Journal of Computational Physics 309 (2016) 207–226

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A computational method for the coupled solution of reaction–diffusion equations on evolving domains and manifolds: Application to a model of cell migration and chemotaxis

G. MacDonald [a], J.A. Mackenzie [a,*], M. Nolan [a], R.H. Insall [b]

[a] Department of Mathematics and Statistics, University of Strathclyde, Glasgow, G1 1XH, United Kingdom
[b] The Beatson Institute for Cancer Research, Garscube Estate, Switchback Road, Glasgow, G61 1BD, United Kingdom

ARTICLE INFO

Article history:
Received 28 March 2015
Received in revised form 16 December 2015
Accepted 17 December 2015
Available online 21 December 2015

Keywords:
Reaction–diffusion
Bulk–surface equations
Cell migration
Chemotaxis
Evolving finite elements
ALE methods
Moving mesh methods

ABSTRACT

In this paper, we devise a moving mesh finite element method for the approximate solution of coupled bulk–surface reaction–diffusion equations on an evolving two dimensional domain. Fundamental to the success of the method is the robust generation of bulk and surface meshes. For this purpose, we use a novel moving mesh partial differential equation (MMPDE) approach. The developed method is applied to model problems with known analytical solutions; these experiments indicate second-order spatial and temporal accuracy. Coupled bulk–surface problems occur frequently in many areas; in particular, in the modelling of eukaryotic cell migration and chemotaxis. We apply the method to a model of the two-way interaction of a migrating cell in a chemotactic field, where the bulk region corresponds to the extracellular region and the surface to the cell membrane.
© 2016 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Coupled bulk–surface problems arise in many areas of engineering and the applied and natural sciences. Examples include crystal growth [29], interfacial fluid flows using soluble and insoluble surfactants [7,27], proton diffusion along biological membranes [34] and cell signalling [47].

Our motivation for the development of a method for coupled problems on evolving domains comes from the study of eukaryotic cell migration and chemotaxis. Chemotaxis, in particular, is essential during embryonic development, immune cell function, and cancer metastasis. Eukaryotic cells typically crawl by protruding pseudopods, which are dynamic structures based on actin fibres, at the front of the cell [26]. Actin is a globular protein which spontaneously polymerises into linear filaments that make up a large fraction of the cell's cytoskeleton. The key step limiting actin polymerisation is the slow initiation of new filaments. Actin assembly can therefore be stimulated by "nucleating factors" which generate new actin filaments. In most cells, actin filaments are concentrated just beneath the cellular membrane and crosslinked into a relatively rigid cortex. Motor proteins such as myosin II bind to actin filaments in the cortex, crosslinking and contracting them, causing cortical tension and mechanical resistance, which together are key determinants of cells' overall behaviour. New actin polymerisation occurs between the cortex and the membrane, giving rise to a pressure pushing the cell membrane

* Corresponding author.
E-mail address: j.a.mackenzie@strath.ac.uk (J.A. Mackenzie).

http://dx.doi.org/10.1016/j.jcp.2015.12.038
0021-9991/© 2016 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

outward at the pseudopod; in other regions, cortical tension pulls along the remainder of the cell body. Together these processes lead to cell movement.

In [40] we developed a preliminary model of cell migration and chemotaxis where we only considered processes taking place on or close to the cell membrane. The model was based on a system of three reaction–diffusion equations posed on an evolving curve in two dimensions. An evolving surface finite element method was used to approximate the solution of the PDEs and a level set method was used to move the cell boundary. The method was later made more robust and more efficient by replacing the level set method by a parameterised finite element approach [39]. The model was shown to predict many aspects of real cell behaviour such as cell polarisation, persistent random walk migration in the absence of external signals and directed migration in the presence of a gradient in an external chemotactic field [41].

The modelling of a number of important biological processes however also requires the ability to solve model equations in the extracellular and intracellular domains which are then coupled to equations posed on the cell membrane. It is also well appreciated that migrating cells have the ability to shape external chemotactic fields through the use of membrane-bound enzymes that degrade the ligand field and by the self-secretion of chemoattractants that allow signal relay to neighbouring cells [50]. The modelling of these important effects require the computational ability of solve partial differential equations on evolving bulk–surface domains.

Some previous computational studies looking at biological signalling have been carried out assuming fixed cell boundaries [36,30,42]. Methods have also been proposed to treat the deforming cell shape; for a recent review see [18]. For example, a phase-field approach has been used to model the combined effect of intracellular actin-flow, cell adhesions and morphology on cell motility [48]. Normally phase-field methods use isotropic stationary meshes which makes their implementation relatively straightforward. However, as the phase-field is smoothed over a small width of size ε, the computational mesh needs to be fine enough to resolve this spatial scale which can make the method computationally expensive unless some form of local mesh adaptation is used. Another well known embedding method is the level set method (LSM) [44]. This approach uses an evolution equation to move a signed distance function; the zero level set of this function is then identified as the moving cell boundary. LSMs have been used successfully in cell motility models [49,54]. The disadvantage of LSMs is the requirement to maintain a high quality signed distance function which can become difficult if the evolving cell morphology becomes complicated. Bulk–surface reaction–diffusion systems on stationary domains have also been addressed using a simple point cloud method in [31]. An analysis of a finite element method for a steady state bulk–surface problem is presented in [13]. Analysis and computations of the diffusion-driven instability properties of coupled bulk–surface reaction–diffusion systems on stationary domains is presented in [32,33].

To approximate the solution of bulk–surface systems of equations we will use an Arbitrary Lagrangian–Eulerian (ALE) approach. Traditionally, ALE methods have been used to solve problems on moving domains using a reformulation of the original problem with respect to an alternative reference frame rather than the standard fixed Eulerian frame. For fluid dynamics problems one could decide to use a Lagrangian transformation to follow the fluid flow. More generally however, there may be no obvious or preferred reference frame, and if the domain moves in time one may simply be satisfied with a transformation from a fixed stationary domain Ω_c onto the physically evolving domain $\Omega(t)$. The ALE formulation was introduced for this purpose and it has been used successfully to tackle a number of physical applications such as fluid–structure interaction systems (see [24,11]). In this paper we will use a conservative finite element ALE scheme. An advantage of the finite element framework is it allows the natural incorporation of flux boundary conditions linking the solution components in the bulk and surface domains. Another potential advantage of the ALE approach is the ability to accommodate arbitrary mesh movements which are not necessarily Lagrangian. This may help the robustness, accuracy and efficiency of the method with fewer instances of remeshing being required.

Crucial to the success of the front-tracking approach used here is the robust generation of cell body-fitted meshes. For this purpose we use an adaptive moving mesh procedure based on the solution of a moving mesh partial differential equations (MMPDEs) [9,19,4,8,23]. Although adaptive moving meshes have been used successfully to solve a range of physical problems [5,51,6], to our knowledge they have not been applied to the generation of meshes for curves evolving according to a geometric evolution law. An important contribution of this paper therefore is the development of an MMPDE method for the tangential motion of grid nodes along a curve which is moving with a specified normal velocity. The proposed method is motivated by the approach of Pan and Wetton [45], who devised a finite difference scheme for the generation of meshes which equidistribute the arc-length between grid nodes. The new method allows a degree of control over the generation of the surface mesh which is then used as known boundary conditions for the generation of the bulk mesh. The additional control of the boundary mesh also adds considerable robustness to the grid generation algorithm leading to far fewer instances for the need of global remeshing due to poor mesh quality.

The layout of this paper is as follows. In the next section we introduce the necessary notation for the description of coupled bulk–surface conservation laws on evolving domains which are then reformulated using a conservative weak ALE formulation. In section 3 we present a moving finite element discretisation of the coupled equation system. We then describe the generation of computational meshes covering the evolving bulk and surface domains in section 4. The complete algorithm is presented in section 5 and applied in section 6 to the simulation of cell migration in a chemotactic field. Finally, we draw some conclusions and directions for further research in section 7.

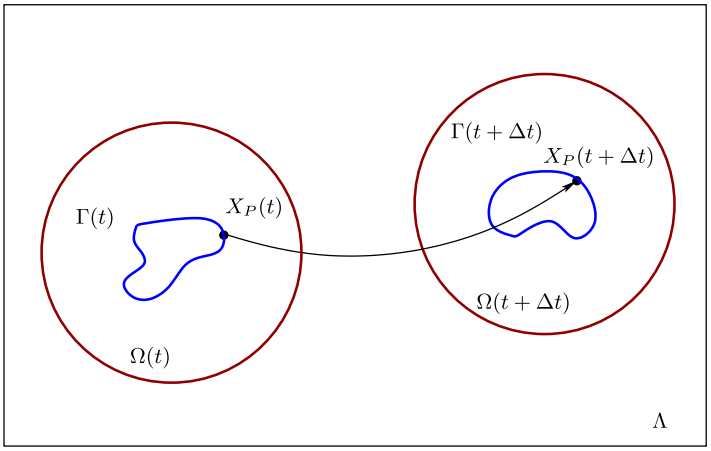


Fig. 1. We consider the simulation of a motile cell through a fixed lab frame of reference Λ. The cell membrane is denoted by $\Gamma(t)$ and the extracellular region close to the cell is denoted by $\Omega(t)$. After a time interval of size Δt, the material point located at $\boldsymbol{X}_p(t)$ on the cell membrane $\Gamma(t)$ evolves to the new location $\boldsymbol{X}_p(t+\Delta t)$.

2. Model system equations

The physical layout for the cell migration simulations considered later is shown in Fig. 1. The cell membrane will be denoted by the evolving curve $\Gamma(t)$. The cell will be assumed to be moving through fixed lab frame of reference Λ. To improve computational efficiency, the governing equations for the extracellular region will only be computed over the time-dependent two-dimensional domain $\Omega(t)$, which is centred on the centroid of the cell. We will assume that a material particle P located at $X_p(t)$ on $\Gamma(t)$ has velocity $\dot{\boldsymbol{X}}_p(t)$. Therefore, we assume that there exists a velocity field $\boldsymbol{u}$ so that points on $\Gamma(t)$ evolve with a velocity field $\dot{\boldsymbol{X}}_p(t) = \boldsymbol{u}(\boldsymbol{X}_p(t), t)$. Let $\boldsymbol{n} = (n_1, n_2)$ denote the unit outward normal to $\Gamma(t)$ and let $\mathcal{N}(t)$ be any open subset of $\mathbb{R}^2$ containing $\Gamma(t)$. For any function ζ which is differentiable in $\mathcal{N}(t)$, we define the tangential gradient on $\Gamma(t)$ by $\nabla_\Gamma \zeta = \nabla\zeta - (\nabla\zeta \cdot \boldsymbol{n})\boldsymbol{n}$, where $\cdot$ denotes the usual scalar product and $\nabla\zeta$ denotes the usual gradient on $\mathbb{R}^2$. For a vector function $\boldsymbol{\zeta} = (\zeta_1, \zeta_2) \in \mathbb{R}^2$, the tangential divergence is defined by

$$\nabla_\Gamma \cdot \boldsymbol{\zeta} = \nabla \cdot \boldsymbol{\zeta} - \sum_{i=1}^{2} (\nabla\zeta_i \cdot \boldsymbol{n}) n_i.$$

The Laplace–Beltrami operator on $\Gamma(t)$ is defined as the tangential divergence of the tangential gradient $\Delta_\Gamma \zeta = \nabla_\Gamma \cdot (\nabla_\Gamma \zeta)$.

Reaction–diffusion systems of the type studied in pattern formation generally exclude cross-diffusion and are only coupled by the reaction kinetics terms. Therefore, we consider the behaviour of a single chemical species with a straightforward generalisation to a system of interacting chemicals. Let us define

$$Q_T = \{(\boldsymbol{x}, t) \in \mathbb{R}^3 : \boldsymbol{x} \in \Omega(t), t \in (0, T]\}. \tag{1}$$

Application of Reynolds transport theorem to the equation for mass conservation for a chemical C, which diffuses with constant D, undergoing reaction at a rate $f(c)$, gives

$$\frac{\partial c}{\partial t} = D\Delta c + f(c), \quad (\boldsymbol{x}, t) \in Q_T, \tag{2}$$

where $c(\boldsymbol{x}, t)$ is the concentration at position $\boldsymbol{x} \in \Omega(t)$ at time t.

We also consider the evolution of a chemical species C_s, that resides on the boundary $\Gamma(t)$. The bulk species C will be coupled to C_s through the generally nonlinear flux boundary condition

$$\overbrace{-D\left.\frac{\partial c}{\partial \boldsymbol{n}}\right|_{\Gamma(t)}}^{\text{Diffusive flux}} + \overbrace{(\boldsymbol{u}\cdot\boldsymbol{n})c|_{\Gamma(t)}}^{\text{Advective flux}} = \overbrace{g(c|_{\Gamma(t)}, c_s)}^{\text{Rate of surface reaction}}, \tag{3}$$

where $c_s(\boldsymbol{x}, t)$ denotes the concentration of C_s at the point $\boldsymbol{x} \in \Gamma(t)$ and $\boldsymbol{n}$ is the outward unit normal to $\Gamma(t)$. At the front of a motile cell $(\boldsymbol{u}\cdot\boldsymbol{n}) > 0$, leading to an advective flux onto $\Gamma(t)$, and at the back of the cell $(\boldsymbol{u}\cdot\boldsymbol{n}) < 0$ leading to a flux off of $\Gamma(t)$. This so-called windshield effect can potentially have a bearing on the ability of highly motile cells to chemotax efficiently in shallow gradients unless some additional mechanisms are employed by the cell such as surface ligand degradation or receptor internalisation [16].

In an analogous manner to the equation for the bulk species, we will assume that the boundary species evolves such that

$$\frac{\partial c_s}{\partial t} + \nabla_\Gamma \cdot (\boldsymbol{u} c_s) = D_s \Delta_\Gamma c_s + g(c|_{\Gamma(t)}, c_s) + h(c_s), \quad (\boldsymbol{x}, t) \in \Gamma(t) \times (0, T), \tag{4}$$

where D_s is the boundary diffusion coefficient, $h(c_s)$ is a surface reaction term.

2.1. ALE reformulation

For the reasons mentioned earlier, when the domain is moving a common frame of reference adopted for computational purposes is the Arbitrary Lagrangian Eulerian (ALE) frame. Let $\mathcal{A}_t$ be a family of bijective mappings, which at each $t \in I = [0, T]$, map points in a reference or computational configuration Ω_c with coordinates $\boldsymbol{\xi} = (\xi, \eta)$, to points in the current physical configuration $\Omega(t)$ with coordinates $\boldsymbol{x} = (x, y)$, so that

$$\mathcal{A}_t : \Omega_c \subset \mathbb{R}^2 \to \Omega(t) \subset \mathbb{R}^2, \qquad \boldsymbol{x}(\boldsymbol{\xi}, t) = \mathcal{A}_t(\boldsymbol{\xi}). \tag{5}$$

The computational configuration could simply be the initial physical configuration $\Omega(0)$. We leave the discussion on how to construct the mapping $\mathcal{A}_t$ to section 4.

For an arbitrary function $g : Q_T \to \mathbb{R}$, defined on the fixed Eulerian frame, its temporal derivative in the ALE frame is defined as

$$\left.\frac{\partial g}{\partial t}\right|_{\boldsymbol{\xi}} : Q_T \to \mathbb{R}, \quad \left.\frac{\partial g}{\partial t}\right|_{\boldsymbol{\xi}} (\boldsymbol{x}, t) = \frac{\partial \hat{g}}{\partial t}(\boldsymbol{\xi}, t), \quad \boldsymbol{\xi} = \mathcal{A}_t^{-1}(\boldsymbol{x}), \tag{6}$$

where $\hat{g} : \Omega_c \times I \to \mathbb{R}$ is the corresponding function in the ALE frame; that is $\hat{g}(\boldsymbol{\xi}, t) = g(\boldsymbol{x}(\boldsymbol{\xi}, t), t) = g(\mathcal{A}_t(\boldsymbol{\xi}), t)$. Taking the time derivative of the ALE mapping defines the ALE velocity $\boldsymbol{w}$ as

$$\boldsymbol{w}(\boldsymbol{x}, t) = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{\xi}} (\mathcal{A}_t^{-1}(\boldsymbol{x}), t). \tag{7}$$

It is important to note that the ALE velocity $\boldsymbol{w}$ on $\Gamma(t)$ will, in general, be different from the material velocity $\boldsymbol{u}$. When $\boldsymbol{w} = \boldsymbol{u}$ the ALE transformation will be purely Lagrangian in nature. To relate the time derivatives with respect to the ALE transformation to the material derivative, a standard application of the chain rule gives

$$\left.\frac{\partial c}{\partial t}\right|_{\boldsymbol{\xi}} = \left.\frac{\partial c}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{w} \cdot \nabla c. \tag{8}$$

The reformulation of (2) in terms of the ALE reference frame therefore takes the form

$$\left.\frac{\partial c}{\partial t}\right|_{\boldsymbol{\xi}} - \boldsymbol{w} \cdot \nabla c = D \Delta c + f(c), \qquad \boldsymbol{\xi} \in \Omega_c. \tag{9}$$

Similarly, on the boundary the ALE formulation of (4) takes the form

$$\left.\frac{\partial c_s}{\partial t}\right|_{\boldsymbol{\xi}} + \nabla_\Gamma \cdot (\boldsymbol{u} c_s) - \boldsymbol{w} \cdot \nabla_\Gamma c = D_s \Delta_\Gamma c + g(c|_{\Gamma(t)}, c_s) + h(c_s), \qquad \boldsymbol{\xi} \in \partial\Omega_c. \tag{10}$$

The ALE reformulated equations (9) and (10) remain coupled through the flux boundary condition (3).

2.2. A conservative weak ALE formulation

To construct a weak formulation of (9), we consider a space of admissible test functions defined on the reference domain made of functions $\hat{v} \in H^1(\Omega_c)$. The ALE mapping then defines a set $\mathcal{H}(\Omega(t))$ of test functions on the domain $\Omega(t)$, as follows:

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

A weak formulation of (9) can be obtained using Reynolds transport formula which states that if $\psi(\boldsymbol{x}, t)$ is a function defined on $\Omega(t)$, and $V_t \subseteq \Omega(t)$ such that $V_t = \mathcal{A}_t(V_c)$ with $V_c \subseteq \Omega_c$, then

$$\frac{d}{dt} \int_{V_t} \psi(\boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{x} = \int_{V_t} \left(\left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{\xi}} + \psi \nabla \cdot \boldsymbol{w} \right) \mathrm{d}\boldsymbol{x} = \int_{V_t} \left(\left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{x}} + \nabla \psi \cdot \boldsymbol{w} + \psi \nabla \cdot \boldsymbol{w} \right) \mathrm{d}\boldsymbol{x}. \tag{11}$$

As functions $\hat{v} \in H^1(\Omega_c)$ do not depend on time, then for any $v \in \mathcal{H}(\Omega(t))$ we can establish from (11) that

$$\frac{d}{dt}\int_{\Omega(t)} v\,\mathrm{d}\boldsymbol{x} = \int_{\Omega(t)} v\nabla\cdot\boldsymbol{w}\,\mathrm{d}\boldsymbol{x} \tag{12}$$

and

$$\frac{d}{dt}\int_{\Omega(t)} v\psi\,\mathrm{d}\boldsymbol{x} = \int_{\Omega(t)} v\left(\left.\frac{\partial\psi}{\partial t}\right|_{\boldsymbol{\xi}} + \psi\nabla\cdot\boldsymbol{w}\right)\mathrm{d}\boldsymbol{x}. \tag{13}$$

Multiplying (9) by a test function $v\in\mathcal{H}(\Omega(t))$, integrating over $\Omega(t)$ and the use of (3), (12) and (13) gives the conservative weak form: find c such that

$$\begin{aligned}&\frac{d}{dt}\int_{\Omega(t)} cv\,\mathrm{d}\boldsymbol{x} - \int_{\Omega(t)}(\nabla\cdot(\boldsymbol{w}c))\,v\,\mathrm{d}\boldsymbol{x} + D\int_{\Omega(t)}\nabla c\cdot\nabla v\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma(t)}(g-(\boldsymbol{u}\cdot\boldsymbol{n})c)v\,\mathrm{d}s\\ &\quad=\int_{\Omega(t)} f v\,\mathrm{d}\boldsymbol{x},\quad \forall v\in\mathcal{H}(\Omega(t)).\end{aligned} \tag{14}$$

Similarly, on the boundary we have the conservative weak formulation: find c_s such that

$$\begin{aligned}&\frac{d}{dt}\int_{\Gamma(t)} c_s v_s\,\mathrm{d}s + \int_{\Gamma(t)}(\nabla_\Gamma\cdot((\boldsymbol{u}-\boldsymbol{w})c_s))\,v_s\,\mathrm{d}s + D_s\int_{\Gamma(t)}\nabla_\Gamma c_s\cdot\nabla_\Gamma v_s\,\mathrm{d}s\\ &\quad=\int_{\Gamma(t)}(g+h)v_s\,\mathrm{d}s,\quad \forall v_s\in\mathcal{H}_s(\Gamma(t)),\end{aligned} \tag{15}$$

where

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

is the space of test function on $\Gamma(t)$.

3. Moving finite element discretisation

3.1. Spatial semi-discretisation

We will assume that, for each $t\in[0,T]$, the physical and reference domains $\Omega(t)$ and Ω_c are approximated by polygonal domains $\Omega_h(t)$ and $\Omega_{c,h}$, respectively. We will assume that $\Omega_{c,h}$ is covered by a fixed triangulation $\mathcal{T}_{h,c}$ with straight edges, so that $\Omega_{c,h}=\cup_{K\in\mathcal{T}_{h,c}}K$. The approximation of the boundary domain $\Gamma_h(t)$ is chosen to be simply the boundary of $\Omega_h(t)$. The total number of elements of $\mathcal{T}_{h,c}$ will be denoted by N. The total number of vertices of $\mathcal{T}_{h,c}$ will be denoted by $\mathcal{N}$ and the number of vertices on the boundary as $\mathcal{N}_s$. We define the Lagrangian finite element spaces on $\mathcal{T}_{h,c}$ as

$$\begin{aligned}&\mathcal{L}^1(\Omega_{c,h}) = \{\hat{v}_h\in H^1(\Omega_{c,h}) : \hat{v}_h|_K\in\mathbb{P}_1(K),\ \forall K\in\mathcal{T}_{h,c}\},\\ &\mathcal{L}^1_0(\Omega_{c,h}) = \{\hat{v}_h\in H^1(\Omega_{c,h}) : \hat{v}_h|_K\in\mathcal{L}^1(\Omega_{c,h}) : \hat{v}_h=0,\ \boldsymbol{\xi}\in\Gamma_{c,h}\},\end{aligned} \tag{16}$$

where $\mathbb{P}_1(K)$ is the space of linear polynomials on K.

In section 4 we will describe a procedure for evolving the nodal positions of the triangulation covering $\Omega_h(t)$. Given the location of the mesh nodes, the ALE mapping will be interpolated using piecewise linear elements giving rise to a discrete mapping $\mathcal{A}_{h,t}\in\mathcal{L}^1(\Omega_{c,h})^2$ of the form

$$\boldsymbol{x}_h(\boldsymbol{\xi},t) = \mathcal{A}_{h,t}(\boldsymbol{\xi}) = \sum_{i=1}^{\mathcal{N}}\boldsymbol{x}_i(t)\hat{\phi}_i(\boldsymbol{\xi}), \tag{17}$$

where $\boldsymbol{x}_i(t)=\mathcal{A}_{h,t}(\boldsymbol{\xi}_i)$ denotes the position of node i at time t, and $\hat{\phi}_i$ is the associated nodal basis function in $\mathcal{L}^1(\Omega_{c,h})$. The discretised ALE velocity therefore takes the form

$$\boldsymbol{w}_h(\boldsymbol{\xi},t) = \sum_{i=1}^{\mathcal{N}}\dot{\boldsymbol{x}}_i(t)\hat{\phi}_i(\boldsymbol{\xi}).$$

Let $\mathcal{T}_{h,t}$ be the image of the reference triangulation $\mathcal{T}_{h,c}$ under the discrete ALE mapping $\mathcal{A}_{h,t}$. Since the mapping is linear, each K_t which is the image of a triangle $K\in\mathcal{T}_{h,c}$, is also a triangle with straight edges. Using the ALE mapping, the finite element test space on $\Omega_h(t)$ is therefore defined as

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

The finite element spatial discretisation of the conservative ALE formulation (14) therefore takes the form: find $c_h(t) \in \mathcal{H}_h(\Omega_h(t))$ such that

$$\frac{d}{dt}\int_{\Omega_h(t)} c_h v_h \,\mathrm{d}\boldsymbol{x} - \int_{\Omega_h(t)} (\nabla\cdot(\boldsymbol{w}_h c_h)) v_h \,\mathrm{d}\boldsymbol{x} + D\int_{\Omega_h(t)} \nabla c_h\cdot\nabla v_h \,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_h(t)} (g - (\boldsymbol{u}\cdot\boldsymbol{n})c_h) v_h \,\mathrm{d}s$$
$$= \int_{\Omega_h(t)} f v_h \,\mathrm{d}\boldsymbol{x},\quad \forall v_h \in \mathcal{H}_h(\Omega_h(t)). \tag{18}$$

Similarly, on the boundary, we have the weak formulation: find $c_{s,h} \in \mathcal{H}_{s,h}(\Gamma_h(t))$ such that

$$\frac{d}{dt}\int_{\Gamma_h(t)} c_{s,h} v_{s,h} \,\mathrm{d}s + \int_{\Gamma_h(t)} (\nabla_\Gamma\cdot((\boldsymbol{u}-\boldsymbol{w}_h)c_{s,h}))\, v_{s,h} \,\mathrm{d}s + D_s\int_{\Gamma_h(t)} \nabla_\Gamma c_{s,h}\cdot\nabla_{\Gamma_h} v_{s,h} \,\mathrm{d}s$$
$$= \int_{\Gamma_h(t)} (g+h) v_{s,h} \,\mathrm{d}s,\quad \forall v_{s,h} \in \mathcal{H}_{s,h}(\Gamma_h(t)). \tag{19}$$

The finite element approximation of the bulk and surface species can be expressed as

$$c_h(\boldsymbol{x},t) = \sum_{j=1}^{\mathcal{N}} c_j(t)\phi_j(\boldsymbol{x},t),\quad \text{and}\quad c_{s,h}(\boldsymbol{x},t) = \sum_{j=1}^{\mathcal{N}_s} c_{s,j}(t)\phi_{s,j}(\boldsymbol{x},t),$$

where $\{\phi_j(\boldsymbol{x},t)\}_{j=1}^{\mathcal{N}}$ and $\{\phi_{s,j}(\boldsymbol{x},t)\}_{j=1}^{\mathcal{N}_s}$ are the time-dependent bulk and surface nodal basis functions. If $\boldsymbol{C}(t) = \{c_j(t)\}_{j=1}^{\mathcal{N}}$ and $\boldsymbol{C}_s(t) = \{c_{s,j}(t)\}_{j=1}^{\mathcal{N}_s}$, we can express (18) as the system of ordinary differential equations

$$\frac{d}{dt}(M(t)\boldsymbol{C}(t)) + [K(t) + A(t,\boldsymbol{w}_h(t)) - B(t,\boldsymbol{w}_h(t))]\boldsymbol{C}(t) + \boldsymbol{D}(\boldsymbol{C}(t),\boldsymbol{C}_s(t)) = \boldsymbol{F}(\boldsymbol{C}(t)), \tag{20}$$

where

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

is the (time-dependent) mass matrix, while

$$[K(t)]_{ij} = D\int_{\Omega_h(t)} (\nabla\phi_j(t)\cdot\nabla\phi_i(t)) \,\mathrm{d}\boldsymbol{x},$$

$$[A(t,\boldsymbol{w}_h(t))]_{ij} = \int_{\Gamma_h(t)} [(\boldsymbol{u}-\boldsymbol{w}_h)\cdot\boldsymbol{n}]\phi_i(t)\phi_j(t) \,\mathrm{d}s,$$

$$[B(t,\boldsymbol{w}_h(t))]_{ij} = -\int_{\Omega_h(t)} [\boldsymbol{w}_h\cdot\nabla\phi_i(t)]\phi_j(t) \,\mathrm{d}\boldsymbol{x},$$

$$[\boldsymbol{D}(\boldsymbol{C}(t),\boldsymbol{C}_s(t))]_i = \int_{\Gamma_h(t)} [g(c_h(t),c_{s,h}(t)) - (\boldsymbol{u}\cdot\boldsymbol{n})c_h(t)]\phi_i(t) \,\mathrm{d}s,$$

and the load vector

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

Note that the vector $\boldsymbol{D}$ will be sparse as only those values of i corresponding to boundary vertices will be non-zero. The spatial discretisation of the boundary equation (19) results in a system of ODEs

$$\frac{d}{dt}(M_s(t)\boldsymbol{C}_s(t)) + [K_s(t) + A_s(t,\boldsymbol{w}_h(t))]\boldsymbol{C}_s(t) = \boldsymbol{D}_s(\boldsymbol{C}(t),\boldsymbol{C}_s(t)) + \boldsymbol{H}(\boldsymbol{C}_s(t)), \tag{21}$$

where

$$[M_s(t)]_{ij} = \int_{\Gamma_h(t)} \phi_{s,i}(t)\phi_{s,j}(t)\,\mathrm{d}s,$$

$$[K_s(t)]_{ij} = D_s \int_{\Gamma(t)} (\nabla_\Gamma \phi_{s,j}(t) \cdot \nabla_\Gamma \phi_{s,i}(t))\,\mathrm{d}s,$$

$$[A_s(t, \boldsymbol{w}_h(t))]_{ij} = \int_{\Gamma_h(t)} \left([\nabla_\Gamma \cdot (\boldsymbol{u} - \boldsymbol{w}_h)]\phi_{s,i}(t)\phi_{s,j}(t) + [(\boldsymbol{u} - \boldsymbol{w}_h) \cdot \nabla_\Gamma \phi_{s,j}(t)]\phi_{s,i}(t)\right)\,\mathrm{d}s,$$

$$[\boldsymbol{H}(\boldsymbol{C}_s(t))]_i = \int_{\Gamma_h(t)} h(c_{s,h}(t))\phi_{s,i}(t)\,\mathrm{d}s,$$

and $\boldsymbol{D}_s$ are the appropriately reordered non-zero elements of $\boldsymbol{D}$.

3.2. Temporal integration

To obtain a temporal discretisation of (20) and (21) we subdivide $[0, T]$ into N_t equal time intervals of size $\Delta t = T/N_t$ and denote $t^n = n\Delta t$, $n = 0, 1, \ldots, N_t$. We will discretise the ALE mapping using linear interpolation between time levels. That is we will define

$$\mathcal{A}_{h,\Delta t}(\boldsymbol{\xi}, t) = \frac{t - t^n}{\Delta t}\mathcal{A}_{h,t^{n+1}}(\boldsymbol{\xi}) + \frac{t^{n+1} - t}{\Delta t}\mathcal{A}_{h,t^n}(\boldsymbol{\xi}), \quad t \in [t_n, t_{n+1}), \tag{22}$$

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

$$\begin{aligned} \boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{\xi}) &= \frac{\mathcal{A}_{h,t^{n+1}} - \mathcal{A}_{h,t^n}}{\Delta t}, \quad t \in [t^n, t^{n+1}), \\ \boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{x}, t) &= \boldsymbol{w}^{n+1}_{h,\Delta t}(\boldsymbol{\xi}) \circ \mathcal{A}^{-1}_{h,\Delta t}(\boldsymbol{x}). \end{aligned} \tag{23}$$

The temporal discretisation of the coupled systems (20) and (21) is obtained using a modified Crank–Nicolson semi-implicit approach. We first predict the boundary solution $\tilde{\boldsymbol{C}}_s^{n+1}$ using a semi-implicit backward Euler method, where the linear diffusion and mesh movement terms are treated implicitly, and the nonlinear reaction and coupling terms are treated explicitly. The predicted boundary solution therefore satisfies the linear system

$$[M_s^{n+1} + \Delta t(K_s^{n+1} + A_s^{n+1})]\tilde{\boldsymbol{C}}_s^{n+1} = M_s^n \boldsymbol{C}_s^n + \Delta t[\boldsymbol{D}_s(\boldsymbol{C}^n, \boldsymbol{C}_s^n) + \boldsymbol{H}(\boldsymbol{C}_s^n)]. \tag{24}$$

The bulk approximation is then updated using a Crank–Nicolson step

$$\begin{aligned} &[M^{n+1} + \tfrac{1}{2}\Delta t(K^{n+1} + A^{n+1} + B^{n+1})]\boldsymbol{C}^{n+1} \\ &\quad = [M^n - \tfrac{1}{2}\Delta t(K^n + A^n + B^n)]\boldsymbol{C}^n + \tfrac{1}{2}\Delta t[\boldsymbol{F}(\boldsymbol{C}^{n+1}) + \boldsymbol{F}(\boldsymbol{C}^n) - \boldsymbol{D}(\boldsymbol{C}^{n+1}, \tilde{\boldsymbol{C}}_s^{n+1}) - \boldsymbol{D}(\boldsymbol{C}^n, \boldsymbol{C}_s^n)]. \end{aligned} \tag{25}$$

The nonlinear system (25) is solved using Newton iteration. Finally, to ensure that the boundary solution is second-order in time, we perform a Crank–Nicolson correction step

$$\begin{aligned} &[M_s^{n+1} + \tfrac{1}{2}\Delta t(K_s^{n+1} + A_s^{n+1})]\boldsymbol{C}_s^{n+1} \\ &\quad = [M_s^n - \tfrac{1}{2}\Delta t(K_s^n + A_s^n)]\boldsymbol{C}_s^n + \tfrac{1}{2}\Delta t[\boldsymbol{D}_s(\boldsymbol{C}^{n+1}, \tilde{\boldsymbol{C}}_s^{n+1}) + \boldsymbol{D}_s(\boldsymbol{C}^n, \boldsymbol{C}_s^n) + \boldsymbol{H}(\tilde{\boldsymbol{C}}_s^{n+1}) + \boldsymbol{H}(\boldsymbol{C}_s^n)]. \end{aligned} \tag{26}$$

Note that this correction step only requires the solution of a linear system of equations even if the reaction terms on the surface, g and h are nonlinear. The linear systems arising above are solved using the iterative method BiCGSTAB [52] and an incomplete LU (ILU) factorisation as a preconditioner.

For the cell migration application considered later, we note that the diffusive time scales in the extracellular region are often much shorter than the time scale associated with cell migration. As the time integration scheme above is fully implicit in the diffusive terms it is therefore robust to the choice of the time step for these applications.

3.3. A model bulk–surface problem on a stationary domain

To get an indication of the spatial and temporal convergence rate of the coupled bulk–surface finite element discretisation, we apply it to the solution of the following model problem:

$$\frac{\partial c}{\partial t} = \Delta c, \qquad \boldsymbol{x} \in \Omega \tag{27}$$

$$\frac{\partial c_s}{\partial t} = \Delta_\Gamma c_s + c - c_s, \quad \boldsymbol{x} \in \Gamma, \tag{28}$$

$$-\frac{\partial c}{\partial n} = c - c_s, \quad \boldsymbol{x} \in \Gamma, \tag{29}$$

where Ω is the unit circle.

This problem can be tackled analytically using polar coordinates $x = r\cos\theta$, $y = r\sin\theta$ so that

$$\frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial r^2} + \frac{1}{r}\frac{\partial c}{\partial r} + \frac{1}{r^2}\frac{\partial^2 c}{\partial \theta^2}, \qquad \boldsymbol{x} \in \Omega$$

$$\frac{\partial c_s}{\partial t} = \frac{\partial^2 c_s}{\partial \theta^2} + c|_{r=1} - c_s, \quad \boldsymbol{x} \in \Gamma,$$

and

$$-\left.\frac{\partial c}{\partial r}\right|_{r=1} = c|_{r=1} - c_s.$$

Similar to Novak et al. [43], we look for a solution in the form

$$c_s(\theta, t) = Ae^{-k^2 t}\cos\theta,$$

and

$$c(r, \theta, t) = \rho(r)c_s(\theta, t).$$

The following exact solution has been used to test the convergence of the finite element discretisation:

$$c(r, \theta, t) = J_1(rk)e^{-k^2 t}\cos\theta$$

and

$$c_s(\theta, t) = \frac{J_1(k)}{2 - k^2}e^{-k^2 t}\cos\theta,$$

where J_1 is the first-order Bessel function of the first kind and $k = 1.177706027$.

Fig. 2 shows the computed approximate solutions in the bulk domain and on the domain boundary using an isotropic mesh with maximal cell diameter $h = 0.1$ and a time step $\Delta t = 2 \times 10^{-4}$. We can see that the method performs well for these values of the discretisation parameters. To test the spatial rate of convergence of the algorithm, simulations were performed on a sequence of increasingly refined isotropic meshes. To ensure that the error was dominated by its spatial component, a sufficiently small time step of $\Delta t = 10^{-3}$ was used. Fig. 3(a) shows the maximum error over all grid nodes for both the bulk and surface numerical solutions and we can see that both converge at the rate of $O(h^2)$, as expected. To investigate the temporal rate of converge, simulations were performed using a fine mesh with $N = 150{,}000$ elements and various time steps. We can see from Fig. 3(b), that the three-step solution procedure (24)–(26) furnishes approximations which are second-order accurate in time. Note that if the surface solution correction step (26) were omitted then, as expected, the resulting approximations were only temporally first-order accurate.

4. Practical evaluation of ALE mapping

For the ALE mapping to be useful, it must satisfy a number of properties. First, it must be inexpensive to construct, relative to the cost of solving the physical problem. Second, the resulting meshes covering $\Omega(t)$ must be of good quality in the sense that mesh elements are adapted to salient features such at steep boundary or interior layers in the physical solution. Finally, the meshes must evolve smoothly in time to avoid the need to use small time steps to maintain numerical stability.

4.1. Bulk domain mesh generation

To avoid potential mesh crossings or foldings, we derive a suitable evolution equation for the inverse ALE mapping $\mathcal{A}_t^{-1}(\boldsymbol{x}) = \boldsymbol{\xi}(\boldsymbol{x}, t)$ rather than $\mathcal{A}_t(\boldsymbol{\xi}) = \boldsymbol{x}(\boldsymbol{\xi}, t)$ (see, for example, the discussion in [12]). As shown in Fig. 4, a mesh $\mathcal{T}_{h,t}$ on $\Omega_h(t)$ can then be generated as the preimage of a fixed mesh $\mathcal{T}_{h,c}$ on $\Omega_{c,h}$. As introduced in [22], we choose the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ corresponding to a fixed value of t in order to minimise the functional

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega_t} [(\nabla\xi)^T G^{-1}(\nabla\xi) + (\nabla\eta)^T G^{-1}(\nabla\eta)]\,\mathrm{d}\boldsymbol{x}, \tag{30}$$

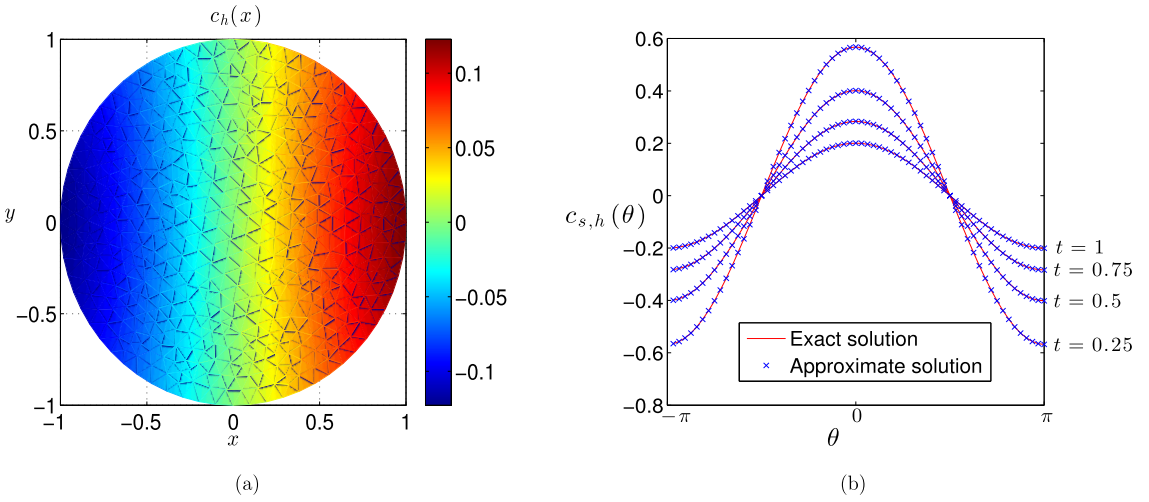


Fig. 2. Numerical solution of coupled model problem (27)–(29) on a stationary unit circle. (a) Approximate bulk solution $c_h(\boldsymbol{x})$ at $t = 1$. (b) Approximate solution $c_{s,h}(\theta)$ on the boundary compared to the exact solution.

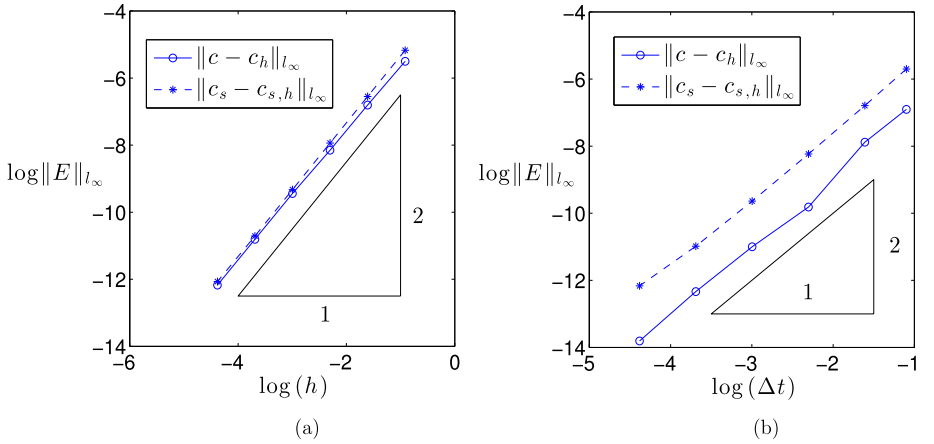


Fig. 3. Second-order convergence of the maximum nodal error as (a) $h \to 0$ and (b) $\Delta t \to 0$ for a coupled model problem on a unit circle.

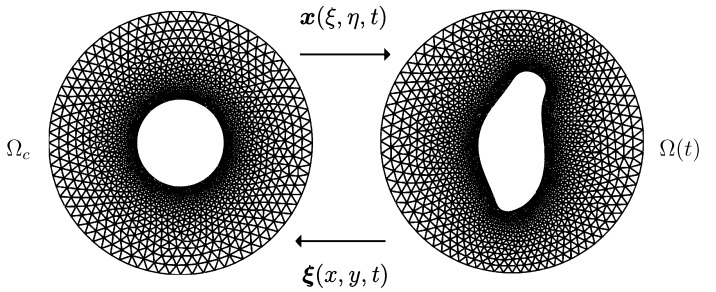


Fig. 4. A moving mesh covering $\Omega(t)$ is the image of a fixed mesh on a reference domain Ω_c through a time-dependent ALE mapping $\mathcal{A}_t(\boldsymbol{\xi}) = \boldsymbol{x}(\boldsymbol{\xi}, t)$.

where G is a 2×2 symmetric positive definite matrix referred to as a monitor matrix and ∇ is the gradient operator with respect to $\boldsymbol{x}$. It can be shown that the functional (30) is coercive and uniformly convex and hence has a unique minimiser [23].

Rather than directly attempt to minimise (30), a more robust procedure is to evolve the mapping according to the modified gradient flow equations

$$\frac{\partial \xi}{\partial t} = \frac{P}{\tau} \nabla \cdot (G^{-1} \nabla \xi), \qquad \text{and} \qquad \frac{\partial \eta}{\partial t} = \frac{P}{\tau} \nabla \cdot (G^{-1} \nabla \eta). \tag{31}$$

Here, $\tau > 0$ is a user-specified temporal smoothing parameter which affects the temporal scale over which the mesh adapts and P is a positive function of $(\boldsymbol{x}, t)$, chosen such that the mesh movement has a spatially uniform time scale [19]. The gradient flow structure of (31) ensures that the mesh evolves smoothly in time and improves robustness in the choice of the initial mesh.

The selection of an appropriate monitor matrix is crucial to the success of mesh adaptation. In this paper, we will consider the monitor matrix proposed by Winslow [53]

$$G = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}, \tag{32}$$

where $M(\boldsymbol{x}, t)$ is a positive weight function called a *monitor function*. A suitable monitor function is essential to the success of adaptive moving mesh methods. The monitor function should be taylored to the type of PDE being solved and the numerical method being used. Monitor functions based on interpolation estimates, a posteriori error estimates and adaptation to solution features have all been proposed [3,8,23]. If no such estimate exists then the monitor function could be any smooth function designed to adapt the mesh towards important solution features such moving interfaces or boundaries (see for example [5]).

In practice, we interchange the roles of the dependent and independent variables in (31), since it's the location of the physical mesh points $\{\boldsymbol{x}_i(t)\}_{i=1}^{\mathcal{N}}$ that defines the ALE map. With a Winslow-type monitor matrix (32) the resulting MMPDEs take the form

$$\tau \frac{\partial \boldsymbol{x}}{\partial t} = P(a\boldsymbol{x}_{\xi\xi} + b\boldsymbol{x}_{\xi\eta} + c\boldsymbol{x}_{\eta\eta} + d\boldsymbol{x}_{\xi} + e\boldsymbol{x}_{\eta}), \quad (\xi, \eta) \in \Omega_c, \tag{33}$$

where

$$a = \frac{1}{M}\frac{x_\eta^2 + y_\eta^2}{J^2}, \quad b = -\frac{2}{M}\frac{(x_\xi x_\eta + y_\xi y_\eta)}{J^2}, \quad c = \frac{1}{M}\frac{x_\xi^2 + y_\xi^2}{J^2},$$
$$d = \frac{1}{(MJ)^2}[M_\xi(x_\eta^2 + y_\eta^2) - M_\eta(x_\xi x_\eta + y_\xi y_\eta)], \quad e = \frac{1}{(MJ)^2}\left[-M_\xi(x_\xi x_\eta + y_\xi y_\eta) + M_\eta(x_\xi^2 + y_\xi^2)\right],$$

and $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the ALE mapping. To complete the specification of the coordinate transformation, the MMPDE must be supplemented by initial and boundary conditions. In all the applications described later, the computational domain $\Omega_{c,h} = \Omega_h(0)$, and the initial mesh over the physical domain is obtained using the Matlab toolbox Distmesh [46]. To allow the mesh points to move along the boundary $\Gamma_h(t)$, they are obtained using a one-dimensional moving mesh approach outlined in section 4.2. Once the boundary nodes have been evolved over one time step, they then become Dirichlet conditions completing the specification for (33).

The numerical solution of (33) requires spatial and temporal discretisation. In space, we discretise using standard linear Galerkin finite elements. In the time direction, we use a backward Euler integration scheme to update the solution at $t = t^{n+1}$, and to avoid solving nonlinear algebraic systems, we evaluate the coefficients $a, c, \ldots, e$ at the time $t = t^n$. We therefore seek $\boldsymbol{x}_h^{n+1} \in (\mathcal{L}^1(\Omega_{c,h}))^2$ such that

$$\tau \int_{\Omega_{c,h}} \left(\frac{\boldsymbol{x}_h^{n+1} - \boldsymbol{x}_h^n}{\Delta t}\right) \cdot \hat{\boldsymbol{v}}_h \, \mathrm{d}\boldsymbol{\xi} + \int_{\Omega_{c,h}} \Big[(\boldsymbol{x}_h^{n+1})_\xi \cdot (a^n \hat{\boldsymbol{v}}_h)_\xi + (\boldsymbol{x}_h^{n+1})_\eta \cdot (c^n \hat{\boldsymbol{v}}_h)_\eta$$
$$+ \frac{1}{2}[(\boldsymbol{x}_h^{n+1})_\xi \cdot (b^n \hat{\boldsymbol{v}}_h)_\eta + (\boldsymbol{x}_h^{n+1})_\eta \cdot (b^n \hat{\boldsymbol{v}}_h)_\xi] - [d^n(\boldsymbol{x}_h^{n+1})_\xi + e^n(\boldsymbol{x}_h^{n+1})_\eta] \cdot \hat{\boldsymbol{v}}_h \Big] \mathrm{d}\boldsymbol{\xi} = 0, \tag{34}$$

for all $\hat{\boldsymbol{v}}_h \in (\mathcal{L}_0^1(\Omega_{c,h}))^2$. The resulting linear systems are again solved using the iterative method BiCGSTAB and an incomplete LU (ILU) factorisation as a preconditioner. An analysis of the performance of this iterative solver for the discretised MMPDE equations can be found in [4].

4.2. Boundary mesh generation

We consider the class of boundary movements given by the following evolution law for the normal velocity

$$\mathcal{V}(\boldsymbol{x}, t) = \alpha\kappa + \beta, \quad \boldsymbol{x} \in \Gamma(t), \tag{35}$$

where κ is the curvature. Within the context of biological cell applications, the functions α and β could model the physical forces exerted on the membrane by cortical tension, and protrusions caused by actin polymerisation, respectively.

We consider the solution of (35) using a Lagrangian approach, where the main idea is to represent the flow of curves by the position vector $\boldsymbol{x}$ which is the solution of the geometric evolution equation

$$\dot{\boldsymbol{x}} = \mathcal{V}\boldsymbol{n} + \mathcal{B}\boldsymbol{t}, \tag{36}$$

where $\boldsymbol{n}$ and $\boldsymbol{t}$ are unit normal and tangent vectors, respectively. Note that the presence of the tangential velocity $\mathcal{B}$ has no effect on the shape of the evolving curve but it is well known that the incorporation of a suitably chosen non-zero value of $\mathcal{B}$ can avoid the major drawback of the Lagrangian approach which is the possibility that grid points merge, resulting in a loss of stability and accuracy [2,1,37].

An embedded regular plane curve Γ can be parameterised by the smooth function $\boldsymbol{x} : S^1 \to \mathbb{R}^2$, i.e. $\Gamma = \text{Image}(\boldsymbol{x}) := \{\boldsymbol{x}(\sigma),\ \sigma \in S^1\}$. Taking into account the periodic boundary conditions at $\sigma = 0, 1$, we shall hereafter identify S^1 with the interval $[0, 1]$. The unit arc-length parameterisation will be denoted by s and it can be shown that $\boldsymbol{x}_s = \boldsymbol{t}$ and $\boldsymbol{x}_{ss} = \kappa \boldsymbol{n}$. The arc-length of the curve from the point $\boldsymbol{x}(\sigma_0)$ to $\boldsymbol{x}(\sigma)$ is given by

$$s(\sigma) = \int_{\sigma_0}^{\sigma} \sqrt{x_\sigma^2 + y_\sigma^2}\, \mathrm{d}\sigma = \int_{\sigma_0}^{\sigma} \|\boldsymbol{x}_\sigma\|\, \mathrm{d}\sigma,$$

where $\|\cdot\|$ denotes the l_2-norm. Using the chain rule

$$\boldsymbol{x}_\sigma = \boldsymbol{x}_s \frac{ds}{d\sigma} = \boldsymbol{x}_s \|\boldsymbol{x}_\sigma\| \tag{37}$$

and differentiation of (37) leads to the relation

$$\boldsymbol{x}_{\sigma\sigma} = \boldsymbol{x}_{ss} \|\boldsymbol{x}_\sigma\|^2 + \boldsymbol{x}_s \|\boldsymbol{x}_\sigma\|_s \|\boldsymbol{x}_\sigma\|. \tag{38}$$

Multiplying through (38) by $\boldsymbol{n}$, we get an expression for κ and hence in terms of the parameterisation σ, we can express the normal velocity as

$$\mathcal{V} = \dot{\boldsymbol{x}} \cdot \boldsymbol{n} = \alpha \left(\frac{\boldsymbol{x}_{\sigma\sigma} \cdot \boldsymbol{n}}{\|\boldsymbol{x}_\sigma\|^2} \right) + \beta. \tag{39}$$

Control of the mesh spacing in the tangential direction can be achieved by ensuring that a weighted arc-length between grid nodes is constant. That is, if $M(\boldsymbol{x}, t) > 0$ is a positive monitor function, then we set

$$M \frac{ds}{d\sigma} = \text{constant}$$

and hence

$$\left(M \frac{ds}{d\sigma} \right)_\sigma = (M \|\boldsymbol{x}_\sigma\|)_\sigma = 0. \tag{40}$$

The differential-algebraic system (39) and (40) could, in theory, be used to evolve the boundary $\Gamma(t)$; the case $\alpha = 1$, $\beta = 0$ and $M = 1$, which aims to construct a uniform arc-length parameterisation, is exactly the equation system used by Pan and Wetton [45]. In [39] we used the parameterised finite element method (PFEM) [2] to evolve the approximation of a curve describing an evolving cell membrane. This method introduces an intrinsic tangential velocity $\mathcal{B}$ such that the arc-length between grid nodes is equidistributed. When using the PFEM to generate boundary node displacements, which are then used as boundary data for meshes covering bulk regions, we have found that the lack of control of the PFEM intrinsic tangential velocity can lead to poor quality bulk meshes which require frequent remeshing. Experience with the generation of adaptive moving meshes suggests however that the introduction of an evolution equation for the tangential velocity, driven by the mesh equidistribution condition (40), can significantly improve stability and robustness [21,20,3]. Therefore, we consider the tangential velocity equation

$$\mathcal{B} = \dot{\boldsymbol{x}} \cdot \boldsymbol{t} = \frac{P}{\tau} (M \|\boldsymbol{x}_\sigma\|)_\sigma, \tag{41}$$

where, as before, τ and P are a temporal smoothing parameter and spatial balancing operator, respectively. The proposed procedure therefore involves the simultaneous solution of (39) and (41).

For simplicity, we discretise (39) and (41) using a finite difference method. The evolving curve at time t^n is represented by the discrete plane points $\boldsymbol{x}_i^n$, $i = 1, \ldots, \mathcal{N}_s$. The linear approximation of the curve is therefore given by the polygon with vertices $\boldsymbol{x}_i^n$, $i = 1, \ldots, \mathcal{N}_s$. Due to the periodicity conditions, we also use the additional values $\boldsymbol{x}_{-1}^n = \boldsymbol{x}_{\mathcal{N}_s - 1}^n$, $\boldsymbol{x}_0^n = \boldsymbol{x}_{\mathcal{N}_s}^n$ and $\boldsymbol{x}_{\mathcal{N}_s+1}^n = \boldsymbol{x}_1^n$. We use a uniform discretisation of the parameterisation interval $\sigma \in [0, 1]$ with step size $\Delta\sigma = 1/\mathcal{N}_s$. We use central finite differences to approximate the spatial derivatives in (39) and a first-order fully implicit temporal discretisation. This leads to a nonlinear system which is solved using Picard iteration. Let $\boldsymbol{x}^{[n,m]}$ denote the approximation of $\boldsymbol{x}$ at time level t^n and iteration level m, then for the normal velocity equations we have the linear system

$$\begin{aligned} &[-\mu_i^{[n+1,m]} \boldsymbol{x}_{i-1}^{[n+1,m+1]} + (1 + 2\mu_i^{[n+1,m]}) \boldsymbol{x}_i^{[n+1,m+1]} - \mu_i^{[n+1,m]} \boldsymbol{x}_{i+1}^{[n+1,m+1]}] \cdot \boldsymbol{n}_i^{[n+1,m]} \\ &\quad = \boldsymbol{x}_i^{[n+1,m]} \cdot \boldsymbol{n}_i^{[n+1,m]} + \Delta t \beta_i, \end{aligned} \tag{42}$$

for $i = 1, \ldots, \mathcal{N}_s$, where $\mu_i^{[n,m]} = 4\Delta t \alpha_i / (\|\boldsymbol{x}_{i+1}^{[n,m]} - \boldsymbol{x}_{i-1}^{[n,m]}\|^2)$. We approximate the unit tangent vector

$$\boldsymbol{t}_i^n = \frac{\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n}{\|\boldsymbol{x}_{i+1}^n - \boldsymbol{x}_{i-1}^n\|} = (t_1^n, t_2^n),$$

and hence we set $\boldsymbol{n}_i^n = (t_2^n, -t_1^n)$. Using a similar discretisation of the tangential velocity equation (41) we have

$$\begin{aligned}
&[-\nu_i^n \boldsymbol{x}_{i-1}^{[n+1,m+1]} + (1 + 2\nu_i^n)\boldsymbol{x}_i^{[n+1,m+1]} - \nu_i^n \boldsymbol{x}_{i+1}^{[n+1,m+1]}] \cdot \boldsymbol{t}_i^{[n+1,m]} \\
&\quad = \boldsymbol{x}_i^n \cdot \boldsymbol{t}_i^{[n+1,m]} + \frac{\Delta t P_i}{4\tau(\Delta\sigma)^2}(M_{i+1}^n - M_{i-1}^n)(\|\boldsymbol{x}_{i+1}^{[n+1,m]} - \boldsymbol{x}_{i-1}^{[n+1,m]}\|),
\end{aligned} \tag{43}$$

for $i = 1, \ldots, \mathcal{N}_s$, where $\nu_i^n = \Delta t M_i^n P_i / (\tau(\Delta\sigma)^2)$. The coupled set of $2\mathcal{N}_s$ equations (42) and (43) are solved for $\boldsymbol{x}^{[n+1,m+1]}$ and the Picard iteration is stopped when

$$\|\boldsymbol{x}^{[n+1,m+1]} - \boldsymbol{x}^{[n+1,m]}\| < \mathrm{TOL}.$$

The last iteration is then used as the approximation $\boldsymbol{x}^{n+1}$.

4.3. Example

To demonstrate the capability of the proposed adaptive grid generation procedures, we consider the evolution of an ellipse under mean curvature flow where $\mathcal{V} = -\alpha\kappa$, and simultaneously we require the bulk and surface meshes evolve to resolve the travelling wave profile

$$u(x, y, t) = \frac{1}{2}\left[1 + \tanh\left(\frac{x + t - 0.7}{0.6}\right)\right].$$

There are many possible choices for the monitor function but for illustrative purposes we will set

$$M(x, y, t) = 1 + \operatorname{sech}^2\left(\frac{x + t - 0.7}{0.6}\right).$$

The monitor function takes its maximum value at the centre of the travelling wave and decreases smoothly to a non-zero constant value far from the front. The resulting adapted mesh should therefore be clustered around the wave but almost uniform in the flat regions. The initial physical domain consists of the region exterior to the ellipse $4x^2 + 16y^2 = 1$ and interior to a unit circular far-field boundary; the fixed computational domain Ω_c is chosen to be the initial physical domain. The computational mesh is obtained using the Matlab toolbox Distmesh [46] and has $N = 9832$ elements and $\mathcal{N}_s = 98$ vertices on the boundary of the ellipse. The simulations are performed over the time interval $t \in [0, 7 \times 10^{-2}]$ using a constant time step $\Delta t = 10^{-3}$. The coefficient of the normal velocity of the evolving curve is $\alpha = 0.75$, and the temporal smoothing parameters in the moving mesh PDEs are both $\tau = 10^{-4}$. The computed meshes at four representative times are shown in Fig. 5. We can see that the initial mesh is adapted towards the wave front which is centred to the right of the ellipse at $x = 0.7$. Initially the boundary mesh points are distributed to equidistribute the arc-length as the monitor function $M \approx 1$ along the ellipse. At $t = 0.02$ and $t = 0.04$, we can see that the bulk meshes have moved to follow the wave front; the boundary meshes have also adapted correctly in the tangential direction and evolved in the normal direction according to the curvature of the boundary. Finally, at $t = 0.05$, the wave front has passed by the inner curve and the boundary mesh relaxes back to equidistribute the arc-length between mesh points. Fig. 6 shows the adapted boundary meshes; for clarity, at the two times when the wave front intersects the boundary, we have indicated its location by a vertical dashed line. We can see that at these times the boundary mesh has adapted well to the wave front and at the other two times the mesh equidistributes the arc-length of the evolving curve. To test the accuracy of the computed moving boundary, Fig. 7 shows the evolution of the exact and approximate area enclosed by the closed curve; the exact area for mean curvature flow takes the form $A(t) = A(0) - 2\pi\alpha t$. We can see that the approximate area decreases linearly in time, and at the times indicated it agrees well with the exact area.

5. The complete algorithm

We now describe the complete algorithm implementing the ALE–FEM scheme and the generation of the surface and bulk meshes. At time $t = t^n$ we assume we have a mesh $\mathcal{T}_{h,t^n}$ and the finite element approximations c_h^n and $c_{s,h}^n$ of the bulk and surface species, respectively. The following steps are then carried out to advance the mesh and the finite element approximations.

1. *Update the physical mesh*
   (a) Using $c_{s,h}^n$, calculate the normal velocity of Γ given by (35).
   (b) Update the location of the manifold Γ_h^{n+1} by solving the simultaneous systems (42) and (43).

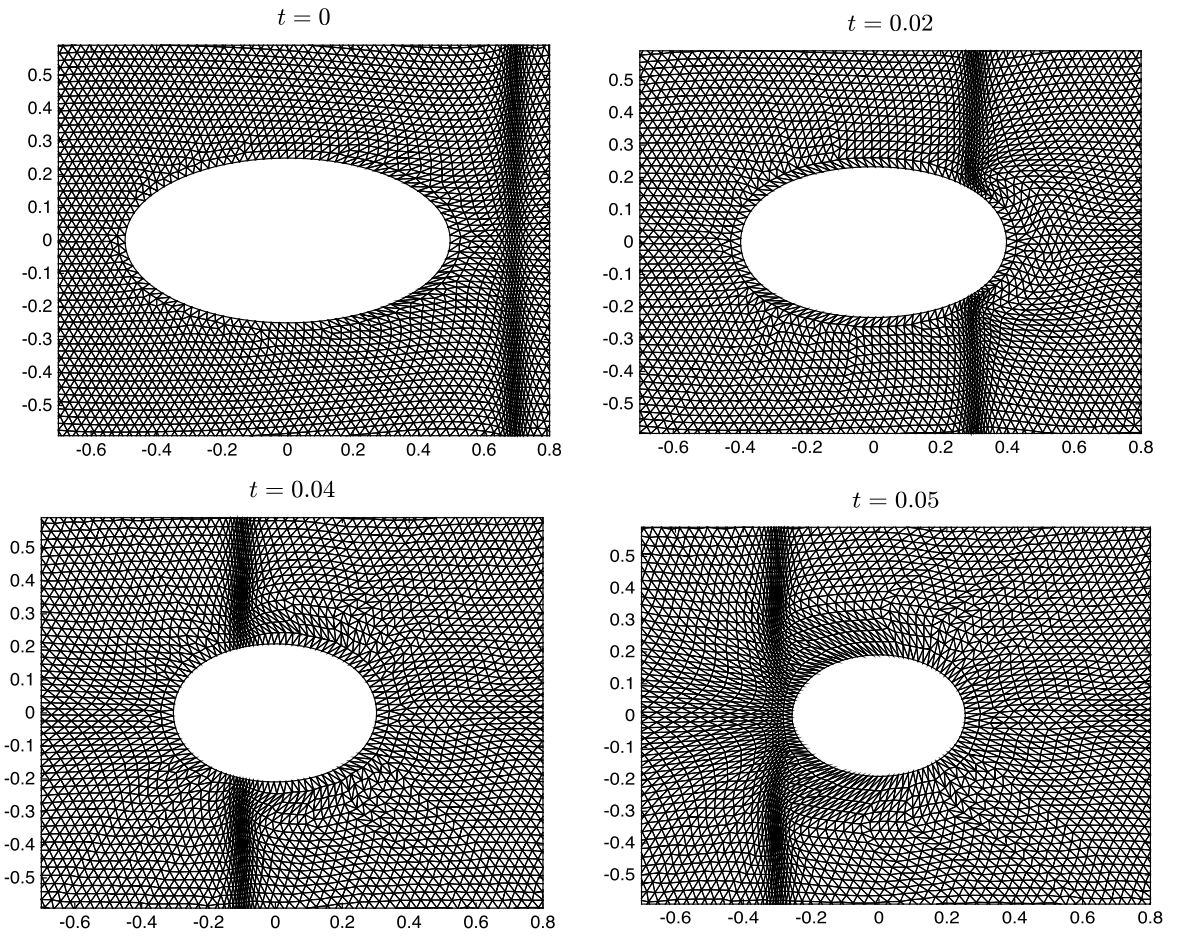


Fig. 5. Adaptive bulk meshes for a time-dependent domain where the inner boundary evolves by mean curvature flow. The meshes are adapted to a travelling wave profile moving across the domain from right to left.

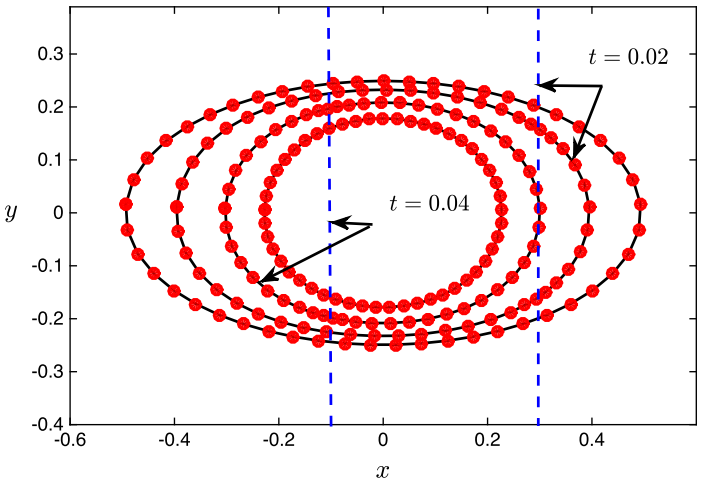


Fig. 6. Computed meshes for a curve evolving in the normal direction by mean curvature flow; the meshes are also adapted in the tangential direction to a travelling wave profile moving across the domain from right to left. Meshes are shown at $t = 0$, 0.02, 0.04 and $t = 0.05$. The dashed vertical lines correspond to the location of the wave front at the times indicated.

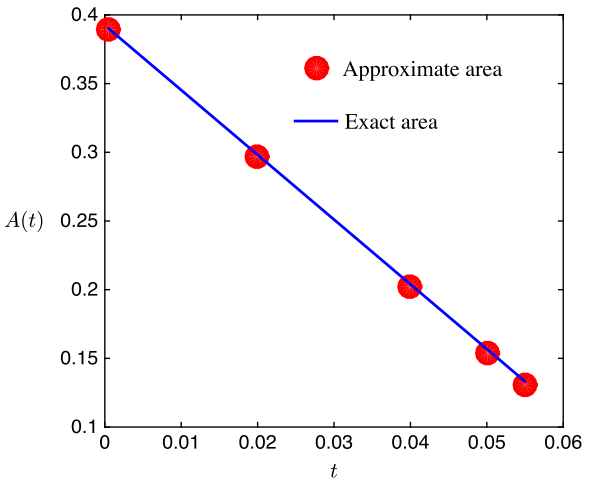


Fig. 7. Comparison of the evolution of the exact and computed area enclosed by a closed curve under mean curvature flow at $t = 0$, 0.02, 0.04, 0.05 and $t = 0.055$.

(c) Using the bulk approximation c_h^n, determine the mesh adaptation monitor function M.
(d) Using the updated boundary points as fixed Dirichlet data, update the interior mesh points by solving (34).
(e) Test for mesh quality. If the mesh is fine then goto 2. If not then re-grid and interpolate the solutions onto the new mesh and re-do the time step.

2. *Update the finite element solutions in the bulk and the surface*
(a) Use the meshes $\mathcal{T}_{h,t^{n+1}}$ and $\mathcal{T}_{h,t^n}$ to define the discrete ALE velocity $\boldsymbol{w}_h$.
(b) Update the solutions $c_{s,h}^{n+1}$ and c_h^{n+1} by solving (24)–(26).

To determine the quality of the bulk meshes we measure the minimum angle over all of the triangles and decide to remesh when this is below a given tolerance. The new mesh covering the physical domain is then used as the fixed computational mesh. The bulk and boundary solutions are then linearly interpolated onto the new mesh to allow further time integration.

6. Cell migration and chemotaxis

We now consider the application of the developed algorithm to the computational modelling of eukaryotic cell migration and chemotaxis. In [40,41] we developed a "pseudopod-centered" [25] model based on a system of reaction–diffusion equations that gives rise to a suitable spatiotemporal activator profile that can be used for the generation of pseudopods without the need for a driving external signal. The following set of equations was derived from a well-established discrete model developed by Meinhardt [35] (M model). The model describes the dynamic interaction between a membrane-bound local autocatalytic activator a, a rapidly distributed global inhibitor b and a local inhibitor c. Assuming that the cell boundary $\Gamma(t)$ moves with velocity $\boldsymbol{u}$, then for $\boldsymbol{x} \in \Gamma(t)$ the equations take the form

$$\frac{\partial a}{\partial t} + \nabla_\Gamma \cdot (\boldsymbol{u} a) = D_a \Delta_\Gamma a + \frac{s(a^2/b + b_a)}{(s_c + c)(1 + s_a a^2)} - r_a a, \tag{44}$$

$$\frac{\partial b}{\partial t} + \nabla_\Gamma \cdot (\boldsymbol{u} b) = D_b \Delta_\Gamma b - r_b b + \frac{r_b}{|\Gamma(t)|} \oint_{\Gamma(t)} a \, \mathrm{d}\boldsymbol{x}, \tag{45}$$

$$\frac{\partial c}{\partial t} + \nabla_\Gamma \cdot (\boldsymbol{u} c) = D_c \Delta_\Gamma c + b_c a - r_c c. \tag{46}$$

Here, r_a, r_b and r_c denote decay rates of the local activator, global inhibitor and local inhibitor, respectively. The corresponding diffusion coefficients are D_a, D_b and D_c. In the nonlinear reaction term in the activator equation, s_a is a saturation coefficient, s_c is a Michaelis–Menten constant and b_a is a basal production rate of the activator. The constant b_c determines the growth of the local inhibitor c in the presence of the activator a. The effect of any external chemotactic field is incorporated in the signal term s. All of the variables appearing in this model are assumed to be non-dimensional. The activators and inhibitors in this model are not intended to represent molecular species. Although some molecular-based models have been successful in describing individual processes, the global morphology of real cells has proven too complex for such a description at present. We therefore have used a top-down approach, where each parameter can represent several molecular species. The pseudopod activator could read out at the level of actin nucleation, for example, through SCAR/WAVE proteins.

We assume that actin polymerisation creates a protrusive pressure that pushes the cell membrane outward in the normal direction. Recent detailed investigation of pseudopod formation suggests that this assumption is valid for cells migrating in

the absence of external cues and in the presence of chemotactic gradients [5,4]. We will assume that the rate of polymerisation is proportional to the concentration of the local activator. At rest, the cell experiences pressure from cortical tension, which maintains the spherical shape of the cell. Using a cortical shell-liquid drop model [7], the pressure generated by the cortical tension is assumed to act normally on the cell membrane and depends on the local surface curvature. The cell boundary is therefore assumed to evolve with the normal velocity

$$\mathcal{V}(\boldsymbol{x},t) = K_{prot}a(\boldsymbol{x},t) - \lambda(t)\kappa, \quad \boldsymbol{x}\in\Gamma(t). \tag{47}$$

To control the area enclosed by $\Gamma(t)$, we have used a spatially constant but time dependent cortical tension factor $\lambda(t)$. Larger values of λ will increase the cortical tension, which will eventually result in a decrease in the cell area. Conversely, by decreasing λ the cortical tension is weakened, leading to an increase in cell area. There are many possible forms that a dynamic equation for λ could take, but we have found through numerical experimentation that the following works well:

$$\frac{d\lambda}{dt} = \frac{\lambda_0\lambda(A - A_0 + dA/dt)}{A_0(\lambda+\lambda_0)} - \beta\lambda. \tag{48}$$

Here, λ_0 and β are positive parameters and A_0 is the initial prescribed area of the cell. Equation (48) is solved numerically using an explicit Euler method.

The simulations of cell chemotaxis presented in [40,41] were obtained using an approximation of the local fractional receptor occupancy, i.e. the ratio of the local number of ligand-bound receptors to the total number of receptors. The receptor occupancy was estimated using a ligand–receptor binding model which assumed the extracellular ligand field was unaffected by the binding process and the fact that the cell is moving and constantly changing morphology. While the predictions in [40,41] displayed many attributes of real cell chemotaxis, we now consider a model to account for the interaction of the cell on the external chemotactic field.

In the extracellular region, $\Omega(t)$, we assume that the concentration of ligand molecules evolves according to a linear diffusion equation. At the cell membrane, $\Gamma(t)$, we assume that a chemoattractant ligand L binds reversibly to a receptor R to form a receptor–ligand complex LR. The coupled bulk–surface system for the evolution of the ligand concentration, l, and the concentration of bound receptors, l_s, therefore takes the form

$$\frac{\partial l}{\partial t} = D\Delta l, \qquad \boldsymbol{x}\in\Omega(t) \tag{49}$$

$$\frac{\partial l_s}{\partial t} + \nabla_\Gamma\cdot(\boldsymbol{u}l_s) = D_s\Delta_\Gamma l_s + k_1(R_{tot}-l_s)l - k_{-1}l_s, \qquad \boldsymbol{x}\in\Gamma(t). \tag{50}$$

Here, D and D_s are the diffusion coefficients for the ligand and receptor–ligand complex, respectively. We assume that the total concentration of bound and unbound receptors is constant and takes the value R_{tot}. The constant k_1 is the rate of ligand association and k_{-1} the rate of disassociation. The normal flux boundary condition between the extracellular region and the cell membrane takes the form

$$-D\frac{\partial l}{\partial n} - [(\boldsymbol{u}\cdot\boldsymbol{n})]l = k_1(R_{tot}-l_s)l - k_{-1}l_s, \qquad \boldsymbol{x}\in\Gamma(t). \tag{51}$$

Determining the concentration of bound receptors, l_s allows the estimation of the local fractional receptor occupancy

$$R_o(\boldsymbol{x},t) = \frac{l_s(\boldsymbol{x},t)}{R_{tot}}.$$

It has been observed that some cells move randomly in the absence of any external cues. We have therefore included an intrinsic noise component to our system that is independent of the external chemotactic signal. For this purpose we will assume that the intrinsic noise η^t satisfies a stochastic differential equation of mean reverting type [40]. The combined effect of the response to the external signal and random intrinsic noise is modelled by the term

$$s(\boldsymbol{x},t) = r_a(\eta^t + R_o(\boldsymbol{x},t)),$$

which feeds in multiplicatively to the autocatalytic activator equation (44).

6.1. Migration in a linear gradient

The following numerical experiments were performed using the parameters found in Table 1 and a uniform time step $\Delta t = 0.1$. We first simulate a cell moving in a linear gradient of the chemoattractant field given initially by

$$l(x,y) = 5.3 + 8.5(x+0.1). \tag{52}$$

With the receptor–ligand disassociation constant $K_d = k_{-1}/k_1 = 30$, the receptor occupancy initially at the back of the cell $R_o^b = 0.15$, and at the front of the cell $R_o^f = 0.19$. To improve computational efficiency when there is significant cell movement, the diffusion equation is solved over a time-dependent annular region $\Omega(t)$, where the internal boundary represents

Table 1
Non-dimensional default parameter values for cell migration simulations. Parameters for the pseudopod-centred cell migration model are taken from [40].

Quantity	Symbol	Value
Decay rate of activator	r_a	2×10^{-2}
Basic production rate of activator	b_a	1×10^{-1}
Saturation of activator autocatalysis	s_a	5×10^{-4}
Diffusion coefficient of activator	D_a	4×10^{-7}
Production & decay rate of global inhibitor	r_b	3×10^{-2}
Diffusion coefficient of global inhibitor	D_b	4×10^{-5}
Production rate of local inhibitor	b_c	7×10^{-3}
Decay rate of local inhibitor	r_c	1.3×10^{-2}
Diffusion coefficient of local inhibitor	D_c	2.8×10^{-6}
Michaelis–Menten constant	s_c	2×10^{-1}
Scaling of protrusive velocity	K_{prot}	1×10^{-5}
Cortical tension factor	λ_0	2×10^{-6}
Cortical tension factor	β	2×10^{-2}
Bulk ligand diffusion coefficient	D	10
Ligand association rate	k_1	$\frac{1}{30}$
Ligand disassociation rate	k_{-1}	1
Ligand–receptor complex diffusion coefficient	D_s	1×10^{-6}
Total number of receptors on cell	$2\pi r_0 R_{tot}$	7×10^4

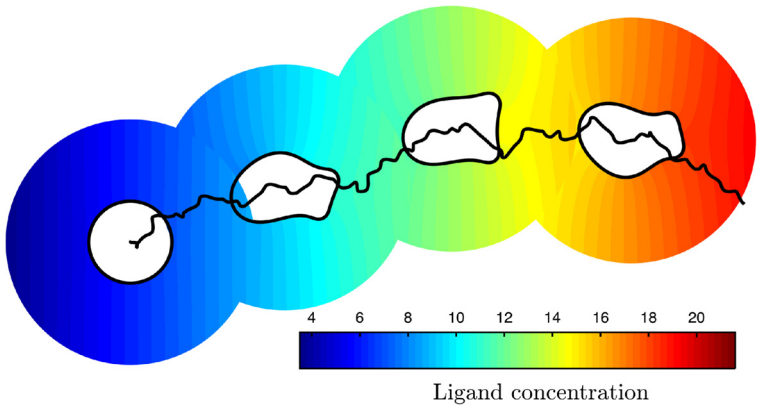


Fig. 8. Simulation of single-cell chemotaxis. Although the cell's morphology is constantly changing by pseudopod extension and retraction, the cell's pathway exhibits a strong degree of directional persistence up the chemoattractant gradient to the right.

the cell membrane, and the outer far-field boundary is a circle centred on the moving cell. The radius of the far-field boundary $r_f = 3r_0$, where the initial radius of the cell $r_0 = 0.1$. At the far-field boundary the ligand concentration is given by (52). The initial mesh has $N = 6250$ elements and is adapted isotropically towards the cell membrane; this mesh is then kept fixed as the initial computational mesh covering Ω_c. Fig. 8 shows four snap-shots of a simulated cell moving in the correct direction to the right. The cell migrates by the generation and splitting of pseudopods at the front of the cell while retracting the back. The generated mesh at one particular time is shown in Fig. 9. We can see that the bulk and boundary meshes are of high quality and follow the changing morphology of the migrating cell. Of the 5×10^5 time steps used in this simulation, only 42 remeshing steps were required demonstrating the robustness of the grid generation algorithms. The computational method has been implemented in MATLAB to make use of the Distmesh algorithm for the generation of the initial mesh and meshes when re-gridding is needed. The simulations in this section took approximately 3 hours of computing time on a desktop machine using an Intel i7 2600 quad core processor running at 3.4 GHz. In this time the simulated cell had migrated a distance of approximately 10 cell diameters. No major attempt was made to optimise the code; it can be expected that run times would be reduced dramatically by computing in an appropriate low-level language.

6.2. Migration in an initially homogeneous field

We now consider the possibility of a cell to shape an initially homogeneous ligand field and investigate the feedback of this interaction on the cell's migratory pattern. The biological interest in this problem comes from recent experimental observations that some migrating cells locally self-generate chemotactic gradients thus leading to increased persistence of migration or sustained directed migration [10,38]. Fig. 10 shows a simulation of a typical cell, where the initial ligand concentration is constant throughout the extracellular domain. We can see that movement of the cell, and changes to the cell morphology affect the ligand field (a). In particular, when a new pseudopod is created (b), the cell pushes out

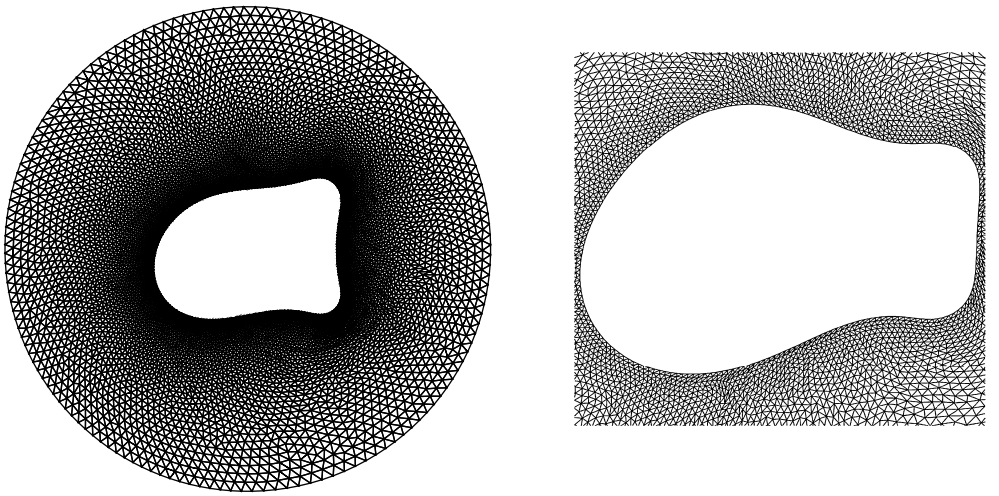


Fig. 9. Global and close up view of a typical mesh used for the simulation of cell chemotaxis. The mesh follows faithfully the migrating cell and is of good quality close to the cell membrane.

its membrane and this results in the local dilution of the concentration of the receptor–ligand complex (c). As the total receptor concentration is assumed constant, this means that additional unbound receptors become available to bind ligand molecules at the cell membrane (d). The bulk concentration at the cell surface is then depleted resulting in a local gradient in the bulk ligand field towards the cell (e). Conversely, in areas where the cell membrane is retracted such as the back of the cell and at retracting pseudopods, the local concentration of receptor–ligand complex increases slightly and this creates a gradient in the bulk ligand field away from the cell surface which drives ligand molecules away from the cell surface in these areas. The overall effect of these interactions between the extracellular ligand field and the surface receptor binding kinetics is to enhance the polarity and the persistence of the migrating cell even though the unperturbed background field is homogeneous. Additional reactions such as the degradation of ligand molecules by membrane bound enzymes and receptor internalisation could also play a major role in the generation of self-generated gradients and we intend to consider these possibilities in the future.

7. Conclusions and further work

In this paper, we have developed a computational framework for the solution of coupled bulk–surface reaction–diffusion equations in two dimensions. The proposed algorithm is based on a conservative finite element ALE scheme to approximate the solution of the PDEs, and an adaptive moving mesh method for grid generation. A novel MMPDE approach has been developed to simulate a curve moving in the normal direction by a geometric evolution law that also allows control of the tangential distribution of mesh points. The overall algorithm has been shown to work well when applied to model problems with known analytical solutions. The method has also been applied to a model of cell migration and chemotaxis and shown to predict a possible novel mechanism for cells to increase their persistence and polarity by generating their own local gradients of chemoattractants. The numerical algorithm has proved to be very robust and requires few global remeshing steps even though the simulated cell morphology is constantly changing while it is migrating. The methodology is simple to implement using standard finite element procedures and freely available routines for initial grid generation.

In future, we aim to use the developed computational framework to tackle intracellular cell processes. The main issue there will be the proper description of the deformation of material points within the moving cell. This is of course a highly non-trivial issue although there have been attempts to model the cell mechanics as an active viscoelastic medium [28,17].

The computational approach presented here has been applied to a model of single cell migration and chemotaxis. It is of course of great interest how populations of cells migrate and interact with their micro-environment and each other. To apply the approach used here would be a major challenge if many cells were to be simulated given the fitted nature of the bulk mesh covering the highly dynamic extracellular region. We think the approach presented here is best suited to obtaining detailed information from simulations of single cells or the interaction of a small number of cells. This information can then be used to properly inform population models based on partial differential equations for the density of cells using upscaling techniques.

Ultimately, we would like to extend our approach to model cell migration and chemotaxis in three dimensions. Using a suitable variational formulation, the adaptive moving mesh method used here to evolve grid nodes in the tangential direction should extend to two-dimensional surfaces. This equation system could then be coupled to a geometric evolution law for the surface normal velocity. The MMPDE approach could then be used to generate evolving meshes for three-dimensional bulk domains. The ALE surface finite element method has already been utilised for solving PDEs on evolving surfaces [14,15] and the bulk ALE–FEM method proposed here also extends naturally to three dimensions. The main challenge in three

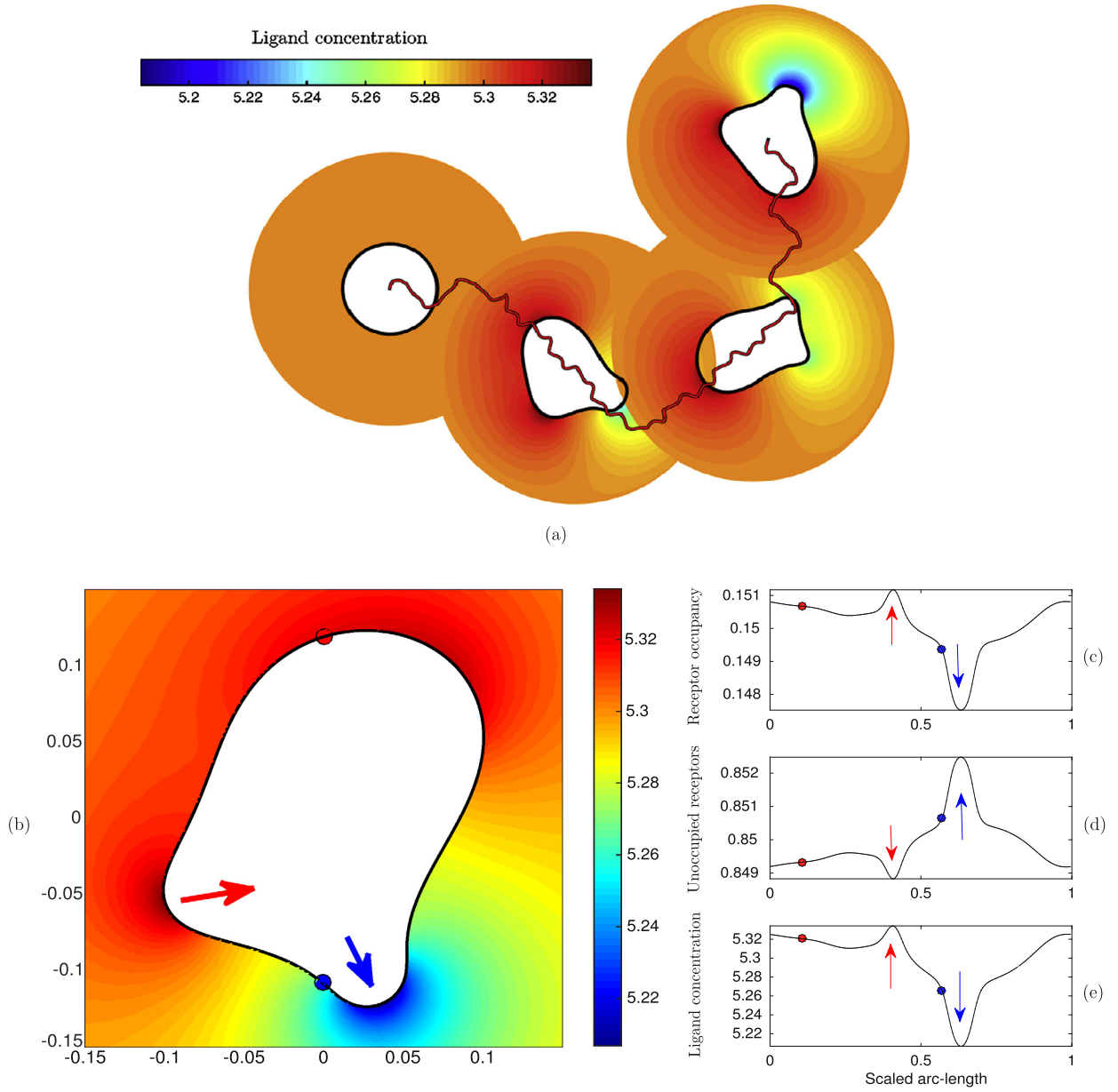


Fig. 10. Simulation of cell migration in an initially homogeneous chemotactic field. (a) Cell outlines and ligand field at equal time intervals; (b) Close-up showing the generation of a pseudopod pushing outwards (blue arrow) and a retracting older pseudopod (red arrow); (c) Receptor occupancy on the cell membrane; (d) Proportion of the total number of receptors that are unoccupied and (e) the external ligand field at the cell membrane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

dimensions however is not likely to be the development of efficient computational techniques but the increased complexity of modelling the interaction of migrating cells with their extracellular environment. The insights gained from modelling attempts in two dimensions would therefore be an essential first step in this direction.

Acknowledgements

GM and MN gratefully acknowledge the financial support from the UK Engineering and Physical Sciences Research Council (EPSRC) and Cancer Research UK for postgraduate studentship awards. Core support for RHI was provided by Cancer Research UK, grant number 15672. This work was completed while JAM was a Leverhulme Trust Research Fellow, supported by the award RF-2014-522.

References

[1] M. Balazovjech, K. Mikula, A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force, SIAM J. Sci. Comput. 33 (5) (2011) 2277–2294.
[2] J.W. Barrett, H. Garcke, R. Nürnberg, On the variational approximation of combined second and fourth order geometric evolution equations, SIAM J. Sci. Comput. 29 (2007) 1006–1041.
[3] G. Beckett, J.A. Mackenzie, A. Ramage, D.M. Sloan, On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution, J. Comput. Phys. 167 (2) (2001) 372–392.
[4] G. Beckett, J.A. Mackenzie, A. Ramage, D.M. Sloan, Computational solution of two-dimensional unsteady PDEs using moving mesh methods, J. Comput. Phys. 182 (2) (2002) 478–495.
[5] G. Beckett, J.A. Mackenzie, M.L. Robertson, A moving mesh finite element method for the solution of two-dimensional Stefan problems, J. Comput. Phys. 168 (2) (2001) 500–518.
[6] G. Beckett, J.A. Mackenzie, M.L. Robertson, An r-adaptive finite element method for the solution of the two-dimensional phase-field equations, Commun. Comput. Phys. 1 (5) (2006) 805–826.
[7] M. Booty, M. Siegel, A hybrid numerical method for interfacial fluid flow with soluble surfactant, J. Comput. Phys. 229 (10) (2010) 3864–3883.
[8] W. Budd, C.J. Huang, R.D. Russell, Adaptivity with moving grids, Acta Numer. 18 (2009) 111–241.
[9] W. Cao, W. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (2) (1999) 221–244.
[10] E. Donà, J.D. Barry, G. Valentin, C. Quirin, A. Khmelinskii, A. Kunze, S. Durdu, L.R. Newton, A. Fernandez-Minan, W. Huber, M. Knopp, D. Gilmour, Directional tissue migration through a self-generated chemokine gradient, Nature 503 (2013) 285–289.
[11] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions, Comput. Methods Appl. Mech. Eng. 33 (1–3) (1982) 689–723.
[12] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (2) (1991) 450–476.
[13] C.M. Elliott, T. Ranner, Finite element analysis for a coupled bulk–surface partial differential equation, IMA J. Numer. Anal. 33 (2013) 377–402.
[14] C.M. Elliott, B. Stinner, C. Venkataraman, Modelling cell motility and chemotaxis with evolving surface finite elements, J. R. Soc. Interface 9 (76) (2012) 3027–3044.
[15] C.M. Elliott, V. Styles, An ALE ESFEM for solving PDEs on evolving surfaces, Milan J. Math. 80 (2) (2012) 469–501.
[16] R.G. Endres, N.S. Wingreen, Accuracy of direct gradient sensing by single cells, Proc. Natl. Acad. Sci. USA 105 (41) (2008) 15749–15754.
[17] U.Z. George, A. Stéphanou, A. Madzvamuse, Mathematical modelling and numerical simulations of actin dynamics in the eukaryotic cell, J. Math. Biol. 66 (3) (2013) 547–593.
[18] W.R. Holmes, L. Edelstein-Keshet, A comparison of computational models for eukaryotic cell shape and motility, PLoS Comput. Biol. 8 (2012) e10002793.
[19] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, J. Comput. Phys. 171 (2) (2001) 753–775.
[20] W. Huang, Y. Ren, R.D. Russell, Moving mesh methods based on moving mesh partial differential equations, J. Comput. Phys. 113 (2) (1994) 279–290.
[21] W. Huang, Y. Ren, R.D. Russell, Moving mesh PDEs based on the equidistribution principle, SIAM J. Numer. Anal. 31 (3) (1994) 709–730.
[22] W. Huang, R.D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, SIAM J. Sci. Comput. 20 (3) (1999) 998–1015.
[23] W. Huang, R.D. Russell, Adaptive Moving Mesh Methods, vol. 174, Springer, 2010.
[24] T.J.R. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Eng. 29 (3) (1981) 329–349.
[25] R.H. Insall, Understanding eukaryotic chemotaxis: a pseudopod-centred view, Nat. Rev. Mol. Cell Biol. 11 (2010) 453–458.
[26] R.H. Insall, L.M. Machesky, Actin dynamics at the leading edge: from simple machinery to complex networks, Dev. Cell 17 (3) (2009) 310–322.
[27] A.J. James, J. Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, J. Comput. Phys. 201 (2) (2004) 685–722.
[28] E. Kuusela, W. Alt, Continuum model of cell adhesion and migration, J. Math. Biol. 58 (1–2) (2009) 135–161.
[29] Y.-I. Kwon, J.J. Derby, Modeling the coupled effects of interfacial and bulk phenomena during solution crystal growth, J. Cryst. Growth 230 (1) (2001) 328–335.
[30] H. Levine, D.A. Kessler, W.-J. Rappel, Directional sensing in eukaryotic chemotaxis: a balanced inactivation model, Proc. Natl. Acad. Sci. USA 103 (26) (2006) 9761–9766.
[31] C.B. Macdonald, B. Merriman, S.J. Ruuth, Simple computation of reaction–diffusion processes on point clouds, Proc. Natl. Acad. Sci. USA 110 (23) (2013) 9209–9214.
[32] A. Madzvamuse, A.H.W. Chung, C. Venkataraman, Stability analysis and simulations of coupled bulk–surface reaction–diffusion systems, Proc. R. Soc., Math. Phys. Eng. Sci. 471 (2175) (2015) 20140546.
[33] A. Madzvamuse, A.H.W. Chung, The bulk–surface finite element method for reaction–diffusion systems on stationary volumes, Finite Elem. Anal. Des. 108 (2016) 9–21.
[34] E.S. Medvedev, A.A. Stuchebrukhov, Proton diffusion along biological membranes, J. Phys. Condens. Matter 23 (23) (2011) 234103.
[35] H. Meinhardt, Orientation of chemotactic cells and growth cones: models and mechanisms, J. Cell Sci. 112 (1999) 2867–2874.
[36] J. Meyers, J. Craig, D.J. Odde, Potential for control of signaling pathways via cell size and shape, Curr. Biol. 16 (17) (2006) 1685–1693.
[37] K. Mikula, M. Remešíková, P. Sarkoci, D. Ševčovič, Manifold evolution with tangential redistribution of points, SIAM J. Sci. Comput. 36 (4) (2014) A1384–A1414.
[38] A.J. Muinonen-Martin, O. Susanto, Q. Zhang, E. Smethurst, W.J. Faller, D.M. Veltman, G. Kalna, C. Lindsay, D.C. Bennett, O.J. Sansom, et al., Melanoma cells break down LPA to establish local gradients that drive chemotactic dispersal, PLoS Biol. 12 (10) (2014) e1001966.
[39] M.P. Neilson, J.A. Mackenzie, S.D. Webb, R.H. Insall, Use of the parameterised finite element method to robustly and efficiently evolve the edge of a moving cell, Integr. Biol. 2 (11–12) (2010) 687–695.
[40] M.P. Neilson, J.A. Mackenzie, S.D. Webb, R.H. Insall, Modelling cell movement and chemotaxis using pseudopod-based feedback, SIAM J. Sci. Comput. 33 (3) (2011) 1035–1057.
[41] M.P. Neilson, D.M. Veltman, P.J.M. van Haastert, S.D. Webb, J.A. Mackenzie, R.H. Insall, Chemotaxis: a feedback-based computational model robustly predicts multiple aspects of real cell behaviour, PLoS Biol. 9 (5) (2011) e1000618.
[42] S.R. Neves, P. Tsokas, A. Sarkar, E.A. Grace, P. Rangamani, S.M. Taubenfeld, C.M. Alberini, J.C. Schaff, R.D. Blitzer, I.I. Moraru, R. Iyengar, Cell shape and negative links in regulatory motifs together control spatial information flow in signaling networks, Cell 133 (4) (2008) 666–680.
[43] I.L. Novak, F. Gao, Y.S. Choi, D. Resasco, J.C. Schaff, B.M. Slepchenko, Diffusion on a curved surface coupled to diffusion in the volume: application to cell biology, J. Comput. Phys. 226 (2) (2007) 1271–1290.
[44] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
[45] Z. Pan, B. Wetton, A numerical method for coupled surface and grain boundary motion, Eur. J. Appl. Math. 19 (3) (2008) 311–327.
[46] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Rev. 46 (2) (2004) 329–345.

[47] A. Rätz, M. Röger, Turing instabilities in a mathematical model for signaling networks, J. Math. Biol. 65 (6) (2012) 1215–1244.
[48] D. Shao, H. Levine, W. Rappel, Coupling actin flow, adhesion, and morphology in a computational cell motility model, Proc. Natl. Acad. Sci. USA 109 (18) (2012) 6851–6856.
[49] W. Strychalski, D. Adalsteinsson, T.C. Elston, Simulating biochemical signaling networks in complex moving boundaries, SIAM J. Sci. Comput. 32 (5) (2010) 3039–3070.
[50] R.H. Sucgang, C.J. Weijer, F. Siegert, J. Franke, R.H. Kessin, Null mutations of the *Dictyostelium* cyclic nucleotide phosphodiesterase gene block chemotactic cell movement in developing aggregates, Dev. Biol. 192 (1) (1997) 181–192.
[51] R.T. Tenchev, J.A. Mackenzie, T.J. Scanlon, M.T. Stickland, Finite element moving mesh analysis of phase change problems with natural convection, Int. J. Heat Fluid Flow 26 (4) (2005) 597–612.
[52] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644.
[53] A. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform triangular mesh, J. Comput. Phys. 1 (2) (1966) 149–172.
[54] C.W. Wolgemuth, M. Zajac, The moving boundary node method: a level set-based, finite volume algorithm with applications to cell motility, J. Comput. Phys. 229 (19) (2010) 7287–7308.