School of Mathematical
and Physical Sciences

University of Reading

Department of Mathematics and Statistics

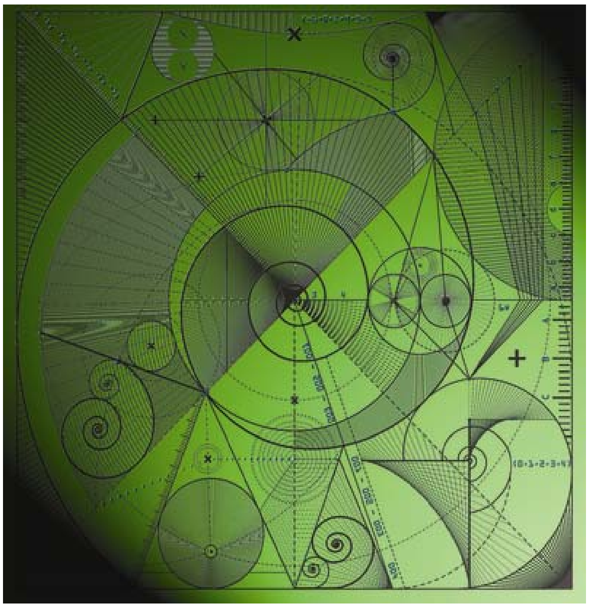
Preprint MPS-2011-03

28 March 2011

A moving mesh approach for modelling avascular tumour growth

by

T.E. Lee, M.J. Baines, S. Langdon and M.J. Tindall



A moving mesh approach for modelling avascular tumour growth

T. E. Lee[a], M. J. Baines[a], S. Langdon[a], M. J. Tindall[a,b,c]

[a] *Department of Mathematics and Statistics, University of Reading, UK*
[b] *Institute for Cadiovascular and Metabolic Research, University of Reading, UK*
[c] *School of Biological Sciences, University of Reading, UK*

Abstract

A key step in many numerical schemes for time-dependent partial differential equations with moving boundaries is to rescale the problem to a fixed numerical mesh. An alternative approach is to use a moving mesh which can be adapted to focus on specific features of the model. In this paper we present and discuss three different velocity-based moving mesh methods applied to a two-phase model of avascular tumour growth formulated by Breward et al. (2002) *J. Math. Biol.* 45(2), 125-152. An obvious advantage of moving mesh methods, either velocity-based or transformation-based, is their ability to track moving boundaries. Each method has one moving node which tracks the moving boundary. To move the internal nodes: our first method moves the nodes proportional to the boundary movement such that nodes remain equidistant at each time step; our second method assumes the nodes move proportional to the local spatial cell velocity; whilst our third method uses local conservation of mass. We demonstrate that when the nodes are moved according to the velocity of the outer boundary, or when local mass fractions are invoked, similar results are obtained to when the governing equations are scaled to a fixed numerical mesh. However, when the mesh nodes move according to the local velocity profile there are difficulties in obtaining accurate stable solutions. Our results demonstrate that a moving mesh approach can produce accurate results without fundamentally altering the form of the governing equations, whilst offering greater flexibility and higher resolution where desired.

Keywords: , finite difference methods, velocity-based moving meshes, avascular tumour growth
2010 MSC: , 65M06, 92-08, 92C99

1. Introduction

Systems of time-dependent partial differential equations (PDEs) occur in many branches of applied mathematics. In certain cases such systems may be prescribed in a region in which the outer boundary moves with time as a result of the internal dynamics of the problem. A common approach to solving such models numerically involves rescaling of the domain with respect to the outer boundary position to a fixed region. However, such a procedure alters the structure of the governing equations, in particular the conservation property of the spatial derivative. Alternatively, a moving mesh approach has the flexibility to preserve the structure whilst tracking the outer boundary. In this paper we utilise a recent model of tumour growth [1] to test a number of moving mesh methods, the solutions of which are compared with those obtained on a rescaled mesh.

Moving mesh methods can be location based or velocity based [2]. An overview is given in [3]. In this paper we use velocity based methods to draw our comparisons. Such methods are familiar from Lagrangian fluid dynamics, and their numerical application to PDEs can be found for example in the Moving Finite Element method [4], the Deformation method [5], the GCL method [6], and the Conservation method [7]. An obvious advantage of moving mesh methods, either velocity based or location based, is their ability to track moving boundaries.

Each moving mesh method we consider has a fixed number of nodes that move with the solution itself, with the precise nature of their movement differing for each scheme. The first method moves the nodes uniformly with respect to the movement of the boundary so that nodes remain equidistant at each time step. We find that this approach produces results that are akin to results from a scaled mesh approach. However, moving the nodes in this geometric manner does not take full advantage of the flexibility of a moving mesh. Our other moving mesh methods use specific features of the model to determine the mesh movement. The second method moves the nodes according to the local cell velocity; the third method moves the nodes using conservation of normalised cell volumes, which is a finite difference version of the finite element approach described in [7]. A similar finite difference approach was used in [8]. The second and third methods conserve local mass, which is not the case when the problem is rescaled to a fixed numerical mesh. Furthermore, we show that for the third method the mesh offers better resolution where the solution is growing, which in the avascular tumour growth model is near the boundary.

Mathematical models of tumour growth can offer effective and efficient ways to advance our understanding of cancer research; see, for example, the survey papers [9, 10]. In recent years there has been a large increase in the number of PDE models describing solid tumour growth.

Whilst differences between such models exist, many exhibit the following features:

- Equations describing the diffusion of nutrients or growth factors in and around the tumour region (generally parabolic in type);
- Mass transfer equations describing the dynamic variation in tumour tissue (generally hyperbolic);
- Mass balance equations describing the growth of the tumour (generally elliptic).

All of these equations are generally coupled via nonlinear interactions. For instance the growth dynamics of a specific cell type may depend in a nonlinear way on a specific nutrient or growth factor. Examples include Ward and King [11] who developed a two phase model of a growing multicellular tumour spheroid (MCTS) in which cells were considered to exist in either a live or dead state, whilst Please *et al.* [12, 13] considered the two phases to be live cells and water, respectively. In contrast Tindall and Please [14] considered a three phase model to account for proliferating and quiescent cells and dead cell material. The complexity of such nonlinear mathematical models means they are most often solved and investigated numerically. Given the coupling between the various equation types (parabolic, hyperbolic and elliptic) it is important that such methods are robust and accurate.

In this work we consider a recent two phase model of tumour growth developed by Breward *et al.* [1], which is a specific form of the two-phase model in [15]. The two phases, cell and water, each have an associated velocity, pressure and volume-fraction-averaged stress tensor. We utilise the model to compare a number of moving mesh strategies with the commonly employed fixed numerical mesh approach. Although three phase models may incorporate more detail our aim here is to demonstrate that moving mesh methods are an effective tool for the numerical solution of problems such as tumour growth models, and for this purpose a two phase model suffices. The extension to three phase models is technical but straightforward in principle. We focus on a two-phase model to clearly demonstrate the velocity-based moving mesh schemes, which can be adapted to numerically solve more sophisticated models.

In the next section we present the normalised one-dimensional model proposed in [1], followed by §3 where we surmise the fixed numerical mesh method used in [1], so as to compare results with the three moving mesh strategies. The details of these strategies are given in §4, where we solve the tumour growth model numerically using each one in turn. The results from the fixed mesh method and the three moving mesh methods are discussed in §5. Finally, in §6 we conclude that a moving mesh method can prove to be an elegant and accurate numerical approach that updates the mesh smoothly with the solution of the orginal model, whilst preserving chosen features of the model such as local mass balance, or relative partial masses, (for self-similar problems, similarity can be preserved). However, since the mesh depends upon the model, care must be taken when choosing a feature of the model to preserve.

2. A mathematical model of tumour growth

The model assumes the tumour consists of two phases, water and live cells, which are treated as incompressible fluids whose densities are equal, to leading order. The model is derived by applying mass balance to the cell and water phases. Further assumptions made are that inertial effects are negligible, no external forces act on the system, and, on the timescale of interest, the cell and water phases can be treated as viscous and inviscid fluids respectively. The model is applied to a tumour whose growth is parallel to the x-axis, and is symmetric about its midpoint.

From [1] the non-dimensional model, in Cartesian form, for the volume fraction of cells $\alpha(x,t) \in (0,1)$, with $t > 0$ and $x \in [0, \ell(t)]$, where $\ell(t)$ is the tumour radius, comprises

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x}(u_c \alpha) = \frac{(1+s_1)\alpha(1-\alpha)C}{1+s_1 C} - \frac{s_2 + s_3 C}{1+s_4 C}\alpha =: S(\alpha, C), \quad (1)$$

$$\frac{\partial}{\partial x}\left[\mu\alpha\frac{\partial u_c}{\partial x} - \alpha\frac{\alpha-\alpha^*}{(1-\alpha)^2}\mathrm{H}(\alpha-\alpha_{\min})\right] = \frac{k\alpha u_c}{1-\alpha}, \quad (2)$$

$$\frac{\partial^2 C}{\partial x^2} = \frac{Q\alpha C}{1+\hat{Q}_1 C}, \quad (3)$$

where $u_c(x,t)$ is the cell velocity, $C(x,t)$ is the nutrient concentration and H is the Heaviside function. The volume fraction of water is $1-\alpha$. The first term of $S(\alpha, C)$ in (1) represents cell growth due to mitosis (cell division), and the second term represents cell death. The parameters μ (a combination of the shear and bulk viscosities), k (the drag coefficient), and s_1, s_2, s_3, s_4, Q and $\hat{Q}_1$ are all positive constants. In addition, $\alpha_{\min}$ and α^* (a natural cell packing density) are constants such that $0 < \alpha_{\min} < \alpha^* < 1$. We remark that equation (1) arises from the global mass balance equation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_0^{\ell(t)} \alpha(x,t)\,\mathrm{d}x = \int_0^{\ell(t)} S(\alpha, C)\,\mathrm{d}x. \quad (4)$$

The normalised model has initial and boundary conditions

$$\ell = 1, \quad \alpha = \alpha^0(x) \qquad \text{at} \qquad t = 0, \quad (5)$$

$$u_c = \frac{\partial C}{\partial x} = 0 \qquad \text{at} \qquad x = 0, \; t > 0, \quad (6)$$

$$\mu\frac{\partial u_c}{\partial x} - \frac{\alpha-\alpha^*}{(1-\alpha)^2}\mathrm{H}(\alpha-\alpha_{\min}) = 0, \quad C = 1, \quad \frac{\partial \ell}{\partial t} = u_c \qquad \text{at} \qquad x = \ell, \; t > 0. \quad (7)$$

In the next three sections we show that moving the mesh to preserve features of the model can produce results in line with [1]. We also present results which demonstrate that the local feature of the model used to track the nodes needs to be carefully chosen.

3. Rescaling to a fixed numerical mesh

In [1] the moving domain $x \in [0, \ell(t)]$ is mapped to a fixed numerical domain $\xi \in [0, 1]$ by the transformation $\xi = x/\ell(t)$, $\tau = t$. Using the chain rule to differentiate $\alpha(\xi, \tau)$ with respect to time τ, the transformed problem is

$$\begin{aligned} \frac{\partial \alpha}{\partial \tau} - \frac{\xi}{\ell}\frac{\mathrm{d}\ell}{\mathrm{d}\tau}\frac{\partial \alpha}{\partial \xi} + \frac{1}{\ell}\frac{\partial}{\partial \xi}(u_c \alpha) &= S(\alpha, C), & (8)\\ \frac{\partial}{\partial \xi}\left(\mu\alpha\frac{\partial u_c}{\partial \xi} - \ell\alpha\frac{\alpha - \alpha^*}{(1-\alpha)^2}\mathrm{H}(\alpha - \alpha_{\min})\right) &= \frac{k\ell^2\alpha u_c}{1-\alpha}, & (9)\\ \frac{\partial^2 C}{\partial \xi^2} &= \frac{Q\ell^2\alpha C}{1 + \hat{Q}_1 C}, & (10) \end{aligned}$$

with initial and boundary conditions

$$\begin{aligned} \ell = 1, \;\; \alpha = \alpha^0(x) \quad &\text{at} \quad \tau = 0, & (11)\\ u_c = \frac{\partial C}{\partial \xi} = 0 \quad &\text{at} \quad \xi = 0,\; \tau > 0, & (12)\\ \mu\frac{\partial u_c}{\partial \xi} - \ell\frac{\alpha - \alpha^*}{(1-\alpha)^2}\mathrm{H}(\alpha - \alpha_{\min}) = 0, \;\; C = 1, \;\; \frac{\mathrm{d}\ell}{\mathrm{d}\tau} = u_c \quad &\text{at} \quad \xi = 1,\; \tau > 0. & (13) \end{aligned}$$

We note that in this approach the spatial derivatives in equation (8), unlike in the original equation (1), are not in divergence form, i.e. there is an additional term on the left-hand side that is not a total derivative with respect to ξ. This changes the structure of the equation which can lead to inaccurate numerical approximation.

Although details of the numerical method are not given in [1], in order to compare our results to those in [1] we surmise their numerical method to produce similar results. Many authors utilise the National Algorithms Group (NAG) routine D02RAF, which uses a finite difference approach [11]. Using the above equations we postulate an algorithm in which we choose a time step $\Delta\tau > 0$ and divide the region $(0, 1)$ into N equal cells of size $\Delta\xi = 1/N$. We define $\xi_j = j\Delta\xi$, $j = 0, 1, \ldots, N$, and $\tau^m = m\Delta\tau$, $m = 0, 1, \ldots$, and approximations $\alpha_j^m \approx \alpha(\xi_j, \tau^m)$, $\ell^m \approx \ell(\tau^m)$, $u_j^m \approx u_c(\xi_j, \tau^m)$, $C_j^n \approx C(\xi_j, \tau^m)$, and $S_j^m \approx S(\alpha(\xi_j, \tau^m), C(\xi_j, \tau^m))$. Given α_j^m, we compute C_j^m, u_j^m and ultimately α_j^{m+1} by a series of steps (labelled Steps F1–F4 below):

Step F1: Find C_j^m by applying central finite differences to (10),

$$\frac{C_{j-1}^m - 2C_j^m + C_{j+1}^m}{(\Delta\xi)^2} = \frac{Q(\ell^m)^2\alpha_j^m C_j^m}{1 + \hat{Q}_1 C_j^m}, \tag{14}$$

for $j = 0, 1, \ldots, N-1$, where from (12) and (13), we take $C_{-1}^m = C_1^m$ and $C_\ell^m = 1$. Newton's method is used to solve the subsequent system of nonlinear equations when $\hat{Q}_1 \neq 0$.

Step F2: Find u_j^m by applying central finite differences to (9),

$$\frac{1}{\Delta\xi}\left\{\alpha^m_{j+\frac{1}{2}}\left[\mu\frac{u^m_{j+1}-u^m_j}{\Delta\xi}-\ell^m\frac{\alpha^m_{j+\frac{1}{2}}-\alpha^*}{(1-\alpha^m_{j+\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{j+\frac{1}{2}}-\alpha_{\min})\right]\right.$$
$$\left.-\alpha^m_{j-\frac{1}{2}}\left[\mu\frac{u^m_j-u^m_{j-1}}{\Delta\xi}-\ell^m\frac{\alpha^m_{j-\frac{1}{2}}-\alpha^*}{(1-\alpha^m_{j-\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{j-\frac{1}{2}}-\alpha_{\min})\right]\right\}=\frac{k(\ell^m)^2\alpha^m_j}{1-\alpha^m_j}u^m_j, \quad (15)$$

for $j = 1, 2, \dots, N-1$, where $\alpha^m_{j+\frac{1}{2}} = \frac{1}{2}(\alpha^m_j + \alpha^m_{j+1})$ and $\alpha^m_{j-\frac{1}{2}} = \frac{1}{2}(\alpha^m_{j-1} + \alpha^m_j)$, leading to a linear system of equations. At the inner boundary $u^m_0 = 0$, as given by (12). To determine u^m_N, we discretise the boundary condition (13) by taking values $[\cdot]^m_{N-\frac{1}{2}}$ and $[\cdot]^m_{N+\frac{1}{2}}$ (the average about $[\cdot]^m_N$) to obtain

$$\frac{1}{2}\left[\mu\frac{u^m_{N+1}-u^m_N}{\Delta\xi}-\ell^m\frac{\alpha^m_{N+\frac{1}{2}}-\alpha^*}{(1-\alpha^m_{N+\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N+\frac{1}{2}}-\alpha_{\min})\right]$$
$$-\ \frac{1}{2}\left[\mu\frac{u^m_N-u^m_{N-1}}{\Delta\xi}-\ell^m\frac{\alpha^m_{N-\frac{1}{2}}-\alpha^*}{(1-\alpha^m_{N-\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N-\frac{1}{2}}-\alpha_{\min})\right]=0. \quad (16)$$

We then adapt (15) for $j = N$, using (16) to replace the first term in square brackets, leading to

$$-\frac{\alpha^m_{N+\frac{1}{2}}+\alpha^m_{N-\frac{1}{2}}}{\Delta\xi}\left[\mu\frac{u^m_N-u^m_{N-1}}{\Delta\xi}-\ell^m\frac{\alpha^m_{N-\frac{1}{2}}-\alpha^*}{(1-\alpha^m_{N-\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N-\frac{1}{2}}-\alpha_{\min})\right]=\frac{k(\ell^m)^2\alpha^m_N}{1-\alpha^m_N}u^m_N,$$

where $\alpha^m_{N+\frac{1}{2}} + \alpha^m_{N-\frac{1}{2}} = 2\alpha^m_N$, which yields a complete set of linear equations for the velocity u^m_j, $j = 1, \dots, N$.

Step F3: Discretise (8) using an explicit Euler time-stepping scheme and a central difference approximation in space, giving

$$\frac{\alpha^{m+1}_j-\alpha^m_j}{\Delta t} = \frac{ju^m_N(\alpha^m_{j+1}-\alpha^m_{j-1})}{2\ell^m}-\frac{u^m_{j+1}\alpha^m_{j+1}-u^m_{j-1}\alpha^m_{j-1}}{2\ell^m\Delta\xi}+S^m_j,$$

for $j = 1, 2, \dots, N-1$. One-sided approximations are used at the boundaries.

Step F4: Since the tumour radius moves with the cell velocity at the boundary, we calculate the tumour radius at the new time level using

$$\ell^{m+1} = \ell^m + \Delta t u^m_N.$$

We then return to Step F1 to complete the next time step. This numerical scheme produces results in line with those in [1] (see §5). Although this is a perfectly reasonable scheme, in the next section we solve the same problem numerically using a velocity-based moving mesh approach in which the velocities are defined by three different strategies.

4. Moving mesh methods

The key component of a velocity-based moving mesh method is the criterion used to define the mesh velocity. We investigate three different choices here, in which we move the mesh in the following ways:

Method A - proportional to the boundary position $\ell(t)$. This construction is geometrical in nature and is very similar to the method described above;

Method B - proportional to the local cell velocity u_c, i.e. based on a feature which is observed over the whole tumour;

Method C - in such a way as to conserve local mass fractions of the solution α in time. Like Method B, this is based on a prevalent feature of the model.

For all of these moving mesh methods (and in contrast to some fixed mesh methods), the final mesh node tracks the tumour radius.

In the model given by equations (1)–(7), x is an independent variable. We introduce the dependent variable $\tilde{x}_j(t)$, $j = 0, \ldots, N$, to represent the $N + 1$ nodes of the mesh, which are dependent on t. The mesh is initially equally-spaced; however, unlike the fixed mesh, re-scaling the grid points leads to them becoming, in general, irregularly separated. We define the velocity of the j-th node to be

$$v(\tilde{x}_j, t) = \frac{\mathrm{d}\tilde{x}_j}{\mathrm{d}t}. \qquad (17)$$

We choose a time step $\Delta t > 0$ and define $t^m = m\Delta t$, $m = 0, 1, \ldots$. We denote $\tilde{x}_j(t^m)$ by x^m_j, and use the approximations $\alpha^m_j \approx \alpha(\tilde{x}_j, t^m)$, $u^m_j \approx u_c(\tilde{x}_j, t^m)$, $C^m_j \approx C(\tilde{x}_j, t^m)$, and $v^m_j \approx v(\tilde{x}_j, t^m)$. For a given x^m_j and α^m_j, $j = 0, \ldots, N$, we compute C^m_j, u^m_j, v^m_j, x^{m+1}_j and α^{m+1}_j by the following algorithm:

Step 1: Find C^m_j by approximating (3) (with boundary conditions given by (6) and (7)) using central finite differences on the non-uniform mesh $\{x^m_0, \cdots, x^m_N\}$. The resulting set of equations is similar to (14), of the form

$$T^l C^m_{j-1} + T^d C^m_j + T^u C^m_{j+1} \quad = \quad \frac{Q\alpha^m_j C^m_j}{1 + \hat{Q}_1 C^m_j}, \qquad j = 0, 1, \ldots, N-1,$$

where

$$T^l = \frac{2}{(x^m_j - x^m_{j-1})(x^m_{j+1} - x^m_{j-1})}, \qquad T^d = \frac{-2}{(x^m_{j+1} - x^m_j)(x^m_j - x^m_{j-1})}, \qquad T^u = \frac{2}{(x^m_{j+1} - x^m_j)(x^m_{j+1} - x^m_{j-1})},$$

and where $x^m_{-1} = -x^m_1$, $C^m_{-1} = C^m_1$ and $C^m_N = 1$, from the boundary conditions (6) and (7).

Step 2: Find u^m_j by applying central finite differences to (2) on the non-uniform mesh $\{x^m_0, \ldots, x^m_N\}$ with boundary conditions given by (6) and (7). The resulting set of equations is similar to (15) and takes the form

$$\frac{1}{x^m_{j+\frac{1}{2}} - x^m_{j-\frac{1}{2}}} \left\{ \alpha^m_{j+\frac{1}{2}} \left[\mu \frac{u^m_{j+1} - u^m_j}{x^m_{j+1} - x^m_j} - \frac{\alpha^m_{j+\frac{1}{2}} - \alpha^*}{(1 - \alpha^m_{j+\frac{1}{2}})^2} \mathrm{H}(\alpha^m_{j+\frac{1}{2}} - \alpha_{\min}) \right] \right.$$
$$\left. - \alpha^m_{j-\frac{1}{2}} \left[\mu \frac{u^m_j - u^m_{j-1}}{x^m_j - x^m_{j-1}} - \frac{\alpha^m_{j-\frac{1}{2}} - \alpha^*}{(1 - \alpha^m_{j-\frac{1}{2}})^2} \mathrm{H}(\alpha^m_{j-\frac{1}{2}} - \alpha_{\min}) \right] \right\} = \frac{k\alpha^m_j}{1 - \alpha^m_j} u^m_j, (18)$$

where $x^m_{j+\frac{1}{2}} - x^m_{j-\frac{1}{2}} = \frac{1}{2}(x^m_{j+1} - x^m_{j-1})$, $j = 1, 2, \dots, N-1$, and $u^m_0 = 0$ (from (6)). As with the fixed numerical mesh method, to determine the boundary value u_N we discretise the boundary condition (7) in a similar way to (16) by taking the average at $[\cdot]^m_{N-\frac{1}{2}}$ and $[\cdot]^m_{N+\frac{1}{2}}$, giving

$$\begin{aligned} &\frac{1}{2}\left[\mu\frac{u^m_{N+1} - u^m_N}{x^m_{N+1} - x^m_N} - \frac{\alpha^m_{N+\frac{1}{2}} - \alpha^*}{(1-\alpha^m_{N+\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N+\frac{1}{2}} - \alpha_{\min})\right] \\ - \quad &\frac{1}{2}\left[\mu\frac{u^m_N - u^m_{N-1}}{x^m_N - x^m_{N-1}} - \frac{\alpha^m_{N-\frac{1}{2}} - \alpha^*}{(1-\alpha^m_{N-\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N-\frac{1}{2}} - \alpha_{\min})\right] = 0. \end{aligned} \tag{19}$$

We then adapt (18) for $j = N$ using (19) to replace the first term in square brackets, leading to

$$-\frac{\alpha^m_{N+\frac{1}{2}} + \alpha^m_{N-\frac{1}{2}}}{x^m_{N+\frac{1}{2}} - x^m_{N-\frac{1}{2}}}\left[\mu\frac{u^m_N - u^m_{N-1}}{x^m_N - x^m_{N-1}} - \frac{\alpha^m_{N-\frac{1}{2}} - \alpha^*}{(1-\alpha^m_{N-\frac{1}{2}})^2}\mathrm{H}(\alpha^m_{N-\frac{1}{2}} - \alpha_{\min})\right] = \frac{k\alpha^m_N}{1-\alpha^m_N}u^m_N,$$

where $\alpha^m_{N+\frac{1}{2}} + \alpha^m_{N-\frac{1}{2}} = 2\alpha^m_N$, which yields a complete set of linear equations for the velocity u^m_j, $j = 1, \dots, N$.

Step 3: Calculate the mesh velocity v^m_j. This step will differ for each of Methods A, B and C, and is detailed below.

Step 4: Update the mesh points by the explicit Euler scheme applied to (17)

$$x^{m+1}_j = x^m_j + \Delta t v^m_j, \qquad j = 0, 1, \dots, N, \tag{20}$$

with v^m_j obtained from Step 3.

Step 5: Calculate α^{m+1}_j. The details of this step will again differ for each method used, and are given in §4.1, 4.2 and 4.3 respectively.

When comparing this scheme to the fixed numerical mesh algorithm in §3, we see that the first two steps are essentially the same, except with a non-uniform mesh. However, whereas the third step of the algorithm in §3 calculates the solution α immediately from (8) on the transformed mesh, the moving mesh methods calculate the nodal positions first and then recover the solution α. Another distinction between the fixed numerical mesh method of §3 and the moving mesh methods is that the latter methods preserve a local mass balance through being written in divergence form (since the chain rule was not applied to the original model).

We now give details of each moving mesh method.

4.1. Method A

For Method A we move the nodes in Step 3 with a velocity proportional to the velocity of the boundary, i.e.

$$v^m_j = \frac{x^m_j}{x^m_N}u^m_N, \qquad j = 0, 1, \dots, N.$$

This velocity-based strategy is similar to the numerical mapping in §3, see Remark 1 below. It is geometrical in nature and draws only on information from the boundary of the tumour to determine how to move the nodes. Once the mesh velocity is defined, the new mesh is determined as in Step 4 above.

Now consider Step 5. To recover α on the new mesh we take an integral-based approach. First define the partial masses $\Theta_j(t)$ by

$$\Theta_0(t) \quad = \quad \int_{\tilde{x}_0(t)}^{\tilde{x}_1(t)} \alpha(x,t)\,\mathrm{d}x, \tag{21}$$

$$\Theta_j(t) \quad = \quad \int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} \alpha(x,t)\,\mathrm{d}x, \quad j = 1,\ldots,N-1 \tag{22}$$

$$\Theta_N(t) \quad = \quad \int_{\tilde{x}_{N-1}(t)}^{\tilde{x}_N(t)} \alpha(x,t)\,\mathrm{d}x,. \tag{23}$$

The values $\Theta_j(0)$ are known from the initial data. To calculate $\Theta_j(t)$, we begin by constructing $\dot{\Theta}_j(t)$. For ease of explanation we give the explicit formulae for $j = 1,\ldots,N-1$ only, but we note that similar formulae hold for $j = 0, N$. We differentiate (22) using Leibnitz' integral rule to give

$$\dot{\Theta}_j(t) = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} \alpha(x,t)\,\mathrm{d}x \quad = \quad \int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} \frac{\partial\alpha}{\partial t}\,\mathrm{d}x + \Big[\alpha(\cdot,t)v(\cdot,t)\Big]_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)}.$$

Substituting $\partial\alpha/\partial t$ from (1) gives

$$\dot{\Theta}_j(t) \quad = \quad \int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} S(\alpha, C)\,\mathrm{d}x + \Big[\alpha(\cdot,t)(v(\cdot,t) - u_c(\cdot,t))\Big]_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)}. \tag{24}$$

We use a mid-point approximation of the integral to obtain a discrete form of (24) at time $t = t^m$,

$$\dot{\Theta}_j^m \quad = \quad (x_{j+1}^m - x_{j-1}^m)S_j^m + \alpha_{j+1}^m(v_{j+1}^m - u_{j+1}^m) - \alpha_{j-1}^m(v_{j-1}^m - u_{j-1}^m), \qquad j = 1,\ldots,N-1,\tag{25}$$

where $\dot{\Theta}_j^m \approx \dot{\Theta}_j(t^m)$. This equation allows us to determine $\Theta_j^{m+1} \approx \Theta_j(t^{m+1})$ in the same manner that x_j^{m+1} is calculated in Step 4, by the explicit Euler scheme $\Theta_j^{m+1} = \Theta_j^m + \Delta t\dot{\Theta}_j^m$.

Once an approximation to the updated partial masses Θ_j^{m+1} has been determined, the final step for Method A is to recover the solution α_j^{m+1} using a mid-point approximation of (22) at time level $m + 1$, i.e.

$$\alpha_j^{m+1} = \frac{\Theta_j^{m+1}}{x_{j+1}^{m+1} - x_{j-1}^{m+1}}, \qquad j = 1,\ldots,N-1.$$

As noted above, similar formulae hold for $j = 0, N$.

Remark 1. *The velocity of Method A corresponds to the transformation-based method of §3 in the sense that the transformation is effected exactly by the boundary velocity. However, when α is calculated in §3 using a velocity derived from the transformation, a quasi-Lagrangian form of the mass balance equation is used in which the velocity is incorporated using a chain rule. The result is an extra term which cannot be written in divergence form. By contrast, in Method A we have preferred to use an integral approach which already incorporates local conservation.*

4.2. *Method B*

Under this strategy, in Step 3 the velocity of each node is determined by the cell velocity at that node, i.e.

$$v_j^m \quad = \quad u_j^m, \qquad\qquad j = 0, 1, \ldots, N. \tag{26}$$

This way of moving the nodes relates to the tumour model more than Method A as it uses local cell information rather than just information from the tumour boundary. Once the mesh velocity has been determined, the new mesh is computed as in Step 4.

In Step 5, as with Method A, we define the partial mass fractions $\Theta_j(t)$ as in (22), and follow Method A to completion, noting that $v = u_c$ at the nodes. In particular, (25) reduces to

$$\dot{\Theta}_j^m \quad = \quad (x_{j+1}^m - x_{j-1}^m) S_j^m \qquad\qquad j = 1, \ldots, N-1. \tag{27}$$

Note that this method corresponds with the mass balance equation (4) over arbitrary subintervals,

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} \alpha(x,t)\ \mathrm{d}x = \int_{\tilde{x}_{j-1}(t)}^{\tilde{x}_{j+1}(t)} S(\alpha, C)\ \mathrm{d}x.$$

4.3. *Method C*

Method C moves the nodes so as to conserve local mass fractions. Like Method B, this method also uses a feature of the model to move the nodes in such a way that information about the distribution of cells within the tumour is carried in time.

Let the total mass be

$$\theta(t) = \int_0^{\ell(t)} \alpha(x,t)\ \mathrm{d}x. \tag{28}$$

We define γ_j to be the mass fraction, so that

$$\gamma_j = \frac{1}{\theta(t)} \int_0^{\tilde{x}_j(t)} \alpha(x,t)\ \mathrm{d}x, \tag{29}$$

and calculate $\tilde{x}_j(t)$ such that γ_j remains constant with respect to time. The total mass θ will be required in order to approximate α, so we first determine $\dot{\theta}$ by differentiating (28) using Leibnitz' integral rule,

$$\dot{\theta}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \int_0^{\ell(t)} \alpha(x,t)\ \mathrm{d}x \quad = \quad \int_0^{\ell(t)} \frac{\partial \alpha}{\partial t}\ \mathrm{d}x + \Big[\alpha(\cdot,t) v(\cdot,t)\Big]_0^{\ell(t)}.$$

Substituting $\partial\alpha/\partial t$ from (1), and using the boundary conditions (6)–(7), gives

$$\dot{\theta}(t) \quad = \quad \int_0^{\ell(t)} S(\alpha, C)\ \mathrm{d}x. \tag{30}$$

It is worth noting that equation (30) corresponds exactly to the global mass balance result (4).

Equation (30) can be approximated directly once Step 1 has been carried out. We define the approximation $\dot{\theta}^m \approx \dot{\theta}(t^m)$ and apply a trapezoidal rule approximation to (30),

$$\dot{\theta}^m = \sum_{j=0}^{N-1} \frac{1}{2} (x_{j+1}^m - x_j^m)(S_{j+1}^m + S_j^m). \tag{31}$$

The updated total mass $\theta^{m+1} \approx \theta(t^{m+1})$ is then found using (31) and the same time-stepping approach used in Step 4, i.e. $\theta^{m+1} = \theta^m + \Delta t\dot{\theta}^m$.

To derive an expression for the mesh velocity, we again use Leibnitz' integral rule on (29) to calculate

$$\gamma_j\dot{\theta}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\int_0^{\tilde{x}_j(t)} \alpha(x,t)\ \mathrm{d}x = \int_0^{\tilde{x}_j(t)} \frac{\partial\alpha}{\partial t}\ \mathrm{d}x + \Big[\alpha(\cdot,t)v(\cdot,t)\Big]_0^{\tilde{x}_j(t)}.$$

Substituting $\partial\alpha/\partial t$ from (1), and using the boundary condition $u_0^m = v_0^m = 0$ (from (6)), gives

$$\gamma_j\dot{\theta}(t) \quad = \quad \int_0^{\tilde{x}_j(t)} S(\alpha, C)\ \mathrm{d}x - u_c(\tilde{x}_j,t)\alpha(\tilde{x}_j,t) + \alpha(\tilde{x}_j,t)v(\tilde{x}_j,t).$$

Thus, for $\alpha(\tilde{x}_j,t) \neq 0$, the mesh velocity is given by

$$v(\tilde{x}_j,t) \quad = \quad \frac{\gamma_j\dot{\theta}(t)}{\alpha(\tilde{x}_j,t)} - \frac{1}{\alpha(\tilde{x}_j,t)}\int_0^{\tilde{x}_j(t)} S(\alpha,C)\ \mathrm{d}x + u_c(\tilde{x}_j,t). \tag{32}$$

We use the composite trapezoidal rule on the integral to obtain a discrete form of (32) at time $t = t^m$,

$$v_j^m \quad = \quad \frac{\gamma_j\dot{\theta}^m}{\alpha_j^m} - \frac{1}{\alpha_j^m}\sum_{i=0}^{j-1}\frac{1}{2}(x_{i+1}^m - x_i^m)(S_{i+1}^m + S_i^m) + u_j^m. \tag{33}$$

Using (33), the new mesh x_j^{m+1} is computed as in Step 4. To approximate the updated solution α_j^{m+1} in Step 5, we consider (29) for $\tilde{x}_{j+1}$ and $\tilde{x}_{j-1}$, and subtract them from each other to give a partial mass. Equating this partial mass definition at times $t = t^{m+1}$ and $t = 0$ gives

$$\frac{1}{\theta^{m+1}}\int_{x_{j-1}^{m+1}}^{x_{j+1}^{m+1}} \alpha(x,t^{m+1})\ \mathrm{d}x \quad = \quad \frac{1}{\theta^0}\int_{x_{j-1}^0}^{x_{j+1}^0} \alpha(x,0)\ \mathrm{d}x.$$

Using a mid-point approximation for the integrals we obtain the numerical approximation

$$\alpha_j^{m+1} = \frac{\theta^{m+1}}{\theta^0}\ \frac{\left(x_{j+1}^0 - x_{j-1}^0\right)}{\left(x_{j+1}^{m+1} - x_{j-1}^{m+1}\right)}\ \alpha_j^0.$$

Note that at any given time the relative discrete mass

$$\frac{1}{\theta(t^{m+1})}(x_{j+1}^{m+1} - x_{j-1}^{m+1})\alpha_j^{m+1}$$

is preserved, and is equal to

$$\frac{1}{\theta(0)}(x_{j+1}^0 - x_{j-1}^0)\alpha_j^0.$$

5. Numerical Results

In this section we solve the tumour growth model numerically using the methods of §3 and §4, and compare the outcomes from each approach. In our experiments we used two sets of parameters from [1], which were chosen so as to focus to on the qualitative nature of the model equations. A purpose of [1] was to examine the effect of altering the tension constant (by altering k and μ). To compare our moving mesh methods to the commonly used fixed mesh method, we choose only the parameters from [1] that correspond to plots of u, v and ℓ over time. Both sets of parameters take

$$Q = 0.5, \quad \hat{Q}_1 = 0, \quad s_1 = s_4 = 10, \quad s_2 = s_3 = 0.5, \quad \alpha^0(x) = \alpha^* = 0.8, \tag{34}$$

with

$$k = 1, \qquad \mu = 1, \qquad \alpha_{\min} = 0.8, \tag{35}$$

in the first case, and

$$k = 0.25, \qquad \mu = 0.25, \qquad \alpha_{\min} = 0.6, \tag{36}$$

in the second case. The first case does not include the effects of cellular attraction, whilst the second case does. Furthermore, the second case has smaller k and μ than the first case, which corresponds to a larger tension constant. Figures 1–8 show results obtained with the method described in §3, with $N = 80$, $\Delta t = 7.5 \times 10^{-3}$ and final time $t = 75$, i.e. 10,000 time steps. Figures 1–4 use parameters (34)–(35) and display a travelling wave solution. Figures 5–8 use the second set of parameters, (34) and (36), and show the tumour radius settling to a steady state. Figures 1–3 and 5–7 closely resemble the results shown in [1] (results for the nutrient concentration were not included in [1]).

Next we examine the convergence of the moving mesh methods of §4 for the parameter set (34) and (35), as N increases and Δt decreases. We solve for $t \in [0, 4]$ and compute results for $N = 10 \times 2^{n-1}$, $n = 1, \ldots, 6$. In order to compare results for different values of n, we denote the points of the mesh for a particular value of n by $x_{j,n}(t)$, $j = 0, \ldots, (10 \times 2^{n-1})$. We then compute both $x_{2^{n-1}i,n} = x_{2^{n-1}i,n}(4)$ and $\alpha_{2^{n-1}i,n} \approx \alpha(x_{2^{n-1}i,n}, 4)$ for each $i = 0, \ldots, 10$ as n increases. To balance the spatial and temporal errors, and recalling that we have used explicit Euler time-stepping, we choose $\Delta t = O(1/N^2)$, precisely $\Delta t = 0.02/(4^n)$. We take the results computed with $n = 6$ (i.e. $N = 320$) as our reference mesh and solution. We anticipate that the pointwise 'errors' $|\alpha_{32i,6} - \alpha_{2^{n-1}i,n}|$ and $|x_{32i,6} - x_{2^{n-1}i,n}|$ will decrease as n increases, for each $i = 0, \ldots, 10$.

As a measure of the errors, we calculate

$$E_N(\alpha) = \sqrt{\frac{\sum_{i=0}^{10}(\alpha_{32i,6} - \alpha_{2^{n-1}i,n})^2}{\sum_{i=0}^{10}(\alpha_{32i,6})^2}}, \qquad E_N(\tilde{x}) = \sqrt{\frac{\sum_{i=0}^{10}(x_{32i,6} - x_{2^{n-1}i,n})^2}{\sum_{i=0}^{10}(x_{32i,6})^2}},$$

for $n = 1, \ldots, 4$ (i.e. $N = 10, 20, 40, 80$). We investigate the hypothesis that

$$E_N(\alpha) \sim \frac{1}{N^p} \qquad \text{and} \qquad E_N(\tilde{x}) \sim \frac{1}{N^q}$$

for large N, where p and q are the estimated orders of convergence for $E_N(\alpha)$ and $E_N(\tilde{x})$ approximated respectively by

$$p_{2N} = -\log_2\left(\frac{E_{2N}(\alpha)}{E_N(\alpha)}\right) \qquad q_{2N} = -\log_2\left(\frac{E_{2N}(\tilde{x})}{E_N(\tilde{x})}\right).$$

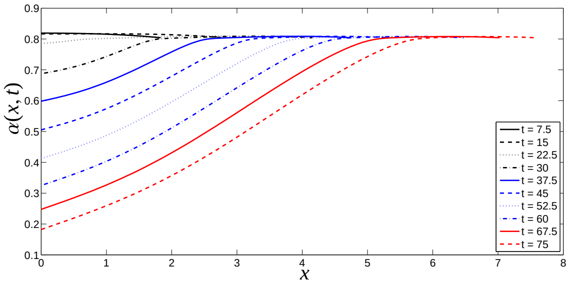


Figure 1: Cell volume fraction $\alpha(x,t)$ using the fixed numerical mesh method and parameter set (34) and (35).

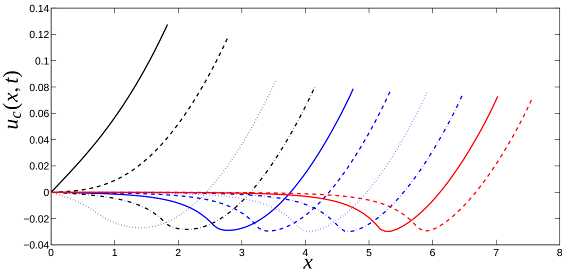


Figure 2: Cell velocity $u_c(x,t)$ using the fixed numerical mesh method and parameter set (34) and (35), legend as in Figure 1.

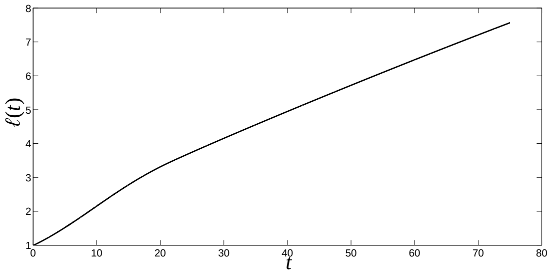


Figure 3: Tumour radius $\ell(t)$ using the fixed numerical mesh method and parameter set (34) and (35).

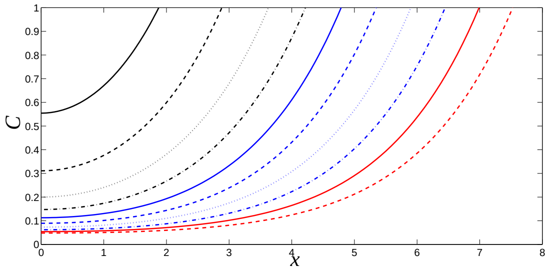


Figure 4: Nutrient concentration C using the fixed numerical mesh method and parameter set (34) and (36), legend as in Figure 1.

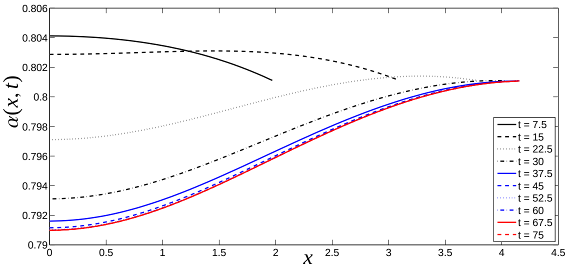


Figure 5: Cell volume fraction $\alpha(x,t)$ using the fixed numerical mesh method and parameter set (34) and (36).

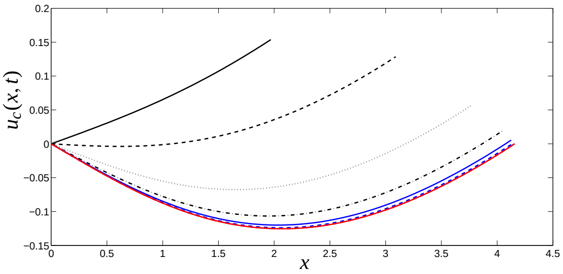


Figure 6: Cell velocity $u_c(x,t)$ using the fixed numerical mesh method and parameter set (34) and (36), legend as in Figure 5.

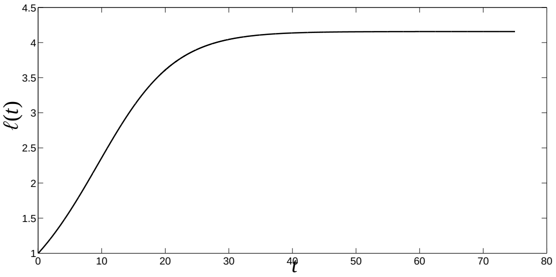


Figure 7: Tumour radius $\ell(t)$ using the fixed numerical mesh method and parameter set (34) and (36).

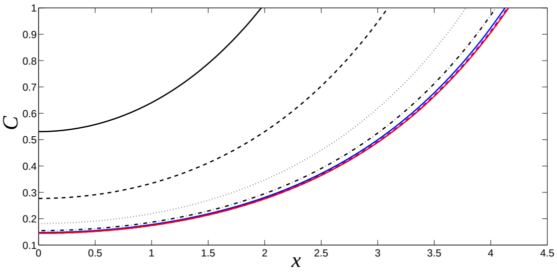


Figure 8: Nutrient concentration C using the fixed numerical mesh method and parameter set (34) and (36), legend as in Figure 5.

Since each step of our scheme is second order in space and first order in time, and recalling that $\Delta t = O\left(\frac{1}{N^2}\right)$, we might expect to see $p, q \approx 2$, although since our meshes are generally non-uniform and varying in time, this is only an approximate hypothesis. Convergence results are shown in Table 1. We see that $E_N(\alpha)$ and $E_N(\tilde{x})$ decrease as N increases for each of the moving mesh methods. This strongly suggests that as the number of nodes increases, both the solution α and the position of the nodes $\tilde{x}_j$ are converging. For Methods A and B, the p-values presented in this table indicate superlinear convergence of α, and the q-values suggest second-order convergence of $\tilde{x}$. For Method C, the p and q values suggest second-order convergence of both α and $\tilde{x}$.

Method	N	$E_N(\alpha)$	p_{2N}	$E_N(\tilde{x})$	q_{2N}
A	10	2.034×10^{-4}	-	1.275×10^{-5}	-
	20	8.346×10^{-5}	1.3	3.306×10^{-6}	1.9
	40	3.547×10^{-5}	1.2	8.478×10^{-7}	2.0
	80	1.471×10^{-5}	1.3	2.050×10^{-7}	2.0
B	10	2.299×10^{-4}	-	6.207×10^{-4}	-
	20	9.293×10^{-5}	1.3	1.109×10^{-4}	2.5
	40	3.891×10^{-5}	1.3	3.043×10^{-5}	1.9
	80	1.600×10^{-5}	1.3	7.224×10^{-6}	2.1
C	10	1.448×10^{-5}	-	1.819×10^{-5}	-
	20	3.645×10^{-6}	2.0	1.944×10^{-6}	3.2
	40	8.807×10^{-7}	2.0	7.148×10^{-7}	1.5
	80	2.090×10^{-7}	2.1	1.880×10^{-7}	1.9

Table 1: Relative errors for α and $\tilde{x}$ with rates of convergence using the explicit Euler time-stepping scheme.

Having established convergence of our moving mesh schemes we now compare the numerical results from the methods of §4 with those of the method described in §3.

We generate results using the parameters detailed in (34) and (35). All three methods were investigated with $N = 80$, $\Delta t = 7.5 \times 10^{-3}$, and final time $t = 75$, i.e. 10,000 time-steps. Each of Methods A and C produce very similar results, so only the results from Method C and Method B are plotted below. Figures 9–11 are due to Method C and display the same travelling wave characteristics as the results in [1] for the same parameters (closely resembling Figures 1–3). The value of α near the free boundary remains fairly constant, and α at the centre of the tumour decreases at a steady rate as time increases. The velocity peaks near the boundary, but the velocity at the boundary appears to stay constant with respect to time for $t \geq 37.5$. This coincides with the tumour radius growing steadily, Figure 11. The minima are subtly different to that of [1]; the troughs in Figure 2, which resemble those in [1], are slightly less rounded than those shown in Figure 10. Interestingly, Method A (a locally conservative version of the method in §3) also presented rounder minima, identical to those in Figure 10.

Figures 12–14 show that Method B appears to behave like Method A and C (and [1]) at early times. However, 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. Furthermore, the velocity at the boundary decreases considerably, with the tumour radius nearly reaching a steady state at $t = 75$. This behaviour is not apparent in [1], nor from Methods A and C. The plots from Method B are less smooth, despite the same number of nodes being used for each method. There is a considerable kink in α and u_c for $t = 45$ which appears to dampen at later times. The solution α does not

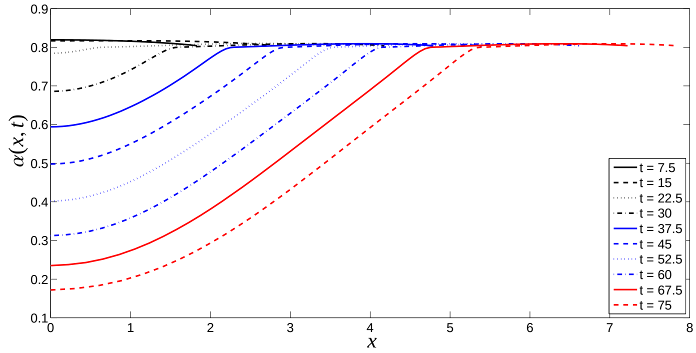


Figure 9: Cell volume fraction $\alpha(x,t)$ using Method C and parameter set (34)–(35).

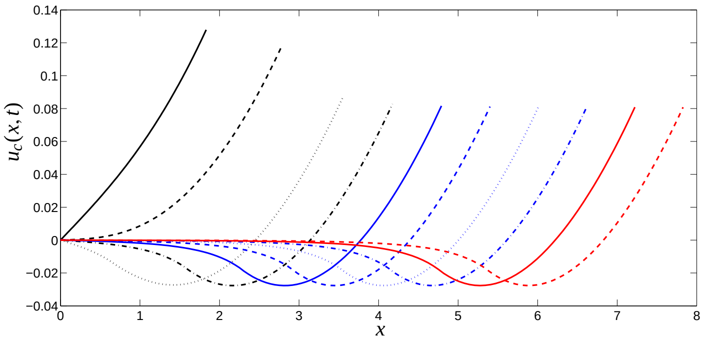


Figure 10: Cell velocity $u_c(x,t)$ using Method C and parameter set (34)–(35), legend as in Figure 9.

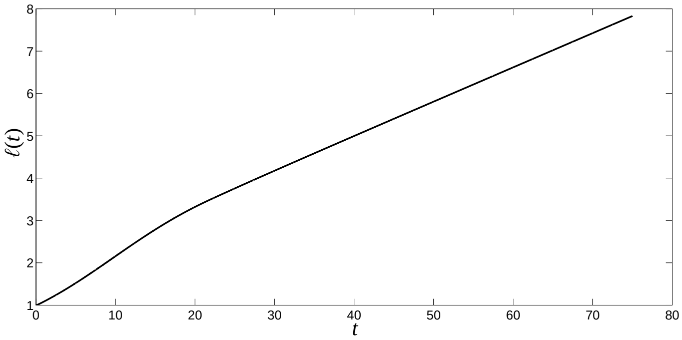


Figure 11: Tumour radius $\ell(t)$ using Method C and parameter set (34)–(35).

drop below 0.4 at the centre of the tumour, even for $t = 100$ (not shown here). This erratic behaviour remains with a smaller Δt, and when using an adaptive second and third order Runge-Kutta method for the time-stepping (see Remark 2 below), suggesting that this behaviour is due to the choice of the velocity in the numerical method. The processes of Method A and Method B are very similar, and because Method A behaves as in Figures 9–11, it is reasonable to conclude that tracking the cell velocity with the mesh nodes, as in Method B, results in the mesh becoming too coarse in some areas, and too fine in others. This is a problem that could be compounded over time, especially in the area where the cell velocities vary between positive and negative; resulting in nodes moving in opposite directions, leaving a considerable gap in between. Indeed if we look at Figure 13 for $t = 75$, we see that the velocity is mostly negative, so that most of the nodes are moving to the left.

As a further example, we use the parameter set (34) and (36), and again present results for the method of §2 and the moving mesh methods in §3. Once the steady state is reached at $t \approx 40$, all cells within the region have negative velocity, i.e. the cells are moving inwards. The comparisons between the methods had similar outcomes: the results for Methods A and C (Figures 15–17) resembled the results in [1] (as shown in Figures 5–7); Method B moves the nodes evenly for early times, but once negative spatial velocities occur, the nodes become clustered to the left, as shown in Figures 18–20. When the tumour radius settles to a steady state, the internal cells continue moving. This feature means that the mesh for Method B never settles to a steady state, whereas the meshes for Methods A and C do.

Finally, we examine exactly how the mesh moves for each of the different moving mesh methods. We take the parameters that produce a steady travelling-wave profile, (34) and (35). By definition, the nodes with Method A remain equally spaced over time, and move to the right uniformly with the tumour growth, as shown in Figure 21. The mesh for Method B, Figure 22, begins by spreading out fairly equally. However, at later times when negative velocities are introduced, the nodes cluster 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 tumour centre. The node at the boundary is then significantly separated from the others, causing an unsatisfactory coarseness at the edge. When the nodes are moved by Method C, Figure 23, the nodes behave similar to the nodes of Method A for $t < 30$. For larger times, the nodes near the tumour centre spread. We would expect the spread to be more prominent as the tumour grows, i.e. the nodes naturally spread where α is low, and cluster where α is larger. Moreover, each node only moves to the right as the tumour grows. When comparing Figures 21 and 23 it becomes apparent why they produce nearly the same results, especially for $t \leq 30$.

Remark 2. *For the moving mesh methods we also considered using a time-stepping scheme based on an adaptive second and third order predictor-corrector Runge-Kutta method, which chooses the time step automatically to minimise the error (specifically, we used ODE23 in Matlab). When using this scheme, we took a maximum $\Delta t = O(1/N)$ to balance the spatial and temporal errors, precisely* $\max \Delta t = 0.02/(2^n)$. *The results from the Runge-Kutta method were very similar to results from the explicit Euler time-stepping scheme, indicating that our approach is robust to different time-stepping approaches, and is not particularly stiff.*

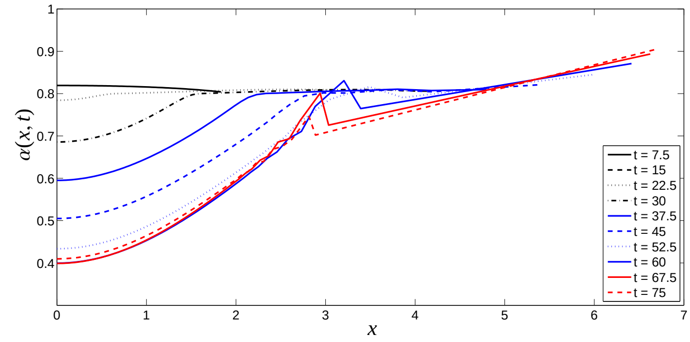


Figure 12: Cell volume fraction $\alpha(x,t)$ using Method B and parameter set (34)–(35).

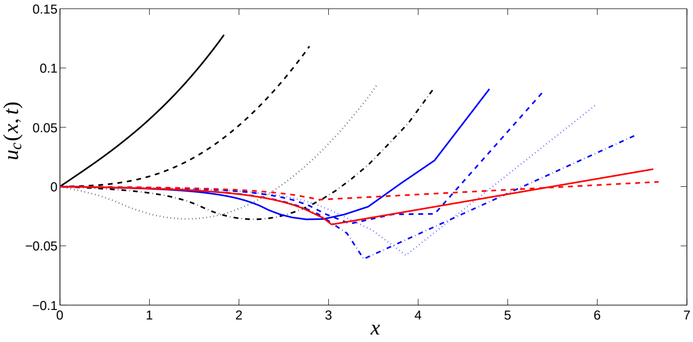


Figure 13: Cell velocity $u_c(x,t)$ using Method B and parameter set (34)–(35), legend as in Figure 12.

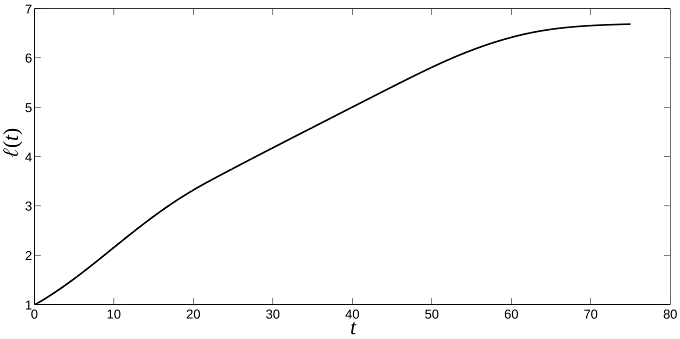


Figure 14: Tumour radius $\ell(t)$ using Method B and parameter set (34)–(35).

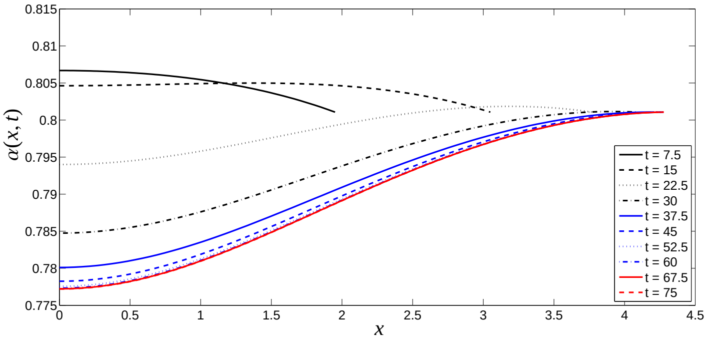


Figure 15: Cell volume fraction $\alpha(x,t)$ using Method C and parameter set (34) and (36).

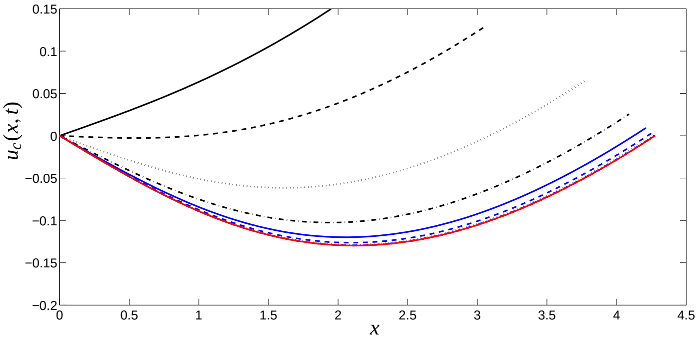


Figure 16: Cell velocity $u_c(x,t)$ using Method C and parameter set (34) and (36), legend as in Figure 15.

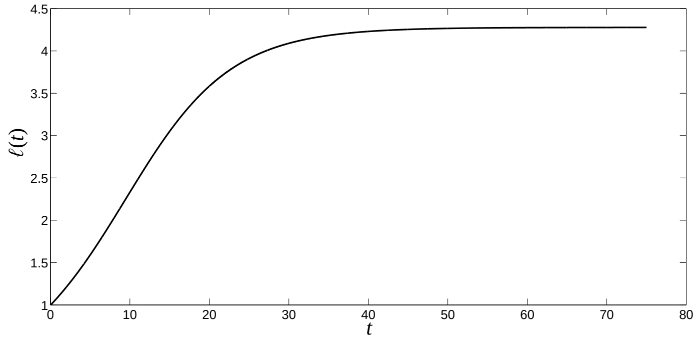


Figure 17: Tumour radius $\ell(t)$ using Method C and parameter set (34) and (36).

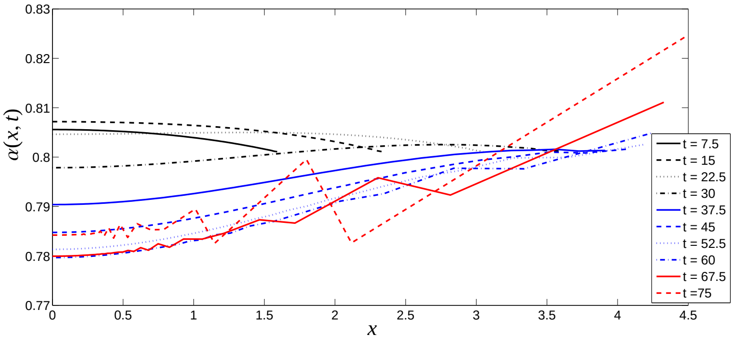


Figure 18: Cell volume fraction $\alpha(x,t)$ using Method B and parameter set (34) and (36).

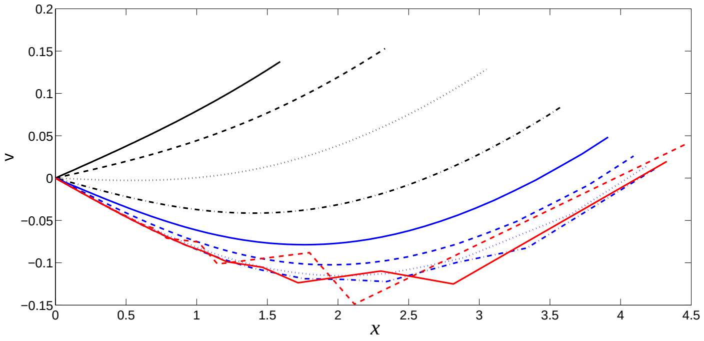


Figure 19: Cell velocity $u_c(x,t)$ using Method B and parameter set (34) and (36), legend as in Figure 18.

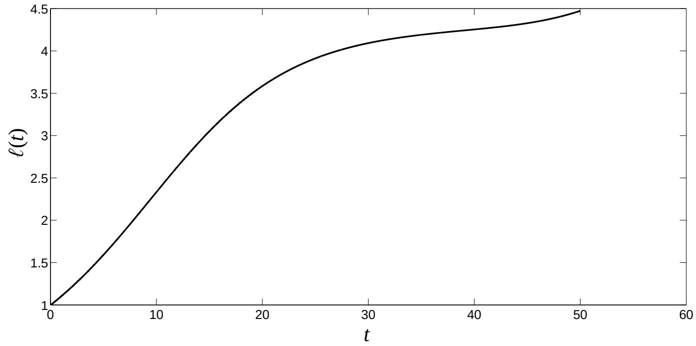


Figure 20: Tumour radius $\ell(t)$ using Method B and parameter set (34) and (36).

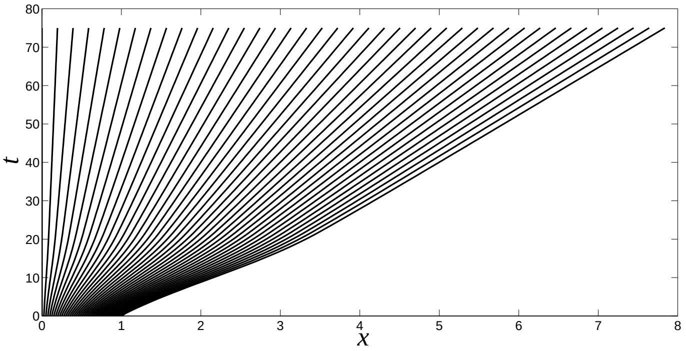


Figure 21: The position of nodes for Method A, $N = 40$, parameter set (34) and (35)

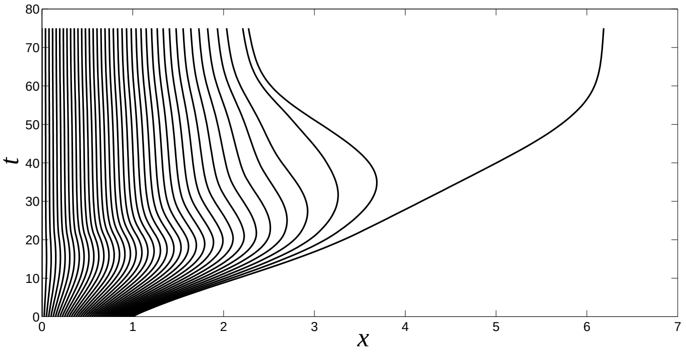


Figure 22: The position of nodes for Method B, $N = 40$, parameter set (34) and (35)

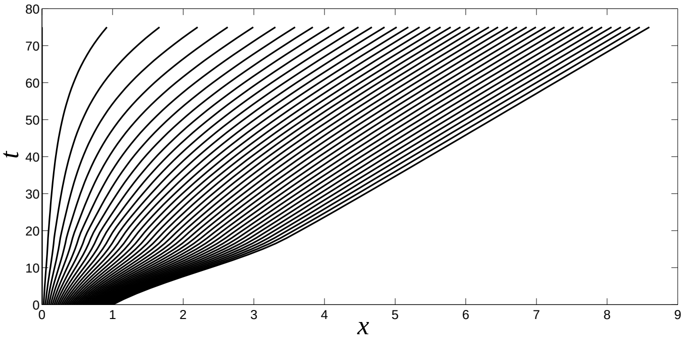


Figure 23: The position of nodes for Method C, $N = 40$, parameter set (34) and (35)

.

6. Conclusions

We have numerically solved the non-dimensionalised form of an avascular tumour growth model given in [1] using three different moving mesh methods. Working with the original non-dimensionalised form of the model, we have replicated the results of [1] and presented three different velocity-based approaches to move the mesh. The different approaches to define the mesh velocity are: (A) proportional to the boundary movement; (B) following the cell velocity; (C) conserving local mass fractions. To advance in time, each of the three methods used either explicit Euler time-stepping or adaptive second and third order Runge-Kutta formulas. Each method, with explicit Euler time-stepping, appears to be convergent for small times. Methods A and C continue to work well for larger times and replicate results in [1], but Method C has the added advantage that the nodes move in a manner that preserves a feature of the model, specifically local mass fractions. However, care is required when choosing a feature of the model to determine the mesh velocity, as evidenced by the poor resolution apparent when using Method B over longer times. Method C is an especially effective method when solving problems with self-similar solutions as it preserves similarity.

An obvious advantage of moving mesh methods, either velocity-based or transformation-based, is their ability to track moving boundaries. In the two-phase model studied here the outer boundary is accurately followed. Moreover, the results show that by moving the nodes in a manner that conserves local mass fractions (Method C), we obtain very similar results to when the mesh is moved geometrically, with the advantage that the mesh movement is controlled by a specific feature of the model which preserves local mass balance and offers higher resolution at the boundary. More recent three-phase models that take into account proliferating, quiescent and necrotic cells can be treated in a similar way, even though these models cannot be reduced to the study of a single component such as α. However, in a two-phase situation the necrotic core can alternately be modelled as a separate inner region between inner and outer moving boundaries.

Acknowledgement

MJT was supported by a Research Council UK Fellowship during the time in which this work was undertaken.

References

[1] 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. Math. Biol*, **45(2)**, 125–152.

[2] Cao, W., Huang, W. and Russell, R. D. (2003). Approaches for generating moving adaptive meshes: location versus velocity. *Appl. Numer. Math.*, **47**, 121-138.

[3] Budd, C., Huang, W., and Russell, R. D. (2009). Adaptivity with moving grids, *Acta Numer.*, 111–241.

[4] Baines, M. J. (1994). *Moving Finite Elements*, Oxford University Press.

[5] Liao, G. and Anderson, D. (1992). A new approach to grid generation. *Appl. Anal.*, **44** 285-298.

[6] Cao, W., Huang, W. and Russell, R. D. (2002). A moving-mesh method based on the geometric conservation law, *SIAM J. Sci. Comput.*, **24** 118-142.

[7] Baines, M. J., Hubbard, M. E. and Jimack, P. K. (2004). A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. *Appl. Numer. Math.*, **54**, 450–469.

[8] Blake, K. and Baines, M. J. (2002). A moving mesh method for non-linear parabolic problems. *Numerical Analysis Reports, University of Reading*, **2/2002**.

[9] Araujo, R. P. and McElwain, D. L. S. (2004). A history of the study of solid tumour growth: The contribution of mathematical modelling. *Bull. Math. Biol.*, **66**, 1039–1091.

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

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

[12] Please, C. P., Pettet, G. J. and McElwain, D. L. S. (1998). A new approach to modelling the formation of necrotic regions in tumours. *Appl. Math. Lett.*, **11**, 89–94.

[13] 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.

[14] Tindall, M. J. and Please, C. P. (2007). Modelling the Cell Cycle and Cell Movement in Multicellular Tumour Spheroids. *Bull. Math. Biol.*, **69**, 1147–1165.

[15] Byrne, H. M., King, J. R., McElwain, D. L. S and Preziosi, L. (2003). A two-phase model of solid tumour growth. *App. Math. Lett.*, **16**, 567–573.