method in [8].

We are able to observe all the varied effects of diffusion, logistic growth or decline and aggregation, and we are also able to generate sensible interface movement. We use the parameters from [29] in order to be confident that the choices are sensible. We are able to make comparisons between the aggregating and non aggregating two-phase models. Figure 7.3 shows a non-aggregating model (with the α values set to zero); this is exactly equiva-

lent to the competition diffusion model in Chapter 5 (5.2). With this choice of parameters, we observe no interface movement in the non aggregating model. The only development observed is in the shape of the solution near the interface, which is driven by diffusion. However, when we introduce aggregation, both species attempt to move away from the interface, resulting in a differently shaped solution (gure 7.6). We can see from gure 7.4 that the survivability index E_1 for species 1 is raised near the interface due to low population density, but then is very low in the domain occupied by species 2. We see in gure 7.5 how the q value takes a longer range average, so that despite the low population density near the interface, species 1 has an ideal velocity away from the interface. In gure 7.6 we observe that as both species vacate the area close to the interface, the changed interface dynamics favour species 2 and the interface moves to the left. Interestingly, in this particular scenario the increased 'intelligence' of the individuals does not help their longer term survival, because these additional movements cause mild overcrowding which offsets the reduced rate of competition at the interface. This suggests that the parameters given by [29] are potentially not the most representative, when this full model with the interface is constructed. With the large number of parameters at our disposal, the range of dynamics we could produce is limitless and very varied. We argue therefore that this model could be of real use to biologists in the eld studying any spatially segregated competition system.

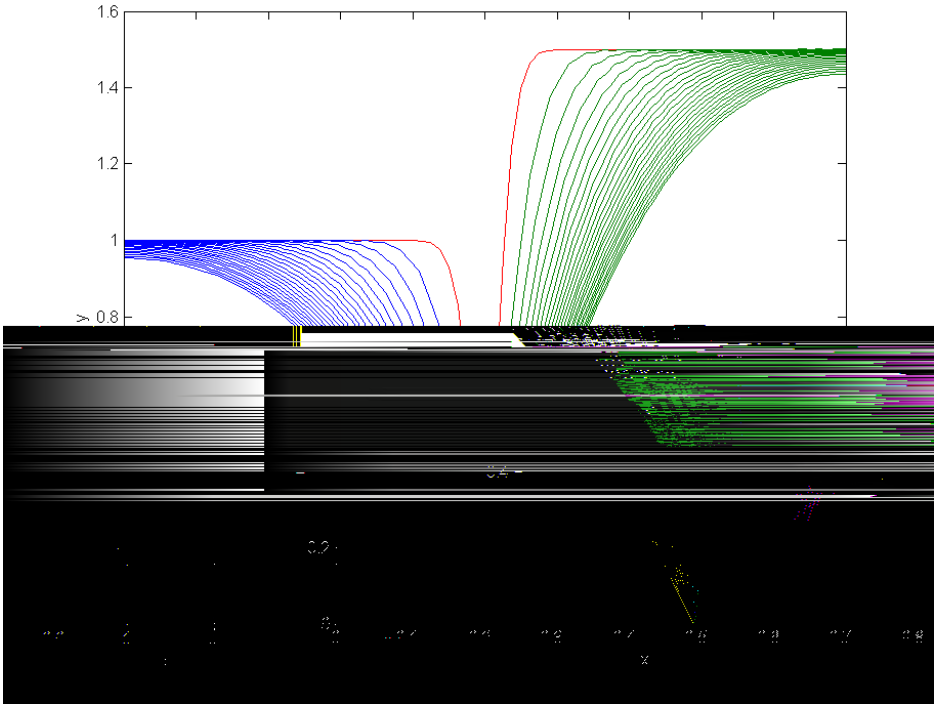


Fig. 7.3 The two phase competition model without aggregation, $n=a=0$. 24, using the parameters from Grindrod. Time steps are every 0.01s. We see stable population densities as the external boundaries, and an evolving shape to the interface.

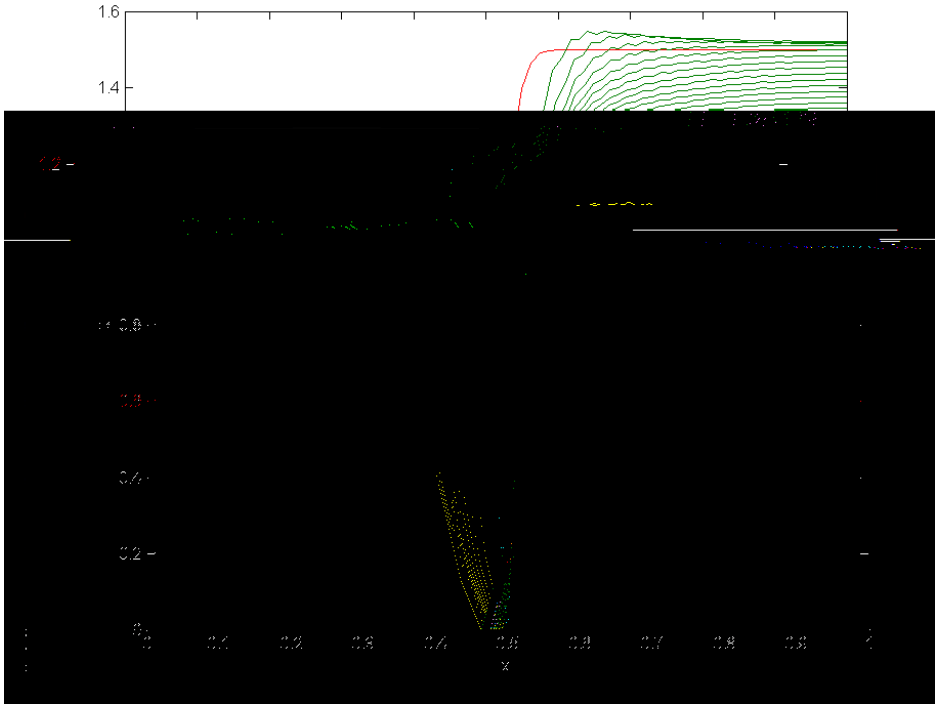


Fig. 7.6 Population decline in the two phase model with aggregation at $t = 0.16$. Time steps are every $0.01s$. We observe decreased movement towards the interface compared to the non-aggregating model. We see initially higher population densities a short distance away from the interface as the individuals resist moving towards it. The resulting overcrowding reduces overall survival rates, for this scenario.

7.3 2-D competition-aggregation-diffusion in a two phase model

We now consider the two dimensional version of the combination model in two phases, which is of additional interest because of the aggregating behaviour possible in 2-D. Reminding ourselves of the driving PDEs, we have

$$\frac{\P u_1}{\P t} = d_1 \tilde{N}^2 u_1 - \tilde{N}(u_1 \tilde{N} q_1) + r u_1 E_1 \qquad t > 0; x \, 2 \, W_1(t) \qquad (7.75)$$

and

$$\frac{\P u_2}{\P t} = d_2 \tilde{N}^2 u_2 - r\,\tilde{N}(u_2 \tilde{N} q_2) + r u_2 E_2 \qquad t > 0; x \, 2 \, W_2(t): \qquad (7.76)$$

We consider a xed domain Ω

in the distribution of species 1. This asymmetry gives us the opportunity to explore the two

We obtain

$$\int_{W_1(t)} w_i \frac{\partial u_1}{\partial t}\, dW = \int_{W_1(t)} d_1 w_i \tilde{N}^2 u_1 dW - \int_{W_1(t)} w_i \tilde{N}\cdot(u_1 \tilde{N} q_1)\, dW + \int_{W_1(t)} w_i r_1 u_1 (A - a u_1)\, dW \tag{7.86}$$

and

$$\int_{W_2(t)} w_i \frac{\partial u_2}{\partial t}\, dW = \int_{W_2(t)} d_2 w_i \tilde{N}^2 u_2\, dW - \int_{W_2(t)} w_i r \tilde{N}\cdot(u_2 \tilde{N} q_2)\, dW + \int_{W_2(t)} w_i r_2 u_2 (B - b\, u_2)\, dW. \tag{7.87}$$

We define the total population of a species as q_k, given by

$$q_k(t) = \int_{W_k(t)} u_k\, dW \tag{7.88}$$

where $W_k(t)$ is the moving domain inhabited by that species. Since mass is not conserved in general, we will use the concept conserving relative mass. We write a relative conservation principle in terms of q, introducing the weight function w_i,

$$\frac{1}{q_k(t)} \int_{W_k(t)} w_i u_k\, dW = c_{k_i} \tag{7.89}$$

or

$$\int_{W_k(t)} w_i u_k\, dW = c_{k_i} q(t) = c_{k_i} \int_{W_k(t)} u_k\, dW. \tag{7.90}$$

Here the constant c_{k_i} is determined by the choice of w_i, which should be chosen to provide a partition of unity. A distributed conservation of mass principle is now given by equation (7.90). Note that

$$\frac{d}{dt} \int_{W_k(t)} w_i u_k\, dW = c_{k_i} q_k. \tag{7.91}$$

We differentiate (7.90) with respect to time using the Leibnitz integral rule on our moving frame $W_k(t)$ to give

$$\frac{d}{dt} \int_{W_k(t)} w_i u_k\, dW = \int_{W_k(t)} \frac{\partial (w_i u_k)}{\partial t} + \tilde{N}\cdot(w_i u_k x)\, dW. \tag{7.92}$$

We require that the basis functions w_i move with the domain. Hence the basis functions also

have velocity x and therefore

$$\frac{\partial w_i}{\partial t} + x \cdot \nabla w_i = 0. \tag{7.93}$$

We obtain

$$\frac{d}{dt} \int_{W_k(t)} w_i u_k \, dW = \int_{W_k(t)} w_i \left(\frac{\partial u_k}{\partial t} + \nabla \cdot (u_k x) \right) dW \tag{7.94}$$

or

$$\frac{d}{dt} \int_{W_k(t)} w_i u_k \, dW - \int_{W_k(t)} w_i \nabla \cdot (u_k x) \, dW = \int_{W_k(t)} w_i \frac{\partial u_k}{\partial t} \, dW. \tag{7.95}$$

We write this in terms of q_k and the constants c_{k_i} to give

$$c_{k_i} q_k - \int_{W_k(t)} w_i \nabla \cdot (u_k x) \, dW = \int_{W_k(t)} w_i \frac{\partial u_k}{\partial t} \, dW. \tag{7.96}$$

We introduce the velocity potential f, defined by

$$x = \nabla f \tag{7.97}$$

so that

$$c_{k_i} q_k - \int_{W_k(t)} w_i \nabla \cdot (u_k \nabla f) \, dW = \int_{W_k(t)} w_i \frac{\partial u_k}{\partial t} dW \tag{7.98}$$

or, after integration by parts

$$c_{k_i} q_k + \int_{W_k(t)} u_k \nabla w_i(t$$

Integration by parts on the right leads to

$$c_{1_i} q_1 + \int_{W_1(t)} u_1 \tilde{\nabla} w_i \cdot \tilde{\nabla} f \; dW - \int_{S_1(t)} u_1 w_i \tilde{\nabla} f \cdot \hat{n}_1 \; dS = - \int_{W_1(t)} d_1 \tilde{\nabla} w_i \cdot \tilde{\nabla} u_1 dW$$

After considering the boundary conditions (7.81) and (7.82) the remaining expression is

$$c_{2_i} q_2 + \int_{W_2(t)} u_1 \tilde{N} w_i \cdot \tilde{N} f \; dW = - \int_{W_2(t)} d_2 \tilde{N} w_i \cdot \tilde{N} u_2 dW + \int_{S_m(t)} w_i d_2 \tilde{N} u_2 \cdot \hat{n}_2 \; dS + \int_{W_2(t)} r \, u_2 \tilde{N} w_i \cdot \tilde{N} q_2 \; dW + \int_{W_2(t)} w_i r_2 u_2 (B - b \, u_2) \; dW. \tag{7.107}$$

If we have a given q_1, q_2, q_1 and q_2, we may solve (7.103) and (7.107) for r. This will allow us to subsequently recover the nodal velocities. In order to obtain q_1 and q_2, we refer to equations (7.79) and (7.80), (7.77) and (7.78). Combining (7.77) and (7.79) we obtain

$$A - a u_1 - b u_2 = e_1 \tilde{N}^2 q_1 + q_1. \tag{7.108}$$

We write this in weak form, using a weight function w_i to give

$$\int_W w_i A \; dW - \int_W w_i a u_1 \; dW - \int_W w_i b u_2 \; dW = \int_W w_i e_1 \tilde{N}^2 q_1 \; dW + \int_W w_i q_1 \; dW. \tag{7.109}$$

After integration by parts on the right-hand side we obtain

$$\int_W w_i A \; dW - \int_W w_i a u_1 \; dW - \int_W w_i b u_2 \; dW = e_1 \int_{S_e} w_i \tilde{N} q_1 \cdot \hat{n}_1 \; dS - e_1 \int_W \tilde{N} w_i \cdot \tilde{N} q_1 \; dW + \int_W w_i q_1 \; dW. \tag{7.110}$$

The first term on the right-

in equation (7.103),

$$\sum_i c_{1_i} q_1 + \int_{W_1(t)} \sum_i [u_1 \tilde{N} w_i \quad \tilde{N} f \,]\, dW = \int_{W_1(t)} \sum_i [d_1 \tilde{N} w_i \quad \tilde{N} u_1]\, dW + \int_{S_m(t)} \sum_i w_i d_1 \tilde{N} u_1 \quad \hat{n}_1 \, dS + \int_{W_1(t)} \sum_i [\tilde{N} w_i \quad \tilde{N} q_1]\, dW + \int_{W_1(t)} \sum_i [w_i r_1 u_1 (A \quad a u_1)]\, dW \qquad (7.113)$$

or

$$q_1 = \int_{S_m(t)} d_1 \tilde{N} u_1 \quad \hat{n}_1 \, dS + \int_{W_1(t)} r_1 u_1 (A \quad a u_1)\, dW \qquad (7.114)$$

and for the sum over equation (7.107)

$$q_2 = \int_{S_m(t)} d_2 \tilde{N} u_2 \quad \hat{n}_2 \, dS + \int_{W_2(t)} r_2 u_2 (B \quad b$$

at the external boundaries can be approximated from their near neighbours because we have Neumann conditions in place.

Before we are able to solve for u_1 and u_2, we require a finite element approximation to obtain a solution for q_1 and q_2 as the first step. We define an approximation to each of our variables in terms of the standard basis functions W_i. We do not require modified basis functions at this stage for the same reasons given for the 1-D case. We do not repeat those definitions here but instead refer to Appendix A, equations (A.1) to (A.4). We substitute those approximations into equation (7.111) and obtain

$$\int_W W_i A \, dW - a \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{1_j} - b \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{2_j}$$

$$= -e_1 \sum_{j=1}^{N} \int_W \tilde{\nabla} W_i \cdot \tilde{\nabla} W_j \, dW \; Q_{1_j} + \sum_{j=1}^{N} \int_W W_i W_j \, dW \; Q_{1_j}. \qquad (7.125)$$

We may write (7.125) in terms of our mass and stiffness matrices M and K to obtain

$$M\underline{A} - aM\underline{U}_{1_j} - bM\underline{U}_{2_j} = e_1 K\underline{Q}_{1_j} + M\underline{Q}_{1_j}. \qquad (7.126)$$

which we may rewrite in terms of $\underline{Q}_{1_j}$

$$\underline{Q}_1 = (-e_1 K + M)^{-1} M(\underline{A} - a\underline{U}_1 - b\underline{U}_2). \qquad (7.127)$$

Here $\underline{A}$ is a vector with all entries equal to A. In exactly the same manner, we substitute the approximations (A.1) to (A.4) defined in Appendix A into (7.112) and obtain

$$\int_W W_i B \, dW - a \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{1_j} - b \sum_{j=1}^{N} \int_W W_i W_j \, dW \; U_{2_j} \qquad (7.128)$$

$$= -e_2 \sum_{j=1}^{N} \int_W \tilde{\nabla} W_i \cdot \tilde{\nabla} W_j \, dW \; Q_{2_j} + \sum_{j=1}^{N} \int_W W_i W_j \, dW \; Q_{2_j} \qquad (7.129)$$

which is, in matrix form

$$\underline{Q}_2 = (-e_2 K + M)^{-1} M(\underline{B} - a\,\underline{U}_1 - b\,\underline{U}_2); \qquad (7.130)$$

where $\underline{B}$ is a vector with all entries equal to B. We can now recover $\underline{Q}_1$ and $\underline{Q}_2$ by solving equations (7.127) and (7.130). We now consider the ALE system which will allow us to obtain U_1 and U_2. For this system we must use the modified weight functions $\tilde{W}_i$ of Chapter 4

(gure 4.11). This will allow us to obey the principle of relative conservation of mass and yet impose a velocity on the interface and on the external boundaries. We take equations (7.103) and (7.107) and substitute into them the approximations (A.1) to (A.14) as necessary. We obtain the following, with all variables now expressed in terms of their piecewise linear

Similarly, (7.132) can be expressed as

$$\tilde{K}(\underline{U}_2)\underline{F}_2 = \underline{\tilde{f}}_2 \tag{7.135}$$

with the vector $\underline{\tilde{f}}_2$ containing entries $\tilde{f}_{2_i}$ given by

$$\begin{aligned}\tilde{f}_{2_i} = &-\tilde{c}_{2_i} q_2 - \sum_{j \in Z_2} \int_{W_2(t)} d_2 \nabla \tilde{W}_i \nabla W_j \, dW \; U_{2_j} \\ &+ \int_{S_m(t)} \tilde{W}_i d_2 \nabla U_2 \cdot \hat{n}_2 \, dS + \sum_{j \in Z_2} \int_{W_2(t)} r U_2 \nabla \tilde{W}_i \cdot \nabla W_j \, dW \; Q_{2_j} \\ &+ \sum_{j \in Z_2} \int_{W_2(t)} \tilde{W}_i W_j r_2 B \, dW \; U_{2_j} - \int_{W_2(t)} r_2 b \, \tilde{W}_i U_2^2 \, dW.\end{aligned} \tag{7.136}$$

The nonlinear terms in (7.134) and (7.136) can be calculated using Gaussian quadrature (see Appendix B). We can now obtain $\underline{F}_1$ and $\underline{F}_2$ by solving these matrix systems. Since the weighted stiffness matrices $\tilde{K}(\underline{U}_1)$ and $\tilde{K}(\underline{U}_2)$ are singular, we have an infinity of solutions available and we set $\nabla F \cdot \hat{n}_k = 0$ at all external boundary nodes to obtain a single solution, where $\hat{n}_k$ is the normal to the boundary for either species $k \in [1, 2]$. Note that summing over the rows of (7.133) and (7.135) will give the expressions for q_1 (7.114) and q_2 (7.115).
To recover X, we use the approximation

$$X(x,t) = \sum_{j \in Z_1 \cup Z_2} X_j(t) W_j(x,t). \tag{7.137}$$

To obtain the finite element form, we substitute this into equation (7.120),

$$\sum_{j \in Z_1 \cup Z_2} \int_{W_k(t)} \tilde{W}_i W_j \, dW \; X_j = \sum_{j \in Z_1 \cup Z_2} \int_{W_k(t)} \tilde{W}_i \nabla W_j \, dW \; F_j. \tag{7.138}$$

In matrix form this is

$$\tilde{M}\underline{X} = \tilde{B}\underline{F}. \tag{7.139}$$

We impose $x \cdot \hat{n}_k = 0$ on the external boundaries. We impose the interface velocity obtained from (7.119). Since we are using modified basis functions we will not interfere with the conservation of relative mass by doing so. We solve (7.139) for the remaining velocities.

Time integration

We move the nodes using Euler's scheme. Using the same scheme, we update the values of q_1 and q_2 from the values of q_1 (7.114) and q_2 (7.115).

Obtaining the solution U_1 and U_2

We may now recover the values of U_1 and U_2. The relative conservation of mass equations (7.123) and (7.124) allow us to obtain u on the updated grid. We substitute the familiar piecewise linear approximations (A.1) and (A.2) into (7.123) and (7.124), and obtain the matrix forms

$$\tilde{M}_1 \underline{U}_1 = \underline{\tilde{c}}_{1_i} q_1(t) \qquad (7.140)$$

and

$$\tilde{M}_2 \underline{U}_2 = \underline{\tilde{c}}_{2_i} q_2(t): \qquad (7.141)$$

We begin 2

6. Update the values of $q_1(t + dt)$ and $q_2(t + dt)$ from the values of $q_1(t)$ (7.114) and $q_2(t)$ (7.115);

7. Find the solutions $U_1(t + dt)$ and $U_2(t + dt)$ by solving the conservation equations (7.140) and (7.141).

7.3.2 Results

The model is implemented in MATLAB on a square domain with 33 nodes along each side. We are able to produce plausible behaviour. We are able to observe the moving interface exhibiting different behaviour at different points along its length, according to the population dynamics either side (see gure 7.10). The interface moves according to condition derived from the high competition limit, and the population densities adjacent to the interface are subject to signi cant increases or decreases because of this motion. Unfortunately, we run into the problem of internal node tangling at the point when more interesting behaviour begins to emerge. Figures 7.11 and 7.12 shown the state of the system shortly before this occurs. This is likely to be a fundamental weakness of this complex implementation of the MMFEM. The MMFEM keeps the node order and connectivity intact, no matter how much movement is occurring, and so cannot easily cope with highly distorted grids. In this particular MMFEM, we have an interface condition which is only indirectly related to the dynamics of the majority of the system. The interface is free to make large and sudden movnements because the calculation of its velocity takes place separately to that of the velocities elsewhere. This freedom has the potential to have a negative impact on the stability of the rest of the domain. It may be possible to nd a set of parameters which are more stable. It is certainly possible to run a steady state system but it is of little interest. However, to make useful progress from the point we have reached, the sensible approach would be to further research the interface condition from both an ecological and a mathematical perspective. In this way it may be possible to nd a consnioannonsnioan6(be)-287(possible)-286

repellant forces between nodes. Alternatively, a smarter way of incorporating the interface condition may help. From an ecological perspective, there may be a different, simpler interface condition that we could use, which may ease this dif culty.

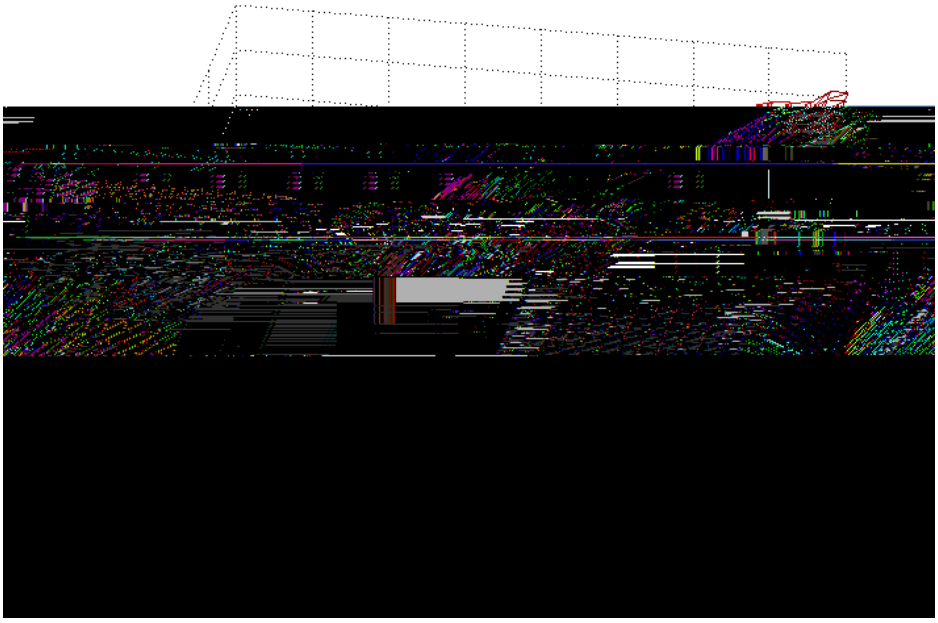
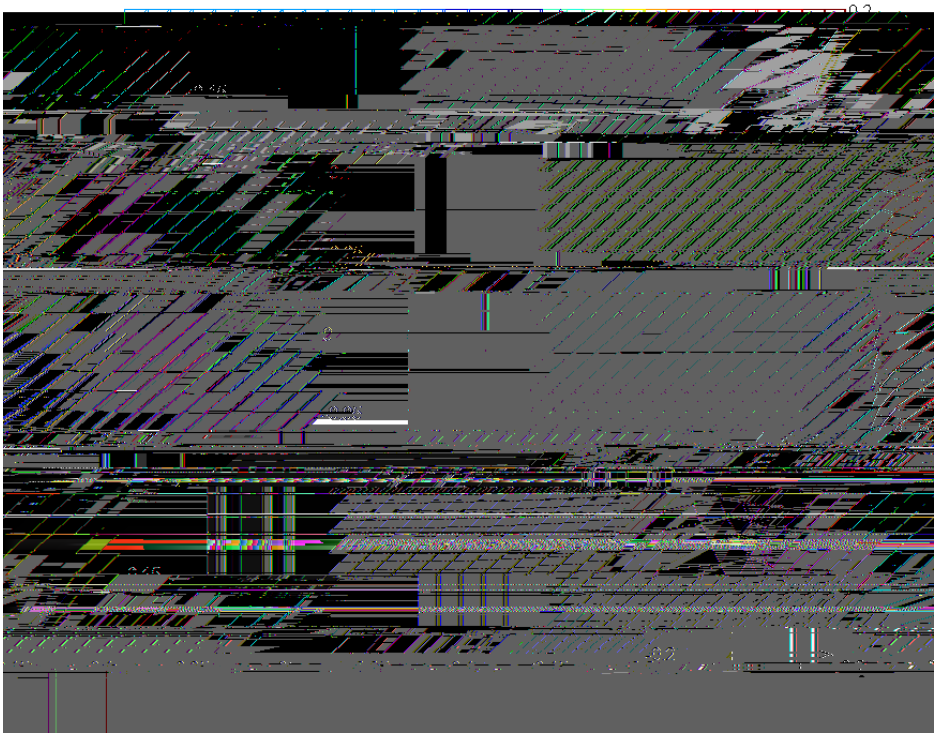


Fig. 7.10 The solution of the 2-D competition-aggregation-diffusion model in two phases with a moving interface at $t = 2 \times 10^{-5}$. The sum of both species is plotted, although they are segregated completely with the population consisting of only species 1 to the left of the interface and only species 2 to the right of the interface. We observe heterogenous movement of the interface, which no longer aligns with $x = 0$. We observe a small building of population density adjacent to it (near $x = 0.05, x = -0.2$) as a result. The parameters used are $d_1 = d_2 = 0.01, k_1 = 1, k_2 = 1=3, r_1 = r_2 = 1, A = 1.5, B = 2, a = 1, b = 2, a = 3$ $b = 1$, and $e_1 = e_2 = 0.001$.



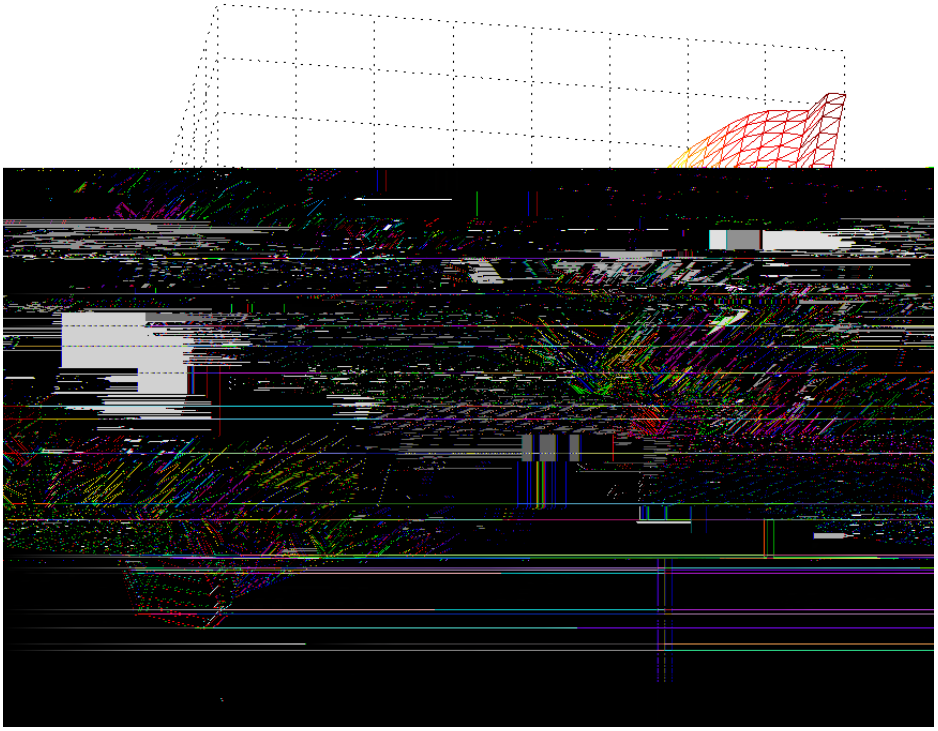


Fig. 7.12 The solution of the 2-D competition-aggregation-diffusion model in two phases with a moving interface at $t = 2.3 \times 10^{-5}$. The heterogenous movement of the interface has produced three distinct areas of high population density. Node tangling occurs soon after. The parameters used are $d_1 = d_2 = 0.01$, $k_1 = 1$, $k_2 = 1.3$, $r_1 = r_2 = 1$, $A = 1.5$, $B = 2$, $a = 1, b = 2, a = 3 b = 1$, and $e_1 = e_2 = 0.001$.

Chapter 8

Summary

We will now summarize the material covered in this thesis and discuss the next steps for this research.

In Chapters 1 and 2, we introduced the concept of moving mesh methods and outlined the various approaches. We discussed the history and development of a variety of velocity-based methods which formed a pathway towards the moving mesh nite element method. We then introduced the Lotka-Volterra competition equations.

In Chapter 3, we outlined, in general terms, the process for applying the MMFEM in either 1 or 2 dimensions. We examined the existing body of work performed using the MMFEM, looking at both classic examples and others that require modi cations to the method.

In Chapter 4, we began to demonstrate new applications for moving mesh methods. We illustrated the equidistribution method with a model of a column of water undergoing wind sheer at the surface, and which is also subject to Coriolis forces. We then illustrated the MMFEM, applying it to the Fisher's equation of blow-up or combustion. We built this model in both 1 and 2 dimensions, and with both a free and xed boundary. We compared the xed boundary model to a nite difference implementation, and found that we were able to resolve the blow up peak at a higher, narrower stage. For the free boundary case we used modi ed basis functions, and considered how best to construct a stiffness matrix in terms of the modi ed basis functions. For this model, we also made a switch between basis systems using the ALE form. We found that we resolved a higher peak, but we lost accuracy in the time at which the blow-up occurs. We also applied the MMFEM to the Keller-Segel model, which has both a substrate and a reactant, building this model in 2-D. We found that the accuracy of this model is dependent on the shapes of the triangles in the mesh, which is determined by the initial node distribution. For the better node distributions, the MMFEM

suffers from node tangling whenever more dynamic mesh movements are produced.

The potential exists for useful further work to extend the research in this thesis. In particular, the population models and simulations in chapters 6 and 7 are novel, and are also suitable for application to real-world situations. There are three useful dimensions for further work.

- Validate existing models. It would be extremely interesting to compare the behaviour of the models against an empirical data set. The models easily lend themselves to adaptions in the sizes and shapes of the domains, alterations to the logistic terms

Appendix A

$$U_L(x;t) = \sum_{j=0}^{N+1} W_j(x;t)U_{L_j}(t) \quad \text{(in 1D)} \qquad U_L(x;t) = \sum_{j=1}^{N} W_j(x;t)U_{L_j}(t) \quad \text{(in 2D)} \tag{A.6}$$

$$F(x;t) = \sum_{j=0}^{N+1} W_j(x;t)F_j(t) \quad \text{(in 1D)} \qquad F(x;t) = \sum_{j=f}^{N}$$

4.34 0 Td [(t)]TJ/F101 052 Tf 4.136 0 Td [())-221(

Appendix B

Gaussian quadrature

For a nonlinear term such as

$$f_i = \int_{w_e} W_i U^2 \, dW \tag{B.1}$$

we may write the contribution to f_i from a triangular element w_e corresponding to the weight function W_i as

$$\int_{w_e} W_i U^2 \, dW = \int_{w_e} W_i \left(\sum_{j=1}^{3} U_j W_j \right)^2 dW. \tag{B.2}$$

The right hand side may be evaluated using three point Gaussian quadrature. This is exact for quadratics and has a higher order of accuracy than the approximation of any of the PDEs in this thesis, so will not affect the numerical accuracy obtainable. Suitable sets of weights and integration points are widely published, for example in [42]. A neat choice uses the same piecewise linear weight functions W_j as are used throughout this work, those of figure 3.3. The locations of the integration points x_1, x_2 and x_3 which correspond to these weights are given in figure B.1. The values of U at these points can be calculated from the values of U at the vertices x_A, x_B and x_C as follows,

$$U_1 = U(x_1) = \frac{2U(x_A)}{3} + \frac{U(x_B)}{6} + \frac{U(x_C)}{6} \tag{B.3}$$

$$U_2 = U(x_2) = \frac{U(x_A)}{6} + \frac{2U(x_B)}{3} + \frac{U(x_C)}{6} \tag{B.4}$$

$$U_3 = U(x_3) = \frac{U(x_A)}{6} + \frac{U(x_B)}{6} + \frac{2U(x_C)}{3}. \tag{B.5}$$

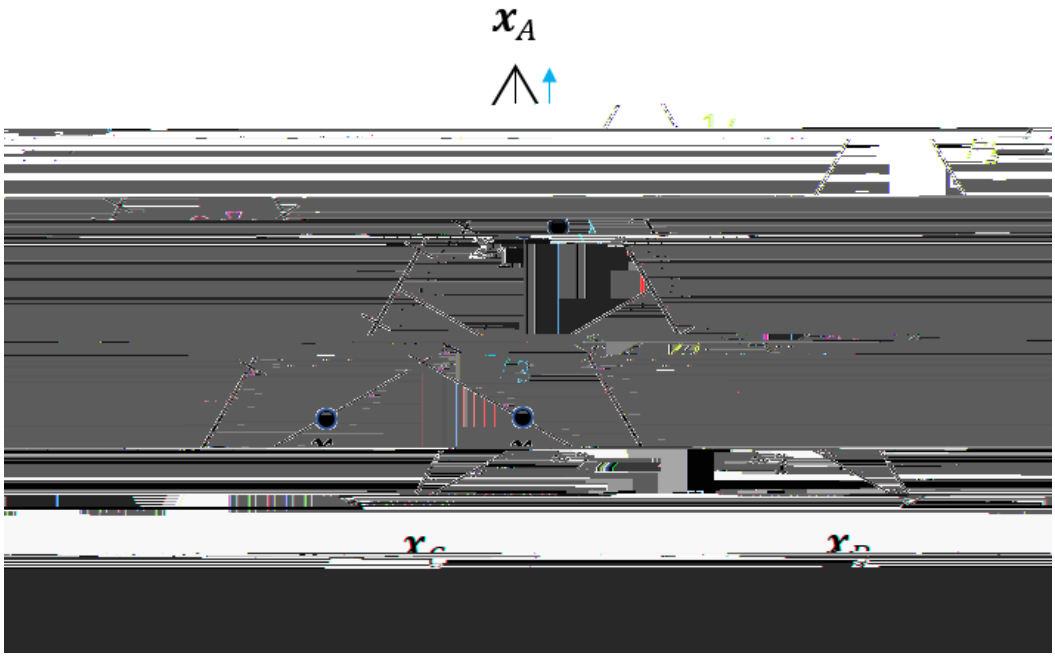


Fig. B.1 Location of integration points x_1, x_2 and x_3 for three point Gaussian quadrature for triangle with vertices at x_A, x_B and x_C. Each integration point lies $\frac{1}{3}$ of the way along the line connecting a vertex to the midpoint of the opposite edge.

Selecting piecewise linear weight function $W_i = W_1$ centered at node A, we can now calculate (B.2) as follows

$$\int_{w_e} W_1 \left(\sum_{j=1}^{3} U_j W_j \right)^2 dW = \frac{\text{Area}_e}{3} \left(2U_1^2 + \frac{U_2^2}{6} + \frac{U_3^2}{6} \right) : \qquad \text{(B.6)}$$

Likewise, if $W_i = W_2$ centered at node B, (B.2) becomes

$$\int_{w_e} W_2 \left(\sum_{j=1}^{3} U_j W_j \right)^2 dW = \frac{\text{Area}_e}{3} \left(\frac{U_1^2}{6} + 2U_2^2 + \frac{U_3^2}{6} \right) \qquad \text{(B.7)}$$

and if $W_i = W_3$ centered at node C, (B.2) becomes

$$\int_{w_e} W_3 \left(\sum_{j=1}^{3} U_j W_j \right)^2 dW = \frac{\text{Area}_e}{3} \left(\frac{U_1^2}{6} + \frac{U_2^2}{6} + 2U_3^2 \right) : \qquad \text{(B.8)}$$

References

[1] Adjerid, S. and Flaherty, J. (1986). A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations. SIAM Journal on Numerical Analysis, 23(4):778–796.

[2] Baines, M. (1994). Moving finite elements. Oxford University Press, Inc.

[3] Baines, M. (1998). Grid adaptation via node movement. Applied Numerical Mathematics, 26(1):77–96.

[4] Baines, M. (2015). Explicit time-stepping for moving meshes. Journal of Mathematical Study, 48(2):93–105.

[5] Baines, M., Hubbard, M., and Jimack, P. (2005). A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. Applied Numerical Mathematics, 54(3):450–469.

[6] Baines, M., Hubbard, M., and Jimack, P. (2011). Velocity-based moving mesh methods for nonlinear partial differential equations. Communications in Computational Physics, 10(3):509–576.

[7] Baines, M., Hubbard, M., Jimack, P., and Jones, A. (2006). Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions. Applied Numerical Mathematics, 56(2):230–252.

[8] Baines, M., Hubbard, M., Jimack, P., and Mahmood, R. (2009). A moving-mesh finite element method and its application to the numerical solution of phase-change problems. Communications in Computational Physics, 6(3):595–624.

[9] Baines, M. and Lee, T. (2014). A large time-step implicit moving mesh scheme for moving boundary problems. Numerical Methods for Partial Differential Equations, 30(1):321–338.

[10] Bird, N. (2014). A moving-mesh method for high order nonlinear diffusion. PhD thesis, University of Reading, Department of Mathematics.

[11] Bonan, B., Baines, M., Nichols, N., and Partridge, D. (2016). A moving-point approach to model shallow ice sheets: a study case with radially symmetrical ice sheets. The Cryosphere, 10(1):1–14.

[12] Brenner, S. and Scott, R. (2007). The mathematical theory of finite element methods, volume 15. Springer Science & Business Media.

[13] Budd, C., Carretero-González, R., and Russell, R. (2005). Precise computations of chemotactic collapse using moving mesh methods. Journal of Computational Physics, 202(2):463–487.

[14] Budd, C., Chen, J., Huang, W., and Russell, R. (1996). Moving mesh methods with applications to blow-up problems for pdes. Pitman Research Notes in Mathematics Series (1996), pages 1–18.

[15] Budd, C., Huang, W., and Russell, R. (2009). Adaptivity with moving grids. Acta Numerica, 18:111–241.

[16] Budd, C. and Williams, J. (2006). Parabolic Monge–Ampère methods for blow-up problems in several spatial dimensions. Journal of Physics A: Mathematical and General, 39(19):5425.

[17] Cao, W., Huang, W., and Russell, R. (2002). A moving mesh method based on the geometric conservation law. SIAM Journal on Scienti c Computing, 24(1):118–142.

[18] Cao, W., Huang, W., and Russell, R. (2003). Approaches for generating moving adaptive meshes: location versus velocity. Applied Numerical Mathematics, 47(2):121–138.

[19] Carlson, N. and Miller, K. (1998a). Design and application of a gradient-weighted moving nite element code i: In one dimensions. SIAM Journal on Scienti c Computing, 19(3):728–765.

[20] Carlson, N. and Miller, K. (1998b). Design and application of a gradient-weighted moving nite element code ii: In two dimensions. SIAM Journal on Scienti c Computing, 19(3):766–798.

[21] Cole, S. (2009). Blow-up in a chemotaxis model using a moving mesh method. Master's thesis, University of Reading, Department of Mathematics.

[22] Conway, E. and Smoller, J. (1977). Diffusion and the predator-prey interaction. SIAM Journal on Applied Mathematics, 33(4):673–686.

[23] Courant, R. (1994). Variational methods for the solution of problems of equilibrium and vibrations. Lecture Notes in Pure and Applied Mathematics, pages 1–1.

[24] Edgington, M. (2011). Moving mesh methods for semi-linear problems. Master's thesis, University of Reading, Department of Mathematics.

[25] Ekman, V. W. (1905). On the in uence of the earth's rotation on ocean current 2.9643on,s

[29] Grindrod, P. (1991).Patterns and waves: The theory and applications of reaction-diffusion equationsOxford University Press, USA.

[30] Heun, K. (1900). Neue methoden zur approximativen integration der differentialgle-

[45] Strang, G. and Fix, G. (1973). *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ.

[46] Turner, M., Clough, R., Martin, H., and Topp, L. (1956). Stiffness and deflection analysis of complex structures. *J. Aeronaut. Sci.*, 23(9):805–823.

[47] Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560.

[48] Volterra, V. (1928). Variations and fluctuations of the number of individuals in animal species living together. *J. Cons. Int. Explor. Mer*, 3(1):3–51.

[49] Weizhang, H. and Russell, R. (2010). *Adaptive moving mesh methods*, volume 174. Springer Science & Business Media.

[50] Wells, B. (2004). *A moving mesh finite element method for the numerical solution of partial differential equations and systems*. PhD thesis, University of Reading, Department of Mathematics.

[51] Wells, B., Baines, M., and Glaister, P. (2005). Generation of arbitrary lagrangian–eulerian (ale) velocities, based on monitor functions, for the solution of compressible fluid equations. *International Journal for Numerical Methods in Fluids*, 47(10-11):1375–1381.

[52] White, Jr, A. B. (1979). On selection of equidistributing meshes for two-point boundary-value problems. *SIAM Journal on Numerical Analysis*, 16(3):472–502.

[53]