A Moving Mesh Approach to Avascular Tumour Growth

Tamsin lee

January 29, 2010

Abstract

A two-phase mathematical model of an avascular tumour proposed by Breward, Byrne and others is discussed. Instead of mapping the moving domain to a fixed domain, a moving mesh approach is taken. The report compares three moving mesh strategies, two of which arise from biological considerations. The understanding gained suggests new approaches to modelling the growth of the tumour.

Contents

Chapter 1

A brief background on cancer growth

A tumour is a group of cancer cells. Like all cells, these cells gain nutrients, such as glucose and oxygen, from the surrounding environment. There is disagreement as to what causes a normal cell to turn into a cancer cell. Nonetheless, it is generally accepted that once a tumour is initiated it has three successive growth stages that it can possibly go through.

The first stage is referred to as the AVASCULAR stage. At this early stage the tumour has a large surface area in relation to its size. Consequently, a large proportion of the cells benefit from the surrounding nutrients and proliferate, causing the tumour to grow rapidly. This can only continue to a certain size. As the tumour grows the external nutrients cannot diffuse into the cells in the centre. The cells in the centre go into a quiescent state. In this state, they are dormant - but are able to proliferate again should nutrients become available to them. As the cells on the edge continue to proliferate, i.e. the tumour grows, the proliferating region expands and the cells in the centre die creating a necrotic core. At this stage, there is a balance between the maximum possible size of the tumour and its surrounding environment, the key reason being the limited ability of the majority of cells to obtain nutrients. However, some mathematical models include other relevant factors such as surface tension [15], [12], attractive cell forces [4], residual stress [1] and contractility (possibly due to the wound-healing process) [13].

During the avascular stage, tumours are malignant and are unlikely to affect the host. However, once the tumour obtains a dependable blood supply from a nearby capillary, then it advances to the more aggressive VASCULAR stage. A nutrient rich capillary is drawn into the tumour, onsetting the rapid

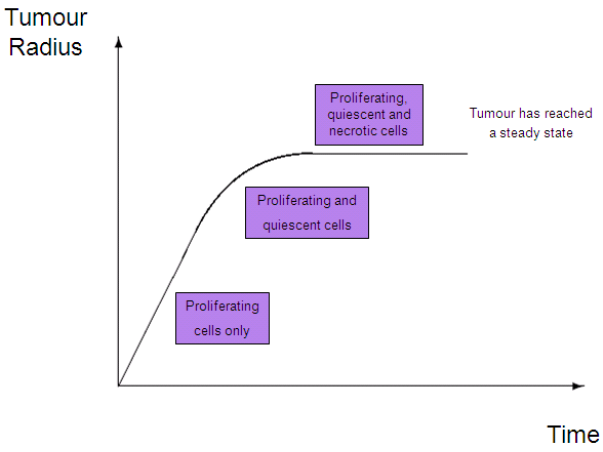

Figure 1.1: The growth of an avascular tumour

.

growth of cancerous cells. The process by which the tumour obtains its own blood supply is called ANGIOGENESIS and preventing this from occurring is of particular interest to drug development. This is because once the tumour has obtained a blood supply the tumours can leave its primary location via the circulatory system (metastasis) and settle in multiple areas of the body. The METASTATIC stage is the final stage of tumour growth, and the most difficult to treat.

From the moment normal cells mutate to cancer cells, there are three distinct stages to cancer. The different stages have different characteristics so require individual investigation. We shall study the primary stage, avascular tumour growth.

1.1 Avascular tumours

As previously mentioned, the later stages of tumour growth are more critical since it is usually not until after angiogenesis that cancer is detrimental to the hosts' health. During the avascular stage, the tumour is malignant. Indeed, following a study of human cancers in mice [18] there is recent controversial hypothesis that we all have small dormant avascular tumours in our bodies.

Regardless of this clinical viewpoint, avascular tumour growth warrants the interest of scientists. It is beneficial to understand the simple system and its components prior to attempting analysis of a more complex system. Vascular tumours have many of the same characteristics as avasuclar tumours, but

the quantity and quality of data on avascular tumours is of a higher standard. This is because it is comparatively easier and cheaper to reproduce high quality avascular tumour experimental evidence in in vitro form.

In summary, we will be investigating a model for avascular tumours (see Figure 1.1) as they are simpler to model and help give an insight into the mechanisms of vascular tumour growth.

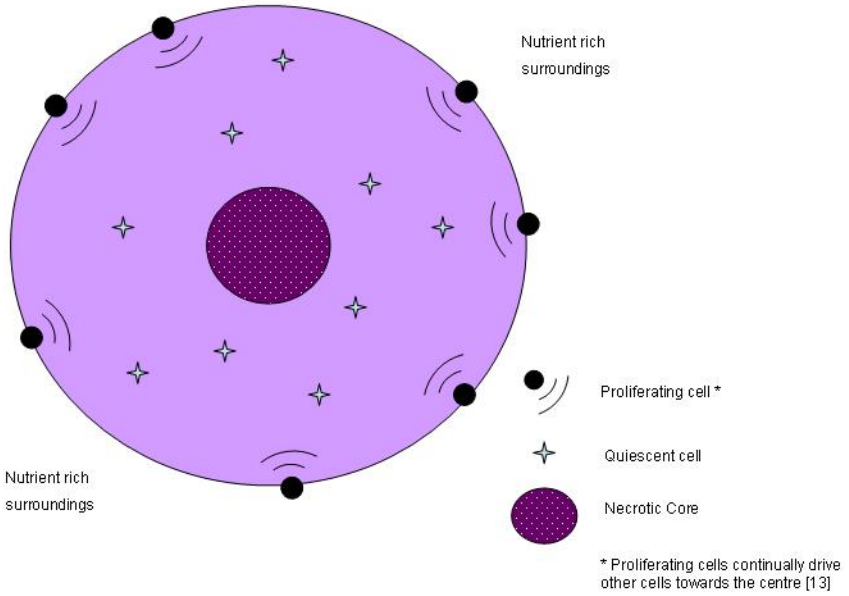

Figure 1.2: An avascular tumour

Chapter 2

The role of mathematics in cancer research

Ever since complex life evolved, it has been susceptible to cancer. The oldest description of cancer in humans was found in an Egyptian papyrus written between 3000-1500 BC. Today specialists are still extensively researching and experimenting in attempts to find cures and improve treatments. In the United Kingdom, one in four people will die of cancer, whilst one in three will be diagnosed to have cancer at some point in their life [20]. These figures illustrate how rife cancer is in modern society - and thereby it gives an indication into the vast size of the cancer research industry.

Despite the considerable volumes of time and money that is invested in this industry, the tools of mathematics have not really been exploited. The UK's first specialist cancer research organisation was set up in 1902, yet mathematics only started making a major contribution in this field from the early seventies. Most of the research is in molecular biology, cell biology and drug delivery. However, the use of mathematics to aid cancer research is increasing by way of computational modelling, as well as analysis on the large library of experimental data.

Indeed, it has been noted that a conceptual framework within which all these new (and old) data can be fitted is lacking [11]. In [10] it states that 'clinical oncologists and tumour biologists posses virtually no comprehensive theoretical model to serve as a framework for understanding, organising and applying these data'. By being educated as to which mechanisms are critical to the essence of tumour growth, these could possibly be manipulated to our advantage. As Byrne (1999a) remarks, 'In order to gain such insight, it is usually necessary to perform large numbers of time-consuming and intricate

experiments - but not always. Through the development and solution of mathematical models that describe different aspects of solid tumour growth, applied mathematics has the potential to prevent excessive experimentation' [6].

Ideally, experiments and modelling work hand-in-hand. The experiments can not only prove to be costly, but the subtleties of the many intricate processes can easily be overlooked. By modelling tumour growth to mimic data already collected, potentially pivotal characteristics can be identified. This can ensure that these interactions are monitored closely in future experiments. Ultimately, the aim for applied mathematics in tumour growth is to enlighten biologists as to the key processes, so that these can be artificially altered in a manner that eradicates (or manages) the disease.

2.1 Parameterisation

Gaining parameters for tumour growth is a challenge within itself. There are many variables on varying scales - some of which can be difficult to measure. For example, the *in vivo* measurement of a pressure that is probably very low ($\sim$ 10 mmHg) in a sample that is very small in size (max 1mm) is technically very difficult [16]. 'An important role of modelling in this respect is to determine, via sensitivity and/or bifurcation analysis, on which parameters the behaviour of the model crucially depends, thereby identififying which parameters need to be measured correctly' [16]. For example, when parameterising the concentration of nutrients surrounding the tumour, the value may refer only to the glucose concentration as this is the main ingredient. Or perhaps another nutrient that only appears in trace amounts (hence may be overlooked during experimentation) has a considerable effect on cancer cells.

When choosing parameters, the values may be chosen to show qualitative predictions. An example is given in [9] where a phenomenon was shown to be largely independent of the specific parameter values. All the same, if parameters are available then a well-parameterised model can make quantative *and* qualitative predictions.

Chapter 3

A two-phase model of solid tumour growth

Byrne et al. [7] formulated a two-phase model of solid tumour growth as a more general version of two different pre-exisiting models for solid tumour growth - [11] and [17]. Although full details of the modelling are not given in this paper, the biological reasoning and assumptions that contribute to the model are explicitly described. This is the model that we will be discussing and solving numerically in this report.

Our first task is to non-dimensionalise the model. This involves the partial or full removal of units by a suitable substitution of variables. Non-dimensionalisation can simplify a problem by reducing the number of variables. It also aids analysis of the behaviour of a system by recovering characteristic properties. In our case, the key motivator to non-dimensionalising the system is to enable us to take advantage of parameterisations studied elsewhere.

In this report we shall approximately solve the non-dimensionalised moving boundary problem by applying a moving mesh approach. We move the mesh in three different ways: by ensuring that mass fractions in an element remain constant over time; by moving the mesh with the cell velocity; by driving mesh movement iin proportion to that of the moving boundary.

The results generated from these methods are discussed and compared with previous results.

3.1 Model formulation

In [7] it is assumed that a tumour consists of cells and water, with respective volume fractions α and β (with $\alpha + \beta = 1$). The two phases have an associ-

ated velocity, pressure and volume-fraction-averaged stress tensor denoted by (v_c, p_c, σ_c) for the cell phase and (v_w, p_w, σ_w) for the water phase.

The cell and water phases are treated as incompressible fluids whose densities are equal to leading order. Consequently, by applying mass balances to the cell and water phases, two partial differential equations (PDEs) for α and β are generated. These PDEs are stated in terms of the respective velocities and the volume conversion rates $S_c = -S_w$ (since the net cell proliferation rate is zero; thus cells are viewed as being made essentially of water). These PDEs are simplified to the one-dimensional forms

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(\alpha v_c) = S_c(\alpha, C), \qquad \frac{\partial}{\partial x}(\alpha v_c + (1-\alpha)v_w) = 0, \tag{3.1}$$

where S_c is consistent with [19] and is taken to be of the form

$$S_c(\alpha, C) = \left(\frac{s_0 C}{1 + s_1 C}\right)\alpha(1-\alpha) - \left(\frac{s_2 + s_3 C}{1 + s_4 C}\right)\alpha$$

for positive constants s_0, s_1, s_2, s_3 and s_4, and where C is a function of (x, t) that represents the nutrient concentration within the tumour.

The formulation of the model continues by applying momentum balances to the two phases, assuming that inertial effects are negligible and no external forces act on the system. This reduces the momentum conservation laws to force balances. The resulting equations contribute to closing the model.

It is further assumed that on the timescale of interest, the cell and water phases can be treated as viscous and inviscid fluids respectively. This allows the volume-fraction-averaged stress tensors σ_c, σ_w to be written in terms of shear and bulk viscosity coefficients of the cell phase, μ_c, λ_c, and pressure for the cell phase, p_c.

In [7] a three-dimensional model of avascular tumour growth is developed. It is then assumed that the tumour undergoes one-dimensional growth, parallel to the x-axis, that occupies the region $-x_N(t) \leq x \leq x_N(t)$ at time t and is symmetric about its midpoint $x = 0$. In addition to (3.1) the final model consists of the momentum balance equations

$$\frac{\partial}{\partial x}\left(-p - \alpha\Sigma_c(\alpha) + (2\mu_c + \lambda_c)\alpha\frac{\partial v_c}{\partial x}\right) = 0, \qquad -(1-\alpha)\frac{\partial p}{\partial x} = k(\alpha)(v_w - v_c), \tag{3.2}$$

and the nutrient diffusion equation

$$\frac{\partial^2 C}{\partial x} - Q_C(\alpha, C) = 0, \tag{3.3}$$

where

- $\Sigma_c(\alpha)$ is the pressure difference between the two phases and may include contributions due to, for example, cell-cell interactions and membrane stress. It is defined by

$$\Sigma_c(\alpha) = \begin{cases} 0 & 0 \leq \alpha < \alpha_{min} \\ \frac{\hat{\Sigma}_c |\alpha - \alpha^*|^{r-1}}{(1-\alpha)^q}(\alpha - \alpha^*) & \alpha_{min} \leq \alpha < 1 \end{cases}$$

  for positive constants $q, r, 0 < \alpha_{min} < \alpha^* < 1$ and $\hat{\Sigma}_c$.
  When specifying $\Sigma_c(\alpha)$, α^* denotes a natural cell packing density: if $\alpha > \alpha^*$ cells move to reduce their stress, while if $\alpha < \alpha^*$, they aggregate, if they are not too sparsely populated ($\alpha \geq \alpha_{min}$) By definition we have $\Sigma_c(0) = 0$,

- $p = p_c - \Sigma_c(\alpha)$ where p_c is the pressure for the cell phase,

- $k(\alpha)$ is a drag coefficient of the form,

$$k(\alpha) = k_0 \alpha (1 - \alpha),$$

  for a positive constant k_0,

- $Q_c(\alpha, C)$ models the rate at which the tumour cells consume nutrients, and is of the form

$$Q_c(\alpha, C) = \frac{Q_0 C \alpha}{1 + Q_1 C} \geq 0,$$

  for positive constants Q_0 and Q_1.

By definition, the tumour occupies the domain in which $\alpha > 0$. At the boundary its growth rate is governed by the cell velocity at $x = x_N(t)$, so

$$\frac{dx_N}{dt} = v_c(x_N, t). \tag{3.4}$$

These equations are defined on a moving domain, and in the model are subject to the boundary conditions and initial conditions below,

$$v_c = v_w = \frac{\partial C}{\partial x} = 0 \qquad \text{at } x = 0, \tag{3.5}$$

$$p = 0, \quad (2\mu_c + \lambda_c)\frac{\partial v_c}{\partial x} - \Sigma_c(\alpha) = 0, \quad C = C_\infty \qquad \text{at } x = x_N(t) \tag{3.6}$$

$$\alpha = \alpha_0(x), \quad x = x_0 \qquad \text{at } t = 0. \tag{3.7}$$

Equations (3.5) ensure symmetry about $x = 0$. In (3.6), C_∞ denotes the nutrient concentration in the medium surrounding the tumour and we assume that C is continuous across the moving boundary $x = x_N(t)$. Equation (3.6) also guarantees continuity of the normal component of the cell and water stress tensors across $x = x_N(t)$, with the ambient pressure outside the tumour normalised so that $p = 0$ there and the tumour boundary is stress-free. Finally, equations (3.7) specify the initial cell distribution within the tumour and its initial size.

By integrating the second part of (3.1) and the first part of (3.2), subject to (3.5) and (3.6), expressions for v_w and p are deduced. These expressions can be substituted into the second part of (3.2) to give the single equation

$$\frac{\partial}{\partial x}\left((2\mu_c + \lambda_c)\alpha\frac{\partial v_c}{\partial x} - \alpha\Sigma_c(\alpha)\right) = \frac{k(\alpha)}{(1-\alpha)^2}v_c \tag{3.8}$$

for v_c.

The model thus comprises the diffusion equation describing the nutrient concentration (3.3), an equation describing the velocity (3.8) (derived from the second part of (3.1)), and a PDE for the volume fraction of cells (3.1), subject to the boundary and initial conditions given in (3.5) to (3.7).

3.2 Non-Dimensionalisation

The model put into a simple non-dimensional form by taking a typical length scale associated with the initial tumour size to be

$$L = x_N(0),$$

and a time scale associated with the outside nutrient concentration to be

$$T = \frac{1 + s_1 C_\infty}{s_0 C_\infty}.$$

This scaling provides us with the normalised variables

$$\begin{array}{lllll}
\hat{\alpha} = & \alpha & & \Rightarrow \alpha = \hat{\alpha}, & \text{volume fraction} \\
\hat{C} = & \frac{C}{C_\infty} & & \Rightarrow C = C_\infty \hat{C}, & \text{concentration} \\
\hat{t} = & \frac{1}{T}t = & \frac{s_0 C_\infty}{1+s_1 C_\infty}t & \Rightarrow t = \frac{1+s_1 C_\infty}{s_0 C_\infty}\hat{t}, & \text{time} \\
\hat{x} = & \frac{1}{L}x = & \frac{1}{x_N(0)}x & \Rightarrow x = x_N(0)\hat{x}, & \text{length} \\
\hat{x}_N(\hat{t}) = & \frac{1}{L}x_N(\hat{t}) = & \frac{1}{x_N(0)}x_N(t) & \Rightarrow x_N(t) = x_N(0)\hat{x}_N(\hat{t}), & \text{length} \\
v = & \frac{T}{L}v_c = & \frac{1+s_1 C_\infty}{x_N(0)s_0 C_\infty}v_c & \Rightarrow v_c = \frac{x_N(0)s_0 C_\infty}{1+s_1 C_\infty}\hat{v}. & \text{velocity}
\end{array}$$

We thus obtain the following non-dimensionalised system

$$\frac{\partial \hat{\alpha}}{\partial \hat{t}} + \frac{\partial}{\partial \hat{x}}(\hat{\alpha} v) = \frac{(1+\hat{s}_1)\hat{C}}{1+\hat{s}_1\hat{C}}\hat{\alpha}(1-\hat{\alpha}) - \frac{\hat{s}_2 + \hat{s}_3\hat{C}}{1+\hat{s}_4\hat{C}}\hat{\alpha} = \hat{S}(\hat{\alpha}, \hat{C}) \tag{3.9}$$

$$\frac{\partial}{\partial \hat{x}}\left[\hat{\alpha}\mu\frac{\partial v}{\partial \hat{x}} - \hat{\alpha}\chi(\hat{\alpha})\right] = \hat{k}\frac{\hat{\alpha}}{1-\hat{\alpha}}v \tag{3.10}$$

$$\frac{\partial^2 \hat{C}}{\partial \hat{x}^2} - \frac{Q\hat{C}\hat{\alpha}}{1+\hat{Q}_1\hat{C}} = 0 \tag{3.11}$$

with initial and boundary conditions

$$\hat{x}_N(0) = 1, \;\; \alpha = \alpha_0 \qquad \text{at} \quad \hat{t} = 0 \tag{3.12}$$

$$\hat{v} = \frac{\partial \hat{C}}{\partial \hat{x}} = 0 \qquad \text{at} \quad \hat{x}_0 = 0 \tag{3.13}$$

$$\Rightarrow \mu \frac{\partial \hat{v}}{\partial \hat{x}} - \chi(\hat{\alpha}) = 0, \quad \frac{\partial \hat{x}_N}{\partial \hat{t}} = \hat{v}, \quad \hat{C} = 1 \qquad \text{at} \quad \hat{x} = \hat{x}_N \tag{3.14}$$

In equations (3.9) to (3.14) we have introduced the parameters

$$\hat{s}_1 = s_1 C_\infty, \quad \hat{s}_2 = \frac{(1 + s_1 C_\infty)}{1 + s_1 C_\infty} s_2, \quad \hat{s}_3 = \frac{1 + s_1 C_\infty}{s_0} s_3, \quad \hat{s}_4 = s_4 C_\infty,$$

$$\hat{k} = \frac{k_0 x_N^2(0) s_0 C_\infty}{1 + s_1 C_\infty}, \quad \mu = (2\mu_c + \lambda_c) \frac{s_0 C_\infty}{(1 + s_1 C_\infty)},$$

$$\chi(\hat{\alpha}) = \begin{cases} 0 & 0 \le \hat{\alpha} < \alpha_{min} \\ \frac{\hat{\Sigma}_c |\hat{\alpha} - \alpha^*|^{r-1}}{(1-\hat{\alpha})^q} (\hat{\alpha} - \alpha^*) & \alpha_{min} \le \hat{\alpha} < 1 \end{cases},$$

$$Q = Q_0 x_N^2(0), \quad \hat{Q}_1 = Q_1 C_\infty.$$

In what follows the hats ($\hat{\cdot}$) are dropped from the variables and parameters.

In this study we shall solve this model numerically using three moving mesh methods.

Chapter 4

Moving meshes

Generally, for the numerical solution of time-dependent differential equations with a moving boundary a static or a moving mesh method can be used. Either of these can be considered as an adaptive mesh method. With a static mesh the discrete solution and equation are initially defined on a given mesh. At each time step a new mesh (which may have a different number of nodes to the previous mesh) is generated and the solution is interpolated from the old mesh to the new. Static methods are generally robust, but the computation can be slow.

When using a moving mesh, a mesh with a fixed number of nodes moves smoothly with the solution itself. Often, a mesh equation and the differential equation are solved simultaneously to generate the new nodes and solution. With a moving mesh, interpolation of dependent variables from the old mesh to the new mesh is unnecessary. It is possible to map a moving domain onto a

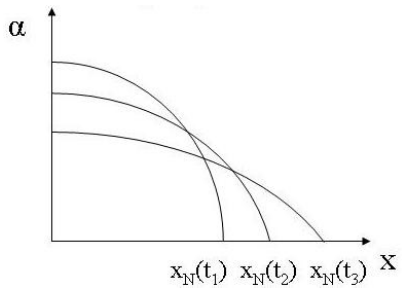

Figure 4.1: A moving boundary problem

fixed domain at the expense of an additional variable to cover for the boundary movement (Figure 5.2). This is the approach used in [4]. In this work we use a moving mesh strategy.

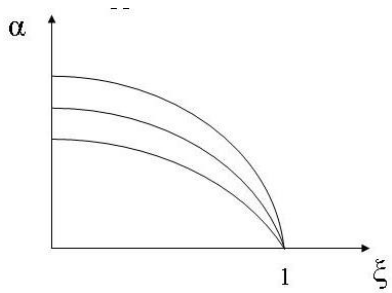

Figure 4.2: A moving boundary problem having been mapped onto a fixed domain

4.1 Conservative and Non-Conservative methods

It is possible to approach the problem in this study by finding the time rate of change $\dot{\alpha}$ of the solution α on a moving mesh having a given velocity $\dot{x}$. To do this the chain rule is used to differentiate $\alpha(x,t)$ with respect to time, giving

$$\dot{\alpha} \quad = \quad \frac{\partial \alpha}{\partial t} + \frac{\partial \alpha}{\partial x}\dot{x} \quad = \quad S(\alpha, C) - \frac{\partial}{\partial x}(\alpha v) + \frac{\partial \alpha}{\partial x}\dot{x}.$$

This technique generates an additional term $\left(\frac{\partial \alpha}{\partial x}\dot{x}\right)$ due to the mesh movement. However, it also makes the moving equation non-conservative in the sense that

$$\int \dot{\alpha} \ dx \neq \frac{d}{dt}\int \alpha \ dx,$$

which is inconsistent with the mass conservation equation (3.1).
A conservative approach is to differentiate the *integral* of the solution $\int \alpha \ dx$ with respect to time, taking into account the time-variation of the limits. To do this we apply Leibnitz integral rule and substitute in the PDE (3.1), giving

$$\begin{aligned} \frac{d}{dt}\int \alpha \ dx &= \int \frac{\partial \alpha}{\partial t} \ dx + \Big[\alpha \dot{x}\Big] \\ &= \int \left[\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(\alpha \dot{x})\right] \ dx \\ &= \int S_c(\alpha, C) \ dx + \Big[\alpha(\dot{x} - v_c)\Big]. \end{aligned}$$

The moving mesh methods used in this study are based on the latter approach, which is conservative.

4.2 Different methods to move the mesh

We will investigate three strategies for moving the mesh, i.e., different ways to define the mesh velocity $\dot{x}$. The three mesh velocities will be:

A: based on conserving mass fractions,

B: the cell velocity v,

C: proportional to the boundary movement $\frac{dx_N}{dt}$.

As with all moving mesh methods, the methods here retain the information at the free boundary. For this problem, that means that the final mesh node x_N tracks the tumour radius.

4.3 The General Algorithm

For all three methods, we use the following algorithm:

(1) Calculate the concentration C.

(2) Calculate the cell velocity v.

(3) Calculate the mesh velocity $\dot{x}$.

(4) Update the mesh $x(t)$ by using an explicit time-stepping scheme.

(5) Using the new mesh, calculate the solution of α .

Steps (1) and (2) are therefore common to the three methods, and we discuss these first.

Chapter 5

The numerical processes

Firstly in this chapter we give details of the numerical process for finding the C and v (Steps 1 and 2), both of which are calculated using finite differences on an irregular mesh. Then we look at solving the PDE for α using the three different moving mesh approaches.

5.1 Calculating the nutrient concentration $C(x,t)$ (Step 1)

We have the nonlinear PDE (3.11)

$$\frac{\partial^2 C}{\partial x^2} - \frac{QC\alpha}{1+Q_1 C} = 0,$$

with boundary conditions given in (3.13) and (3.14)

$$\begin{aligned} \frac{\partial C}{\partial x} &= 0 \quad \text{at} \quad x = 0, \\ C &= 1 \quad \text{at} \quad x = x_N. \end{aligned}$$

We discretise the differential equation (3.11) on a variable mesh $\{x_j\}$, and rearrange the discretisation so that the coefficients for C_j of $\frac{\partial^2 C}{\partial x^2}$ are on the left hand side and distinct, i.e.

$$\frac{\frac{C_{j+1}-C_j}{x_{j+1}-x_j} - \frac{C_j - C_{j-1}}{x_j - x_{j-1}}}{\frac{1}{2}(x_{j+1} - x_{j-1})} = \frac{QC_j\alpha_j}{1+Q_1 C_j}$$

which can be written as

$$T_j^l C_{j-1} + T_j^d C_j + T_j^u C_{j+1} \quad = \quad w(C_j) \quad (j = 1, 2, ..., N-2), \tag{5.1}$$

where

$$\begin{aligned} T_j^l &= \frac{2}{(x_j - x_{j-1})(x_{j+1} - x_{j-1})}, \\ T_j^d &= \frac{-2}{(x_{j+1} - x_j)(x_j - x_{j-1})}, \\ T_j^u &= \frac{2}{(x_{j+1} - x_j)(x_{j+1} - x_{j-1})} \end{aligned}$$

and

$$w(C_j) = \frac{QC_j\alpha_j}{1 + Q_1 C_j}.$$

Equation (5.1) does not hold for $j = N$ as this is known. For $j = 0, N-1$, we need to take account of the boundary conditions.

We write the non-linear system (5.1) in matrix form as

$$T\mathbf{C} = \mathbf{w}(\mathbf{C}) \tag{5.2}$$

where

$\mathbf{C}$ is a vector with entries C_0 to C_{N-1} $\quad (j = 0, 1, ..., N-1)$,

$\mathbf{w}(\mathbf{C})$ is a vector with entries $w(C_j)$ $\quad (j = 0, 1, ..., N-1)$,

and

T is a tridiagonal matrix of the C_j coefficients given in (5.1),

$$T = \begin{pmatrix} T_0^d & T_0^u & 0 & \cdots & 0 \\ T_1^l & T_1^d & T_1^u & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & T_{N-2}^l & T_{N-2}^d & T_{N-2}^u \\ 0 & \cdots & 0 & T_{N-1}^l & T_{N-1}^d \end{pmatrix}.$$

Boundary Conditions: $j = 0$

The boundary condition (3.13) states that $\frac{\partial C}{\partial x} = 0$ at $x = 0$, meaning that

the function is symmetrical about x_0. Hence, we conclude that at $x_1 = -x_{-1}$ we have $C_1 = C_{-1}$. Substituting these values into (5.1) for $j = 0$ gives

$$\frac{-2}{x_1^2 - x_0^2} C_0 + \frac{2}{x_1^2 - x_0^2} C_1 \quad = \quad \frac{QC_0\alpha_0}{1 + Q_1 C_0}. \tag{5.3}$$

Therefore, we have values for T_0^d, T_0^u and $w_0(C)$

$$\begin{aligned} T_0^d \;=\; -T_0^u \quad &= \quad \frac{-2}{x_1^2 - x_0^2}, \\ w(C_0) \quad &= \quad \frac{QC_0\alpha_0}{1 + Q_1 C_0}. \end{aligned}$$

Boundary Conditions: $j = N - 1$

For the right boundary we return again to (5.1), this time for $j = N - 1$ with the substitution $C_N = 1$ (from (3.14)),

$$\begin{aligned} &\frac{2}{(x_{N-1} - x_{N-2})(x_N - x_{N-2})} C_{N-2} - \frac{2}{(x_N - x_{N-1})(x_{N-1} - x_{N-2})} C_{N-1} \\ &\qquad = \frac{QC_{N-1}\alpha_{N-1}}{1 + Q_1 C_{N-1}} - \frac{2}{(x_N - x_{N-1})(x_N - x_{N-2})}. \end{aligned}$$

So T_{N-1}^l and T_{N-1}^d remain as defined by equation (5.1), but the final entry in $\mathbf{w}(\mathbf{C})$ has an extra term due to the boundary condition.

Now we have our complete matrix system, T, to obtain C_j ($j = 0, 1, ...N-1$), but note that the right hand side is nonlinear.

Numerically solving the discretized PDE

For the solution of (5.2) we use Newton's method, where the residual vector $\mathbf{R}$ of (5.2) is

$$\mathbf{R} \quad = \quad T\mathbf{C} - \mathbf{w}(\mathbf{C}).$$

We seek $\mathbf{C}$ such that $\mathbf{R} = \mathbf{0}$, so equation (5.2) holds. Note that if $Q_1 = 0$ the equations are linear and no iteration is needed. Otherwise, we carry out Newton's Method.

The Sub-Algorithm for Step (1)

Preliminary: Make an initial guess for $\mathbf{C}$:
For $p = 0, 1, 2, ...$

(a): Use $\mathbf{C}^p$ to find $\mathbf{w}(\mathbf{C}^p)$;

(b): Find the residual, $\mathbf{R}^p = T\mathbf{C}^p - \mathbf{w}(\mathbf{C}^p)$.

(c): Calculate the Jacobian of $\mathbf{R}^p$,

$$J^p = \frac{\partial \mathbf{R}^p}{\partial \mathbf{C}^p} = T - \left\{ \frac{\partial w_i^p}{\partial C_j^p} \right\}$$

where $i, j = 0, 1, ..., N-1$ and $\left\{ \frac{\partial w_i^p}{\partial C_j^p} \right\}$ is the diagonal matrix

$$\frac{\partial w_i}{\partial C_j} = \begin{pmatrix} \frac{Q\alpha_0}{(1+Q_1C_0)^2} & 0 & \cdots & 0 \\ 0 & \frac{Q\alpha_1}{(1+Q_1C_1)^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{Q\alpha_{N-1}}{(1+Q_1C_{N-1})^2} \end{pmatrix}.$$

(d): Find $\mathbf{H}^p = (J^p)^{-1}\mathbf{R}^p$.

(e): Set $\mathbf{C}^{p+1} = \mathbf{C}^p - \mathbf{H}^p$.

(f): With the new approximation $\mathbf{C}^{p+1}$, return to (a) and repeat until (e) converges, as measured by $\|\mathbf{C}^{p+1} - \mathbf{C}^p\|_2 < 1 \times 10^{-6}$.

5.2 Calculating the cell velocity $v(x,t)$ (Step 2)

We have the linear PDE (3.10)

$$\frac{\partial}{\partial x}\left[\alpha\mu\frac{\partial v}{\partial x} - \alpha\chi(\alpha)\right] = k\frac{\alpha}{1-\alpha}v \tag{5.4}$$

with boundary conditions given in (3.13) and (3.14)

$$v = 0 \quad \text{at} \quad x = 0, \tag{5.5}$$

$$\mu\frac{\partial v}{\partial x} - \chi(\alpha) = 0 \quad \text{at} \quad x = x_N(t). \tag{5.6}$$

We discretise (5.4) on the same variable mesh in a manner that gives us a system of equations for v_j

$$\frac{1}{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}} \left\{ \left[\alpha\mu \frac{\partial v}{\partial x} - \alpha\chi(\alpha) \right]_{j+\frac{1}{2}} - \left[\alpha\mu \frac{\partial v}{\partial x} - \alpha\chi(\alpha) \right]_{j-\frac{1}{2}} \right\} = k \frac{\alpha_j}{1-\alpha_j} v_j$$

$$\text{for } j = 1, 2, ..., N-1,$$

leading to

$$\begin{aligned} &\alpha_{j+\frac{1}{2}} \mu \left(\frac{v_{j+1} - v_j}{x_{j+1} - x_j} \right) - \alpha_{j+\frac{1}{2}} \chi(\alpha_{j+\frac{1}{2}}) - \alpha_{j-\frac{1}{2}} \mu \left(\frac{v_j - v_{j-1}}{x_j - x_{j-1}} \right) + \alpha_{j-\frac{1}{2}} \chi(\alpha_{j-\frac{1}{2}}) \\ &= k \frac{\alpha_j}{1-\alpha_j} (x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}) v_j. \end{aligned}$$

We may write (5.7) as

$$A_j^l v_{j-1} + A_j^d v_j + A_j^u v_{j+1} = b(\alpha_j) \tag{5.7}$$

where

$$\begin{aligned} A_j^l &= \frac{(\alpha_j + \alpha_{j-1})\mu}{2(x_{j+1} - x_{j-1})}, \\ A_j^d &= -\frac{1}{2} \left(\frac{(\alpha_j + \alpha_{j-1})\mu}{x_j - x_{j-1}} + \frac{(\alpha_j + \alpha_{j+1})\mu}{x_{j+1} - x_j} - k \frac{\alpha_j}{2(1-\alpha_j)} (x_{j+1} - x_{j-1}) \right), \\ A_j^u &= \frac{(\alpha_j + \alpha_{j+1})\mu}{2(x_{j+1} - x_j)} \end{aligned}$$

and

$$b(\alpha_j) = \frac{1}{2}(\alpha_j + \alpha_{j+1}) \chi \left(\frac{1}{2}(\alpha_j + \alpha_{j+1}) \right) - \frac{1}{2}(\alpha_j + \alpha_{j-1}) \chi \left(\frac{1}{2}(\alpha_j + \alpha_{j-1}) \right).$$

Next we investigate the boundary conditions at for $j = 0$ and $j = N$.

Boundary Conditions: $j = 0$

When $j = 0$ we have $v_0 = 0$ as stated in (3.5). This means that our system of equations above omits $j = 0$ and so begins with $j = 1$.

Boundary Conditions: $j = N$

The right boundary condition, at $j = N$, requires more work. First we dis-

cretise (5.6) at N by taking the average at $N-\frac{1}{2}$ and $N+\frac{1}{2}$,

$$0 = \frac{1}{2}\left[\mu\frac{\partial v}{\partial x} - \chi(\alpha)\right]_{N+\frac{1}{2}} + \frac{1}{2}\left[\mu\frac{\partial v}{\partial x} - \chi(\alpha)\right]_{N-\frac{1}{2}}. \tag{5.8}$$

Now let us return to (5.7) for $j = N$,

$$\frac{1}{x_{N+\frac{1}{2}} - x_{N-\frac{1}{2}}}\left\{\left[\alpha\mu\frac{\partial v}{\partial x} - \alpha\chi(\alpha)\right]_{N+\frac{1}{2}} - \left[\alpha\mu\frac{\partial v}{\partial x} - \alpha\chi(\alpha)\right]_{N-\frac{1}{2}}\right\}$$
$$= k\frac{\alpha_N}{1-\alpha_N}v_N$$

Substituting (5.8) into the above equation we have

$$-(\alpha_{N+\frac{1}{2}} + \alpha_{N-\frac{1}{2}})\left[\mu\frac{\partial v}{\partial x} - \chi(\alpha)\right]_{N-\frac{1}{2}} = k\frac{\alpha_N}{1-\alpha_N}(x_{N+\frac{1}{2}} - x_{N-\frac{1}{2}})v_N,$$

which gives

$$-(\alpha_{N+\frac{1}{2}} + \alpha_{N-\frac{1}{2}})\left[\mu\left(\frac{v_N - v_{N-1}}{x_N - x_{N-1}}\right) - \chi(\alpha_{N-\frac{1}{2}})\right]$$
$$= k\frac{\alpha_N}{1-\alpha_N}(x_{N+\frac{1}{2}} - x_{N-\frac{1}{2}})v_N.$$

This can be written

$$A_N^l v_{N-1} + A_N^d v_{c_N} = b(\alpha_N) \tag{5.9}$$

where $A_N^l = \frac{2\alpha_N\mu}{x_N - x_{N-1}}$, $A_N^d = -\frac{2\alpha_N\mu}{x_N - x_{N-1}} - k\frac{\alpha_N}{1-\alpha_N}(x_N - x_{N-1})$, and

$b(\alpha_N) = -2\alpha_N\chi\left(\frac{1}{2}(\alpha_N + \alpha_{N-1})\right)$.

The Sub-Algorithm for Step (2)

Let A_j^l, A_j^d and A_j^u $(j = 1, 2, ..., N)$ be the respective entries to the lower, main and upper diagonals of a matrix A, and $b(\alpha_j)$ be the entries of $\mathbf{b}(\alpha)$ $(j = 1, 2, ..., N)$. Hence, the cell velocity v is determined by solving

$$A\mathbf{v} = \mathbf{b}(\alpha). \tag{5.10}$$

5.3 Finding α using a moving mesh

Once C and v are determined over the region, we seek the solution of the time-dependent PDE (3.9) using a moving mesh approach. We will examine three different ways to move the mesh. For all three methods, the updated mesh is obtained from the mesh velocity used in an explicit time-stepping scheme.

The first way of moving the mesh (Method A) defines the mesh movement by keeping the cell mass fractions constant in time. The second uses the cell velocity as the mesh velocity (Method B). Both these methods move the mesh based on biological properties and so are intuitively relevant to the modelling of the problem. The third method (Method C) is more geometrical in nature, moving the nodes proportionally to the boundary movement. We describe each method in turn.

5.3.1 Method A

To find the mesh velocity of the nodes we assume conservation of mass fractions,

$$\frac{1}{\theta(t)} \int_0^{x_j(t)} \alpha \ dx \ = \ \gamma_j$$

where

$$\theta(t) = \int_0^{x_N(t)} \alpha \, dx,$$

is the total (variable) mass and γ_j is constant with respect to time, so that

$$\gamma_j \theta(t) \ = \ \int_0^{x_j(t)} \alpha \ dx. \tag{5.11}$$

Differentiating $\theta(t)$ with respect to time using Leibnitz integral rule gives,

$$\dot{\theta} \ = \ \int_0^{x_N(t)} \frac{\partial \alpha}{\partial t} \ dx + \left[\alpha \dot{x}\right]_0^{x_N(t)}.$$

Substituting in the PDE (3.9) and boundary conditions (3.13)- (3.14) gives

$$\begin{aligned} \dot{\theta} \ &= \ \int_0^{x_N(t)} \left\{ S(\alpha, C) - \frac{\partial}{\partial x}(\alpha v) \right\} \ dx + \alpha_N \dot{x}_N - \alpha(0,t)\dot{x}_0, \\ &= \ \int_0^{x_N(t)} S(\alpha, C) \ dx. \end{aligned} \tag{5.12}$$

It is worth noting that this corresponds to the global mass balance result

$$\frac{d}{dt}\int_0^{x_N(t)} \alpha \ dx = \int_0^{x_N(t)} S(\alpha, C) \ dx$$

which leads to equation (3.1).

Now let us return to (5.11) and apply the Leibnitz integral rule again,

$$\frac{d}{dt}\int_0^{x_j(t)} \alpha \ dx \ = \ \int_0^{x_j(t)} \frac{\partial \alpha}{\partial t} \ dx + \left[\alpha \dot{x}\right]_0^{x_j(t)}.$$

Substituting in the PDE (3.9) and using $\dot{x_0} = 0$, (the node x_0 does not move) we get

$$\begin{aligned}
\frac{d}{dt}\int_0^{x_j(t)} \alpha \ dx &= \int_0^{x_j(t)} \left\{ S(\alpha, C) - \frac{\partial}{\partial t}(\alpha v) \right\} \ dx + \alpha_j \dot{x_j} \\
&= \int_0^{x_j(t)} S(\alpha, C) \ dx - \left[\alpha v\right]_{x_j} + \alpha_j \dot{x}_j \\
&= \int_0^{x_j(t)} S(\alpha, C) \ dx - \alpha_j v_j + \alpha_j \dot{x}_j \\
&= \gamma_j \dot{\theta}.
\end{aligned}$$

By (3.1) again, and from (5.11) and (5.12),

$$\begin{aligned}
\dot{x}_j &= \frac{1}{\alpha_j} \left\{ \gamma \dot{\theta} - \int_0^{x_j(t)} S(\alpha, C) \ dx + \alpha_j v_j \right\} \\
&= \frac{\gamma_j}{\alpha_j} \int_0^{x_N(t)} S(\alpha, C) \ dx - \frac{1}{\alpha_j} \int_0^{x_j(t)} S(\alpha, C) \ dx + v_j
\end{aligned}$$

Time-Stepping

At each time level the new x_j is calculated using an explicit time-stepping scheme, such as the forward Euler scheme.

$$x_j^{m+1} = x_j^m + \Delta t \dot{x}_j^m,$$

and similarly θ is updated by

$$\theta^{m+1} = \theta^m + \Delta t \dot{\theta}^m.$$

Recovering the Solution α

To find an equation that allows us to calculate the solution, α, from the mesh we return to (5.11) and equate $\int \alpha \, dx$ at times t and 0 between the two points $(j+1)$ and $(j-1)$, as in

$$\frac{1}{\theta(t)} \int_{x_{j-1}(t)}^{x_{j+1}(t)} \alpha(x,t) \;\; dx \;\; = \;\; \frac{1}{\theta(0)} \int_{x_{j-1}(0)}^{x_{j+1}(0)} \alpha(x,0) \;\; dx.$$

Applying the mean value theorem for integrals and taking the mean value to be at $x_j(t)$, we obtain the approximation

$$\frac{1}{\theta(t)} \left(x_{j+1}(t) - x_{j-1}(t)\right) \alpha(x_j,t) \;\; = \;\; \frac{1}{\theta(0)} \left(x_{j+1}(0) - x_{j-1}(0)\right) \alpha(x_j,0).$$

We rearrange this equation to make α_j the subject, giving the approximation

$$\alpha_j = \alpha(x_j,t) = \frac{\theta(t)}{\theta(0)} \, \frac{(x_{j+1}(0) - x_{j-1}(0))}{(x_{j+1}(t) - x_{j-1}(t))} \, \alpha(x_j,0).$$

Referring back to the general algorithm in section 5.3), we can now give more details about steps (3) and (5) for method A,

Preliminary: Set up an initial mesh and an initial α. Calculate an initial value for $\theta(t)$ using the initial conditions, $\theta(0) = \int_0^{x_N(t)} \alpha(x,0) \;\; dx$.

3(a): Calculate $\dot{\theta}(t) = \int_0^{x_N(t)} S(\alpha, C) \;\; dx$.

3(b): Calculate the mesh velocity $\dot{x}_j$ from

$$\dot{x}_j = \frac{1}{\alpha_j} \left\{ \gamma_j \dot{\theta}(t) - \int_0^{x_j(t)} S(\alpha, C) \;\; dx + \alpha_j v_j \right\} \qquad\qquad (j = 1, 2, ..., N)$$

4(a): Update the θ values using an explicit time-stepping scheme.

4(b): Update the mesh nodes using an explicit time-stepping scheme.

5: Find the updated solution using

$$\alpha_j(t) = \frac{\theta(t)}{\theta(0)} \, \frac{(x_{j+1}(0) - x_{j-1}(0))}{(x_{j+1}(t) - x_{j-1}(t))} \, \alpha_j(0) \qquad\qquad (j = 0, 1, .., N-1)$$

where by symmetry $x_{-1} = x_1$. A one-sided approximation is used to find α_N.

We now move on to Method B.

5.3.2 Method B

Under this strategy the velocity of the boundary is equal to the velocity of the cells at the boundary. Then

$$\begin{aligned} \frac{dx_j}{dt} = \dot{x}_j &= v(x_j, t) \qquad\qquad (j = 1, 2, .., N) \\ \dot{x}_0 &= 0. \end{aligned}$$

Once the mesh velocity is defined as above, the new mesh can be determined by an explicit time stepping scheme, as in Method A.

To recover α on this new mesh *in a conservative manner* we define the partial masses

$$\Theta_j = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \alpha \ dx. \tag{5.13}$$

Differentiating Θ_j with respect to time, using Leibnitz integral rule, where $\dot{x}_j = v_j$,

$$\begin{aligned} \dot{\Theta}_j &= \frac{d}{dt} \int_{x_{j-1}(t)}^{x_{j+1}(t)} \alpha \ dx, \\ &= \int_{x_{j-1}(t)}^{x_{j+1}(t)} \frac{\partial \alpha}{\partial t} \ dx + \left[\alpha \dot{x}\right]_{j-1}^{j+1} \\ &= \int_{x_{j-1}(t)}^{x_{j+1}(t)} \frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(\alpha v) \ dx. \end{aligned}$$

Hence the terms under the integral are equal to one side of the PDE (3.9), so can be replaced by the source term,

$$\begin{aligned} \dot{\Theta}_j &= \int_{x_{j-1}}^{x_{j+1}} \left[\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(\alpha v)\right] \ dx \\ &= \int_{x_{j-1}}^{x_{j+1}} S(\alpha, C). \end{aligned}$$

We update Θ_j at the new time by using an explicit time stepping scheme. We then use the new value for Θ and find the updated solution by same the mid-point approximation as in Method A applied to (5.13),

$$(x_{j+1} - x_{j-1})\alpha_j = \Theta_j$$

giving

$$\alpha_j = \frac{\Theta_j}{x_{j+1} - x_{j-1}}.$$

The detailed algorithm for steps (3) to (5) for Method B is

Preliminary: Set up an initial Θ_j, and initial α. Calculate the initial values for Θ_j by using the initial conditions.

3(a): Define the mesh velocity as

$$\dot{x}_j \quad = \quad v(x_j, t) \qquad\qquad \text{where } j = 1, 2, .., N.$$

3(b): Find $\dot{\Theta}_j = \int_{j-1}^{j+1} S(\alpha, C)\ dx$.

4(a): Update the mesh nodes using an explicit time-stepping scheme

4(b): Update the $\Theta_j(t)$ value using an explicit scheme.

5: Find the updated solution using

$$\alpha_j = \frac{\Theta_j}{x_{j+1} - x_{j-1}} \qquad (j = 1, 2, ..., N-1).$$

A one-sided approximation is used for $j = 0, N$.

We now gives details of Method C.

5.3.3 Method C

As stated in equation (3.14), the velocity of the boundary $x_N(t)$ moves at a rate equal to the velocity of the cells at the boundary. For the third moving mesh method, we will move all of the nodes proportional to the boundary movement,

$$\dot{x}_j = \frac{x_j(t)}{x_N(t)} \dot{x}_N.$$

Once the mesh velocity is defined as above, the new mesh can be determined by an explicit time-stepping scheme. It is worth noting that this mapping is equivalent to the numerical mapping in [4], (see Chapter 7).

As with Method B we recover α on the new mesh by defining

$$\Theta_j = \int_{x_{j-1}(t)}^{x_{j+1}(t)} \alpha\ dx.$$

Differentiating Θ_j with respect to time, using Leibnitz integral rule,

$$\begin{aligned}\dot{\Theta}_i &= \frac{d}{dt}\int_{x_{j-1}}^{x_{j+1}} \alpha \ dx \\ &= \int_{x_{j-1}}^{x_{j+1}} \frac{\partial \alpha}{\partial t} \ dx + [\alpha \dot{x}]_{j-1}^{j+1} \\ &= \int_{x_{j-1}}^{x_{j+1}} S(\alpha, C) \ dx + \Big[\alpha(\dot{x} - v)\Big]_{x_{j-1}}^{x_{j+1}}.\end{aligned}$$

Note that there is an extra term in $\dot{\Theta}_j$ compared to that of Method B, as for this method $\dot{x} \neq v$. We update Θ at the new time by using an explicit time stepping scheme. As with Method B, we find the updated solution by the mid-point approximation,

$$(x_{j+1} - x_{j-1})\alpha_j = \Theta_j$$

giving

$$\alpha_j = \frac{\Theta_j}{x_{j+1} - x_{j-1}}.$$

(This method is conservative and different to the non-conservative method used in [4], (see Chapter 6)). The detailed algorithm for steps (3) to (5) for Method C is

Preliminary: Set up an initial Θ_j, and initial α. Calculate the initial values for Θ_j by using the initial conditions. Set up an initial Θ,

3(a) Define the mesh velocity as

$$\dot{x}_j = \frac{x_j}{x_N}\dot{x}_N \qquad \text{where } j = 1, 2, .., N.$$

3(b) Find $\dot{\Theta} = \int_{x_{j-1}}^{x_{j+1}} S(\alpha, C) \ dx + \Big[\alpha(\dot{x} - v)\Big]_{x_{j-1}}^{x_{j+1}}$.

4(a) Update the mesh nodes using an explicit time-stepping scheme

4(b) Update the $\Theta(t)$ value using an explicit scheme.

5 Find the updated solution using

$$\alpha_j = \frac{\Theta_j}{x_{j+1} - x_{j-1}} \qquad (j = 1, 2, ..., N-1).$$

A one-sided approximation is used for $j = 0, N$.

Chapter 6

Breward et al.'s Method

In [4] the same tumour growth problem is solved by mapping the variable $x(t)$ to a fixed domain $\xi \in [0,1]$ by the transformation $\xi = \frac{x(t)}{\ell(t)}$ and $\tau = t$, where $\ell = x_N(t)$. Using the chain rule of Chapter 4.1, the transformed problem reads

$$\frac{\partial \alpha}{\partial \tau} - \frac{\xi}{\ell}\frac{d\ell}{d\tau}\frac{\partial \alpha}{\partial \xi} + \frac{1}{\ell}\frac{\partial}{\partial \xi}(\alpha v) = \frac{(1+s_1)C}{1+s_1C}\alpha(1-\alpha) - \frac{s_2+s_3C}{1+s_4C}\alpha, \tag{6.1}$$

$$\ell\frac{\partial}{\partial \xi}\big(\alpha\zeta(\alpha)\big) + \frac{k\ell^2\alpha v}{1-\alpha} = \mu\frac{\partial}{\partial \xi}\left(\alpha\frac{\partial v}{\partial \xi}\right), \tag{6.2}$$

$$\frac{\partial^2 C}{\partial \xi^2} = \frac{Q\ell^2\alpha C}{1+Q_1C}, \tag{6.3}$$

with initial and boundary conditions

$$\ell = 1, \;\; \alpha = \alpha_0 \quad \text{at} \quad \tau = 0, \tag{6.4}$$

$$\frac{\partial C}{\partial \xi} \;=\; v \;=\; 0 \quad \text{at} \quad \xi = 0, \tag{6.5}$$

$$\mu\frac{\partial v}{\partial \xi} = \ell\zeta(\alpha), \;\; C = 1 \quad \text{at} \quad \xi = 1, \tag{6.6}$$

$$\frac{d\ell}{d\tau} = v \quad \text{at} \quad \xi = 1, \tag{6.7}$$

where k and μ have the same definition as before, and the pressure difference between the two phases $\chi(\alpha)$ is defined in the special case $\hat{\Sigma}_c = r = 1$ and

$q = 2$, as

$$\chi(\alpha) = \zeta(\alpha) = \begin{cases} 0 & 0 \leq \alpha < \alpha_{min} \\ \frac{\alpha - \alpha^*}{(1-\alpha)^2} & \alpha_{min} \leq \alpha < 1. \end{cases}$$

To compare the results from the moving mesh method to those in [4] we have replicated their results. Using the above set of equations we postulate the algorithm

Preliminary: Obtain an initial C, v, ℓ and α

(1) Find C using the finite difference method of Step 1 on (6.3)

(2) Find v using the finite difference method of Step 2 on (6.2)

(3) Find α at the new time level by explicitly time-stepping (6.1)

(4) Find the tumour radius at the new time level by explicitly time-stepping $\dot{\ell} = v_N$

When comparing this to the algorithm in Chapter 4.3, we see that Steps 1 and 2 are essentially the same. However for this algorithm we then go straight to calculating the solution α on the transformed mesh, and then update the tumour radius. By contrast, with the moving mesh methods A, B and C of the previous section, after finding C and v, the nodes' positions were calculated (including the final node which represents the tumour radius), and then the integral of the solution is recovered via $\int \alpha \, dx$, (see Chapter 4.1). Another distinction between Breward et al.'s method and the moving mesh method is between non-conservative and conservative formulations.

6.1 Algorithm for numerically solving the transformed problem

Finding the concentration C

Let us discretise (6.3),

$$C_{j-1} - 2C_j + C_{j+1} = w(C_j)$$

where $w(C_j) = \frac{Q\ell^2 \alpha_j C_j}{1+Q_1 C_j}(\Delta\xi)^2$ for $j = 1, 2, ..., N-2$. Note that as the mesh is equally spaced, the differences in ξ_j can be taken to be $\frac{1}{N}$. At the boundaries we

have $C_{-1} = C_1$ and $C_N = 1$, from (6.5) and (6.6) respectively. To correspond with these conditions, the above equation for the special cases $j = 0$ and $j = N - 1$ are

$$\begin{aligned} -2C_0 + 2C_1 &= \frac{Q\ell^2\alpha_j C_0}{1+Q_1 C_0}(\Delta\xi)^2 &= w(C_0) \\ C_{N-2} - 2C_{N-1} &= \frac{Q\ell^2\alpha_j C_{N-1}}{1+Q_1 C_{N-1}}(\Delta\xi)^2 - 1 &= w(C_{N-1}) \end{aligned}$$

As in Chapter 5.1, we write the non-linear system as

$$T\mathbf{C} = \mathbf{w}(\mathbf{C})$$

where

$\mathbf{C}$ is a vector of C_0 to C_{N-1}, $j = 0, 1, ..., N-1$,

$\mathbf{w}(\mathbf{C})$ is a vector of $w(C_j)$, $j = 0, 1, ..., N-1$,

and

T is a tridiagonal matrix of the C_j coefficients,

$$T = \begin{pmatrix} -2 & 2 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix}.$$

The sub-algorithm for calculating C is the same as in Chapter 5, namely

Preliminary: Make an initial guess for $\mathbf{C}$.

1(a): Use $\mathbf{C}^p$ to find $\mathbf{w}(\mathbf{C}^p)$;

1(b): Find the residual, $\mathbf{R}^p = T\mathbf{C}^p - \mathbf{w}(\mathbf{C}^p)$.

1(c): Calculate the Jacobian of $\mathbf{R}^p$,

$$J^p = \frac{\partial \mathbf{R}^p}{\partial \mathbf{C}^p} = T - \left\{ \frac{\partial w_i^p}{\partial C_j^p} \right\}$$

where $i, j = 0, 1, ..., N-1$ and $\left\{ \frac{\partial w_i^p}{\partial C_j^p} \right\}$ is the diagonal matrix

$$\frac{\partial w_i}{\partial C_j} = \begin{pmatrix} \frac{Q\ell^2\alpha_0}{(1+Q_1C_0)^2}(\Delta\xi)^2 & 0 & \cdots & 0 \\ 0 & \frac{Q\ell^2\alpha_1}{(1+Q_1C_1)^2}(\Delta\xi)^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{Q\ell^2\alpha_{N-1}}{(1+Q_1C_{N-1})^2}(\Delta\xi)^2 \end{pmatrix}.$$

1(d): Find $\mathbf{H}^p = (J^p)^{-1}\mathbf{R}^p$.

1(e): Set $\mathbf{C}^{p+1} = \mathbf{C}^p - \mathbf{H}^p$.

1(f): With the new approximation $\mathbf{C}^{p+1}$, return to (a) and repeat until (e) converges, as measured by $\|\mathbf{C}^{p+1} - \mathbf{C}^p\|_2 < 1 \times 10^{-6}$.

In step (c) the entries to the diagonal matrix $\frac{\partial w_i^p}{\partial C_j^p}$ contain an extra factor of $\ell^2(\Delta\xi)^2$ when compared to Chapter 5, to accommodate the different $w(C_j)$.

Finding the velocity v

We find the velocity in the same manner as in Chapter 5.2. We discretise (6.2) and rearrange so that

$$A_j^l v_{j-1} + A_j^d v_j + A_j^u v_{j+1} = b(\alpha_j) \qquad (j = 2, ..., N-1). \tag{6.8}$$

where $A_j^l = \frac{(\alpha_j+\alpha_{j-1})\mu}{2(\Delta\xi)^2}$, $A_j^d = -\left(-\frac{(\alpha_j+\alpha_{j-1})\mu}{2(\Delta\xi)^2} - \frac{(\alpha_j+\alpha_{j+1})\mu}{2(\Delta\xi)^2} - \frac{k\ell^2\alpha_j}{1-\alpha_j}\right)$,

$A_j^u = \frac{(\alpha_j+\alpha_{j+1})\mu}{2(\Delta\xi)^2}$ for $j = 1, 2, ..., N-1$ and

$b(\alpha_j) = \frac{\ell}{4\Delta\xi}(\alpha_j + \alpha_{j+1})\zeta\left(\frac{1}{2}(\alpha_j + \alpha_{j+1})\right) - \frac{\ell}{4\Delta\xi}(\alpha_j + \alpha_{j-1})\zeta\left(\frac{1}{2}(\alpha_j + \alpha_{j-1})\right)$.

As with the moving mesh equations, $v_0 = 0$. For the case when $j = N$ we take the average of the boundary condition (6.6) and substitute it into the discrete form of (6.2) to get

$$A_N^l v_{N-1} + A_N^d v_N \;=\; b(\alpha_N),$$

where $A_N^l = \frac{2\mu\alpha_N}{(\Delta\xi)^2}$, $A_N^d = \frac{-2\mu\alpha_N}{(\Delta\xi)^2} - \frac{k\ell^2\alpha_N v_N}{1-\alpha_N}$ and $b(\alpha_N) = \frac{-\ell\alpha_{N+1}(\zeta(\alpha_N)-\zeta(\alpha_{N-1}))}{\Delta\xi}$.
Note that we are *assuming* that the boundary conditions are imposed in this way in [4].

Let A_j^l, A_j^d and A_j^u $(j = 1, 2, ..., N)$ be the respective entries to the lower, main and upper diagonals of matrix A, and $b(\alpha_j) = \mathbf{b}(\alpha)$ $(j = 1, 2, ..., N)$. Hence Step 2, finding the cell velocity v, is determined by solving

$$A\mathbf{v} = \mathbf{b}(\alpha).$$

As intended we have calculated C and v on a fixed mesh in the same way as we calculated them on a moving mesh. For the fixed mesh there is no mesh velocity to define, but we still need to compute the change in the tumour radius.

Finding the solution α

Finally, to obtain α on the fixed mesh, we discretise (6.1) explicitly in time, with a central difference approximation in space,

$$\begin{aligned}
\frac{\alpha_j^{i+1} - \alpha_j^i}{\Delta t} &- \frac{\xi v_N}{\ell}\frac{\alpha_{j+1}^i - \alpha_{j-1}^i}{2\Delta\xi} + \frac{v_{j+1}^i\alpha_{j+1}^i - v_{j-1}^i\alpha_{j-1}^i}{2\ell\Delta\xi} \\
&= \frac{(1+s_1)\alpha_j^i(1-\alpha_j^i)C_j^i}{1+s_1C_j^i} - \frac{s_2+s_3C_j^i}{1+s_4C_j^i}\alpha_j^i, \\
\Rightarrow \alpha_j^{i+1} &= \left(\frac{\xi v_N}{\ell}\frac{(\alpha_{j+1}^i - \alpha_{j-1}^i)}{2\Delta\xi} - \frac{v_{j+1}^i\alpha_{j+1}^i - v_{j-1}^i\alpha_{j-1}^i}{2\ell\Delta\xi}\right. \\
&+ \left.\frac{(1+s_1)\alpha_j^i(1-\alpha_j^i)C_j^i}{1+s_1C_j^i} - \frac{s_2+s_3C_j^i}{1+s_4C_j^i}\alpha_j^i\right)\alpha_j^i, \qquad (6.9) \\
&\qquad\qquad (j = 0, 1, ..., N).
\end{aligned}$$

A one-sided approximation to α_ξ is used at the boundaries. This scheme is non-conservative (see Chapter 4.1). We are *assuming* that the α are updated in this way in [4].

Finding the tumour radius

By (6.7) the tumour radius ℓ grows at the same rate as the cell velocity at the boundary v_N. We find the tumour radius at the new time level by using the explicit Euler time-stepping scheme,

$$\ell^{(i+1)} = \ell^{(i)} + \Delta t\, v_N.$$

6.2 Numerical Results for Breward et al.'s Method

It is important to remark that the numerical algorithm specified in this chapter is a surmise based upon the transformed problem given in [4]. However, solving the transformed problem using this numerical process, we successfully generate plots which replicate those shown in [4]. We can therefore be reasonably confident in our preliminary calculations in Chapter 4 where we find the concentration C and cell velocity v using finite differences.

The numerical results given in [4] focus on the qualitative nature of the solution of the model equations. Most of the figures concentrate on varying the viscosity and drag effects, μ and k. As that is not our primary aim here we concentrate on Figure 3 and Figure 8 from [4]. These two figures portray three different plots each, for two different sets of parameter values. Each figure shows,

- the volume fraction of cells in space (α against x) evolving over time,
- the cell velocity in space (v against x) evolving over time,
- the tumour radius over time.

In Figure 3 of [4] the parameter values exclude the effects of cellular attraction ($\alpha^* = \alpha_{min}$). The following description is taken from [4]. Initially the tumour cell volume fraction α increases toward the centre of the tumour. The tumour can relieve the stress at its outer boundary by expanding outwards and, thus, reducing the volume fraction there. Following this early phase, the tumour becomes sufficiently large that cells at the centre to start to die, so that a central necrotic core forms. The system then settles to a steady, travelling-wave solution, in which cells close to the edge of tumour move outwards (to relieve their stress), whilst cells nearer to the tumour centre move inwards. The plot of the tumour radius over time confirms that the system indeed settles to a steady, travelling-wave profile, after an initial period of growth, the speed with which the leading edge of the tumour advances is approximately constant.

In comparison to the parameter values used to produce Figure 3, the parameter values for Figure 8 have smaller μ and k (viscosity and drag), and more importantly $\alpha_{min} < \alpha^*$. Physically, this means that the cells sense each other using their filopodia and are attracted toward each other. As with Figure 3, α initially increases over the whole region (most notably at the centre of the

tumour), before decreasing at the centre only. When $\alpha_{min} = 0.6$, the tumour radius reaches a steady state before a necrotic core is established.

6.2.1 Convergence

We expect that as we take more nodes (N increases), the solution should converge. We choose the following set of parameters from [4]

$$\begin{aligned} Q = 0.5, \quad Q_1 = 0, \quad p = 1, \quad q = 2, \quad \hat{\Sigma}_c = 1, \\ s_1 = s_4 = 10, \quad s_2 = s_3 = 0.5, \\ k = 1, \quad \mu = 1, \quad \alpha_0 = \alpha_{min} = \alpha^* = 0.8. \end{aligned}$$

This set of parameter values was used in [4] to create figure 3 from the paper. For the purpose of investigating convergence, we take the solution at the arbitrary time $t = 4$ for various N, and calculate the relative error in the L_2 norm of the solution and the tumour radius to examine convergence. We have used central differences to approximate α, so the solution should be up to second order in $\Delta\xi$. And we use an explicit Euler time-step, so the solution is first order in time. To keep the contributions to the truncation error in balance, the time step is quartered as the number of nodes doubles.

There are two different variables that we test for convergence. The first is $\alpha(t)$, the solution of (6.1), the second is the tumour radius $\ell(t)$.

There is no exact solution to the problem, so we investigate whether a Cauchy convergence criterion is satisfied, where we compare the solution generated from $N = 5, 10, 20, 40$ with the solution from $N = 80$.

The solution is represented as a vector of length $N + 1$,

$$\alpha = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_j \\ \vdots \\ \alpha_N \end{pmatrix}.$$

To enable comparison between solutions for different numbers of nodes, the solution vector α was reduced to a standard size of six by taking the $\frac{mN}{5}$ $(m = 0, 1, ..., 5)$ entries only.

NOTATION: For the purpose of discussing convergence, the notation α^N represents the solution vector α at $t = 4$, calculated with N nodes. Similarly ℓ^N is ℓ at $t = 4$, calculated with N nodes.

We define the relative error of α^N and ℓ^N as

$$\begin{aligned} \text{rel.err.}(\alpha)^N &= \sqrt{\frac{\sum_{j=0}^{N}(\alpha_j^N - \alpha_j^{80})^2}{\sum_{j=0}^{N}(\alpha_j^{80})^2}} \\ \text{rel.err.}(\ell)^N &= \frac{\left|\ell^N - \ell^{80}\right|}{\ell^{80}} \end{aligned}$$

respectively. Let us assume that the relative error is proportional to the order of convergence p by

$$\text{rel. err.}^N \propto \frac{1}{N^p}$$

for large N. Then

$$\frac{\text{rel.err.}^{2N}}{\text{rel.err.}^N} = \frac{N^p}{(2N)^p} = \frac{1}{2^p}.$$

Hence the order of convergence is

$$p = \frac{-\ln(\text{rel.err.}^{2N}/\text{rel.err.}^N)}{\ln 2}. \tag{6.10}$$

So, we can find the order of convergence by taking a ratio of consecutive relative errors, at $t = 4$.

N	Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(ℓ)	$p(\ell)$
5	0.020	8.025×10^{-4}		7.319×10^{-3}	
10	0.005	3.164×10^{-4}	1.3	3.668×10^{-3}	1.0
20	0.0013	1.242×10^{-4}	1.3	1.643×10^{-3}	1.2
40	3.125×10^{-4}	3.963×10^{-5}	1.6	5.623×10^{-4}	1.5

The table above shows that, as expected, the relative error for both α and ℓ decreases as the number of nodes increases, and ℓ has a slightly smaller error. We would expect the ratio of errors to converge as N increases and is compared to $N >> 80$, hence the order of convergence could be determined. From the table above, we can only assess that the order of convergence is larger than one

for both α and ℓ. Nonetheless, we can be reasonably confident that the solution and the tumour boundary converges.

Chapter 7

Numerical results for moving mesh methods

In this section we use the moving mesh methods described earlier to present numerical simulations of the non-dimensionalised model, equations (3.9) to (3.14), in several parameter regimes. Our aims are to compare the three different moving mesh methods, and also to compare the results with existing mesh numerical simulations in [4].

Firstly, we investigate the order of convergence for the three different moving meshes. We also present plots showing how the different moving meshes advance over time. Finally, we compare our moving mesh results with the numerical results in [4]. To do this we show how the volume fraction of cells α, the cell velocity v and the tumour radius x_N evolve over time, for two different sets of parameter values.

Concerning the time-stepping process, the results here are generated from either the explicit Euler time-stepping scheme or a built-in adaptive Matlab solver. Several different Matlab ODE solvers were investigated before settling upon ODE23 with a specified maximum time step, to ensure that the nodes remain ordered and the plots remain stable. Although other Matlab ODE solvers such as ODE45 and ODE15s were successful, ODE23 proved to be sufficient and requires less computation (indicating that the problem is not particularly stiff). It is worth noting that all three moving mesh methods, combined with either the Euler explicit time-stepping scheme or ODE23, have a manageable computation time for $N \leq 80$.

The results given here are the non-dimensionalised values. To convert back to the dimensionalised values, scale back to the non-dimensionalised variables shown in Chapter 3.2.

7.1 Convergence

As with the method in Chapter 6, we would expect that as we take more nodes (N increases), the solution should converge. We choose the same parameters as in Chapter 6.2.1.

Again, we take the solution α at $t = 4$ for various N. As the mesh x_j varies in time, we also require the x_j to converge. By this we mean that as more nodes are used, the mesh settles to particular position. For example, it would be desirable if at time t every other node of a mesh for $N = 80$ lined up with the mesh $N = 40$. The mesh is represented as a vector of nodes of length $N+1$. To compare coarser meshes with finer meshes, the different vectors of nodes are all reduced to be of length six. This is done in the same manner that the solution vector was reduced in Chapter 6.2.1: by taking the $\frac{mN}{5}$ $(m = 0, 1, ..., 5)$ entries only. The various solution vectors α^N are also defined in this fashion. Using the same notation from Chapter 6, the relative errors of α and x for N nodes are taken to be

$$\text{rel. error}(\alpha)^N = \sqrt{\frac{\sum_{j=0}^{N}(\alpha_j^N - \alpha_j^{80})^2}{\sum_{j=0}^{N}(\alpha_j^{80})^2}}$$

$$\text{rel. error}(x)^N = \sqrt{\frac{\sum_{j=0}^{N}(x_j^N - x_j^{80})^2}{\sum_{j=0}^{N}(x_j^{80})^2}},$$

for $N = 5, 10, 20, 40$, and compared with the solution from $N = 80$. To find the order of convergence we use the result (6.10)

$$p = \frac{-\ln(\text{rel.err.}^{2N}/\text{rel.err.}^{N})}{\ln 2}.$$

The first three tables below presents the results for all three methods using ODE23. The maximum time step size was halved as N doubled because ODE23 is second order in time, and the mid-point rule (which is used to recover α) is second order in space. As before, we wish to keep the contributions to the truncation error in balance, hence Δt and Δx decrease at the same rate.

The last three tables present the results when the explicit Euler time-

stepping scheme was used. In this case as the number of nodes doubled, the time-step was quartered. This decision was made as the solution α is recovered using a mid-point approximation, which is second order in space, and the explicit Euler time-stepping scheme is first order in time.

We would expect the solutions to converge quicker where using the ODE23 solver because this uses an approximation based upon Runge-Kutta 2 and 3, which have a higher order of accuracy than Euler time-stepping. However, we should be careful to note that the timesteps are considerably larger when using ODE23.

Table 7.1: Relative errors for α and x with rates of convergence for Method A using Matlab function ODE23.

N	Max. Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	2.262×10^{-3}		5.087×10^{-2}	
10	0.01	1.042×10^{-3}	1.1	2.323×10^{-2}	1.1
20	0.005	4.512×10^{-4}	1.2	9.918×10^{-3}	1.2
40	0.0025	1.534×10^{-4}	1.6	3.300×10^{-3}	1.2

Table 7.2: Relative errors for α and x with rates of convergence for Method B using Matlab function ODE23.

N	Max. Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	4.052×10^{-4}		2.099×10^{-3}	
10	0.01	1.199×10^{-4}	1.8	4.837×10^{-4}	2.1
20	0.005	3.267×10^{-5}	1.9	1.267×10^{-4}	1.9
40	0.0025	8.130×10^{-6}	2.0	2.596×10^{-5}	2.3

Table 7.3: Relative errors for α and x with rates of convergence for Method C using Matlab function ODE23.

N	Max. Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	3.561×10^{-4}		1.661×10^{-3}	
10	0.01	1.033×10^{-4}	1.8	4.580×10^{-4}	1.9
20	0.005	2.755×10^{-5}	1.9	1.148×10^{-4}	2.0
40	0.00025	6.749×10^{-6}	2.0	2.350×10^{-5}	2.3

Examining these tables we observe that when we use the explicit Euler time-stepping scheme, we get slightly quicker convergence. This is most likely due

Table 7.4: Relative errors for α and x with rates of convergence for Method A using the explicit Euler time-stepping scheme.

N	Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	4.368×10^{-4}		1.117×10^{-3}	
10	0.005	1.167×10^{-4}	1.9	2.423×10^{-4}	2.2
20	1.25×10^{-3}	2.725×10^{-5}	2.1	6.549×10^{-5}	1.9
40	3.125×10^{-4}	5.494×10^{-5}	2.3	1.361×10^{-5}	2.3

Table 7.5: Relative errors for α and x with rates of convergence for Method B using the explicit Euler time-stepping scheme.

N	Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	7.534×10^{-4}		2.524×10^{-3}	
10	0.005	2.788×10^{-4}	1.4	3.142×10^{-4}	3.0
20	1.25×10^{-3}	1.001×10^{-4}	1.5	9.662×10^{-5}	1.7
40	3.125×10^{-4}	3.096×10^{-5}	1.7	1.931×10^{-5}	2.3

Table 7.6: Relative errors for α and x with rates of convergence for Method C using the explicit Euler time-stepping scheme.

N	Δt	**rel.err.**(α)	$p(\alpha)$	**rel.err.**(x)	$p(x)$
5	0.02	6.460×10^{-4}		6.727×10^{-5}	
10	0.005	2.438×10^{-4}	1.4	1.255×10^{-5}	2.4
20	1.25×10^{-3}	8.942×10^{-5}	1.4	3.101×10^{-6}	2.0
40	3.125×10^{-4}	2.804×10^{-5}	1.7	6.428×10^{-7}	2.3

to the smaller Δt used.

In all six tables we note that the mesh x has a slightly higher order of convergence than the solution α. This is possibly because α comes from a midpoint approximation that depends on x, so the order of convergence of α cannot be higher than that of the mesh. The difference of convergence rates between α and x was more apparent when Euler time-stepping was implemented (Tables 8.4 to 8.6).

We see that Method A, together with ODE23 (Table 8.1), has the slowest convergence, whereas the same method, but with the explicit Euler time-stepping scheme converges significantly faster (Table 8.4). Moreover, Method A with the explicit Euler time-stepping scheme (Table 8.4) converges at a rate very similar to that of Method B and Method C, using ODE23 (Tables 8.2 and 8.3). As with Method A, Methods B and C show little similarity in their

convergence behaviour when comparing Euler time-stepping and using ODE23. Yet, within a chosen time-stepping scheme, Method B and C have nearly identical convergence rates - especially when when using ODE23 (Tables 8.2 and 8.3). When comparing Methods B and C with the explicit Euler time-stepping (Tables 8.5 and 8.6), we see that the α converge similarly, but the order of convergence of x for Method B (Table 8.5) appears to be behaving erratically, albeit for the small data sample obtained here. The mesh from Method C, together with Euler's time-stepping (Table 8.6), seems to have the highest rate of convergence.

It appears that generally the mesh approaches an order of convergence larger than two, whilst α may prove to be of second order convergence. However, we cannot be sure of the order of convergence in any of the cases without having more data, and comparing the solutions to data retrieved using $N >> 80$. Even so, we can be reasonably confident that the solution and mesh converge for all three moving mesh methods.

7.2 Comparison with Breward et al.'s method [4]

7.2.1 Comparing Figure 3 from [4]

We wish to compare our numerical methods to the mesh method used in [4]. We generate results using the same parameters above to compare our results with Figure 3 in [4]. All three methods were investigated, using both the explicit Euler time-stepping scheme and ODE23. Throughout this section, we take $N = 80$, $\Delta t = 0.0075$ and run until $t = 75$.

Methods A and C produce very similar plots to each other, regardless of the time-stepping approach. For this reason, only the results from Method A are included here.

The explicit Euler time-stepping scheme and ODE23 generate very similar solutions for α. However, due to the nature of ODE23 it is less straightforward to produce plots for the velocity, as it is only an intermediary stage to calculate α. For this reason, all the results presented here are generated using the explicit Euler time-stepping scheme.

Figures 8.1, 8.3 and 8.5 are due to Method A and display the same characteristics as the results in [4] for the same parameters. For α, the boundary values remain fairly constant, and α at the centre of the tumour decreases at a steady rate. The velocity increases rapidly as it gets closer to the boundary, but the velocity at the boundary itself stays constant over time for $t \geq 37.5$.

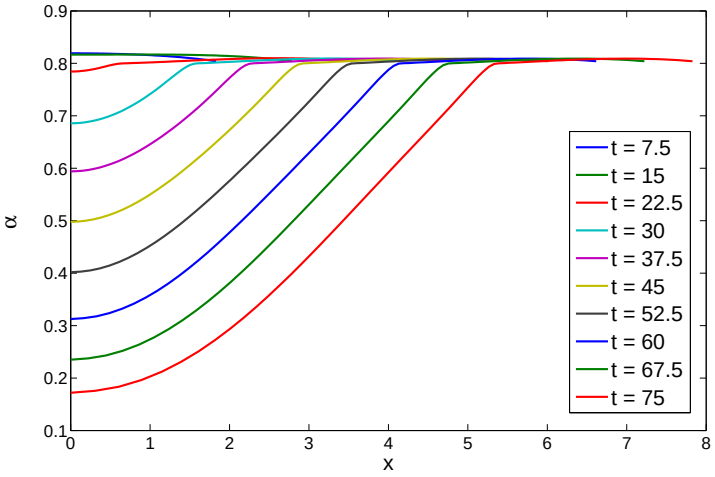

Figure 7.1: Solution α for Method A using explicit Euler time-stepping - α varying with x at different times

.

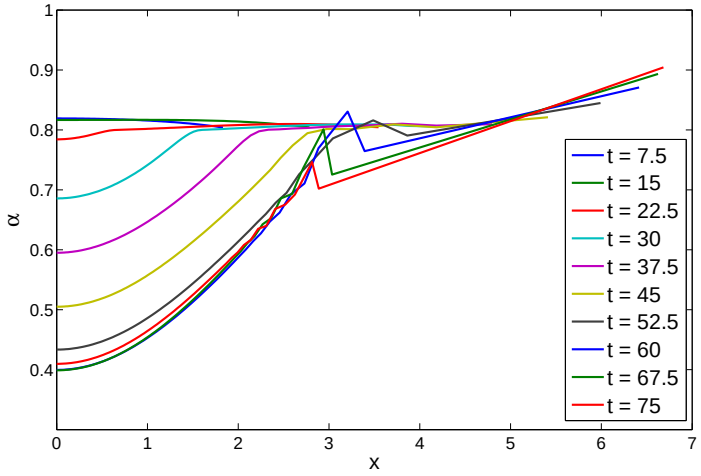

Figure 7.2: Solution α for Method B using explicit Euler time-stepping - α varying with x at different times

.

The plots for α and v present the travelling wave characteristics apparent in [4]. The region of negative velocity is subtly different to the same region in [4], where the minimum is slightly less rounded than shown in Figure 8.3. Interestingly, Method C also presented rounder minima, but the method in Chapter 6 presented troughs that resembles those in [4].

Method B appears to behave like Method A and [4] at earlier times, but after approximately $t = 45$, α appears to grow at the boundary, and no longer decreases at a regular rate at the centre of the tumour. The plots from Method B are less smooth, despite the same number of nodes used for both methods. There is a considerable kink in α and v for $t = 45$ which appears to dampen for later times. The solution α does not drop below 0.4 at the centre of the tumour, even for $t = 100$ (not shown here). The key characteristics remain even for smaller Δt, and also when using ODE23 - suggesting that this behaviour is due to the numerical method. The processes of Method B and Method C are very similar, and as Method C behaves as in Figures 8.1 and 8.3, it is reasonable to conclude that tracking the cell velocity with the mesh nodes can result in the mesh becoming too coarse in some areas. This is a problem that could be compounded over time, especially where the cell velocities vary between positive and negative. At this point, the nodes would be moving in opposite directions, leaving a considerable deficit in between. Indeed if we look at Figure 8.4 for $t = 75$ we see that the velocity is mostly negative, so the nodes are moving to the left.

The differences between Method B and the results presented in [4] are more apparent in Figure 8.4. We can see more clearly that it appears to become less smooth when t is larger than 45. We see the velocity at the boundary decreasing considerably with time. This behaviour is not seen in [4], nor in Methods A and C.
The plots of the tumour radius (Figures 8.5 and 8.6) are consistent with [4]. As expected, the tumour is steadily growing. However, the differences from the results from Method B are apparent in Figure 8.6 because the growth is slowing down. This corresponds to Figure 8.4 where we can see that the velocity is nearly constant for $t = 75$, meaning that in Method B the nodes would not be moving much.

7.2.2 Comparing Figure 8 from [4]

To align our parameters with those used to produce Figure 8 in [4] we only change the following values:

$$\alpha_{min} = 0.6, \qquad \mu = k = 0.25.$$

Again, we take $N = 80$ with $\Delta t = 0.0075$ for Methods A and C, which produce very similar plots, so only results from Method A are included here. For Method B we take the same number of nodes $N = 80$, but with $\Delta t = 0.005$. Also, as

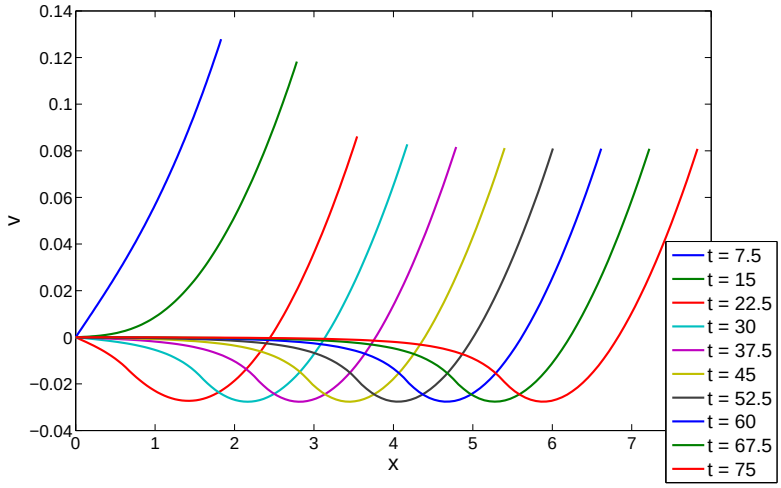

Figure 7.3: Velocity for Method A using explicit Euler time-stepping - v varying with x at different times

.

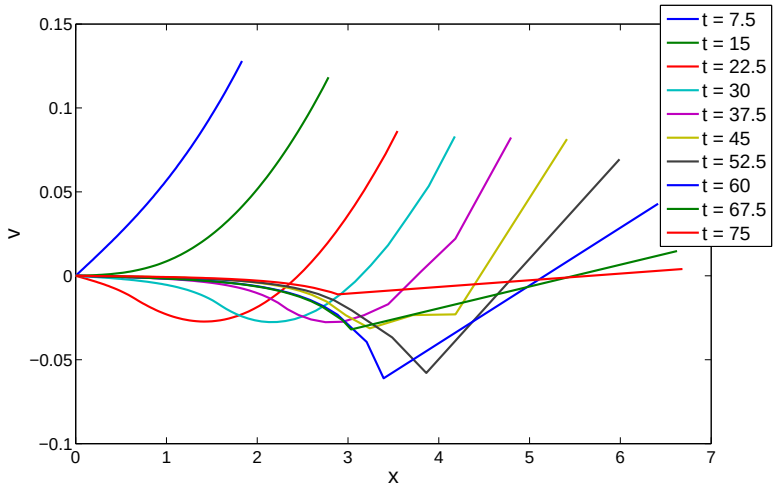

Figure 7.4: Velocity for Method B using explicit Euler time-stepping - v varying with x at different times

.

the ODE solver and the explicit Euler time-stepping scheme produce similar plots, only results from the explicit Euler time-stepping scheme are included.

We see that, as in [4] the tumour reaches a steady-state - with the tumour growth remaining constant after $t = 40$ (Figure 8.11). Figure 8.9 shows that once the steady state is reached (the cell velocity at the boundary drops to

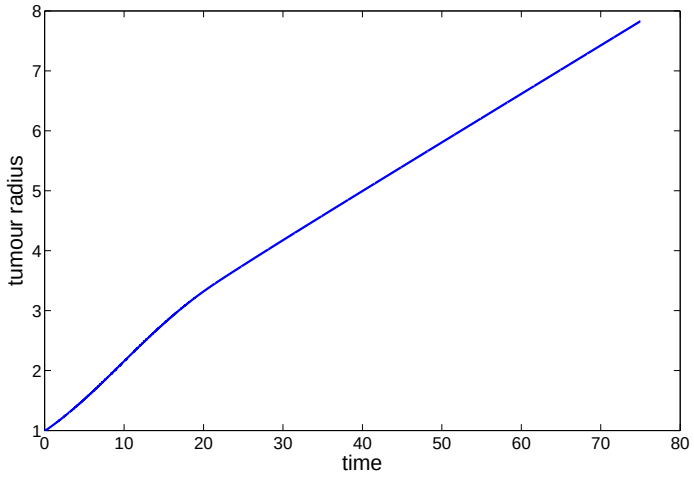

Figure 7.5: The tumour radius evolving over time for Method A using explicit Euler time-stepping.

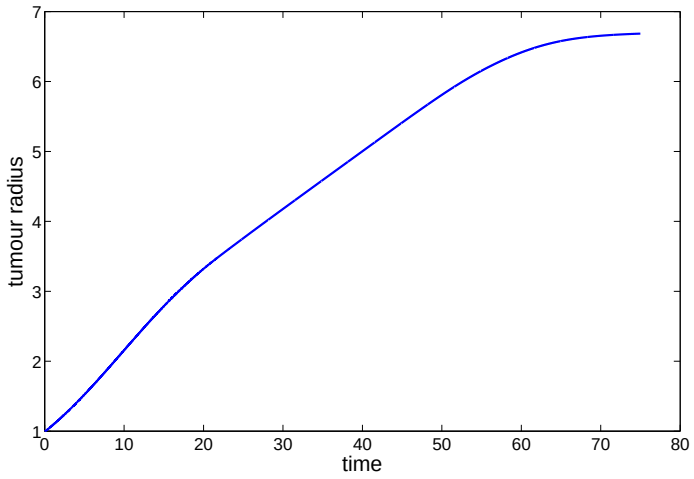

Figure 7.6: The tumour radius evolving over time for Method B using explicit Euler time-stepping.

zero), all cells within the region have negative velocity, i.e. the cells are moving inwards.

This movement of cells has serious consequences when moving the mesh as in Method B. We see in Figures 8.8, 8.10 and 8.12 that for $t > 40$ the method struggles to produce believable results. As the nodes are tracking the cell velocity, at each time level they are clustering more and more to the left, which produces erratic results.

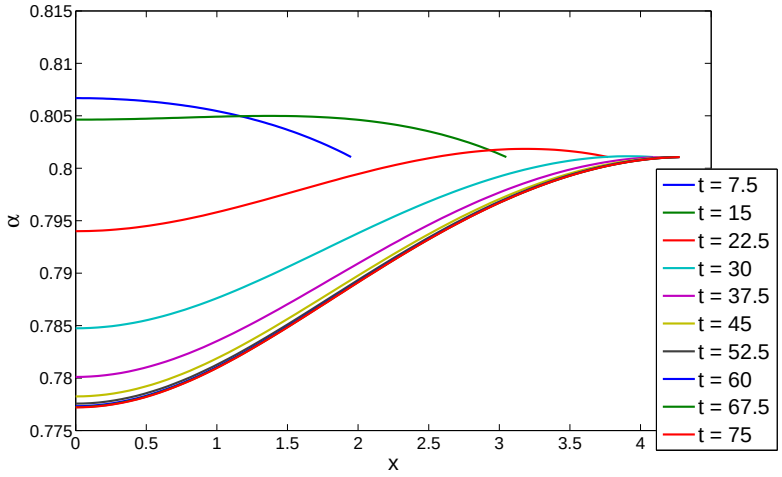

Figure 7.7: Solution α for Method A using explicit Euler time-stepping - α varying with x at different times

.

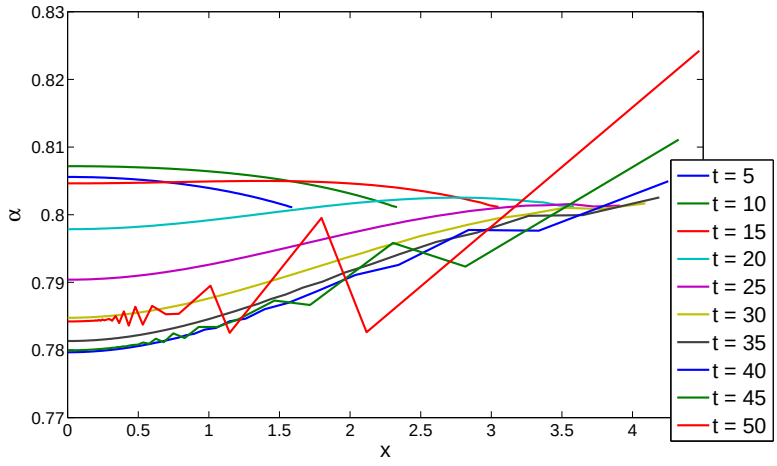

Figure 7.8: Solution α for Method B using explicit Euler time-stepping - α varying with x at different times

.

7.2.3 Examining how the nodes move

Let us define the computational space $\xi \in [0, 1]$. We examine how the mesh moves for different approaches of defining the mesh velocity. We take the parameters that produce a steady travelling-wave profile ($\alpha_{min} = 0.8$, $k = \mu = 1$). When the nodes move by conserving relative mass, Method A, the nodes remain

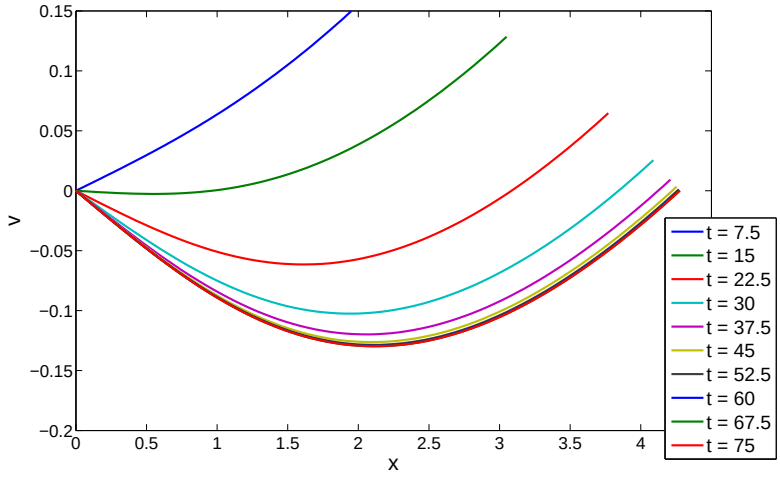

Figure 7.9: Velocity for Method A using explicit Euler time-stepping - v varying with x at different times

.

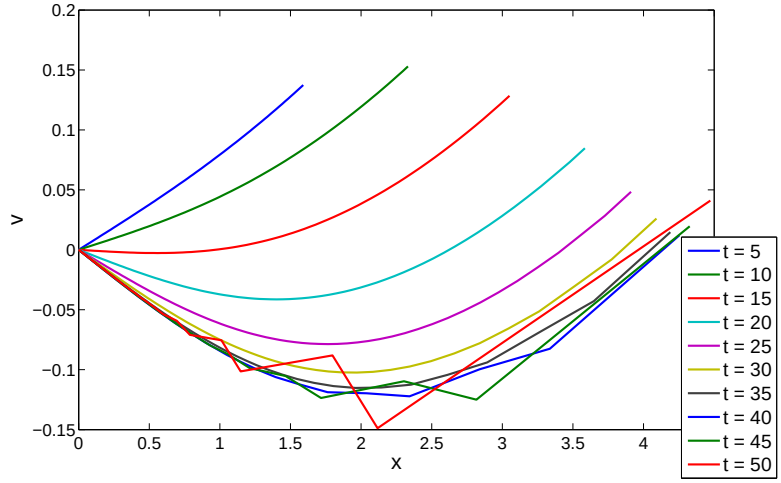

Figure 7.10: Velocity for Method B using explicit Euler time-stepping - v varying with x at different times

.

nearly equally spread with time up to $t = 60$, only gathering very slightly for $0.2 < \xi < 0.5$. This gathering is more noticeable for longer time. We would expect the gathering to be more prominent and start closer to $\xi = 0$ as time continues, because nodes appear to always be moving out as the tumour grows, i.e. each node only moves to the right as the tumour grows. Method C moves

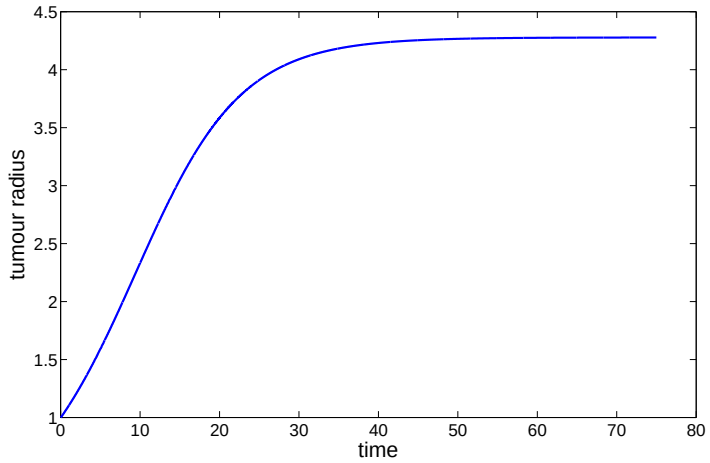

Figure 7.11: The tumour radius evolving over time for Method A using explicit Euler time-stepping.

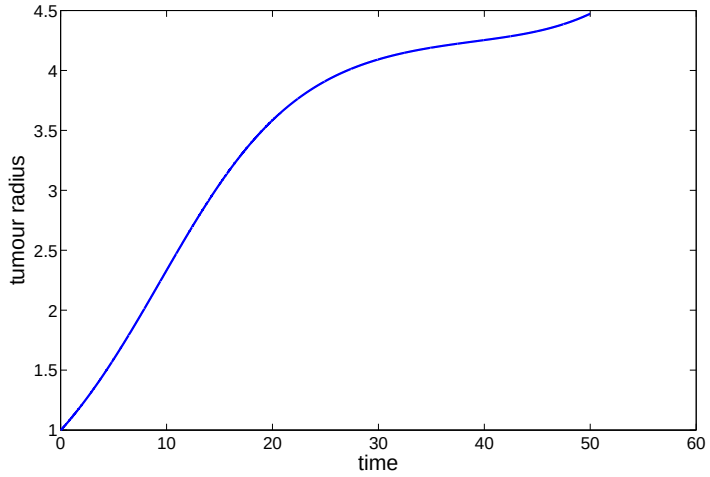

Figure 7.12: The tumour radius evolving over time for Method B using explicit Euler time-stepping.

the nodes proportional to the boundary movement, so they remain equally spaced over time, also only moving to the right with the tumour growth. When comparing Figures 7.13 and 7.15 it becomes apparent why they produce nearly the same results, especially when considering $t \leq 60$.

The mesh for Method B, where the nodes move with the cell velocity, begin by spreading out fairly equally. However, at later times when negative velocities are introduced, the nodes gather nearer the centre of the tumour. Indeed, it can be seen that most nodes will initially move out with the tumour growth, but then return to the left. The node at the boundary is then significantly apart from the others, causing an unsatisfactory coarseness at the edge. For example, at $t = 75$ the final three x values are 2.21, 2.27 and 6.18.

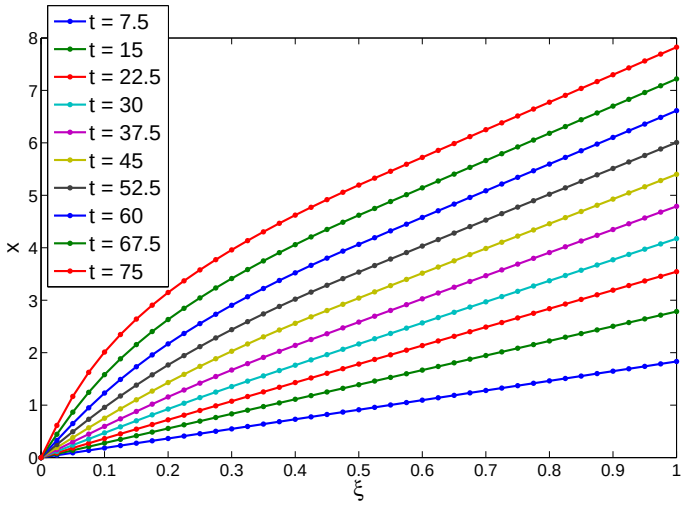

Figure 7.13: Method A using ODE23 - The position of the 40 nodes at different times

.

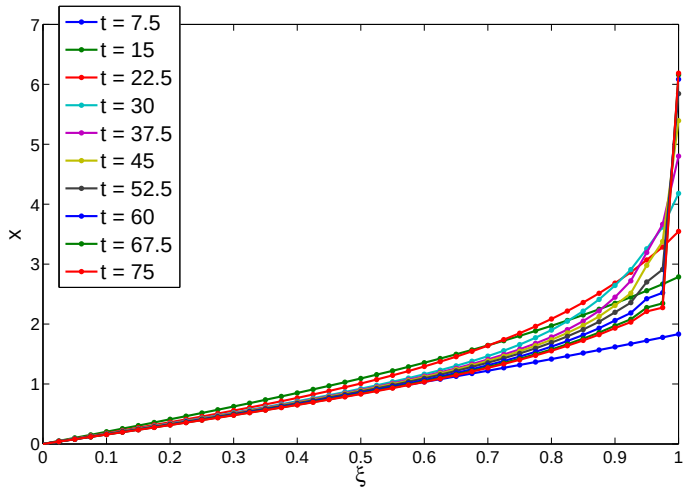

Figure 7.14: Method B using ODE23 - The position of the 40 nodes at different times

.

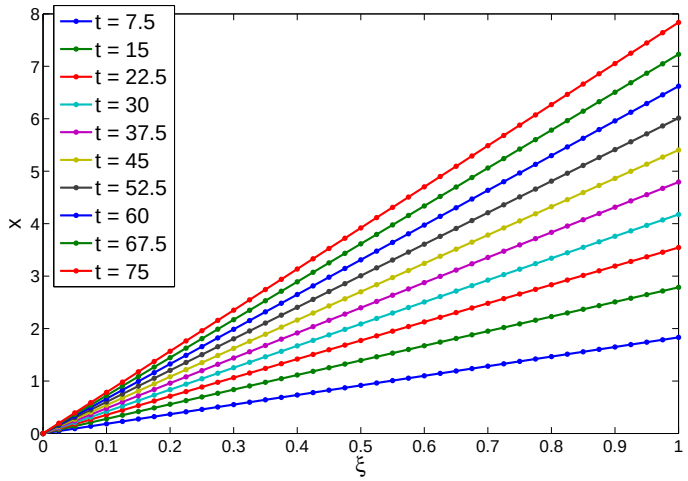

Figure 7.15: Method C using ODE23 - The position of the 40 nodes at different times

.

Chapter 8

Comparing Method C and the mesh method in [4]

In Chapter 6 we solved the non-conservative form of the tumour growth problem, as stated in [4],

$$\frac{\partial \alpha}{\partial \tau} - \frac{\xi}{\ell}\frac{d\ell}{d\tau}\frac{\partial \alpha}{\partial \xi} + \frac{1}{\ell}\frac{\partial}{\partial \xi}(\alpha v) = S(\alpha, C).$$

In [4] there is no mention of the numerical process. So we solved (6.1) by explicitly time-stepping α by Euler's method,

$$\alpha^{new} = \alpha^{old} + \Delta t \left(S^{new} - \frac{1}{\ell^{old}}(\alpha^{old} v^{new})_\xi + \frac{\dot{\ell}}{\ell^{old}} \xi \alpha_\xi^{old} \right). \tag{8.1}$$

Let us compare this to the moving mesh Method C in Section 6.3.3. Here we use the conservative form $\left(\frac{d}{dt}\int \alpha \, dx = \int S \, dx \right)$ of the PDE

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(\alpha v) = S(\alpha, C)$$

to find the integral of α with respect to space. We define $\Theta = \int \alpha \ dx$, thus discretely

$$\alpha = \frac{\Theta}{\Delta x}. \tag{8.2}$$

Instead of explicitly time-stepping α we explicitly time-stepped Θ, by Euler's method in the form

$$\Theta_j^{new} = \Theta_j^{old} + \Delta t \left(\int_{x_{j-1}}^{x_{j+1}} S^{new} \ dx - [\alpha^{old} v^{new}]_j + [\alpha^{old} \dot{x}]_j \right).$$

From (8.2) we can substitute $\Theta = \alpha\Delta x$ into the above. This gives

$$\alpha_j^{new}\Delta x_j^{new} = \alpha_j^{old}\Delta x_j^{old} + \Delta t\left(S^{new}\Delta x^{new} - [\alpha^{old}v^{new}] + [\alpha^{old}\dot{x}] + [\alpha^{old}\dot{x}]\right)_j.$$

Hence

$$\alpha_j^{new} = \alpha_j^{old}\frac{\Delta x_j^{old}}{\Delta x_j^{new}} + \Delta t\left(S^{new} - \frac{[\alpha^{old}v^{new}]}{\Delta x^{new}} + \frac{[\alpha^{old}\dot{x}]}{\Delta x^{new}}\right)_j.$$

With Method C, Δx was defined as a proportion of the moving boundary x_N. For the purpose of comparing Method C to the method in Chapter 6, we use $x_N = \ell$. Therefore

$$\Delta x = \ell\Delta\xi \qquad \text{and} \qquad \dot{x} = \dot{\ell}\Delta\xi$$

where ξ is the computational space $[0, 1]$. The equation above in terms of ξ and ℓ is then

$$\alpha_j^{new} = \alpha_j^{old}\frac{\ell^{old}}{\ell^{new}} + \Delta t\left(S^{new} - \frac{1}{\ell^{new}}(\alpha^{old}v^{new})_\xi + \frac{\dot{\ell}}{\ell^{new}}(\xi\alpha^{old})_\xi\right)_j. \tag{8.3}$$

When we compare (8.3) with (8.1) we see that the differences between the two are subtle. In Method C, α^{old} is multiplied by $\frac{\ell^{old}}{\ell^{new}}$ and the two end terms are divided by ℓ^{new} as opposed to ℓ^{old}. As this is nearly one for small time steps, this is not a significant difference between the two. More notable is the last term in Method C, which expands (using the product rule) to give

$$\Delta t\frac{\dot{\ell}}{\ell^{new}}(\xi\alpha_\xi + \alpha).$$

Comparing this to the final term in (8.1) shows that when solving the conservative form of the problem we have an additional $\Delta t\frac{\dot{\ell}\alpha}{\ell^{new}}$ term.

This comparison allows us to adapt our work in Chapter 6 so as to make the method conservative and explore the effect of the differences between (8.1) and (8.3). Graphically, the solutions to the non-conservative form (8.3) and the conservative form (8.1) turn out to be the same. Nonetheless, further investigations may present more significant differences.

Chapter 9

Further work

9.1 Altering the cell velocity boundary condition

Throughout this report we have not changed the model presented in [7] and [4]. Let us consider the boundary condition on the cell velocity

$$v = 0 \qquad \text{at } x = 0.$$

Possible future work could involve changing the left boundary condition to

$$\frac{dv}{dx} = 0 \qquad \text{at } x = 0,$$

depending on an internal pressure. Thus $v \neq 0$ at the inner boundary. This would mean that the tumour would still remain symmetrical about $x = x_0$, but the cells in the centre would have a velocity that depends on the viscosity μ, drag k and the nutrient concentration C. When the necrotic core forms, i.e. when $\alpha \to 0$, the region occupied by cells moves away from the origin. The problem would be solved on the region occupied by $\alpha \neq 0$.

9.2 Examining the effect of $\chi(\alpha)$

Let us return to $\chi(\alpha)$ in the form shown in Section 4.2

$$\chi(\alpha) = \begin{cases} 0 & 0 \leq \alpha < \alpha_{min}, \\ \frac{\hat{\Sigma}_c |\alpha - \alpha^*|^{r-1}}{(1-\alpha)^q}(\alpha - \alpha^*) & \alpha_{min} \leq \alpha < 1. \end{cases}$$

For $\alpha_{min} < \alpha^*$ there is discontinuity at α_{min}. This jump may cause inaccuracies when numerically approximating the derivative of $\chi(\alpha)$, used in Chapter 5.2

since, in this section we solved the equation for the cell velocity (3.10) by discretising in space. The right hand side of the matrix system (5.10) was a vector of the numerical approximation of $\frac{d}{dx}\big(\alpha\chi(\alpha)\big)$,

$$\begin{aligned}\frac{d}{dx}\big(\chi(\alpha)\alpha\big) &= \alpha_{j+\frac{1}{2}}\chi(\alpha_{j+\frac{1}{2}}) - \alpha_{j-\frac{1}{2}}\chi(\alpha_{j-\frac{1}{2}}) \\ &= \frac{1}{2}(\alpha_j + \alpha_{j+1})\chi\left(\frac{1}{2}(\alpha_j + \alpha_{j+1})\right) - \frac{1}{2}(\alpha_j + \alpha_{j-1})\chi\left(\frac{1}{2}(\alpha_j + \alpha_{j-1})\right).\end{aligned}$$

By approximating across the whole region in this manner, we are not accounting for the jump in $\chi(\alpha)$ at $\alpha = \alpha_{min}$. This may cause inaccuracies at this point, which might account for the severe oscillations in Figures 7.7 to 7.10. These figures use $\alpha_{min} = 0.6$ and $\alpha^* = 0.8$, and show that the solution is well behaved until near the point where α drops down to 0.6.

To assess this error in our discretisation we identify when $\alpha = \alpha_{min}$ and use a one-sided approximation for $\frac{d}{dx}\big(\alpha\chi(\alpha)\big)$ either side of this point, so as to not discretise across the jump in $\chi(\alpha)$.

As in the velocity calculation, it is necessary to use one-sided approximations at the same point ($\alpha = \alpha_{min}$) when finding the solution α. This is because α is recovered on the moving mesh by using a central difference mid-point approximation, and we are avoiding discretising across the jump in $\chi(\alpha)$. Details of the one-sided approximations are included in Appendix A.

Bibliography

[1] Araujo, R. P. and McElwain, D. L. S. (2004). A history of the study of solid tumour growth: The contribution of mathematical modelling. *Bulletin of Mathematical Biology*, **66**, 1039-1091.

[2] Araujo, R. P. and McElwain, D. L. S (2005). The nature of stresses induced during tissue growth. *App. Math. Lett*, **18**, 1081-1088 .

[3] Blake, K. (2001). Moving Mesh Methods for Non-Linear Parabolic Partial Differential Equations. Ph.D Thesis, *Department of Mathematics, University of Reading.*

[4] Breward, C. J. W., Byrne, H. M and Lewis, C. E. (2002). The role of cell-cell interactions in a two-phase model for avascular tumour growth. *J. of Math. Biol*, **45(2)**, 125-152.

[5] Byrne, H. M (1999). The role of mathematics in solid tumour growth. *Math. Today*, **35**, 59-89.

[6] Byrne, H. M (1999a). Using mathematics to study solid tumour growth. *Proceedings of the 9th General Meetings of European Women in Mathematics*, 81-107.

[7] Byrne, H. M., King, J. R., McElwain, D. L. S and Preziosi, L. (2003). A two-phase model of solid tumour growth. *Applied Mathematics Letters*, **16**, 567-573.

[8] Gatenby, R. A. (1998). Mathematical models of tumour-host interactions. *Cancer J.*, **11**, 289-293.

[9] Gatenby, R. A. and Gawlinski, E. T (1996). A reaction-diffusion model of cancer invasion. *Cancer Res.*, **56**, 5745-5753.

[10] Gatenby, R. A and Maini, P. K. (2003). Cancer summed up. *Nature*, **421**, 321.

[11] H. P. Greenspan (1976). On the growth and stability of cell cultures and solid tumours. *J. Theor. Biol.*, **56**, 229-242.

[12] Landman, K. A. and Please, C. P. (2001). Tumour dynamics and necrosis: Surface tension and stability. *IMA J. Math. Appl. Medicine Biol.*, **18**, 131-158.

[13] Lubkin, S. R. and Jackson, T. (2002). Mulitphase mechanics of capsule formation in tumours. *J. Biomech. Eng.*, **124**, 237-243.

[14] Pettet, G. J., Please, C. P., Tindall, M. J. and McElwain, D. L. S (1998). The migration of cells in multicell tumour spheroids. *Bull. Math.*

Biol, **63**, 231-257.

[15] Please, C. P, Pettet, G. J. and McElwain, D. L. S (1999). Avascular tumour dynamics and necrosis. *Math. Models Methods Appl. Sci*, **9**, 569-579.

[16] Roose, T., Chapman, S. J. and Maini, P. K. (2007). Mathematical models of avascular tumour growth. *SIAM review*, **49**, 179-208.

[17] Sherratt, J. A. and Nowak, M. A. (1992). Oncogenes, anti-oncogenes and the immune response to cancer. *Proc. Roy. Soc. London. B*, **248**, 261-271.

[18] Udagawa, N., Fernandez, A., Achilles, E. G., Folkman, R. J. and D'Amato, R. J. (2002). Persistence of microscopic human cancers in mice: Alterations in the angiogenic balance accompanies loss of tumour dormacy. *The FASEB Jour.*, **16**, 1361-1370.

[19] Ward, J. P. and King, J. R (1997). Mathematical modelling of avascular tumour growth. *IMA J. Math. Appl. Medicine Biol.*, **14**, 39-69.

[20] http://www.cancerresearchuk.com

Appendices

Appendix A

Examining the effect of $\chi(\alpha)$

Let us identify the node just to the left of the point where $\alpha = \alpha_{min}$ as the 'marker node', which we shall denote as x_m. To account for the jump between x_m and x_{m+1} we use a one-sided discretisation of (3.10) at x_m and x_{m+1}.

Downwind discretisation of the velocity (3.10) at x_m

$$\frac{1}{x_m - x_{m-1}} \left\{ \left[\alpha\mu \frac{\partial v}{\partial x} - \alpha\chi(\alpha) \right]_m - \left[\alpha\mu \frac{\partial v}{\partial x} - \alpha\chi(\alpha) \right]_{m-1} \right\} = \frac{k\alpha_{m-\frac{1}{2}}}{1 - \alpha_{m-\frac{1}{2}}} v_{m-\frac{1}{2}}$$

Again, we use a one-sided discretisation on the terms in the square brackets, so as to not approximate a differential across x_m. Also, as before, we use $\alpha_{m-\frac{1}{2}} \approx \frac{1}{2}(\alpha_m + \alpha_{m-1})$.

$$\begin{aligned} &\frac{1}{x_m - x_{m-1}} \left\{ \mu\alpha_m \left(\frac{v_m - v_{m-1}}{x_m - x_{m-1}} \right) - \chi(\alpha_m)\alpha_m - \mu\alpha_{m-1} \left(\frac{v_{m-1} - v_{m-2}}{x_{m-1} - x_{m-2}} \right) + \chi(\alpha_{m-1})\alpha_{m-1} \right\} \\ &\quad = \frac{k(\alpha_m + \alpha_{m-1})}{4 - 2(\alpha_m + \alpha_{m-1})} (v_m + v_{m-1}) \end{aligned}$$

$$\begin{aligned} &\mu\alpha_m \left(\frac{v_m - v_{m-1}}{x_m - x_{m-1}} \right) - \mu\alpha_{m-1} \left(\frac{v_{m-1} - v_{m-2}}{x_{m-1} - x_{m-2}} \right) - \frac{k(\alpha_m + \alpha_{m-1})(x_m - x_{m-1})}{4 - 2(\alpha_m + \alpha_{m-1})} (v_m + v_{m-1}) \\ &\quad = \chi(\alpha_m)\alpha_m - \chi(\alpha_{m-1})\alpha_{m-1}. \end{aligned}$$

Hence

$$A_m^l v_{m-2} + A_m^d v_{m-1} + A_m^u v_m = b(\alpha_m)$$

where $A_m^l = \frac{\mu\alpha_{m-1}}{x_{m-1} - x_{m-2}}$, $\quad A_m^d = -\frac{\mu\alpha_m}{x_m - x_{m-1}} - \frac{\mu\alpha_{m-1}}{x_{m-1} - x_{m-2}} - \frac{k(\alpha_m + \alpha_{m-1})(x_m - x_{m-1})}{4 - 2(\alpha_m + \alpha_{m-1})}$,

$A^u_m = \frac{\mu\alpha_m}{x_m+x_{m-1}} - \frac{k(\alpha_m+\alpha_{m-1})(x_m-x_{m-1})}{4-2(\alpha_m+\alpha_{m-1})}$ and $b(\alpha_m) = \chi(\alpha_m)\alpha_m - \chi(\alpha_{m-1})\alpha_{m-1}$.

Upwind discretisation of the velocity (3.10) at x_{m+1}

$$\frac{1}{x_{m+2}-x_{m+1}}\left\{\left[\alpha\mu\frac{\partial v}{\partial x}-\alpha\chi(\alpha)\right]_{m+2}-\left[\alpha\mu\frac{\partial v}{\partial x}-\alpha\chi(\alpha)\right]_{m+1}\right\} = \frac{k\alpha_{m+\frac{3}{2}}}{1-\alpha_{m+\frac{3}{2}}}v_{m+\frac{3}{2}}$$

We use a one-sided discretisation on the terms in the square brackets, so as to not approximate a differential across x_{m+1}. Also, we use $\alpha_{m+\frac{3}{2}} \approx \frac{1}{2}(\alpha_{m+2} + \alpha_{m+1})$.

$$\begin{aligned}\frac{1}{x_{m+2}-x_{m+1}}\Bigg\{&\mu\alpha_{m+2}\left(\frac{v_{m+3}-v_{m+2}}{x_{m+3}-x_{m+2}}\right)-\chi(\alpha_{m+2})\alpha_{m+2}\\ &-\mu\alpha_{m+1}\left(\frac{v_{m+2}-v_{m+1}}{x_{m+2}-x_{m+1}}\right)+\chi(\alpha_{m+1})\alpha_{m+1}\Bigg\} = \frac{k(\alpha_{m+2}+\alpha_{m+1})}{4-2(\alpha_{m+2}+\alpha_{m+1})}(v_{m+3}+v_{m+2})\end{aligned}$$

$$\begin{aligned}&\mu\alpha_{m+2}\left(\frac{v_{m+3}-v_{m+2}}{x_{m+3}-x_{m+2}}\right)-\mu\alpha_{m+1}\left(\frac{v_{m+2}-v_{m+1}}{x_{m+2}-x_{m+1}}\right)-\frac{k(\alpha_{m+2}+\alpha_{m+1})}{4-2(\alpha_{m+2}+\alpha_{m+1})}(v_{m+3}+v_{m+2})\\ &\quad=\chi(\alpha_{m+2})\alpha_{m+2}-\chi(\alpha_{m+1})\alpha_{m+1}.\end{aligned}$$

Hence

$$A^l_{m+1}v_{m+1} + A^d_{m+1}v_{m+2} + A^u_{m+1}v_{m+3} = b(\alpha_{m+1}),$$

where $A^l_{m+1} = \frac{\mu\alpha_{m+1}}{x_{m+2}-x_{m+1}}$, $A^d_{m+1} = -\frac{\mu\alpha_{m+2}}{x_{m+3}-x_{m+2}} - \frac{\mu\alpha_{m+1}}{x_{m+2}-x_{m+1}} - \frac{k(\alpha_{m+2}+\alpha_{m+1})(x_{m+2}-x_{m+1})}{4-2(\alpha_{m+2}+\alpha_{m+1})}$, $A^u_{m+1} = \frac{\mu\alpha_{m+2}}{x_{m+3}-x_{m+2}} - \frac{k(\alpha_{m+2}+\alpha_{m+1})(x_{m+2}-x_{m+1}}{4-2(\alpha_{m+2}+\alpha_{m+1})}$ and $b(\alpha_{m+1}) = \chi(\alpha_{m+2})\alpha_{m+2} - \chi(\alpha_{m+1})\alpha_{m+1}$.

These one-sided approximations will affect the matrix system (5.10), so that

$$A = \begin{pmatrix} A^d_1 & A^u_1 & 0 & \cdots & \cdots & 0 \\ A^l_2 & A^d_2 & A^u_1 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & A^l_m & A^d_m & A^u_m & \ddots & \vdots \\ \vdots & \ddots & A^l_{m+1} & A^d_{m+1} & A^u_{m+1} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & A^l_{N-1} & A^d_{N-1} & A^u_{N-1} \\ 0 & \cdots & \cdots & 0 & A^l_N & A^d_N \end{pmatrix},$$

and

$$\begin{aligned}
\mathbf{b}(\alpha) &= \Big(b(\alpha_1)\ b(\alpha_2)\ \cdots\ b(\alpha_m)\ b(\alpha_{m+1})\ \cdots\ b(\alpha_{N-1})\ b(\alpha_N)\Big)^T \\
\mathbf{v} &= \Big(v_1\ v_2\ \cdots\ v_m\ v_{m+1}\ \cdots\ v_{N-1}\ v_N\Big)^T.
\end{aligned}$$

Recovering α_m and α_{m+1} using one-sided approximations

For consistency, we recover α using a one-sided approximation at x_m and x_{m+1}. Method A:

$$\begin{aligned}
\alpha_m &= \frac{\theta(t)}{\theta(0)}\frac{x_m(0)-x_{m-1}(0)}{x_m(t)-x_{m-1}(t)}\alpha_m \\
\alpha_{m+1} &= \frac{\theta(t)}{\theta(0)}\frac{x_{m+2}(0)-x_{m+1}(0)}{x_{m+2}(t)-x_{m+1}(t)}\alpha_{m+1}.
\end{aligned}$$

Methods B and C:

$$\begin{aligned}
\alpha_m &= \frac{\Theta_m}{x_m - x_{m-1}} \\
\alpha_{m+1} &= \frac{\Theta_{m+1}}{x_{m+2} - x_{m+1}}.
\end{aligned}$$

We should note that the downwind approximation at x_m requires $m \geq 2$, but the position where $\alpha = \alpha_{min}$ occurs at the right hand boundary, so intially m is likely to be smaller than 2.